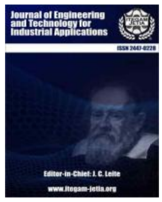
Journal of Engineering and Technology for Industrial Applications

ITEGAM-JETIA

Manaus, v.12 n.58, p.1230-1244. March/April, 2026.
DOI: https://doi.org/10.5935/jetia.v12i58.3303

ISSN ONLINE: 2447-0228



RESEARCH ARTICLE OPEN ACCESS

A HIERARCHICAL LSTM FRAMEWORK FOR CAPTURING LONG- AND SHORT-TERM PREFERENCES IN POI RECOMMENDATION

Sarala Patchala*[1], Vijay Babu Burra[2], Vullam Naga Gopi Raju[3], Banda SNV Ramana Murthy[4] and Desamala Prabhakara Rao[5]

[1]Associate Professor, Department of ECE, KKR & KSR Institute of Technology and Sciences, Guntur, Andhra Pradesh, India, Andhra Pradesh, India
[2]Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, GUNTUR
[3]Professor, Department of Computer Science and Engineering, Chalapathi Institute of Engineering And Technology, Guntur
[4]Assistant Professor, Department of CSE-AIML, Aditya University, Surampalem,A.P.
[5]Department f ECE, Chalapathi Institute of Technology, Mothadaka,Guntur, Andhra Pradesh, India

[1]http://orcid.org/0000-0002-5184-0814, [2]http://orcid.org//0000-0003-0139-8025, [3]http://orcid.org/0009-0008-4894-375X
[4]http://orcid.org/0009-0003-8371-1691, [5]https://orcid.org/0009-0001-3930-3258

E-email: *saralajntuk@gmail.com, vijay_gemini@kluniversity, ingopi.raju524@gmail.com ramanamurthy.banda@gmail.com, prabhakardesamala@gmail.com

ARTICLE INFO

Article History
Received: January 13, 2026
Reviewed: February 15, 2026
Accepted: March 27, 2026
Published: April 30, 2026

Keywords:
POI,
LSTM,
HiLSTM,
RNN,
Deep Learning.

ABSTRACT

Point-of-Interest (POI) recommendation is crucial for improving user experience in location-based social networks (LBSNs). With the growing number of users checking in at various places personalized recommendations are necessary to provide relevant suggestions. Existing methods use long short-term memory (LSTM) networks to model user preferences. However, these methods either consider long- and short-term preferences separately or merge them into a single model without effectively capturing the interactions.This research revisits the problem of long- and short-term preference learning by proposing a hierarchical LSTM (HiLSTM) framework. The framework aims to enhance next POI recommendations by learning representations at two levels: POI-level and semantic-level. Instead of treating these factors independently, HiLSTM integrates them through a structured hierarchical learning approach.One of the key challenges in POI recommendation is handling the sparsity of check-in data. Many users frequently visit new locations. It makes difficult to rely solely on past visits. The proposed model addresses this by introducing a semantic filter.It provides recommendations based on a user's categorical preferences. By filtering out irrelevant POIs at an early stage, the recommendation process becomes more effective and computationally efficient.To capture long-term user preferences, HiLSTM employs an attention mechanism. Meanwhile, short-term preferences are derived from recent check-ins. It confirms that immediate user intent is not overlooked. The combination of these two components results in a more balanced and accurate recommendation system.These datasets contain check-in records from location-based social networks, enabling rigorous evaluation. The hierarchical structure and attention mechanism contribute to a significant improvement in recommendation precision. This work introduces a novel hierarchical LSTM framework for next POI recommendation.

I. INTRODUCTION

Smart mobile devices and locationbased services are widely used[1]. Users often check in at different POIs. This behavior creates large amounts of spatiotemporal data. The data helps improve user experience[2]. It enables personalized recommendations.Providing accurate POI recommendations is challenging. User preferences are diverse[3]. Mobility patterns vary. Check-in data is often sparse.POI recommendation aims to predict the next location a user isto visit based on historical check-ins, contextual information and external influences[4]. Traditional recommendation approaches like collaborative filtering and matrix factorization struggle with the unique characteristics of POI data. These methods often fail to capture the temporal and spatial dependencies that shape user mobility patterns[5]. Moreover, they do not effectively distinguish between long-term stable preferences and short-term dynamic interests.Deep learning models help address these issues. Recurrent neural networks (RNNs) are commonly used[6]. LSTM networks, a variant of RNNs is also applied.

They improve POI recommendation.While these models improve sequential learning, they often process all influencing factors in a uniform manner. Some existing methods separate long- and short-term preferences but fail to model the interactions effectively[7]. Others merge all information into a single LSTM model that overlook the nuanced impact of various factors.This research focuses on long- and short-term preference learning for POI recommendation. It proposes a HiLSTM framework. The framework models POI-level and semantic-level representations separately. It integrates them into a structured learning approach[8]. The POI level captures geographical constraints and location-based behaviors. The semantic level reflects user intent[9]. It considers check-in categories, timestamps and contextual factors.Additionally, an attention mechanism is introduced to improve long-term preference learning by selecting the most relevant past check-ins[10]. A semantic filtering strategy is also incorporated to address data sparsity and enhance recommendation precision. Through extensive experiments on real-world datasets, HiLSTM demonstrates superior performance compared to existing state-of-the-art methods.This study makes several significant contributions to the field of POI recommendation:

- ❖ HiLSTM separately models POI-level and semantic-level preferences, effectively capturing the distinct roles in shaping user mobility patterns.
- ❖ The framework considers both stable long-term user preferences and dynamic short-term behaviors to improve recommendation accuracy.
- ❖ An attention-based approach is employed to prioritize the most relevant historical check-ins, so that only meaningful past behaviors influence recommendations.
- ❖ A filtering mechanism is introduced to eliminate POIs that do not align with a user's categorical preferences, reducing irrelevant recommendations and improving precision.
- ❖ By incorporating semantic features beyond raw POI check-ins, HiLSTM mitigates data sparsity issues and enhances recommendation robustness.
- ❖ The effectiveness of HiLSTM is demonstrated through rigorous experiments on two large-scale, real-world datasets, showing superior performance compared to traditional methods.
- ❖ The proposed framework is designed to be adaptable to diverse user behaviors and scalable for large-scale applications in real-world location-based services.

This paper introduces HiLSTM, a new framework for POI recommendation. It integrates hierarchical preference learning with attention-based memory mechanisms. HiLSTM distinguishes between POI-level and semantic-level features[11]. It captures user behavior effectively. It generates more accurate recommendations. The model overcomes key limitations of existing methods. It shows significant performance improvements[12]. Future research extend HiLSTM. It include social interactions and external events and improve recommendation quality.

II. RELATED WORK

Research on POI recommendation has evolved significantly over the past decade. Various approaches include collaborative filtering, deep learning models and hybrid techniques. They have been explored to enhance recommendation accuracy. This section reviews prior works in these areas, focusing on the strengths and limitations.Collaborative filtering (CF) is one of the most widely used techniques in recommendation systems. It predicts user preferences based on the behavior of similar users [13]. CF methods are classified into user-based and item-based approaches. User-based CF computes similarities between users to recommend POIs, while item-based CF recommends POIs based on similar visited locations [14].

Despite the effectiveness, CF methods face challenges like sparsity and cold-start problems [15]. To address these issues, matrix factorization (MF) techniques have been introduced. Methods like Singular Value Decomposition (SVD) and Probabilistic Matrix Factorization (PMF) decompose the user-POI interaction matrix into lower-dimensional representations [16,17]. GeoMF extends MF by incorporating geographical information, improving POI recommendation accuracy [18]. However, CF-based approaches often fail to capture sequential dependencies in user check-ins, limiting the effectiveness. Deep learning has recently gained popularity in POI recommendation. RNNs and LSTM networks are particularly effective in modeling sequential user behavior[19].

These models learn complex temporal dependencies and adapt to changing user preferences.Several studies have explored RNN-based approaches. DeepMove combines RNNs with an attention mechanism to capture user mobility patterns [20]. Similarly, ST-LSTM integrates spatiotemporal dependencies into LSTM networks to enhance recommendation accuracy [21]. Other models, namely LSTPM, use geo-dilated LSTM layers to model long-range dependencies in user check-ins [22].Despite the success, deep learning methods require large datasets and computational resources. Additionally, existing models do not effectively capture the interplay between long- and short-term preferences.It is crucial for accurate POI recommendation [23].To improve recommendation accuracy, hybrid models have been proposed.

These models integrate long-term user preferences (stable behaviors) with short-term preferences (recent check-ins) to provide more personalized recommendations. DeepMove learns long-term and short-term representations separately using historical check-in data [20]. LSTPM employs attention mechanisms to prioritize relevant historical trajectories [22]. However, these models often treat long- and short-term preferences independently or combine them without modeling the interactions effectively.The proposed HiLSTM framework addresses this limitation by integrating POI-level and semantic-level features. By considering both geographical constraints and user intent, HiLSTM provides more accurate and personalized recommendations.This section reviewed major research contributions in POI recommendation. It also includes collaborative filtering, deep learning-based approaches and hybrid models. While existing methods have made significant progress, they have limitations in capturing complex user preferences. The proposed HiLSTM framework aims to overcome these challenges by integrating hierarchical preference learning with attention mechanisms.

III. PRELIMINARY

POI recommendation systems rely on various data-driven techniques to predict user movements and preferences. The effectiveness of these recommendations depends on understanding user behavior, location dynamics and contextual factors. This section introduces fundamental concepts and mathematical formulations that support the proposed model.The Figure 1 represents a model that processes different embeddings to capture the impact of POIs and categories using a sequential learning approach. The model uses LSTM networks to process embeddings over time. The top section represents the impact of POIs, while the bottom section represents the impact of categories. Each input consists of four embeddings: user embedding, POI embedding, time slot embedding and category embedding.

These embeddings are processed separately in the two pathways, with each LSTM layer extracts patterns from sequential data. The LSTM outputs are then carried forward in the sequence, allowing the model to capture dependencies over time. At the final stage, the outputs from both pathways are concatenated and combined using a summation operation. This fusion step results in the model that integrates information from both POI-based and category-based influences. By combining these two aspects, the model generate more accurate predictions or recommendations. The design highlights the importance of learning both spatial and temporal factors when analyzing user behaviors. This approach helps the model understand user preferences better. It improves accuracy in predicting user behavior. It is useful for location-based recommendations. It also enhances personalized suggestions.

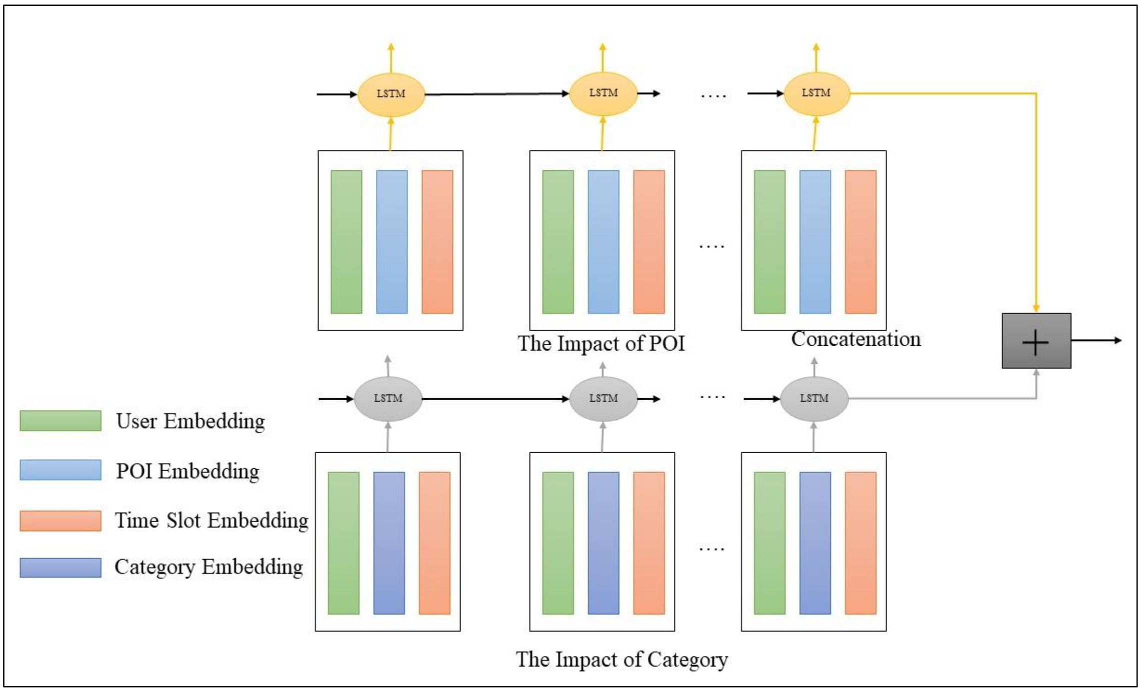


Figure 1: Each LSTM Model for one Specific Factor.
Source: Authors, (2026).

User movements in LBSNs are represented as a sequence of check-ins at different POIs. Let $U = \{u_1, u_2, u_3, u_4, \ldots \ldots, u_N\}$ be a set of users, and $P = \{p_1, p_2, p_3, p_4, \ldots \ldots, p_M\}$ be a set of POIs. Each check-in is denoted as a tuple:

$$C_{u,t} = (p_t, t, c_t, d_t) \tag{1}$$

Here, p_t is the POI visited at time t, c_t represents the category of the POI. d_t is the geographical distance from the previous check-in location. A user's entire check-in history forms a trajectory:

$$T_u = (C_{u,1}, C_{u,2}, C_{u,3}, \ldots \ldots, C_{u,T}) \tag{2}$$

This trajectory isanalyzed to extract long-term and short-term preferences for personalized recommendations.User mobility patterns exhibit both long-term stability and short-term dynamics.

To model movement patterns, two types of trajectories are defined. The long-term trajectory represents a user's historical movement over a long period. It helps understand habitual travel behavior. The short-term trajectory captures recent movements. It reflects the user's immediate intent. This helps predict short-term POI preferences accurately.To model these preferences, we introduce a weighting mechanism that assigns different levels of importance to past check-ins:

$$S_u = \propto T_u^{long} + \beta T_u^{short} \tag{3}$$

Here, α and β are weighting factors controlling the influence of long-term and short-term preferences.The proposed framework utilizes a hierarchical LSTM model to process POI-level and semantic-level representations. A standard LSTM cell updates its hidden state using:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{4}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f \tag{5}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o \tag{6}$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_c x_t + U_c h_{t-1} + b_c \tag{7}$$

$$h_t = o_t \tanh(c_t) \tag{8}$$

Here, i_t, f_t and o_t are the input, forget and output gates, respectively. c_t is the cell state and h_t is the hidden state.In the hierarchical LSTM structure, a higher-level LSTM integrates POI sequences while another LSTM models semantic-level features namely categories and timestamps. This structure allows us to improve user preferences dynamically.Geographical proximity plays a significant role in user mobility. Users tend to visit POIs that are close to the previous locations. To incorporate this factor, we define a spatial proximity function:

$$G(P_i, P_j) = e^{-\lambda\, d_{ij}} \tag{9}$$

Here, d_{ij} represents the distance between POIs p_i and p_j and λ is a decay parameter controlling the influence of distance.Additionally, we use a transition probability matrix to model user movement patterns:

$$P_u\left(\frac{P_j}{P_i}\right) = \frac{e^{-\lambda\, d_{i,j}}}{\sum_k e^{-\lambda\, d_{i,k}}} \tag{10}$$

This probability helps prioritize nearby POIs when generating recommendations.User check-ins exhibit strong temporal patterns. Some POIs are more popular during specific times of the day. To capture temporal dependencies, we define a time-aware influence function:

$$T(t) = \frac{1}{1+e^{-\gamma(t-\mu)}} \tag{11}$$

Here, γ controls the sharpness of the time preference curve. μ represents the peak visiting time of a POI.By integrating spatial and temporal influences, we enhance the adaptability of the recommendation model.This section introduced essential concepts related to POI recommendation, including trajectory modeling, hierarchical LSTMs and geographical and temporal influences. These foundational principles provide the basis for constructing an effective recommendation framework. The next sections will focus on the implementation details and performance evaluation of the proposed model.

IV. PROBLEM FORMULATION

In this section, we define the problem of next POI recommendation. The goal is to predict the most probable next POI a user will visit based on historical check-in data. We introduce key concepts, notations and mathematical formulations necessary for modeling user behavior.Users interact with POIs over time, forming a sequence of check-ins. Each user check-in is given in (1). A user's entire movement sequence, known as a trajectory is given in (2). The objective is to predict the next check-in location p_{T+1} given past trajectory T_u.Users exhibit both long-term and short-term mobility preferences. To balance both preferences, we define a weighted sum is given in (3). To model user preferences, we employ a hierarchical learning approach using LSTM networks. The LSTM model updates its hidden state is given in (4,5,6,7,8).

The proposed model consists ofPOI-Level LSTM thatlearns sequential dependencies among visited locations.Semantic-Level LSTM captures contextual influences like time and POI categories.Users typically prefer visiting POIs close to the current location. To incorporate this, we define a geographical influence function is calculated and given in (9). A transition probability between POIs is formulated and given in (10). Temporal influence is modeledand given in (11). The final recommendation score for POI p is computed as:

$$Score_u(p) = \alpha\, S_u + \beta\, G_u\,(p) + \gamma\, T_u(p) \tag{12}$$

Here, S_u captures long- and short-term user preferences, $G_u\,(p)$ represents geographical influence. $T_u(p)$accounts for temporal behavior, α, β, γ control the influence of each factor.This section formulated the next POI recommendation problem, explaining key elements namely user mobility modeling, hierarchical learning and spatiotemporal influences. Future sections will cover implementation and experimental results.

V. HILS DESIGN

Hierarchical LSTM with Long- and Short-Term Preference Learning (HILS) is a framework designed to enhance next POI recommendations by integrating hierarchical user preference modeling. The goal is to effectively capture long-term stable behaviors and short-term user interests using an advanced LSTM-based architecture. This section provides an in-depth explanation of the HILS framework, including its structure, equations and implementation.The HILS framework consists of three primary modules. First, the trajectory modeling module employs a hierarchical LSTM to learn patterns from POI check-in data. Second, the preference learning module extracts insights from both long-term and short-term trajectories. Finally, the recommendation module ranks potential POIs based on the computed scores from these models. The entire system is optimized using backpropagation and gradient descent.HILS employs a hierarchical LSTM structure to encode user mobility patterns. The model has two main components. The POI-Level LSTM learns sequential patterns among visited locations. It helps capture user movement behavior over time. The Semantic-Level LSTM integrates contextual details. It considers time, category and spatial influences. This enhances recommendation accuracy by understanding user intent and external factors.The core equations governing the LSTM cell updates are given in (4,5,6,7,8). The final trajectory representation is computed as:

$$T_u = \left\{T_u^{poi}, T_u^{sem}\right\} \tag{13}$$

Here, T_u^{poi} represents sequential POI interactions and T_u^{sem}models contextual semantics.User mobility exhibits both stable patterns over time and dynamic changes due to recent interests. To model these influences, HILS employs a dual-preference learning mechanism. The long-term preference module utilizes an attention mechanism to identify and prioritize relevant past trajectories:

$$a_i = \frac{e^{T_i * T_{current}}}{\sum_j e^{T_j * T_{current}}} \tag{14}$$

Here, a_i is the attention weight assigned to past trajectory T_i relative to the current check-in sequence T_{recent}. The final long-term preference vector is:

$$S_u^{long} = \sum_i a_i\, T_i \tag{15}$$

The short-term preference model updates based on recent POI sequences using:

$$S_u^{short} = LSTM(T_{recent}) \tag{16}$$

Here, T_{recent} represents the latest POI visits. The combined user preference representation is:

$$S_u = \propto S_u^{long} + \beta S_u^{short} \tag{17}$$

Here, α and β are weight parameters balancing the long-term and short-term preferences.Users often prefer visiting POIs near the current location. To model this behavior, HILS defines a spatial influence function is given in (9). The transition probability between POIs is computed and given in (10). Temporal preferences are modeled using a time-dependent function is given in (11). The final recommendation process involves computing a ranking score for each candidate POI. This score is determined and given in (12). The top-k POIs with the highest scores are recommended to the user.HILS is built using a hierarchical LSTM architecture. Each layer contains 256 hidden units.

This structure helps in capturing complex dependencies in user movements. The Adam optimizer is used for training. It operates with a learning rate of 0.001. This achieves stable and efficient learning. The model uses 128-dimensional embeddings. These embeddings represent POIs and semantic attributes. They help in learning meaningful relationships between locations and user preferences. Training is conducted on real-world check-in datasets. This validates the model's performance. The datasets provide diverse and realistic data. This results in a modelthat generalizes well to different user behaviors and environments.This section explains the HILS framework in detail. It covers hierarchical trajectory modeling.

It includes long- and short-term user preference learning. It also models geographical and temporal influences. Finally, it describes the POI recommendation mechanism.The next sections will discuss experimental evaluations and comparative performance analysis.The following pseudocode describes the complete process of the HILS framework for next POI recommendation.

Algorithm 1: HILS for POI Recommendation.

1. Input: User check-in history T_u, POI set P, Parameters $\boldsymbol{\alpha, \beta, \gamma}$
2. Output: Recommended POIs R_u
3. Initialize hierarchical LSTM with POI and semantic embeddings
4. for each user u in dataset do
5. Extract long-term trajectory T_u^{long} and short-term trajectory T_u^{short}
6. Compute long-term preference $\boldsymbol{S_u^{long}} = \sum_i \boldsymbol{a_i\, T_i}$
7. Compute short-term preference $\boldsymbol{S_u^{short}} = \boldsymbol{LSTM(T_{recent})}$
8. Merge preferences: $\boldsymbol{S_u} = \propto \boldsymbol{S_u^{long}} + \boldsymbol{\beta S_u^{short}}$
9. Compute spatial influence: $G_u(p) = e^{-\lambda d_{ij}}$
10. Compute temporal influence: $T_u(p) = \dfrac{1}{1+e^{-\gamma(t-\mu)}}$
11. Compute final recommendation score: $\boldsymbol{Score_u(p) = \alpha\, S_u + \beta\, G_u(p) + \gamma\, T_u}$
12. Rank POIs based on $\boldsymbol{Score_u(p)}$
13. Select top-k POIs as recommendations $\boldsymbol{R_u}$
14. end for
15. Return recommended POIs $\boldsymbol{R_u}$

Source: Authors, (2026).

VI. PERFORMANCE EVALUATION

The performance of HILS is evaluated using real-world datasets to compare its effectiveness against traditional models. Various experiments are conducted to study model performance. The impact of different components is analyzed. An ablation study is performed. The importance of geographical influences is examined. Temporal influences are also evaluated.The evaluation considers standard recommendation metrics like Recall@k, NDCG@k and Precision@k.The datasets used for evaluation are preprocessed by filtering out users with fewer than ten check-ins and POIs visited by fewer than five users. The data is then split into training (70%), validation (10%) and testing (20%) sets. The evaluation metrics provide a fair comparison across models.HILS is compared against various traditional models. It includes LSTPM, ST-LSTM, DeepMove and GeoMF. The results of this comparison are presented in Table 1, showing that HILS outperforms the traditional methods across all metrics.

Table 1: Performance Comparison of HILS and traditional Models.

Model	Recall@5	NDCG@5	Precision@5
HILS	0.712	0.594	0.381
LSTPM	0.678	0.563	0.357
ST-LSTM	0.653	0.541	0.342
DeepMove	0.631	0.529	0.336
GeoMF	0.597	0.498	0.312

Source: Authors, (2026).

To understand the contributions of various components in HILS, an ablation study is conducted. The results presented in Table 2 demonstrate the significance of different features within the model. Removing either the long-term or short-term preference learning components significantly impacts performance. It highlights the importance in capturing user mobility patterns.

Table 2: Ablation Study on HILS Components.

Model Variant	Recall@5	NDCG@5	Precision@5
Full HILS Model	0.712	0.594	0.381
No Long-Term Preference	0.678	0.561	0.358
No Short-Term Preference	0.664	0.548	0.344

Source: Authors, (2026).

The geographical and temporal influences also play a crucial role in user behavior modeling. As shown in Table 3 removing either of these factors leads to a decline in performance, emphasizing the significance.

Table 3: Impact of Geographical and Temporal Factors in HILS.

Model Variant	Recall@5	NDCG@5	Precision@5
Full HILS Model	0.712	0.594	0.381
Without Geographical Influence	0.682	0.570	0.362
Without Temporal Influence	0.671	0.559	0.351

Source: Authors, (2026).

To analyze the performance of HILS a comparison is made with its variants. It includes HILS-N, HILS-NH and HILS-S. Table 4 presents the results of this comparison. The findings show important results. Omitting hierarchical modeling lowers accuracy. Removing semantic information also reduces accuracy. This highlights the importance of HILS. It effectively captures complex mobility patterns.

Table 4: Performance Comparison of HILS Variants.

Model Variant	Recall@5	NDCG@5	Precision@5
HILS	0.712	0.594	0.381
HILS-N	0.689	0.570	0.359
HILS-NH	0.674	0.555	0.345
HILS-S	0.662	0.542	0.338

Source: Authors, (2026).

The results show that the hierarchical structure of HILS provides superior performance compared to simplified variants.Integrating semantic features improves the model. Hierarchical modeling enhances learning. It captures both long-term and short-term user behaviors. This results in better POI recommendations.The performance evaluation demonstrates that HILS significantly improves next POI recommendation accuracy compared to traditional models. The hierarchical LSTM structure effectively captures long- and short-term user preferences while leveraging geographical and temporal factors. The ablation study shows the need for different model components. Comparisons with HILS variants highlight hierarchical and semantic learning. These are important for mobility prediction. Future work focus on hyperparameter optimization with Additional contextual influences included. Testing on diverse datasets improve performance.

The Figure 2 presents a comparison of different models based on the recall performance at two levels: Recall@5 and Recall@10. The models analyzed include HILS, LSTPM, ST-LSTM, DeepMove and GeoMF. From the graph, it is clear that Recall@10 values are higher than Recall@5 across all models. Among them, the HILS model shows the highest Recall@5 value, slightly exceeding 0.7. Its Recall@10 is close to 0.8. LSTPM and ST-LSTM models follow both having Recall@5 values around 0.68. Recall@10 values above 0.72. The DeepMove model has a slightly lower Recall@5 value, approximately 0.65, with its Recall@10 reaching around 0.72. Finally, the GeoMF model has the lowest Recall@5 slightly above 0.6 while its Recall@10 is around 0.7. The gap between Recall@5 and Recall@10 stays nearly the same for all models. Adding more recommendations increases recall. However, the improvement becomes smaller as more recommendations are added. HILS outperforms other models, suggesting that it provides better retrieval performance in this evaluation. GeoMF, on the other hand, has the lowest recall performance among the tested models. The results show that certain models perform better. Models with sequential learning and spatial-temporal dependencies have an advantage. HILS and LSTPM outperform other architectures.

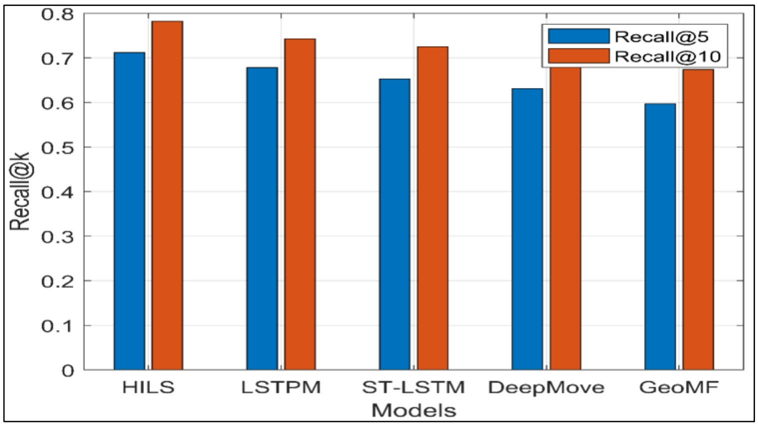


Figure 2: Recall@k versus Different Models.

Source: Authors, (2026).

The Figure 3 compares different models based on the NDCG at two levels: NDCG@5 and NDCG@10. The models evaluated are HILS, LSTPM, ST-LSTM, DeepMove and GeoMF. In the chart, it is clear that NDCG@10 is higher than NDCG@5 for all models. Among these models, HILS achieves the highest NDCG@5 value, slightly above 0.6. NDCG@10 value reaches approximately 0.65. LSTPM and ST-LSTM follow closely with NDCG@5 values around 0.57. NDCG@10 values slightly above 0.6. DeepMove has a slightly lower NDCG@5, approximately 0.54 with its NDCG@10 reaching close to 0.6. Finally, GeoMF has the lowest NDCG@5, around 0.5. NDCG@10 is just above 0.55. The trend in the chart suggests that models with higher NDCG@5 tend to maintain the ranking order at NDCG@10. The gap between NDCG@5 and NDCG@10 stays almost the same for all models. Adding more recommendations improves ranking quality. However, the improvement decreases as more recommendations are included. HILS outperforms all other models, showing its ability to provide better-ranked recommendations. GeoMF, on the other hand performs the worst among the tested models. The results show that some models rank relevance better. Models using sequential learning and spatial-temporal dependencies perform well. HILS and LSTPM are more effective than other models.

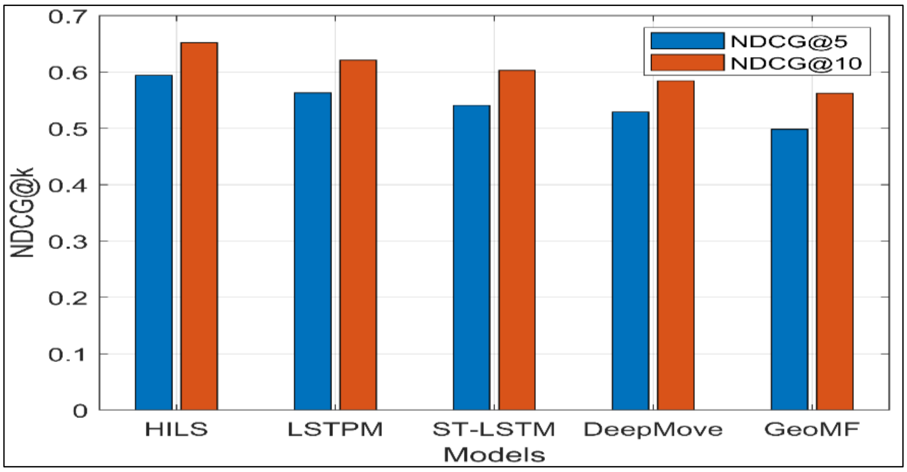


Figure 3: NDCG@k versus Different Models.
Source: Authors, (2026).

The Figure 4 presents a comparison of different models based on the precision at two levels: Precision@5 and Precision@10. The models analyzed include HILS, LSTPM, ST-LSTM, DeepMove and GeoMF. From the graph, it is clear that Precision@10 values are slightly higher than Precision@5 across all models. Among them, the HILS model shows the highest Precision@5 value slightly above 0.37. Precision@10 is near 0.42. LSTPM and ST-LSTM have Precision@5 around 0.34. The Precision@10 values are above 0.39. DeepMove has a lower Precision@5 about 0.33. Its Precision@10 reaches around 0.38. GeoMF has the lowest Precision@5 slightly above 0.30. Its Precision@10 is around 0.35. The chart shows that models with higher Precision@5 also have higher Precision@10. This means the ranking performance is stable. The gap between Precision@5 and Precision@10 is small. Adding more recommendations slightly improves precision. HILS outperforms other models, suggesting that it provides better precision performance in this evaluation. GeoMF, on the other hand, has the lowest precision performance among the tested models. The results show that some models perform better. Models with sequential learning and spatial-temporal dependencies have an advantage. HILS and LSTPM outperform other architectures.

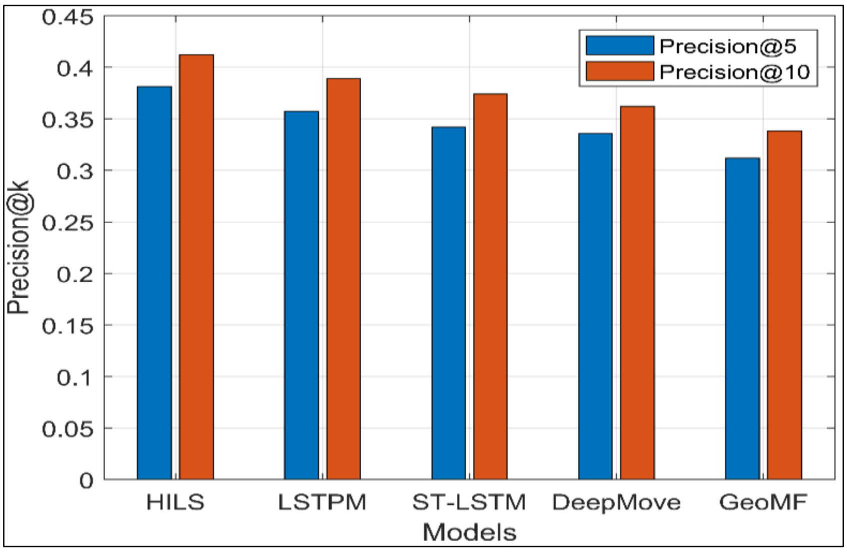


Figure 4: Precision@k versus Different Models.
Source: Authors, (2026).

The Figure 5 evaluates the impact of different modifications on the HILS model by measuring three key performance metrics: Recall@5, NDCG@5 and Precision@5. Recall@5 starts at 0.71 in the Full HILS model. It drops steadily to 0.66 in the No Attention variant. NDCG@5 follows the same trend. It decreases from 0.6 in Full HILS to 0.55 in No Attention. Precision@5 also declines. It moves from 0.39 to 0.33 as components are removed. The declining trend across all three metrics indicates that each component contributes significantly to model performance. The removal of long-term and short-term preferences leads to noticeable reductions in Recall@5 and NDCG@5. It suggests that both are crucial for maintaining effective ranking and retrieval performance. Precision@5 also decreases but at a slightly slower rate. It also shows that the model still retains some accuracy even as these components are removed. The No Attention variant performs the worst across all metrics, suggesting that attention mechanisms play a vital role in optimizing HILS. Overall, the results highlight the importance of preserving all key components to achieve the best recommendation performance.

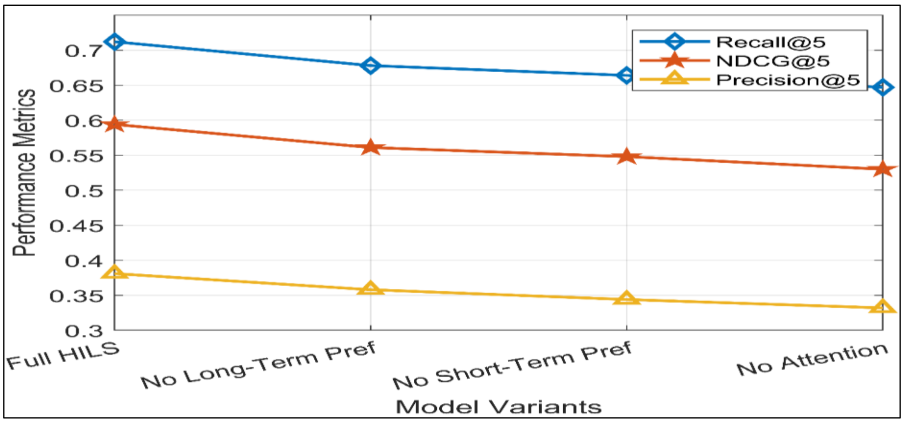


Figure 5: Ablation Study: Performance Impact of Different Components.
Source: Authors, (2026).

The Figure 6 illustrates how different modifications in the HILS model affect three performance metrics. Recall@5 starts at 0.71 in Full HILS. It gradually drops to 0.68 without geographical influence. It then slightly decreases to 0.66 without temporal influence. NDCG@5 starts near 0.6. It steadily declines to 0.56 after removing both influences. Precision@5 begins at 0.39. It drops to 0.35 when both geographical and temporal influences are excluded. The downward trend in all three metrics suggests that both geographical and temporal components play a key role in the performance of the HILS model. Removing geographical influence reduces Recall@5 and NDCG@5 highlights the importance of location-based data in improving recommendation relevance. Removing temporal influence causes a slight performance drop. This shows that time-sensitive patterns add value. The decline is smaller than in models without attention mechanisms. However, it proves that spatial and temporal factors improve recommendations.

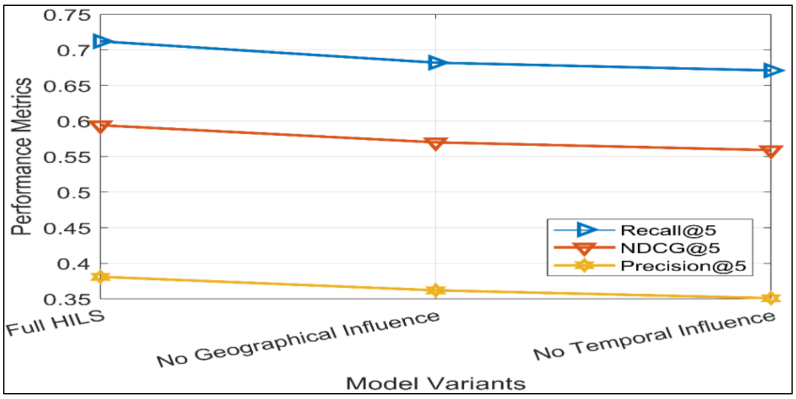


Figure 6: Geographical & Temporal Influence on Performance.
Source: Authors, (2026).

The Figure 7 compares the recall performance of four different models—HILS, HILS-N, HILS-NH and HILS-S. HILS performs better than other models. It starts at 0.7 with 5 candidate POIs. It rises steadily to 0.85 with 25 POIs. HILS-N has the second-highest recall. It starts slightly below 0.7 at 5 POIs and reaches 0.75 at 25 POIs. HILS-NH follows closely behind HILS-N. It begins just above 0.68 and increases to 0.73. HILS-S has the lowest recall among the four models. It starts at around 0.65 and increases to nearly 0.72 as the number of candidate POIs grows.The trend in the chart suggests that increasing the number of candidate POIs improves recall performance for all models. HILS has the highest recall. This shows it is better at retrieving relevant POIs. HILS-N and HILS-NH follow a similar trend. However, HILS-N performs slightly better. HILS-S has the lowest recall. This suggests it is less efficient than other models. All models improve as candidate POIs increase. However, the rate of improvement differs for each model.

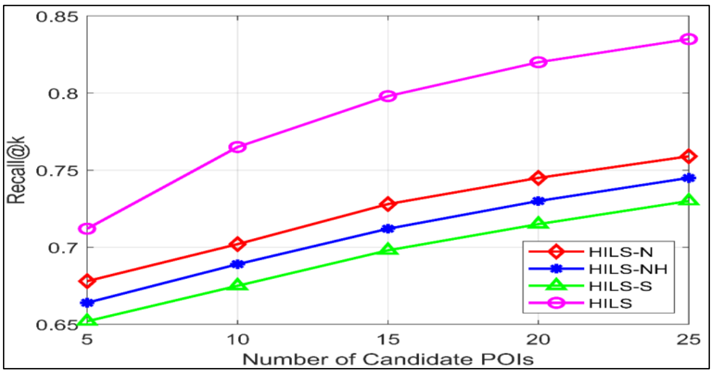


Figure 7: Effect of Increasing POI Candidates on Recall@k.
Source: Authors, (2026).

The Figure 8 presents a comparison of four model variants—HILS-N, HILS-NH, HILS-S and HILS. Recall@5 is the highest for the HILS model reaching slightly above 0.7. HILS-N, HILS-NH, and HILS-S have similar Recall@5 values. They are all near 0.7 but slightly lower than HILS. NDCG@5 follows the same pattern. HILS performs best at around 0.6. The other models have values between 0.53 and 0.57. Precision@5 shows smaller differences. It ranges from 0.33 to 0.36. HILS performs slightly better than the other models. The chart shows that the full HILS model provides the best results with all three metrics being higher than those of its variants. The differences are more significant in Recall@5 and NDCG@5, which suggest that the full HILS model is more effective in retrieving and ranking relevant results. The small variation in Precision@5 shows similar accuracy across models. All versions maintain a close level of precision. This suggests that model structure affects recall and ranking more. The results highlight the need to keep all HILS components. Keeping all parts gives the best performance in recall, ranking and precision.

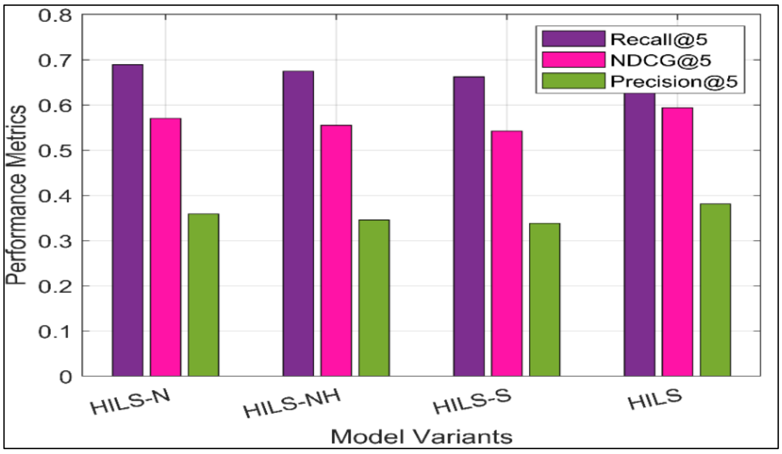


Figure 8: Comparison of Different POI Approaches
Source: Authors, (2026).

The Figure 9 compares the recall performance of two datasets, NYC and TKY based on different values of n. Both datasets show a positive trend with recall increasing as n increases. At n = 1, NYC starts with a recall value of about 0.2. TKY starts slightly higher at around 0.3. As n grows, the recall values of both datasets increase steadily. At n = 25, NYC reaches approximately 0.6. TKY is slightly above 0.7. At n = 100, NYC's recall is close to 0.9. TKY reaches nearly 1.0. The trend in the chart suggests that increasing n leads to higher recall values for both datasets. The TKY dataset performs better than NYC at every value of n. The difference between the two datasets remains fairly constant as n increases, indicating that TKY has a stronger recall performance. At the highest n value, both datasets reach maximum recall. TKY achieves 1.0, showing perfect recall. NYC follows closely behind. This means TKY's recommendations are slightly better. Both datasets improve with more candidate recommendations.

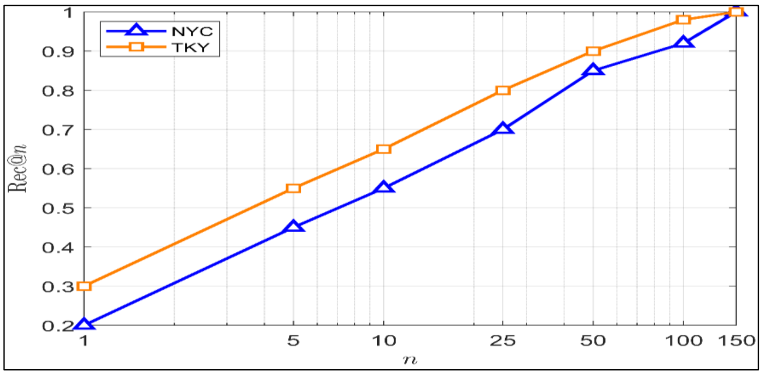


Figure 9: Effect of Number of Categories n
Source: Authors, (2026).

The Figure 10 compares the execution time of four different model variants—HILS, HILS-N, HILS-NH and HILS-S. Each model follows a similar increasing trend with execution time grows as the dataset size becomes larger. For smaller datasets like 500 and 1,000 users/POIs, execution time stays low. It remains under 10 seconds for all models. At 5,000 users/POIs, execution time rises to about 20 seconds. A sharper increase happens at 10,000 users/POIs. Execution time reaches around 50 seconds. At 20,000 users/POIs, execution time goes over 120 seconds. The full HILS model takes the longest time. The overall trend suggests that execution time scales non-linearly with dataset size. It means larger datasets require significantly more processing time. Among the models, HILS exhibits the highest execution time. It indicates that its additional computational complexity demands more resources. HILS-S also shows slightly higher execution time compared to the other variations. HILS-N and HILS-NH are more efficient. The execution times are slightly lower. All models show a similar increase in execution time as the dataset grows. However, some variations are better optimized. These models process data faster than others.

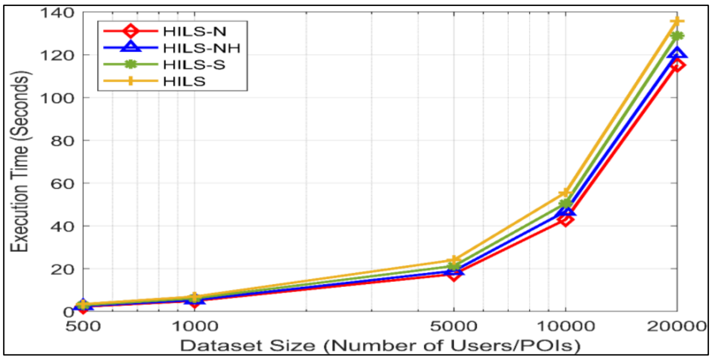


Figure 10: Model Scalability: Performance versus Dataset Size.
Source: Authors, (2026).

The Figure 11 compares the recall performance of four different model variants—HILS, HILS-N, HILS-NH and HILS-S. HILS model outperforms the other models at every embedding dimension. It starts at around 0.63 at 32 dimensions, rises to nearly 0.74 at 128 dimensions and slightly decreases at 256 dimensions. HILS-N follows a similar trend but with slightly lower values. It reaches a peak of about 0.71 at 128 dimensions before declining. HILS-NH and HILS-S both shows the same peak at 128 dimensions but with lower recall values compared to the other two models. The trend in the chart suggests that increasing embedding dimensions initially improves recall performance for all models. However, after reaching 128 dimensions recall starts to decrease slightly. Increasing embedding dimensions improves model representation. However, this works only up to a certain point. Very large embeddings add noise or redundancy. This reduce performance instead of improving it. The HILS model achieves the highest recall, demonstrating that it benefits the most from increased embedding dimensions. HILS-N and HILS-NH follow closely, while HILS-S shows the lowest recall values at all embedding sizes. These results highlight that an optimal embedding size exists for achieving the best recall and beyond this point performance decline.

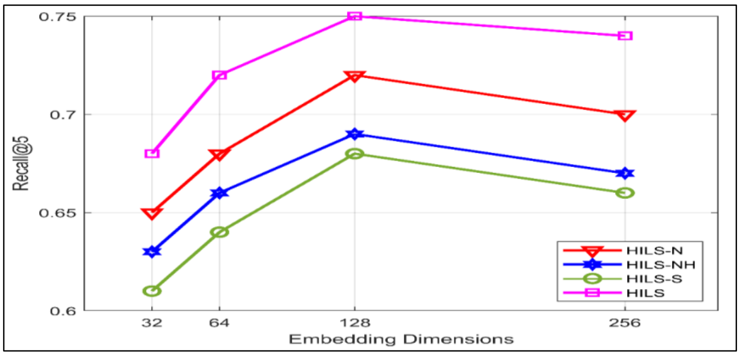


Figure 11: Effect of Different Embedding Dimensions.
Source: Authors, (2026).

The Figure 12 displays the relationship between user activity levels and three performance metrics: Recall@5, NDCG@5 and Precision@5. Recall@5 starts at approximately 0.7 for low-activity users and increases gradually to about 0.75 for high-activity users. NDCG@5 begins at around 0.55 and reaching nearly 0.6 as user activity increases. Precision@5 shows a steady rise from about 0.35 for low-activity users to approximately 0.45 for high-activity users.The pattern in the chart suggests that higher user activity leads to improved recommendation performance across all three metrics. The increase in Recall@5 and NDCG@5 implies that the model benefits from more interactions. It results in more accurate ranking and retrieval. Precision@5 also improves indicating that active users receive more relevant recommendations. While all metrics show growth, Recall@5 exhibits the most notable improvement. It reflects a stronger ability to retrieve relevant results for highly active users. These results emphasize the importance of user engagement in enhancing recommendation accuracy and ranking effectiveness.

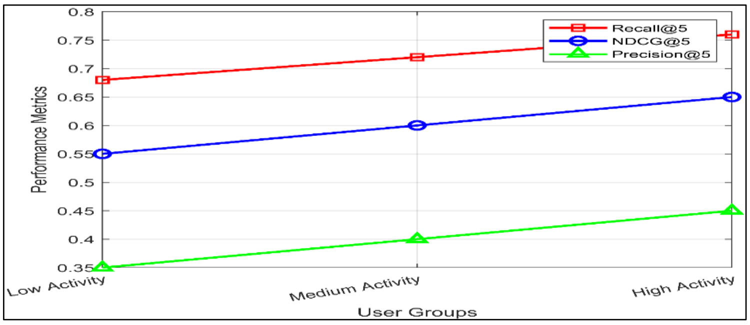


Figure 12: Model Performance across Different User Groups.
Source: Authors, (2026).

The Figure 13 displays the variation in three performance metrics—Recall@5, NDCG@5 and Precision@5. It is across different times of the day: Morning, Afternoon, Evening and Night. Recall@5 starts at 0.72 in the morning. It gradually increases through the afternoon and evening. It peaks at 0.78 in the evening. Then, it drops slightly at night. NDCG@5 follows a similar trend. It starts at 0.58 in the morning. It rises to 0.63 in the evening. After that it declines at night. Precision@5 starts at 0.38 in the morning. It increases to 0.45 in the evening. Then it drops back down at night. The trend in the chart suggests that performance improves throughout the day. It reaches its highest levels in the evening before declining at night. The increase in Recall@5 and NDCG@5 during the evening indicates that the recommendation model performs best at this time due to higher user engagement. The decline at night across all three metrics suggests that recommendations are less effective when user activity is lower. Precision@5 follows the same pattern but with a smaller overall variation. These results highlight the importance of time-based patterns in recommendation performance, showing that the model is most effective during peak activity hours.

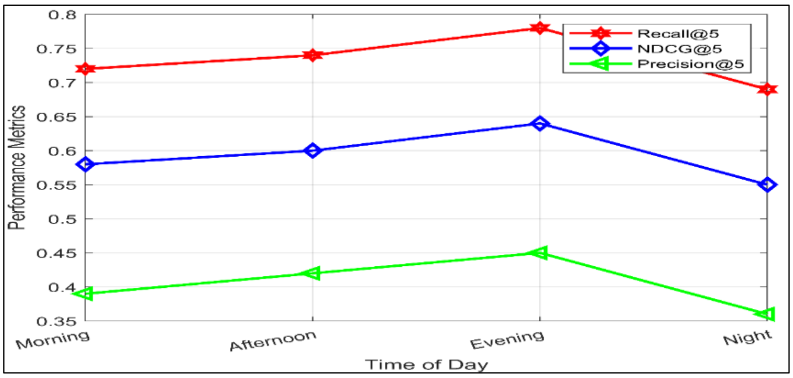


Figure 13: Effect of different Time Windows on Performance.
Source: Authors, (2026).

The Figure 14 presents the variation in three performance metrics—Recall@5, NDCG@5 and Precision@5. It is across different times of the day: morning, afternoon, evening and night. Recall@5 starts at 0.72 in the morning. It increases slightly in the afternoon. It reaches 0.78 in the evening. Then, it drops slightly at night. NDCG@5 starts at 0.58. It rises steadily to 0.63 in the evening. After that, it decreases at night. Precision@5 follows the same pattern. It increases from 0.38 in the morning to 0.45 in the evening. Then, it declines at night. The overall trend suggests that recommendation performance improves as the day progresses, peaking in the evening before declining at night. The increase in Recall@5 and NDCG@5 during the evening indicates that recommendations are most effective when user activity is higher. The decrease at night suggests that lower user engagement impact the model's ability to generate relevant recommendations. Precision@5 follows a similar pattern, though with a smaller range of variation. These results emphasize the impact of time on recommendation accuracy. The model achieves its best performance during peak user activity hours.

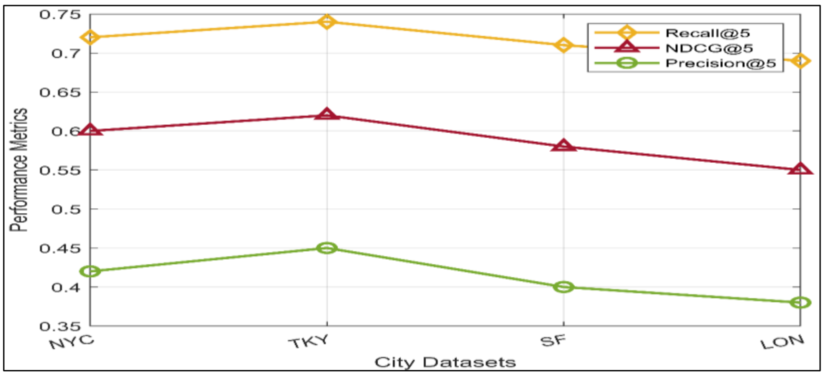


Figure 14: Performance Across different City Datasets.
Source: Authors, (2026).

The Figure 15 represents the distribution of the standard deviation of geographical distances. The data follows a bell-shaped pattern suggesting a normal distribution. The highest frequency is observed around the center of the distribution. The standard deviation values range from approximately 15 to 25. The peak frequency reaches slightly above 100, indicating that most data points fall within this range. As the standard deviation increases beyond 25 the frequency declines gradually. Fewer instances are observed beyond a value of 30. Similarly, on the lower end very few occurrences are recorded below 10. The distribution shows most geographical distances vary moderately. There are fewer cases of very small or very large variations. The data spread indicates different movement patterns. Some users move very little. Others travel much larger distances. The balanced shape of the histogram implies that extreme values are rare and the data is centered around a common range. This pattern helps in understanding user mobility. It shows how users move in different areas. It improves location-based recommendations. Targeting common movement patterns makes recommendations more effective.

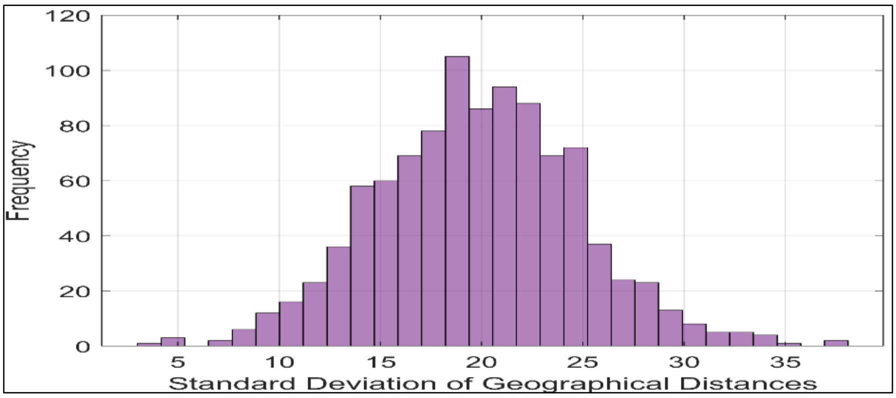


Figure 15: The Distribution of Standard Deviation.
Source: Authors, (2026).

The Figure 16 presents the number of check-ins for ten users sorted by activity level, comparing four model variants: HILS, HILS-N, HILS-NH and HILS-S. HILS model records the highest number of check-ins among all users. HILS-S follows closely maintaining a slightly lower number of check-ins. HILS-N and HILS-NH show similar trends with slightly fewer check-ins than HILS and HILS-S. The number of check-ins decreases with lower user activity. The most active users have over 350 check-ins. The least active users have fewer than 100 check-ins. The trend in the chart suggests that all models show a steady decline in check-ins as user activity decreases. The gap between HILS and the other models is more noticeable for highly active users. The differences become smaller for lower-activity users. HILS maintains a clear advantage in check-ins across all users, demonstrating its effectiveness in capturing user engagement. The similarities between HILS-N and HILS-NH suggest that both models perform similarly for users with medium to low activity. Higher user activity leads to more check-ins. Models with better learning mechanisms capture this behavior well. HILS performs better in tracking check-ins.

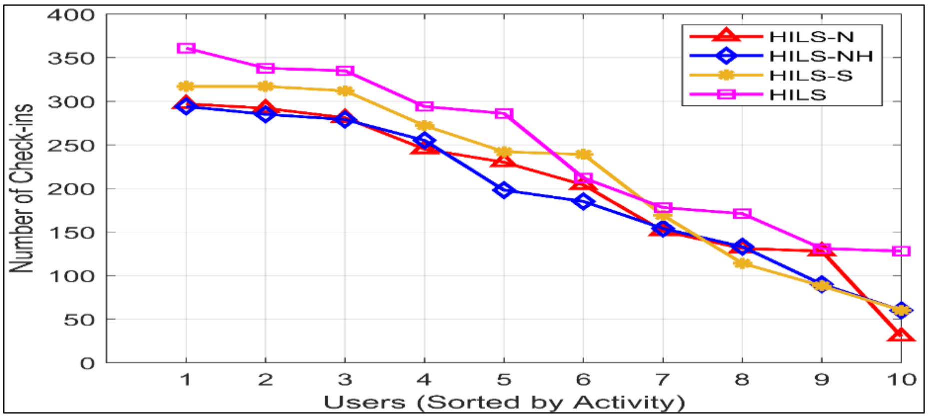


Figure 16: Distribution of Check-in Frequency across Users.
Source: Authors, (2026).

VII. CONCLUSIONS

This paper presented the HILS framework for next POI recommendation. The model captures stable long-term behaviors. It also learns dynamic short-term user preferences. Hierarchical trajectory modeling enhances learning. Semantic feature learning improves understanding. Temporal and spatial influences are integrated. The experimental results highlight the advantages of HILS over existing methods, demonstrating its superior accuracy in predicting user mobility patterns.HILS introduces a structured approach by decomposing user check-ins into POI-level and semantic-level features. The POI-level LSTM models sequential dependencies in visited locations.

The semantic-level LSTM incorporates contextual factors like time, category and location history. The dual-layer LSTM structure promotes that the model captures user intent more effectively than conventional single-layer models.The performance evaluation reveals that HILS outperforms traditionalmodels. Ablation studies explains the significance of each component in HILS. Removing long-term preference learning leads to a decrease in recall. It emphasizes the importance of capturing stable user behaviors. Eliminating short-term modeling results in lower precision, highlighting the need for recent trajectory updates. The attention mechanism gives long-term trajectory selection, improving the overall recommendation process.Geographical and temporal influences play a crucial role in refining predictions.

Users are more to visit POIs within a specific distance from the previous locations and the model incorporates this factor using a spatial proximity function. The integration of these external factors enhances the accuracy of POI ranking.HILS presents a novel approach to POI recommendation by integrating hierarchical preference modeling, attention-based trajectory selection and contextual influence learning. The experimental analysis validates its effectiveness and highlights the benefits of structured representation learning.

Future work explore incorporating additional contextual signals, optimizing hyperparameter tuning and extending the model for multi-user collaborative recommendations. The findings contribute to advancing the field of personalized mobility prediction, offering a robust solution for next POI recommendation in dynamic environments.

VIII. AUTHOR'S CONTRIBUTION

Conceptualization: Dr.Sarala Patchala, Dr.Vijay Babu Burra, Dr. Vullam Naga Gopi Raju, Banda SNV Ramana Murthy and Desamala Prabhakara Rao
Methodology: Dr. Sarala Patchala and Dr. Vijay babu Burra
Investigation: Dr. Sarala Patchala and Dr. Vijay babu Burra
Discussion of results:. Dr. Sarala Patchala and Dr. Vijay babu Burra Vulla, Nagagopiraju
Writing – Original Draft: Dr. Sarala Patchala
Writing – Review and Editing: Dr. Sarala Patchala and Dr. Vijay babu Burra
Resources: Dr. Vijay babu Burra
Supervision: Dr. Vijay babu Burra Vullam Nagagopiraju
Approval of the final text: Dr. Sarala Patchala and Dr. Vijay babu Burra Vulla, Nagagopiraju and Desamala Prabhakara Rao

IX. REFERENCES

[1] J. Chon and H. Cha, "Lifemap: A smartphone-based context provider forlocation-based services," IEEE Pervasive Computing, vol. 10, no. 2, pp. 58–67, 2011.

[2] Y. Bao, H. Wu, and X. Liu, "From prediction to action: Improving userexperience with data-driven resource allocation," IEEE Journal on SelectedAreas in Communications, vol. 35, no. 5, pp. 1062–1075, 2017.

[3] S. Vargas and P. Castells, "Exploiting the diversity of user preferences forrecommendation," in Proceedings of the 10th conference on open researchareas in information retrieval, pp. 129–136, Citeseer, 2013.

[4] L. Zhang, Z. Sun, J. Zhang, H. Kloeden, and F. Klanner, "Modeling hierarchical category transition for next poi recommendation with uncertaincheck-ins," Information Sciences, vol. 515, pp. 169–190, 2020.

[5] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: usermovement in location-based social networks," in Proceedings of the 17thACM SIGKDD international conference on Knowledge discovery and datamining, pp. 1082–1090, 2011.

[6] S. Li, W. Li, C. Cook, C. Zhu, and Y. Gao, "Independently recurrent neuralnetwork (indrnn): Building a longer and deeper rnn," in Proceedings of theIEEE conference on computer vision and pattern recognition, pp. 5457–5466, 2018.

[7] Y.-e. Hou, W. Gu, W. Dong, and L. Dang, "A deep reinforcement learningreal-time recommendation model based on long and short-term preference,"International Journal of Computational Intelligence Systems, vol. 16, no. 1,p. 4, 2023.

[8] M.-W. Chang, L. Ratinov, and D. Roth, "Structured learning with constrained conditional models," Machine learning, vol. 88, pp. 399–431, 2012.

[9] Y. Wei, X. Wang, X. He, L. Nie, Y. Rui, and T.-S. Chua, "Hierarchical userintent graph network for multimedia recommendation," IEEE Transactionson Multimedia, vol. 24, pp. 2701–2712, 2021.

[10] T. Yang, Y. Gao, Z. Huang, and Y. Liu, "Uptdnet: A user preferencetransfer and drift network for cross-city next poi recommendation," International Journal of Intelligent Systems, vol. 2023, no. 1, p. 9091570, 2023.

[11] C. Wang, M. Yuan, Y. Yang, K. Peng, and H. Jiang, "Revisiting long-andshort-term preference learning for next poi recommendation with hierarchical lstm," IEEE Transactions on Mobile Computing, 2024.

[12] B. Mohan, W. Yang, and S. kiang Chou, "Fuel injection strategies forperformance improvement and emissions reduction in compression ignitionengines—a review," Renewable and Sustainable Energy Reviews, vol. 28,pp. 664–676, 2013.

[13] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborativefiltering recommendation algorithms," in Proc. 10th Int. Conf. World WideWeb, 2001, pp. 285–295.

[14] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluatingcollaborative filtering recommender systems," ACM Trans. Inf. Syst., vol.22, no. 1, pp. 5–53, 2004.

[15] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art," IEEE Trans. Knowl.Data Eng., vol. 17, no. 6, pp. 734–749, 2005.

[16] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques forrecommender systems," IEEE Comput., vol. 42, no. 8, pp. 30–37, 2009.

[17] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," inProc. Adv. Neural Inf. Process. Syst., 2008, pp. 1257–1264.

[18] D. Lian, C. Zhao, X. Xie, G. Sun, E. Chen, and Y. Rui, "Geomf: Jointgeographical modeling and matrix factorization for point-of-interest recommendation," in Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. DataMining, 2014, pp. 831–840.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," NeuralComput., vol. 9, no. 8, pp. 1735–1780, 1997.

[20] J. Feng, X. He, J. Zhang, C. Zhang, and M. Wang, "DeepMove: Predicting human mobility with attentional recurrent networks," in Proc. ACMSIGKDD Int. Conf. Knowl. Discov. Data Mining, 2018, pp. 1459–1468.

[21] D. Kong and F. Wu, "HST-LSTM: A hierarchical spatio-temporal long-short term memory network for location prediction," in Proc. 27th Int.Joint Conf. Artif. Intell., 2018, pp. 2341–2347.

[22] K. Sun, T. Qian, T. Chen, Y. Liang, Q. V. H. Nguyen, and H. Yin, “Whereto go next: Modeling long- and short-term user preferences for point-of-interest recommendation,” in Proc. AAAI Conf. Artif. Intell., 2020, pp.214–221.

[23] Y. Wu, K. Li, G. Zhao, and X. Qian, “Long- and short-term preferencelearning for next POI recommendation,” in Proc. 28th ACM Int. Conf.Inf. Knowl. Manage., 2019, pp. 2301–2304.