ECTI Transactions on Computer and Information Technology
Journal homepage: https://ph01.tci-thaijo.org/index.php/ecticit/
Published by the ECTI Association, Thailand, ISSN: 2286-9131

Awareness of Check-In Patterns for an Adaptive Framework in Next POI Recommendation

Oraya Sooknit[1], Jakkarin Suksawatchon[2] and Ureerat Suksawatchon[3]

ABSTRACT

The recommendation of the Next Point-of-Interest (Next POI) is pivotal in the domain of location-based services, as it forecasts a user's subsequent check-in based on their historical movement patterns. Although prior researches have recognized and acknowledged the diversity in individual travel behavior, the methodologies for effectively distinguishing these patterns remain somewhat ambiguous and unclear. This particular challenge becomes more complex when users engage in check-ins at irregular locations or times, which consequently complicates the process of forecasting the Next POI. To address this issue, we aim to analyze the check-in patterns to improve the Next POI recommendation process. To address this particular concern, we analyze check-in patterns to improve the Next POI recommendation process. We propose AFNextPOI (Awareness of Check-In Patterns for an Adaptive Framework in Next POI Recommendation), which enhances check-in pattern analysis through pattern-based features. This research implements strict privacy protection measures, utilizing only anonymized check-in data to ensure no user profile information is accessible or analyzed. Experiments conducted on two real-world datasets demonstrate that the AFNextPOI framework achieves superior performance compared to state-of-the-art models in terms of Recall and NDCG metrics, thereby validating the effectiveness of our approach.

Article information:

Keywords: Next POI Recommendation, Adaptive Framework, Check-in Pattern, Routine Behavior, Non-routine Behavior, Sustainable Tourism

Article history:
Received: February 28, 2025
Revised: July 24, 2025
Accepted: October 9, 2025
Published: October 25, 2025 (Online)

DOI: 10.37936/ecti-cit.2025194.260946

1. INTRODUCTION

In the last few years, the growth of location-based social networking (LBSN) services such as Foursquare, Gowalla, and Yelp has been significant due to the increasing integration of geolocation technologies, social networking, and mobile applications [1]. These networks are driven by technological advances and increasing consumer demand for personalized POI (Point of Interest) [2] [3]. The Next POI recommendation system enhances traditional POI recommendation methodologies by emphasizing the prediction of a user's next location based on his or her route and behavioral patterns as a basis. This approach differs from traditional POI recommendation systems, which typically do not consider sequence data, temporal factors, or contextual details comprehensively. This technology can significantly improve the user experience by providing more accurate, relevant, and personalized suggestions. It is highly beneficial to businesses, local communities, and society.

Recent studies show that the Next POI recommendation systems have been continuously developed by addressing various challenges in greater depth. For instance, the data sparsity and cold start problems are addressed by using multi-source data [4] or by sharing external data with similar users [5]. However, some researches focused on other contextual information to improve the performance of recommendations [6]. Moreover, a critical challenge in Next POI recommendations lies in addressing user preference issues. Researchers employ machine learning techniques to analyse users' travel patterns as sequential data to tackle this issue.

In order to model and categorize the area of hometown check-ins, the HI-LDA model uses the area with the highest frequency of check-ins as their hometown,

[1,2,3]The authors are with the Faculty of Informatics, Burapha University, Seansuk, Mueang, Chonburi, 20131, Thailand, Email: 65810009@go.buu.ac.th, jakkarin@go.buu.ac.th and ureerat@go.buu.ac.th
[3]Corresponding author: ureerat@go.buu.ac.th

while other areas are classified as out-of-town[7]. The PR-UIDT model similarly identifies the user's hometown as the area with the highest number of check-ins, while all other regions are considered as component of the current city [8].

Consequently, certain limitations remain when users exhibit dynamic patterns that reflect hometown behaviors distinct from their out-of-town behaviors. This complication occurs because the data is partitioned solely based on latitude and longitude, which may not account for other important contexts. Temporal factors, such as the time and day of check-ins, are also essential for accurately representing user behaviors[5] [9]. Therefore, we must first correctly distinguish user behaviors to address issues specific to each behavior, including area, day of week, and time. This differentiation is crucial to avoid using a uniform decision-making process from recommending the next location when the current position whether routine or non-routine check-in is indeterminate.

For example, considering the case in Fig. 1, users' historical check-ins have been clustered by area utilizing using the DBSCAN algorithm to identify clusters based on data density [7]. Concurrently, the Isolation Forest can identify anomalies based on these three salient features: area, day, and time of check-in records. It is easily observable that this user engages in routine check-ins at the office, bar, and bus station, characterized by specific days, hours, and areas (areas 0 and 1). However, there are also instances of non-routine check-ins at the bar on varying days, as well as at other great outdoors and stadium categories, which are not part of his regular activities (areas 1, 2, and 3). This observation shows that routine check-ins can manifest across diverse areas, while non-routine check-ins can occur within regions previously associated with routine patterns.

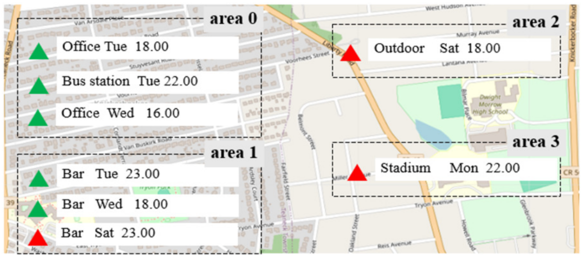


Fig.1: *Classifying Patterns of Check-In Points.*

We observed that the routine pattern is not limited to specific check-in locations but also includes the days, times, and areas frequently visited by users. Although several researches have considered the user preference issue, the model characterizes both long-term and short-term user preferences [5] [10] [11] and [12], as well as the user's awareness of their routine regularity. In addition, many studies have addressed the problem by distinguishing between areas with frequent check-ins and areas outside the user's regular travel zone [7] [13] [14] to differentiate the patterns as well. However, separating the check-in points by each format leads to different outcomes and allows for observable comparisons as shown in Table 1.

To address this issue, we introduce AFNextPOI (Awareness of Check-In Patterns for an Adaptive Framework in Next POI Recommendation), which emphasizes diverse patterns in user trajectories based on area and time frequency. These main contributions are summarized as follows:

- We propose an AFNextPOI for the Next POI recommendation that focuses on identifying two dis-

Table 1: *Comparing the Methods of Dividing Check-In Points.*

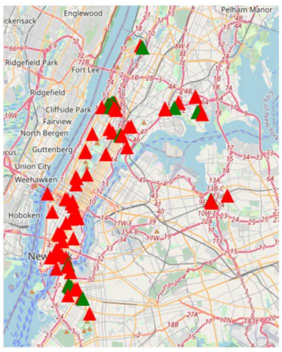
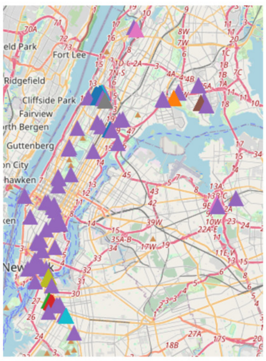
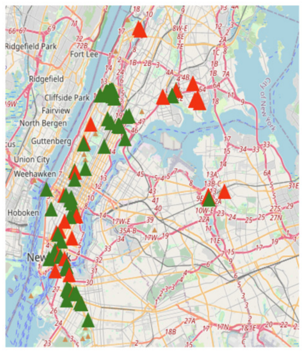
Frequency	DBSCAN	Isolation Forest
		
▲ hoemtown ▲ out-of-town	▲▲▲▲▲▲▲▲▲▲ cluster	▲ routine ▲ non-routine
Data: latitude, longitude	Data: latitude, longitude	Data: area, time, day
Note: The area with the highest check-in frequency, but at different times.	Note: A cluster that is not visited frequently but is repeatedly traveled to.	Note: A user who stays at home only on weekends. Although the check-ins are few, it is a behavior they consistently follow.

***Table 2:** Survey Related Work as Next POI Recommendation.*

Details		Problems			Methodology					Feature					
Year	Acronym	Data sparsity	Context info.	User preference	Markov chain	RNN	LSTM	Graph	Etc.	location	time	day	category	area	pattern
2019	HI-LDA [7]	✓		✓					DBSCAN	✓	✓				
2019	PR-UIDT [8]	✓		✓					Drift and transfer	✓	✓			✓	
2020	LSTPM [11]			✓			✓			✓	✓				
2021	HOPE [13]			✓			✓			✓					
2022	PLSPL [10]			✓			✓			✓			✓		
2022	RTPM [5]			✓			✓			✓	✓	✓	✓		
2023	UPTDNet[14]	✓		✓		✓				✓	✓				
2023	MARAN [12]			✓				✓	NNS	✓	✓			✓	
ours	AFNextPOI	✓	✓	✓			✓		Isolation forest	✓	✓	✓	✓	✓	✓

tinct patterns—routine and non-routine—by using Isolation Forest for anomaly detection and employing separate processes to determine the best next POI for each pattern.

- We design an adaptive network for routine patterns with location preferences based on specific places the user frequently visits, and for non-routine patterns with category preferences to recommend options that match the user's lifestyle.
- We experimented on two real-world datasets with state-of-the-art models in the Next POI field.

2. RELATED WORK

2.1 Next POI Recommendation

The Next POI recommendation has gained significant attention in recent years due to its potential applications in personalized location-based services. The goal is to predict a user's next destination based on historical movement patterns, preferences, and contextual factors such as time and location. Researchers have explored various approaches to address this problem, including data sparsity, cold-start issue, contextual information, and user preferences. The Next POI prediction relies on sequential behavioural patterns, so researchers develop models that can capture temporal dependencies. These models include Markov models, recurrent neural networks (RNNs), long short-term memory networks (LSTMs), gated recurrent units (GRUs), and graph-based approaches. The research uses a Markov model to predict users' next favourite POIs based not only on their current location but also on their previous locations [15]. The UPTDNet model uses an RNN-based model to capture the sequential effects of check-in records. Additionally, the LSTM model is widely used to address user preferences with sequence data [5] [10] [11] [13] [16].

2.2 User Preference problems

Regarding user preferences for next POI prediction, several models address the complexity of temporal preference patterns. These models, namely LSTPM [11], PLSPL[10], and RTPM [5], detect variations in user patterns by partitioning the training data into two distinct components: long-term and short-term preferences. These models employ LSTM networks to learn preferences over extended periods and recent timeframes.

However, preference patterns exhibit inherent complexity across temporal scales. Even within long-term preferences, users exhibit short-term behavioral fluctuations. Similarly, short-term preferences may still contain underlying long-term behavioral tendencies. Research on the HOPE model [13] shows that users exhibit dynamic preferences when engaging in out-of-town behaviours. The HOPE model employs a hybrid approach to address these two distinct user behavior types. Furthermore, the MARAN model [12] considers users' routine regularity alongside short-term preference changes. The AWSBP model [16] focuses on differences in user patterns, separating them by frequency and dynamics. These approaches demonstrate the need for sophisticated pattern analysis to capture the complexity of user mobility behavior.

These models address both temporal and spatial preference variations: long-term and short-term preferences (separated by time period), as well as hometown and out-of-town locations (separated by geographical area). However, data sparsity and incompleteness present significant challenges for learning accurate user preferences. Incorporating additional contextual information can help address these limitations and improve recommendation effectiveness. Specifically, variations in patterns across different areas at the same time and day highlight the importance of identifying when users most frequently remain in specific areas. Therefore, analysing user patterns requires approaches that go beyond simple segmentation by area or time alone. Table 2 presents a comprehensive survey of related work.

3. PRELIMINARY

3.1 Preparing data

Let $U = \{u_1, u_2, \ldots, u_n\}$ denote a set of users and the vector p^u consisting of $[l, c, h, d]$ followed by user check-ins. The summary of all the notations in this paper is listed below in Table 3.

Table 3: *Notation and Description.*

Notation	Description
U	set of users
l	set of [latitude, longitude]
c	set of categories
h	set of hours [0 - 23]
d	day of week [0 - 6]
a	set of area for user check-ins data
P	set of check-in points $[l, c, h, d]$
p^u	set of check-in points of user $[l, c, h, d]$
T^u_n	set of trajectories by the user
t^u_n	set of users' trajectories
s^t_n	set of trajectory sequences
hs^u_n	hidden state
pt^u_n	pattern of check-ins data $[p, a, c]$
cd^u_n	The candidate's points of user
top_a	The top 3 popular areas of user
top_c	The top 3 popular categories of users

3.2 Definition

We generate a new feature for the classification pattern of check-in points, denoted as pt, in a three-dimensional vector $[p, a, c]$, which relates to the user's pattern, area of user, and category. We define two types of p as follows.

Routine pattern (R) represents repetitive user activity patterns, often involving similar areas and time slots. For instance, such patterns may include commuting to work at a consistent time each morning, returning home at the same time in the evening, visiting a regular holiday destination, engaging in evening exercise routines, and shopping at the supermarket every Saturday.

Non-routine pattern (N) reflects irregular, unexpected, or rare actions, such as visiting a hospital, attending a concert, or exploring a new area. These events may occur in areas previously visited but at unusual times, or in entirely new areas.

In Fig.2, this user usually follows a routine pattern on Saturdays in area 0. If they engage in activities elsewhere on a Sunday — such as going to a drug store, stadium, dinner, tearoom, bowling alley, or bar — it is considered a non-routine pattern.

To effectively distinguish between consistent and irregular user behaviors in check-in data, we de-

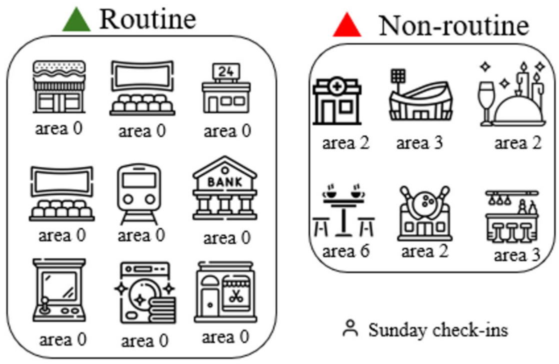


Fig.2: *The Majority Of Sunday Activities Occur In Area 0.*

fine routine and non-routine patterns based on the concept of anomaly detection. In this work, we employ the Isolation Forest algorithm [17, 18] to model these differences, considering non-routine patterns as anomalies with respect to a user's temporal and spatial check-in history. We analysed all check-in points using the *Isolation Forest* algorithm implemented with the *Scikit-learn library (contamination=0.1, random state=42)* based on three features: a, h, and d. We represent each check-in point as a three-dimensional temporal–spatial vector and train an *Isolation Forest* (IF) model to assign an anomaly score to each check-in point. We define output of patterns as equations (1) and (2).

Routine Pattern (R): a check-in point with anomaly score $p_i = 1$.

$$R = \{p_i \in p_u | if(p_i) = +1\} \tag{1}$$

Non-routine Pattern (N): a check-in point flagged as anomalous, with score $p_i = -1$.

$$N = \{p_i \in p_u | if(p_i) = -1\} \tag{2}$$

Binary classification results indicate routine patterns (value = 1) versus non-routine patterns (value = -1). We determine spatial area assignment using DBSCAN clustering performed on location coordinates l = [latitude, longitude] derived from user check-in data. Fig.3 presents the percentage distribution of routine and non-routine check-in points across both datasets. The integrated approach detailed in Section 4.2.2 employs DBSCAN clustering algorithm for spatial analysis combined with temporal pattern classification to simultaneously identify user behavioral patterns and popular geographical regions.

4. METHODOLOGY

Fig.4 illustrates the overview of our proposed model, AFNextPOI. We preprocess data to remove noise and reduce redundancy for accurate processing. Section 5.1, *Data Description*, provides the details. Data are set in the *Embedding Layer* and directed as

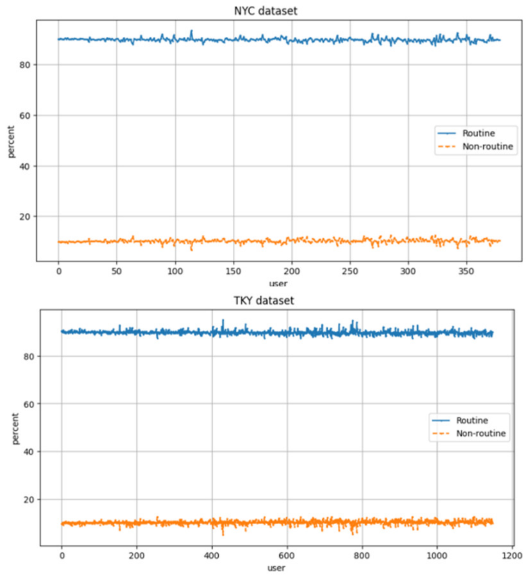


Fig.3: *Percentage Of Patterns Of Check-Ins Point.*

input to the *Preference Layer*, followed by additional processing in the Adaptive Layer. Finally, the proposed model outputs the top@K Next POIs in the *Recommendation Layer*.

4.1 Embedding Layer

This section is dedicated to preparing sequence data for LSTM models. We use c, which represents the user's check-in category, to learn the user's POI category preferences based on their sequential check-in data. In Fig.5, we then ordered the check-in points by trajectory, defining new trajectories whenever the time gap between consecutive check-in points exceeded 24 hours. Next, we transformed each trajectory into fixed-length sequences of 5 points and constructed subsequent sequences by shifting one point forward within the trajectory [16].

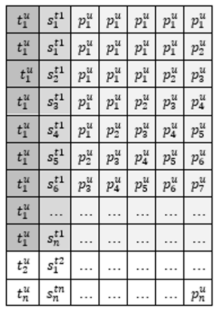


Fig.5: *Preparing Sequence to LSTM Model.*

4.2 Preference Layer

4.2.1 LSTM model

In this layer, the LSTM model [19] inputs the user history check-ins sequences by category index c, with p_c^u to trajectory $T^u \in \{t_1^u, t_2^u, \ldots, t_n^u\}$ and subsequence $s_i^t \in \{s_1^t, s_2^t, \ldots, s_n^t\}$ to the LSTM model and output the hidden state hs_i^u as equations (3)-(5).

$$hs_i^u = o_i \odot \tanh(C_i) \quad (3)$$

$$C_i = f_i \odot C_{i-1} + ip_i \odot \tilde{C}_l \quad (4)$$

$$o_i = \sigma(W_o \cdot [hs_{i-1}^u, s_i] + b_o) \quad (5)$$

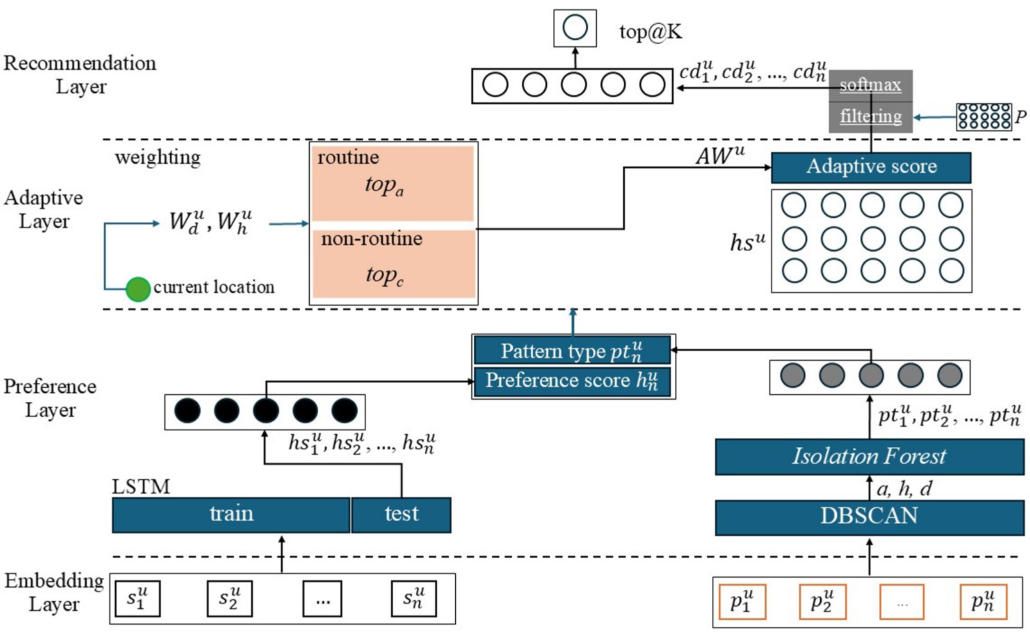


Fig.4: *Our Proposed AFNextPOI Framework.*

$$hs_i^u \in hs_1^u, hs_2^u, \ldots, hs_n^u$$
$$s_i \in s_1^t, s_2^t, \ldots, s_n^t$$
$$t \in t_1^u, t_2^u, \ldots, t_n^u$$

Where hs_i^u is the hidden state at time step (this is the model's output each time i. O_i is output gate activation at time i, C_i is element-wise multiplication, and tanh is the hyperbolic tangent activation function. f_i is forget gate activation, ip_i is input gate activation and $\tilde{C}_l$is the candidate cell state.

The σ is sigmoid activation function, W_0 refer to weight matrix for the output gate, b_0 is bias term for the output gate, hs_{i-1}^u is hidden state from the previous time step and s_i is input at the current time step i.

We applied the best early stop, or a maximum of 200 epochs were run on each person's model using a Python-based integrated computational framework built on TensorFlow/Keras.

4.2.2 Isolation Forest Detection

We cluster the area of users by using DBSCAN (Density-Based Spatial Clustering of Applications with Noise) from vector $p^u[l]$ related to latitude and longitude. DBSCAN clusters points that are closely packed together and marks outliers as noise. The result in equation (6).

$$a = DBSCAN(p^u[l]) \tag{6}$$

$$p^u \in p_1^u, p_2^u, \ldots, p_n^u$$

The Isolation Forest algorithm detects anomalies by isolating them instead of profiling normal data. Equation (7) shows the Isolation Forest result from 3 features of the user check-in point $p^u[a, h, d]$. The Anomaly Score 1 is "routine", score -1 is "non-routine" pattern.

$$pt_i^u = IsolationForest(p^u[a_i, h_i, d_i]) \tag{7}$$

4.3 Adaptive Layer

We calculate the scores of all candidate points in the test set by focusing on the relationship between the current location and each candidate's next location. First, we are finding a pattern of the current check-ins point is routine or non-routine pattern from pt_n^u. Then, we calculated the candidates scores based on different objectives. For the first objective, routine patterns, we considered user preferences for routine activities, focusing on temporal proximity, distance, and personal location preferences. For the second objective, non-routine patterns, we used popular preferences from personal category check-in history instead of POI preferences to recommend the next POI.

4.3.1 High-performing models

We randomly select a current location from the check-in points in the test set, and the ground truth for the next POI consists of the subsequent points following the current location. There can be more than one ground truth next POI. To ensure the availability of ground truth for next POI prediction, we exclude the final check-in in each test sequence from sampling as the current location during evaluation.

4.3.2 Weighting

In this task, we are learning a weight from the current location by distance W_d^u and hour W_h^u [16].

W_d^u is determined by calculating the distance d between two points using the *Haversine Formular* [16] [20]and exponentially for each d from all points. Let $\alpha = -1$ represent the adjustment factor for higher nearby location scores as equation(8).

$$W_{d_i}^u = \frac{\exp(a * d_i)}{\sum_{j=1}^{|hs^u|} \exp(a * d_i)} \tag{8}$$

W_h^u is determined by calculating the time interval between the current time $current_h$ and the next time of the candidate's point h by computing the absolute difference using equation (9).

$$W_{h_i}^u = \frac{\exp(a * |current_h - h_i|)}{\sum_{j=1}^{|hs^u|} \exp(a * |current_h - h_i|)} \tag{9}$$

Finally, we incorporate each historical point preference $\{hs_1^u, hs_2^u, \ldots, hs_n^u\}$ into the weighting of all check-in points, as formulated in equation (10).

$$AW_i^u = hs_i^u + W_{d_i}^u + W_{h_i}^u \tag{10}$$

4.3.3 Routine and Non-routine Framework

The top_a relates to the top 3 most frequently checked-in locations of a user, as described in Algorithm 1. The top_c relates to the top 3 most frequently checked-in categories of a user as described in Algorithm 2.

Algorithm 1: Most popular location of the user

Require: A column of data $pt^u = [pt_1^u a, pt_2^u a, ..., pt_n^u a]$
Ensure: counts value a of user to top_a

```
Initialize empty dictionary C ← {}
for each u ∈ U do
    for each pt_i^u ∈ pt_n^u do
        if pt_i^u a ∈ C then
            C[pt_i^u] ← C[pt_i^u] + 1 {Increment count}
        else
            C[pt_i^u] ← 1 {Initialize count}
        end if
    end for
    Extract sorted unique values A ← keys of C[: 3]
    Extract sorted counts V ← values of C[: 3]
    top_a^u ← (A, V)[: 3]
end for
return top_a
```

Algorithm 2: Most popular category of the user

Require: A column of data $pt^u = [pt_1^u c, pt_2^u c, ..., pt_n^u c]$
Ensure: counts value c of user to top_c
Initialize empty dictionary $C \leftarrow \{\}$
for each $u \in U$ **do**
  for each $pt_i^u \in pt_n^u$ **do**
    if $pt_i^u c \in C$ **then**
      $C[\mathrm{pt}_i^u] \leftarrow C[\mathrm{pt}_i^u] + 1$ {Increment count}
    else
      $C[\mathrm{pt}_i^u] \leftarrow 1$ {Initialize count}
    end if
  end for
  Extract sorted unique values $A \leftarrow$ keys of $C[:3]$
  Extract sorted counts $V \leftarrow$ values of $C[:3]$
  $top_c^u \leftarrow (A, V)[:3]$
end for
return top_c

Based on an analysis of the user's travel routes, we applied DBSCAN on the user's latitude and longitude data to form spatial clusters. We then counted the number of check-ins in each cluster and selected the top 3 clusters with the highest proportions of total check-ins. After that, we calculate top_a from user locations, top_c from the user's categories, and apply Algorithm 3 to adjust the score based on different types of patterns.

Algorithm 3: Adaptive personal preference

Require: p, pt, AW, top_a, top_c
Ensure: AW is computed
for each $u \in U$ **do**
  for $i \in p^u$ **do**
    if $pt_i^u p ==$ "routine" **then**
      if $p_i^u l \in top_a$ **then**
        $AW_i^u \leftarrow AW_i^u * *2$
      end if
    else
      if $p_i^u c \in top_c$ **then**
        $AW_i^u \leftarrow AW_i^u * *2$
      end if
    end if
  end for
end for

4.4 Recommendation Layer

In this phase, we enhance the candidate POI set by incorporating popular locations derived from other users' behavioural patterns while eliminating irrelevant candidates. We then compute the probability distribution across all candidate POIs using the SoftMax function.

Filtering: We selected other locations within a 3.5 km radius from the average next location, ensuring they matched the user's favourite categories at that time from $p_i^u \cap P$. We calculated AW_i^u based on the user's max score. Otherwise, filtering-in points were based on the user's average score. Additionally, we incorporated the location index from p^u, maintaining the same length to align with real locations. We assign the lowest score to candidate locations that are more than 10 km away to exclude long-distance points.

SoftMax: The SoftMax function [21] outputs probabilities that sum up to 1 in equation (11). The variable e is exponentiates each element to ensure positive values. After that, we sorted the candidates cdu by highest rank in the final step.

$$\text{Softmax}(AW_i^u) = \frac{e^{AW_i^u - \max(AW^u)}}{\sum_{j=1}^{n} e^{AW_i^u - \max(AW^u)}} \qquad (11)$$

5. EXPERIMENT

5.1 Data Description

We evaluated our model's performance using two real-world datasets: Foursquare check-ins in New York (NYC) and Tokyo (TKY), collected between April 12, 2012, and February 16, 2013. This dataset was initially employed to investigate the spatial-temporal regularities of user activity in LSBNs [22]. The preprocessing follows the same approach as the comparison model, including the removal of duplicate and noisy data [5, 11]. We handled duplicate check-in data by identifying users with entries at the same location, on the same day, and within the same hour, ensuring the removal of redundant information. We excluded unpopular locations with fewer than 10 check-ins. Additionally, we removed trajectories with fewer than 10 check-ins each and eliminated users with fewer than three trajectories. Next, we organized the data by user and allocated the first 80% of users from each dataset to the training set, reserving the remaining 20% for the testing set. The dataset descriptions are in Table 4.

Table 4: *Dataset Description.*

Dataset	Feature	Original	After Prep.
NYC	♯user	1,083	379
	♯category	398	236
	♯check-ins	227,428	67,547
TKY	♯user	2,293	1,148
	♯category	385	225
	♯check-ins	573,703	228,852

5.2 Evaluation

In this section, we assessed accuracy using Recall@K and NDCG@K metrics. Recall@K measures the model's ability to identify correct POIs within the top K recommendations compared to ground truth locations, while NDCG@K evaluates ranking quality by assessing the correct ordering of POIs within the top K results. We set K = {1, 3, 5}, and defined the evaluation metrics in equation (12).

$$R@K = \frac{1}{|U|}\sum\nolimits_{u=1}^{|U|} R_u@k\frac{r_u \cap t_u}{|t_u|} \tag{12}$$

r_u denotes the set of top K POIs for user u, and t_u represents the set of ground truth POIs chosen by the user for the next location in the test set.

In equations (13) and (14), $IDCG$ represents the maximum possible *Discounted Cumulative Gain (DCG)* for a given recommendation list. The variable rel_i is set to 1 if the POI at position i in the recommendation list has been visited and 0 otherwise. N denotes the number of correctly recommended POIs.

$$NDCG@K = \frac{1}{|U|}\sum\nolimits_{u=1}^{|U|} \frac{DCG_u@k}{IDCG_u@k} \tag{13}$$

$$DCG@k = \sum\nolimits_{i=1}^{N} \frac{2^{rel_i - 1}}{\log_2(i+1)} \tag{14}$$

5.3 Comparison with Baseline Algorithms

We compared our proposed framework, AFNextPOI, with several baseline methods used for Next POI recommendation. To prevent errors when developing other research, we directly used the performance results of two datasets that appeared in the following works.

LSTM [23] a Python-based integrated computational framework built on TensorFlow/Keras, designed for sequential data, similar to our model.

GRU [24] a Python-based integrated computational framework built on TensorFlow/Keras, designed for sequential data, similar to our model.

LSTPM [5] [11] model addresses the limitations of existing POI recommendation models by incorporating both long-term and short-term user preferences.

RTPM [5] analyses the long-term preference, the short-term preference, real-time analysis, the probability calculation, and the recommendation module.

Table 5 compares AFNextPOI with baseline models using Recall and NDCG metrics at top@K [1, 3, 5].

Table 5: *Performance Comparison with Baseline.*

Dataset	NYC					TKY				
Metric	R@1	R@3	R@5	N@3	N@5	R@1	R@3	R@5	N@3	N@5
LSTM	0.1161	0.2621	0.3730	0.1891	0.2337	0.1080	0.2480	0.3514	0.1837	0.2255
GRU	0.0739	0.1974	0.2983	0.1462	0.1814	0.0601	0.1828	0.2891	0.1262	0.1694
LSTPM	0.1836	0.3087	0.3707	0.2559	0.2814	0.2088	0.3492	0.4135	0.2902	0.3168
RTPM	0.1944	0.3182	0.3752	0.2663	0.2898	0.2143	0.3504	0.4151	0.2934	0.3201
AFNextPOI	*0.2137*	*0.3851*	*0.5011*	*0.2989*	*0.3453*	0.2352	*0.4033*	0.4786	0.3228	0.3524
Improvement	*1.93%*	*6.69%*	*12.59%*	*3.26%*	*5.55%*	*2.09%*	*5.29%*	*6.35%*	*2.94%*	*3.23%*

Table 6: *Comparing Current Location with Routine and Non-Routine.*

Dataset	NYC (R:339+N:40 =379)					TKY (R:988+N:160 =1,148)				
Metric	R@1	R@3	R@5	N@3	N@5	R@1	R@3	R@5	N@3	N@5
AFNextPOI-R	0.2094	0.3795	0.4988	0.2929	0.3407	0.2318	0.4096	0.4864	0.3252	0.3554
AFNextPOI-N	0.2500	0.4315	0.5208	0.3495	0.3848	0.2562	0.3646	0.4308	0.3078	0.3338
AFNextPOI	*0.2137*	*0.3851*	*0.5011*	*0.2989*	*0.3453*	*0.2352*	*0.4033*	*0.4786*	*0.3228*	*0.3524*

Note: R = recall, N = NDCG

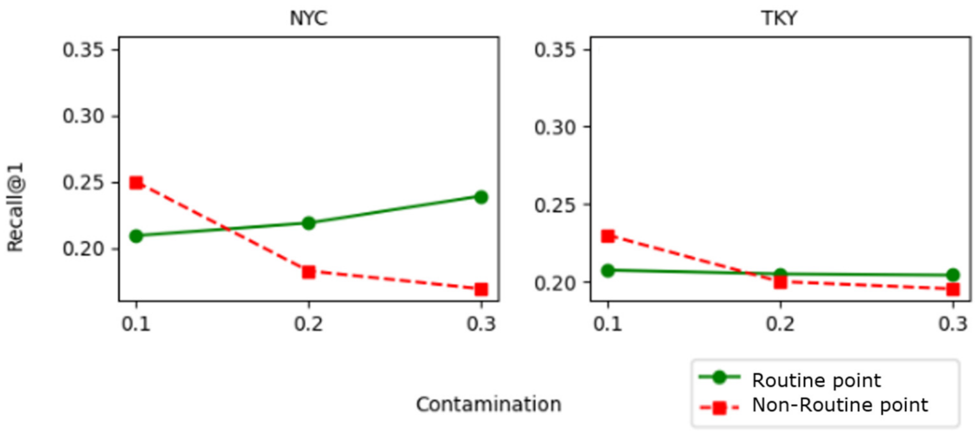


Fig.6: *Evaluate Using Routine and Non-Routine Location with Different Contamination Levels at Recall@1.*

Table 7: *Average Matric Per Run with Recall.*

Time		*1st*	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
NYC	R@1	*0.2137*	0.2032	0.2137	0.2164	0.2111	0.2005	0.2005	0.2375	0.2111	0.2084
	R@3	*0.3851*	0.3707	0.3641	0.3804	0.3865	0.3698	0.3527	0.3907	0.4182	0.3685
	R@5	*0.5011*	0.4901	0.4846	0.4876	0.4959	0.4689	0.4612	0.4967	0.4847	0.4838
TKY	R@1	*0.2352*	0.2239	0.2352	0.2117	0.2334	0.2073	0.2282	0.2186	0.2282	0.2056
	R@3	*0.4033*	0.3823	0.4129	0.4014	0.3997	0.3931	0.4149	0.4011	0.3889	0.3749
	R@5	*0.4786*	0.4679	0.4814	0.4848	0.4803	0.4732	0.4804	0.4719	0.4585	0.4507

The results show that AFNextPOI outperforms the baseline models on both datasets (NYC and TKY). In addition, our model outperforms the best competitor, RTPM, on two datasets in terms of Recall and NDCG metrics. These results indicate that the proposed approach enhances the performance of Next POI recommendation.

5.4 Discussion

In addition to the comparative analysis of the performance of the proposed framework, we identified the optimal performance metric Recall@1 for non-routine points at a contamination level of 0.1, derived from the configurations of the isolation forest, as illustrated in Fig. 6. It is evident that the non-routine points exhibit superior performance across two graphical representations at a contamination level of 0.1. The comparative results pertaining to the contamination level of 0.1 between routine points (AFNextPOI-R) and non-routine points (AFNextPOI-N) are detailed in Table 6. We employed the same current location used in the comprehensive model and categorized users based on the pattern type associated with their current location. The model demonstrates commendable efficacy on non-routine patterns in terms of Recall@1, which studies identify as the most dependable metric. To substantiate the validity of our findings, we computed the average and present the results in Table 7. These findings are in alignment with our experimental trials, which we conducted repetitively 10 times, utilizing randomly selected locations. The experiments were executed with data obtained from 379 users in the NYC dataset and 1,148 users in the TKY dataset, respectively.

6. CONCLUSION AND FUTURE WORK

In this research, we propose AFNextPOI (Awareness of Check-In Patterns for an Adaptive Framework in Next POI Recommendation). Our approach consists of several key components. First, we analyze check-in point patterns using Isolation Forest to identify anomalous behavior. Second, we classify users' check-in points into geographical areas using DBSCAN clustering. Third, we leverage these pattern-based features to adaptively score candidate POIs for recommendation. The check-in patterns are learned from user behavior, incorporating three dimensions, including check-in locations, temporal patterns (days), and time-of-day preferences. Integrating newly identified features that capture user behaviour through the combination of spatial patterns and area classifications significantly enhances the effectiveness of Next POI recommendation. For user preference learning, we structure the input sequences for the LSTM model to capture temporal dependencies effectively. Our experiments demonstrate that the adaptive framework successfully handles diverse check-in patterns across different user types. Overall, the proposed AFNextPOI framework achieves superior recommendation performance compared to existing state-of-the-art methods.

A significant limitation of this study is the incompleteness of user check-in data. Since users did not consistently and reliably log their precise geographical locations over time, the dataset contains gaps that may hinder the model's ability to fully learn mobility patterns. This incompleteness may affect both the accuracy and generalizability of the results. Future work should consider more continuous and comprehensive data collection methods, such as GPS tracking or the integration of data from multiple diverse sources, to significantly enhance the quality of data and improve the overall performance of the predictive model.

In future studies, we aim to investigate how non-routine patterns affect next POI predictions, to develop improvements that enhance recommendation accuracy across all user scenarios, including irregular and spontaneous movement patterns.

ACKNOWLEDGEMENT

The authors would like to express their gratitude to the Thailand scholarship through Rajamangala University of Technology Srivijaya for providing the Ph.D. Scholarship to O.S. Additionally, this study was supported by the Graduate School, Burapha University.

AUTHOR CONTRIBUTIONS

Conceptualization, O.S. and J.S.; methodology, O.S.; software, O.S.; validation, O.S., J.S., and U.S.; formal analysis, O.S.; experiment, O.S. and U.S.; discussion, O.S. and U.S.; visualization, O.S.; writing—original draft preparation, O.S.; writing—review and editing, O.S., J.S., and U.S.; supervision, J.S. and U.S.; funding acquisition, O.S., J.S.,

and U.S. All authors have read and agreed to the published version of the manuscript. Corresponding author: U.S. (email: ureerat@go.buu.ac.th).

References

[1] M. Rizwan, W. Wan and L. Gwiazdzinski, "Visualization, spatiotemporal patterns, and directional analysis of urban activities using geolocation data extracted from LBSN," *ISPRS International Journal of Geo-Information*, vol. 9, no. 2, p. 137, 2020.

[2] T. Qian, B. Liu, Q. V. H. Nguyen and H. Yin, "Spatiotemporal representation learning for translation-based POI recommendation," *ACM Transactions on Information Systems (TOIS)*, vol. 37, no. 2, pp. 1-24, 2019.

[3] X. Wang, Y. Liu, X. Zhou, X. Wang and Z. Leng, "A point-of-interest recommendation method exploiting sequential, category and geographical influence," *ISPRS International Journal of Geo-Information*, vol. 11, no. 2, p. 80, 2022.

[4] L. Sun, "POI recommendation method based on multi-source information fusion using deep learning in location-based social networks," *Journal of Information Processing Systems*, vol. 17, no. 2, pp. 352-368, 2021.

[5] X. Liu, Y. Yang, Y. Xu, F. Yang, Q. Huang and H. Wang, "Real-time POI recommendation via modeling long-and short-term user preferences," *Neurocomputing*, vol. 467, pp. 454-464, 2022.

[6] S. M. Rahimi, B. Far and X. Wang, "Contextual location recommendation for location-based social networks by learning user intentions and contextual triggers," *GeoInformatica*, vol. 26, no. 1, pp. 1-28, 2022.

[7] X. Xiong *et al.*, "Where to go: An effective point-of-interest recommendation framework for heterogeneous social networks," *Neurocomputing*, vol. 373, pp. 56-69, 2020.

[8] J. Ding, G. Yu, Y. Li, D. Jin and H. Gao, "Learning from hometown and current city: Cross-city POI recommendation via interest drift and transfer learning," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 4, pp. 1-28, 2019.

[9] X. Wang, X. Liu, L. Li, X. Chen, J. Liu and H. Wu, "Time-aware User Modeling with Check-in Time Prediction for Next POI Recommendation," *2021 IEEE International Conference on Web Services (ICWS)*, Chicago, IL, USA, pp. 125-134, 2021.

[10] Y. Wu, K. Li, G. Zhao and X. Qian, "Personalized Long- and Short-term Preference Learning for Next POI Recommendation," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 4, pp. 1944-1957, 1 April 2022.

[11] K. Sun, T. Qian, T. Chen, Y. Liang, Q. V. H. Nguyen and H. Yin, "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, 2020.

[12] X. Sun, B. Huang, X. Wang and D. Yu, "MARAN: Supporting awareness of users' routines and preferences for next POI recommendation based on spatial aggregation," *Expert Systems with Applications*, vol. 238, p. 121961, 2024.

[13] H. Sun, J. Xu, R. Zhou, W. Chen, L. Zhao and C. Liu, "HOPE: a hybrid deep neural model for out-of-town next POI recommendation," *World Wide Web*, vol. 24, no. 5, pp. 1749-1768, 2021.

[14] T. Yang, Y. Gao, Z. Huang and Y. Liu, "UPTD-Net: A User Preference Transfer and Drift Network for Cross-City Next POI Recommendation," *International Journal of Intelligent Systems*, vol. 2023, no. 1, p. 9091570, 2023.

[15] S. Liu and L. Wang, "A self-adaptive point-of-interest recommendation algorithm based on a multi-order Markov model," *Future Generation Computer Systems*, vol. 89, pp. 506-514, 2018.

[16] O. Sooknit, J. Suksawatchon and U. Suksawatchon, "Next Point of Interest Recommendation Using Adaptive Weights for Specific Behavioral Patterns," *2024 28th International Computer Science and Engineering Conference (ICSEC)*, Khon Kaen, Thailand, pp. 1-6, 2024.

[17] D. Xu, Y. Wang, Y. Meng and Z. Zhang, "An Improved Data Anomaly Detection Method Based on Isolation Forest," *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, Hangzhou, China, pp. 287-291, 2017.

[18] F. T. Liu, K. M. Ting and Z. -H. Zhou, "Isolation Forest," *2008 Eighth IEEE International Conference on Data Mining*, Pisa, Italy, pp. 413-422, 2008.

[19] S. Hochreiter, "Long Short-term Memory," *Neural Computation MIT-Press*, 1997.

[20] N. Panigrahi, *Computing in geographic information systems.* CRC Press, 2014.

[21] K. Banerjee, R. R. Gupta, K. Vyas and B. Mishra, "Exploring alternatives to softmax function," *arXiv preprint arXiv:2011.11538*, 2020.

[22] D. Yang, D. Zhang, V. W. Zheng and Z. Yu, "Modeling User Activity Preference by Leveraging User Spatial Temporal Characteristics in LBSNs," in *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 1, pp. 129-142, Jan. 2015.

[23] H. N. Bhandari, B. Rimal, N. R. Pokhrel, R. Rimal and K. R. Dahal, "LSTM-SDM: An integrated framework of LSTM implementation for sequential data modeling," *Software Impacts*, vol. 14, p. 100396, 2022.

[24] Z. Zhang, C. Li, Z. Wu, A. Sun, D. Ye and X. Luo, "Next: a neural network framework for next poi recommendation," *Frontiers of Computer Science*, vol. 14, pp. 314-333, 2020.

Oraya Sooknit received the B.B.A. degree in Information Systems in 2004 and the M.Sc. degree in Management Information Technology in 2010. She is currently pursuing the Ph.D. degree with the Faculty of Informatics, Burapha University, Thailand. Her research interests include recommender systems and machine learning.

Jakkarin Suksawatchon received a B.Sc. degree in Computer Science from Burapha University, Chonburi, Thailand, in 1998, and M.Sc. degree in Information Technology from King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand, in 2000. Then he received Ph.D degree in Computer Science from Chulalongkorn University, Thailand, in 2007. Currently, he has been a faculty member of the Faculty of Informatics, Burapha University, Chonburi, Thailand, since 2002. His current research interests include web and mobile application development, pattern recognition, machine learning, and data analytics.

Ureerat Suksawatcho received a B.Sc. degree in Mathematics from Kasetsart University, Bangkok, Thailand, in 1997. Then, she received M.Sc. and Ph.D degree in Computer Science from Chulalongkorn University, Thailand, in 2001 and 2009, respectively. Dr. Ureerat Suksawatchon has been a faculty member of the Faculty of Informatics, Burapha University, Chonburi, Thailand, since 2007. Her current research interests include pattern recognition, machine learning, and data analytics.