LSTM-Based Destination Prediction Method Using Location Information and Temporal Characteristics

Ji Hwan Jeong
Information & Communication Engineering
Chungbuk National University
Cheongju 28644, Republic of Korea
starwlghks7@chungbuk.ac.kr

Won Seok Choi
Research Institude for Computer and Information Communication
Chungbuk National University
Cheongju 28644, Republic of Korea
choiws@cbnu.ac.kr

Hyeon Muk Park
Information & Communication Engineering
Chungbuk National University
Cheongju 28644, Republic of Korea
qga0809@chungbuk.ac.kr

Seong Gon Choi*
Information & Communication Engineering
Chungbuk National University
Cheongju 28644, Republic of Korea
choisg@cbnu.ac.kr

***Abstract*— To reflect personal characteristics in destination recommendations, we propose an LSTM based user destination prediction method that uses the temporal characteristics of user location information. For this purpose, we introduce preprocessing methods for time series GPS data and an LSTM based architecture for user destination prediction. The preprocessing method utilizes GPS records of user location and date related features, applying one-hot encoding and min–max normalization to prepare the inputs for the prediction model. For validation, the proposed pipeline was implemented in Python, and its performance was assessed using the F1-score metric.**

Keywords—Location Information, User Destination Prediction, Temporal Characteristics, Personal Characteristics, LSTM

I. Introduction

With the rapid advancement of autonomous vehicle technologies, there is increasing demand for traffic safety and transportation efficiency. In this context, path planning including obstacle avoidance, trajectory generation, and real-time optimization—constitutes a fundamental component of autonomous driving systems [1][2].

Path planning research is generally categorized into traditional probabilistic approaches and deep learning–based sequential modeling.

Petzold et al. [3] conducted an objective comparison of next-location prediction techniques for indoor location-based services—including dynamic Bayesian networks, multilayer perceptrons, Elman networks, Markov predictors, and state predictors—on a common benchmark. However, they performed an analysis of pure spatial patterns using no temporal context (such as timestamps, weekdays, or holidays) and did not consider temporal variations in location information.

Wu Hao et al. [4] proposed CSSRNN (Constrained State Space RNN) and LPIRNN (Latent Prediction Information RNN) to overcome the limitations of traditional Markov IRL methods, thereby improving the accuracy of predicting transition probabilities between successive trajectory segments in urban road networks. However, these approaches focus on modeling general trajectory patterns across the entire user population, making them unsuitable for predicting trajectories that reflect personal characteristics.

Zhao et al. [5] proposed the TALL model—integrating bidirectional LSTM with an attention mechanism—and the H-TALL model, which applies multi-resolution hierarchical learning, to enhance fine-grained destination prediction based on segmented trajectory intervals. However, these models do not use any temporal features, making them unsuitable for daily location trajectory prediction where time sensitivity is critical [6].

Therefore, we propose a user destination prediction method that uses user location information and temporal characteristics to solve the challenges of temporal changes in location information, trajectory prediction reflecting personal characteristics, and daily location trajectory prediction.

The rest of this paper is organized as follows. Section 2 reviews related work on path planning. Section 3 presents the LSTM-based user destination prediction pipeline. Section 4 details the proposed model's training configuration and experimental setup. Section 5 discusses the results.

II. Related Works

Research on path planning is generally categorized into traditional probabilistic approaches [3][7][8] and deep learning–based sequence modeling [4][5].

First, traditional probabilistic approaches include Markov chain–based location prediction, dynamic Bayesian networks, comprehensive probabilistic models, and the LeZi-Update framework, which applies the LZ78 compression scheme for mobility tracking. Early studies on trajectory prediction were based on Markov assumptions and probabilistic graphical models. Petzold et al. [3] modeled indoor location sequences with a dynamic Bayesian network to predict the next room based solely on past visitation history, whereas Li et al. [7] proposed a fully probabilistic framework that considers all possible paths inferred from brief GPS snippets to reconstruct continuous trajectories. Bhattacharya and Das [8] proposed an adaptive online location update algorithm called LeZi-update based on the Lempel–Ziv (LZ78) compression principle in a PCS network environment and presented a method to simultaneously minimize location update cost and paging cost through path-based update and prediction-based paging. However, these probabilistic models focus solely on aggregate patterns across the entire user population and thus fail to capture location information reflecting personal characteristics.

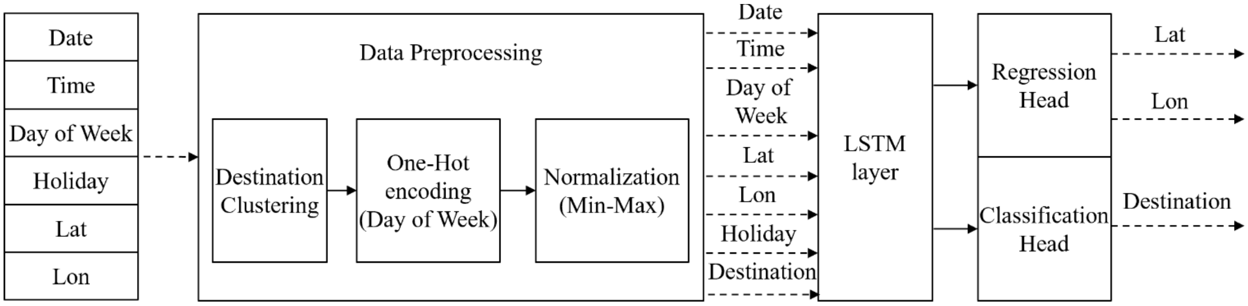

Fig. 1. Proposed LSTM based user destination prediction pipeline

Second, deep learning–based sequence analysis has been extended to trajectory prediction, with various RNN- and LSTM-based models having been proposed. Wu et al. [4] effectively modeled large-scale trajectory datasets by proposing CSSRNN (Constrained State Space RNN), which enforces urban road network topology via a masked softmax, and LPIRNN (Latent Prediction Information RNN), which

jointly learns hidden states and employs lightweight classifiers to predict subsequent transitions. Zhao et al. [5] proposed a TALL model combining bidirectional LSTM and attention mechanisms to predict destinations from sub-

trajectories of user movement paths, and a hierarchical H-TALL model that integrates movement patterns of multiple spatial resolutions and showed that partial path information contributes to improving destination prediction accuracy.

However, these models primarily focus on learning general patterns across the entire user population or omit temporal features (e.g., timestamps, day-of-week, holidays), thus failing to adequately capture individual routine and non-routine mobility patterns. We propose an LSTM-based user destination prediction pipeline that incorporates personal characteristics and temporal sensitivity to overcome the limitations in capturing individual mobility characteristics and routine pattern variations.

III. PROPOSED USER DESTINATION PREDICTION METHOD

This section describes the LSTM-based user destination prediction pipeline and prediction method that uses the temporal characteristics of individual location information.

Figure 1 illustrates the proposed LSTM-based user destination prediction pipeline. The pipeline consists of data input stage, data preprocessing, LSTM layer, a multi-head layer, and Prediction.

In the data input stage, input consists of GPS data combined with temporal characteristics, including date, time, day of the week, holiday indicator, latitude, and longitude.

Data preprocessing is the stage of transforming input data into a form that LSTMs can easily learn before passing it to the LSTM layer. This stage consists of destination clustering, one-hot encoding, and normalization. Destination clustering is the process of generating destinations from input data by examining latitude and longitude values. When repeating intervals of the same latitude and longitude are identified, a new destination ID is assigned to each latitude and longitude pair. Once all data is clustered, pairs assigned destination IDs are compared. If they fall within a certain distance, they are determined to be the same destination and their IDs are combined. The purpose of integrating the destination ID is to prevent excessive fragmentation of destinations. One-hot encoding is the process of converting day-of-week information into a 7-dimensional vector, making it easier for LSTMs to learn. Normalization is the process of converting numerical features with different ranges, such as time (hour + minute), latitude, longitude, holiday, and destination ID, to the range [0, 1] through min-max scaling. Input data that has undergone destination clustering, one-hot encoding, and normalization is then passed to the LSTM layer.

The LSTM layer is a stage of learning data that has undergone data preprocessing. This stage processes a 7-dimensional input vector containing date, time, day of the week, holidays, latitude, longitude, and destination ID in a single step, outputting the final hidden state as a 64-dimensional feature vector. This vector, which compresses temporal and spatial patterns, is then passed to multiple heads.

The multi-head stage uses the 64-dimensional feature vector output by the LSTM to output latitude, longitude, and destination ID. This stage splits the 64-dimensional hidden representation of the LSTM into two dense ranches, which perform different prediction tasks in parallel. The Regression head maps the 64-dimensional input to a 2-dimensional output to predict continuous latitude and longitude values. The Classification head converts the same 64-dimensional input to a 1-dimensional output and then applies a rounding operation to determine the final destination ID as an integer.

IV. IMPLEMENTATION

In this section, we describe the experimental environment and prediction performance of the LSTM-based user destination prediction.

A. Experimental Environment

TABLE I. EXPERIMENTAL ENVIRONMENT

GPU	NVIDIA GeForce RTX 3060 GPU (12GB)
CPU	Intel Core™ i7-10700 CPU (2.90GHz)
Memory	32GB RAM
Software	Python 3.10 PyTorch 1.12 Scikit-learn 1.1
data	Total 1,048,575 samples (Training:Validation = 90:10)

Table 1 presents the experimental environment used to validate the proposed LSTM-based user destination prediction pipeline. The hardware configuration included an NVIDIA GeForce RTX 3060 GPU (12 GB), an Intel Core™ i7-10700 CPU (2.90 GHz), and 32 GB of RAM. The software environment comprised Python 3.10, PyTorch 1.12, and Scikit-learn 1.1. A total of 1,048,575 samples were used, with a 90:10 split between the training and validation sets. The LSTM training configuration was set with a batch size of 64 and 30 epochs.

B. Performance Evaluation

Figure 2 depicts the precision, recall, and F1 scores by destination ID. IDs 1 and 4 attain perfect precision (1.00) and sustain high recall rates of 0.93 and 0.84, respectively, yielding F1 scores of 0.96 and 0.92. In contrast, ID 2 demonstrated an almost perfect recall of 0.99 but a relatively low precision of 0.75, revealing a tendency to misclassify instances not belonging to ID 2 as ID 2. ID 3 achieves a precision of 0.96 and a recall of 0.91, indicating a well-balanced predictive capability that clearly delineates its relative strengths and weaknesses among the destination IDs.

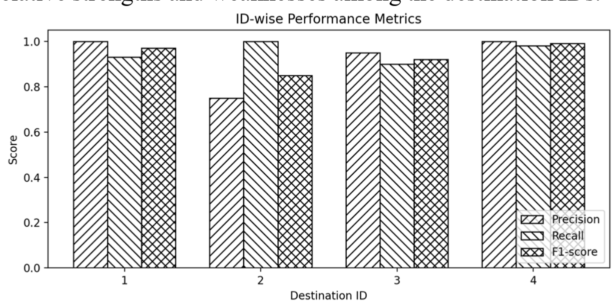

Fig. 2. Comparison of Precision, Recall and F1-score by Destination ID.

```
Accuracy: 0.9332
Macro F1-score: 0.9165
              precision    recall  f1-score   support

           1       1.00      0.93      0.96    570175
           2       0.75      0.99      0.85    172640
           3       0.96      0.91      0.93    249600
           4       1.00      0.84      0.92     56160

    accuracy                           0.93   1048575
   macro avg       0.92      0.92      0.92   1048575
weighted avg       0.95      0.93      0.94   1048575
```

Fig. 3. Summary of classification report for the entire dataset

Figure 3 summarizes the classification report for the entire dataset. The overall accuracy is 93.32%, and the macro F1 score is 91.65%, demonstrating consistently balanced performance across all destination IDs despite the imbalance in sample support. The weighted average F1 score (94.0%) exceeds the macro-average F1 score (92.0%) because the substantial support of IDs 1 and 3 influenced the overall performance. Support refers to the actual number of samples for each ID, and when computing the weighted F1-score, each ID's F1-score is multiplied by its support to determine its contribution to the overall metric; thus, IDs with larger sample counts have a greater impact on the aggregated performance.

These results indicate that the proposed LSTM model achieves robust performance across diverse destination IDs.

V. CONCLUSION

We propose an LSTM-based user destination prediction method that uses the temporal characteristics of user location information. The proposed user destination prediction method uses an LSTM-based end-to-end framework to capture temporal, day-of-week, and holiday patterns from synthetic GPS data, while simultaneously performing latitude-longitude regression and destination classification, achieving 93.3% accuracy and a 91.7% macro F1 score. In particular, latitude-longitude normalization ensures that the predicted values remain within the range of actual location information. These results suggest that destination setting functions that reflect individual characteristics of autonomous vehicle control systems can be implemented realistically in real-world environments.

Future work will involve applying the model to public datasets to validate its generalization in real-world environments, integrating dynamic clustering methods, and conducting sensitivity analyses through noise optimization.

ACKNOWLEDGMENT

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. RS-2024-00397979, Development of tera-level ultra-precision transmission network system technology, 50%) and by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. RS-2020-NR049604, 50%).

*Corresponding author: S.G. Choi(choisg@cbnu.ac.kr)

REFERENCES

[1] Hamidaoui, M., Talhaoui, M. Z., Li, M., Midoun, M. A., Haouassi, S., Mekkaoui, D. E., Smaili, A., Cherraf, A., & Benyoub, F. Z. "Survey of autonomous vehicles' collision avoidance algorithms." *Sensors (Basel, Switzerland)* 25.2 (2025): 395.

[2] Tao, F., Ding, Z., Fu, Z., Li, M., & Ji, B. "Efficient path planning for autonomous vehicles based on RRT* with variable probability strategy and artificial potential field approach." *Scientific Reports* 14.1 (2024): 24698.

[3] Petzold, J., Bagci, F., Trumler, W., & Ungerer, T. "Comparison of different methods for next location prediction." European conference on parallel processing. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.

[4] Wu, H., Chen, Z., Sun, W., Zheng, B., & Wang, W. "Modeling trajectories with recurrent neural networks." IJCAI, 2017.

[5] Zhao, J., Xu, J., Zhou, R., Zhao, P., Liu, C., & Zhu, F. "On prediction of user destination by sub-trajectory understanding: A deep learning based approach." Proceedings of the 27th ACM international conference on information and knowledge management. 2018.

[6] Chekol, Ayele Gobezie, and Marta Sintayehu Fufa. "A survey on next location prediction techniques, applications, and challenges." EURASIP Journal on Wireless Communications and Networking 2022.1 (2022): 29.

[7] Li, Mu, Amr Ahmed, and Alexander J. Smola. "Inferring movement trajectories from GPS snippets." Proceedings of the Eighth ACM International Conference on Web Search and Data Mining. 2015.

[8] Bhattacharya, Amiya, and Sajal K. Das. "Lezi-update: An information-theoretic framework for personal mobility tracking in pcs networks." *Wireless Networks* 8.2 (2002): 121-135.