Rev. Roum. Sci. Techn.– Électrotechn. et Énerg.
Vol. **70**, *1*, pp. 115–120, Bucarest, 2025

NEXT LOCATION PREDICTION VIA DEEP LEARNING SQUEEZE AND EXCITATION BIDIRECTIONAL GATED RECURRENT UNIT

UMA NATARAJAN [1], ANITHA RAMACHANDRAN [2]

Keywords: Deep learning; Next location prediction; Squeeze and excitation; Accuracy.

Next location prediction has recently attracted much attention from researchers due to its application in various domains. Many variables usually affect moving objects, including time, distance, and user configuration. This makes it difficult to predict where moving items will go when these factors are considered. This research proposes a deep learning-based next-location prediction network (DL-NLocP) to increase the accuracy of next-location prediction. Initially, the datasets are pre-processed to enhance the data quality and employ term frequency-inverse document frequency (TF-IDF) with glove word embedding approaches to convert the textual data into real-valued vectors. Afterward, multi-head CNN extracts the vector data's temporal, location, and user behavior features. Finally, squeeze and excitation with the BiGRU network are developed to predict the following location in each trajectory with contextual information. The proposed DL-NLPN model was tested on the Ningbo AIS and Geolife dataset, and experimental results supported the model's validity. The proposed model consistently outperforms current state-of-the-art approaches by 93.75% for Geolife and 94.75% for Ningbo AIS on average accuracy@20. The results show that the proposed approach performs better in Next location prediction than the existing methods.

1. INTRODUCTION

Mobile devices have produced vast amounts of spatial trajectory data showing moving objects' movement due to the quick advancement of location tracking and mobile networking technologies [1]. The implementation of numerous cutting-edge applications is made possible by the capacity to forecast the future location of mobile items [2]. Real-time transportation systems, food delivery, taxi services, and advertising are a few examples of the numerous location-based services we use daily [3]. In these applications, next-location prediction is crucial. Predicting a website visitor's proximate location has garnered much scholarly attention recently [4] due to its potential significance for businesses and consumers.

Typically, the upcoming position of a user is predicted by looking at their location history and seeing if specific patterns occur [5]. Nowadays, path planning, location sharing, location query, and other location-based services are the key offerings. Position prediction uses the user's historical trajectory data to forecast their next position with the highest likelihood [6]. Next location prediction, which involves determining a user's likely following location based on a previous GPS trace, is crucial for many mobile computing applications [7]. For instance, predicting the locations of various users is essential in ride-sharing systems to identify user groups with nearby destinations [8].

Neural networks have been effectively applied to sequence modeling and next-location prediction in several application domains [9]. A deep learning strategy should be suggested to forecast the following location while considering the topology between locations. Moreover, the sparsity problem is resolved by applying embedding techniques [10]. Though generic locational area prediction is helpful, people typically do not need highly accurate locations. Therefore, it might not be enough for noisy text from social media. This paper proposes a deep learning-based next-location prediction network (DL-NLPN).

The research outline is organized as follows: Section 2 includes relevant work. Section 3 explains the proposed design. Section 4 discusses the experiment’s results and discussion. Section 5 covers future work and the conclusion.

2. LITERATURE SURVEY

Several decades have been spent on location prediction, although most research has concentrated on user location prediction. The corresponding location prediction research f will be introduced in the next section.

In 2019, Sassi A. *et al.* [11] introduced a deep neural network-based machine learning model that uses convolutional neural networks to improve input data representation through embedding techniques. They created a position embedding technique known as loc2vec to raise the quality of the input position representation. Additionally, they demonstrate that the loc2vec-CNN model outperforms other models when transfer learning is added.

Zhang R. *et al.* [12] presented a Multi-task Model for Location prediction in 2019. The sequential and temporal properties between the locations of the moving objects are recovered by LSTM, a method proposed after CNN has extracted the spatial features. The suggested model outperforms the prior techniques based on accuracy, recall, precision, and f1-score, according to tests conducted on real datasets.

Qian T. *et al.* (2020) [13] proposed an internal, external trajectory dependency in a cooperative attention-based position prediction network called CABIN. The suggested approach outperforms existing RNN-based techniques regarding efficiency and prediction accuracy, as shown by experimental findings on real-world datasets. According to experimental results, the proposed technique surpasses the latest methods.

Zhang X. *et al.* [14] suggested a Semantic and Attention Spatio-temporal Recurrent method (SASRM) in 2020 to predict the location of anything. To input semantic vectors into the model, the SASRM first proposed a way to encode them and concatenate vectors. The suggested method performs with better accuracy than several cutting-edge models.

To predict position, Wang S. *et al.* (2021) [15] presented the spatial-temporal self-attention network (STSAN), which combines self-attention and spatial-temporal data. To find the dynamic trajectory representation of users, they developed a trajectory attention module in STSAN. The

[1] Department of Information Technology, Sri Venkateswara College of Engineering, Sriperumbudur, Tamil Nadu 602117, India
[2] Department of Computer Science and Engineering, Sri Venkateswara College of Engineering, Sriperumbudur, Tamil Nadu 602117, India
E-mails: ranitha@svce.ac.in, numa@svce.ac.in

DOI: 10.59277/RRST-EE.2025.1.20

studies indicate that spatial-temporal information can significantly enhance the method's functionality. Using the New York City dataset as a baseline, our technique STSAN achieves improvements of approximately 39.8% Acc@1 and 4.4% APR [16,17].

3. **PROBLEM DEFINITION**

This section summarizes the main concepts discussed in this article and explains some key ideas for the following debate. The study aims to predict a user's following location using trajectory data.

Definition 1 (Trajectory). A trajectory can be defined as the chronological order of a user's locations during a given duration. Using user U as an example, let's say that $T\,T^U = \{w_1^U, w_2^U, w_3^U, \dots, w_n^U\}$ is each record w_a^x with three attributes (U, k_a, t_a), where U is the user ID, t_a the timestamp, and k_a and the place that user U visited time t_a.

Definition 2 (POI). A point of interest (POI) is a location in a coordinate system that has both global position information and location identification (v) (latitude and longitude coordinates).

Definition 3 (Semantic). The semantic ξ is a term used to describe why a person visits a stay point. The semantic ξ is found by aligning the stay point with the k-nearest POI. Types of POI make up ξ's dimension. Each consecutive stay point will search for each k-nearest POI as semantic $S_p = \{S_p^1, S_p^2, S_p^3, \dots, S_p^x\}$.

Goal (Next location prediction). Given user U's trajectory T, our objective is to use the trajectory data to learn the next location k_{n+1} .

4. **PROPOSED METHODOLOGY**

In this section, DL-NLPN is proposed. The proposed method is categorized into four phases: 1) Data preprocessing, 2) Word embedding, 3) Feature extraction, and 4) location prediction. Initially, the datasets are pre-processed, and TF-IDF with Glove word embedding approaches are employed to convert the textual data into real-valued vectors. Afterward, Multihead CNN extracts temporal, location, and user behavior features from the vector data. Finally, squeeze and excitation with the BiGRU network are developed to predict the following location in each trajectory with contextual information. The overall workflow is shown in Fig 1.

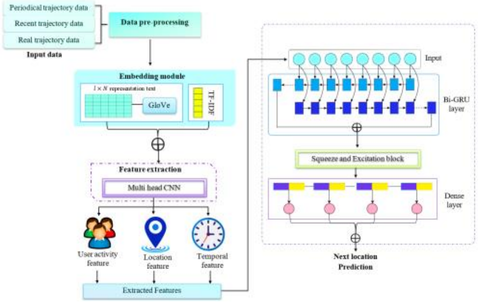


Fig. 1 – Overall block of DL-NLPN methodology.

4.1. DATA PREPROCESSING

This step is essential because it converts the trajectory data into a consistent format, which helps the classifiers perform better. Preprocessing is done on the data to eliminate inconsistent, noisy, and incomplete data. All punctuation and spaces are eliminated from the review text during the preparation stage. The following tasks are part of data preprocessing.

Tokenization. Tokens are blocks of characters used in text data. After being divided into tokens, the papers are utilized for additional processing.

Data Cleaning. Analysis can proceed to obtain an appropriate text review from the retrieved data. The best results are obtained by cleaning the crawled data by removing all special characters (like: ":/.,'#$*^&-"). The crawled content should be cleaned and copied into a CSV file.

4.2. WORD EMBEDDING

TF-IDF. Word embedding weighted averages have been shown to enhance unsupervised natural language processing (NLP) tasks, particularly next-location performance,

$$S_{l,t} = T_f \times \log\left(\frac{Q}{l_f}\right). \quad (1)$$

The term's weight value t is represented by S the document l denoting the number of records, and it is defined as Q, and l_f, T_f indicating the term frequency.

Global vectors (GloVe). Word2vec extracts embeddings from text documents. By pre-training the GloVe word embedding model using 1.2 million vocabulary terms, 27 billion tokens, and 2 billion tweets, a 200-dimensional vector word matrix was produced. The proposed model was evaluated using TF-IDF and GloVe models.

Word vectors of l dimension are created for each input k-word expressed as $T\ (t_1, t_1, \dots, t_n)$. The dimensions space of each word will therefore be, G^l. Each input text is then denoted as, $G^{n\times l}$, and the input text matrix generation is denoted by $T = (t_1, t_1, \dots, t_n) \in G^{n\times l}$ After concatenating $\oplus$ with word embedding, the features vector k_v for the document is displayed as follows:

$$k_v = s_1 \oplus s_2 \oplus s_3 \dots \oplus s_{x-1} \oplus. \quad (2)$$

The proposed approach combines TF-IDF weighing with pre-trained Glove word embedding to enhance text representation,

$$Y_z = S_{l,t} \times k_v, \quad (3)$$

where the document's TF-IDF weighting is represented by, $S_{l,t}$ and the word vector matrix, k_v is one that Glove acquired.

4.3. FEATURE EXTRACTION

A multi-head convolutional neural network (MH-CNN) is utilized to extract the temporal and location-based features of the user after vector representation. The architectural layout of the proposed method is depicted in Fig. 2. The definition of a convolution is as follows:

$$Q = \sum_{p=1}^{L} \sum_{q=1}^{W} k(p,q)g(p,q), \quad (4)$$

where Q represents the output of $k(p,q)$ and $g(p,q)$. With length L and width W, respectively, as the filters. Utilizing one-dimensional convolutions, temporal characteristics from trajectory data are processed. Let us assume that the input signal ρ of the multivariate time series data is represented by the function $\rho \in \mathbb{Z}^{E\times R}$, where E and R denote the time steps and size of features set, respectively. Next, the following expression can be used to represent the b^{th} feature map at the l^{th} layer of the h^{th} head of the multi-head CNN:

$$\rho_{l,b}^{p} = \left(t_{l,b} + \sum_{p=1}^{R} \sum_{q=1}^{g_s} z_{l,b}^{pq}\, \rho_{(l-1)p}^{a+q,h}\right),\ \forall_h = 1,2,3, \qquad (5)$$

where $t_{l,b}$ denotes the feature map's bias, k^h represents the kernel size, h^{th} head, $z_{l,b}^{pq}$ and denotes the matrix weight at layer l , whereas ρ stands for feature maps index at $(l-1)^{th}$ layer. On the other hand, in convolution networks, this component affects the quantity of attributes. Equation (6) determines how many attributes are present in each layer of a conventional MHCNN,

$$G = g_c * g_s * v + \text{bias}, \qquad (6)$$

where g_c denotes the number of filters, g_s the filter's size, and v the final dimension of the resultant vectors obtained from the layer that came before it. A nonlinear function is applied δ to $\rho_{l,b}^{a}$ following each convolutional layer to lessen the network's vanishing explosion/gradient challenge,

$$X_b^h = \delta(\rho_{l,b}^{a}), \qquad (7)$$

where the ReLU activation function is denoted by δ, the feature map X_b^h is passed to the h^{th} head's batch normalizing layer $\Im$,

$$X_{\mathbb{N}}^{h} = \Im(X_b^h). \qquad (8)$$

Following transmission to the convolutional layer in the second convolutional block, the batch normalization layer's output yields the feature set F, which can be written as follows:

$$\mathrm{F}_{l,b}^{h} = \delta\left(t_{l,b} + z_{g_s}^{h} \times X_{\mathbb{N}}^{h}\right), \qquad (9)$$

where t is the h^{th} filter, which has a kernel of the model's size $1 \times g_s$. To decrease the size of the output map $\mathrm{F}_{l,b}^{h}$. The learned features $\mathrm{F}_{l,b}$ of the h^{th} head, are then flattened into a single long vector $\mathbb{Q}$. A vector $\mathcal{H}$ is created by concatenating the feature vector $\mathbb{Q}$ of each h^{th}head,

$$\mathcal{H} = \mathbb{C}(\mathbb{Q}_a^{h_a}), \qquad (10)$$

where the concatenation function is represented by $\mathbb{C}$. The retrieved features are loaded into a weighted word embedding for word representation.

4.4. LOCATION PREDICTION VIA SE-GRU

This section uses a deep learning-based multiple-classification algorithm that can effectively predict the following location by combining heterogeneous features.

Bidirectional Gated Recurrent Unit. The suggested model feeds the word vectors from the trajectory data into the Bi-GRU layer. Each neuron contains two gated recurrent units with opposing training directions, which make up a bidirectional gated recurrent unit (BiGRU).

After calculating the passing word vectors, GRU produces a vector with fixed dimensions. Here is the detailed design. From the equation $\overrightarrow{f_x}$ denoting the forward and $\overleftarrow{f_x}$ reverse layers,

$$\overrightarrow{f_x} = \text{GRU}\left(P_x, \overrightarrow{f_{x-1}}\right), \qquad (11)$$

$$\overleftarrow{f_x} = \text{GRU}\left(P_x, \overleftarrow{f_{x-1}}\right), \qquad (12)$$

$$P_x = z_x \overrightarrow{f_x} + r_x \overleftarrow{f_x} + q_x, \qquad (13)$$

Four computational components go into GRU. To decide which data to discard earlier, GRU first utilizes a reset gate,

$$l_x = \sigma(S_l\, P_x + K_l f_{x-1} + q_l\,). \qquad (14)$$

Next, GRU uses the update gate to update the data that is currently available;

$$c_x = \sigma(S_c\, P_x + K_l f_{x-1} + q_l\,). \qquad (15)$$

Here, q is the bias S, and K is the weight information:

$$\tilde{f}_x = \tan(\,S_f\, P_x + K_f\, l_x \odot f_{x-1} + q_l\,). \qquad (16)$$

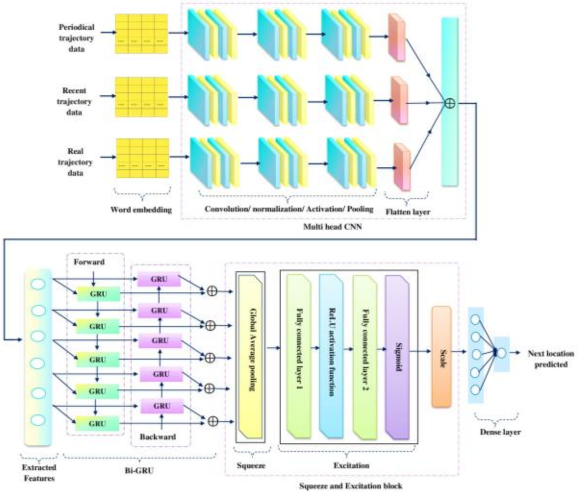


Fig 2 – Architecture of proposed MH-SE BiGRU network.

Lastly, the output from the outcomes above is computed by the GRU:

$$\text{GRU}(P_x, f_{x-1}) = (1 - c_x) \odot f_{x-1} + c_x \tilde{f}_x. \qquad (17)$$

P_x is the network's input value at that precise point in the equations above. The final state of the hidden layer is given as f_{x-1}. The Hadamard product is $\odot$. σ represents the sigmoid function. In the equation, the terms $\mathrm{S}S_l$, K_l, z_x and r_x stands for weights. As a result, the performance of the GRU network can be enhanced by learning discriminative features from the input data.

Squeeze and excitation module (SE). By modeling the relationships between channels, the SE module is used to boost the representational capacity of the network. As a result, the network can enhance optimal features and execute feature recalibration. Nevertheless, SE is a straightforward and effective computational architectural unit. A statistic is produced by compressing S through its spatial dimensions $U \times N$. The g^{th} element of J is then calculated as follows:

$$J_g = T_{sq}(S) = \frac{1}{U \times N} \sum_{a=1}^{U} \sum_{b=1}^{N} S(a,b), \qquad (18)$$

where U and N denote the height and breadth of the input feature map and T_{sq} is the squeeze function. To identify channel-wise dependencies, the excite operation receives the aggregated data from the squeeze operation

$$h = T_{ex}(J,U) = \sigma\big(v(J,U)\big) = \sigma(U_2\delta(U_1 J)) \qquad (19)$$

where δ stands for the ReLU activation function and σ for the Sigmoid function. The gating mechanism is parametrized using two non-linear, fully connected layers to make the model more straightforward and more applicable. By rescaling S with activations s, the block's final output is generated as follows:

$$\widetilde{a_g} = T_{scale}(S_G, P_G) = S_G, P_G \qquad (20)$$

where $T_{scale}(S_G, P_G)$ show the multiplication between the scalar variables channel-wise. P_G and feature map $S_G \in D^{U\times N}$ and $\tilde{A} = [\widetilde{a_1}, \widetilde{a_1}, \dots \widetilde{a_G}\ ]$.

5. RESULT AND DISCUSSION

This section contains tests to compare the proposed model's performance with current approaches and assess the model's correctness under different conditions. F1-score, precision, accuracy, and recall are utilized to evaluate the proposed method and tested using the two datasets. Our model is built using MATLAB 2020b and Python 3.7, and our experiments are carried out on a Windows 11 PC with an Intel(R) Core (TM) i5-1335U CPU operating at 1.30 GHz (8 CPUs) and 8 GB of main memory.

5.1. DATASET DESCRIPTION

The Geolife dataset and the Ningbo AIS trajectory data [12] are the foundation for the suggested method's tests.

Geolife dataset. For more than five years, 182 users in Beijing participated in the Geolife project and provided the Geolife dataset (April 2007 to August 2012). With the time and distance thresholds for stay point detection set at 5 minutes and 200 meters, respectively, we can acquire 43,442 stay points in total. 182 people are down to 50, and eliminating those with stay point records below 200 will obtain 35,960 stay points. Ultimately, 23,775 trajectory sequences are obtained.

Ningbo AIS trajectory Dataset. This dataset compiles all ship data acquired by Ningbo Port AIS. Each entry includes the ship ID, GPS location, and timestamp. The collecting period is 2015.03.01 to 2015.03.31. Due to the massive volume of data, there is extensive coverage and a quick data-collecting frequency.

5.2. EVALUATION METRICS

The suggested method assesses the next location prediction performance using five metrics: the hitting ratio, precision, recall accuracy, and NDCN.

Hitting ratio @N. The hitting ratio calculates the proportion of test trajectories for which the top-k result list correctly recovers the ground-truth position.

Accuracy. Its definition is the proportion of accurate forecasts to all successful predictions,

$$Acc_{@N} = \frac{\text{correct prediction}}{\text{total testing samples}}. \tag{21}$$

Recall@N and Precision@N. It is computed as the proportion of instances where the actual visited location was discovered inside the top N most probable locations within the ranking list.

$$\text{Recall@N} = \frac{1}{|X|}\sum_{x=1}^{|X|}\sum_{p=1}^{S_t}\frac{Y_{x,p}^{N,pre}\cap Y_{x,p}^{vis}}{N}, \tag{22}$$

$$\text{Precision@N} = \frac{1}{|X|}\sum_{x=1}^{|X|}\sum_{p=1}^{S_t}\frac{Y_{x,p}^{N,pre}\cap Y_{x,p}^{vis}}{Y_{x,p}^{vis}}. \tag{23}$$

Here, $Y_{x,p}^{vis}$ it represents the locations a user has visited and S_t the duration of each user's test sequence.

Normalized discounted cumulative gain (NDCG). The discounted cumulative gain (DCG) divided by the NDCG is used to determine the prediction vector's quality (IDCG),

$$\text{NDCG} = \frac{\text{DCG@N}}{\text{IDCG@N}},\ \text{where}$$
$$\text{DCG@N} = \sum_{a=1}^{N}(2^{rel_a} - 1\,)/\log_{2(a+1)}. \tag{24}$$

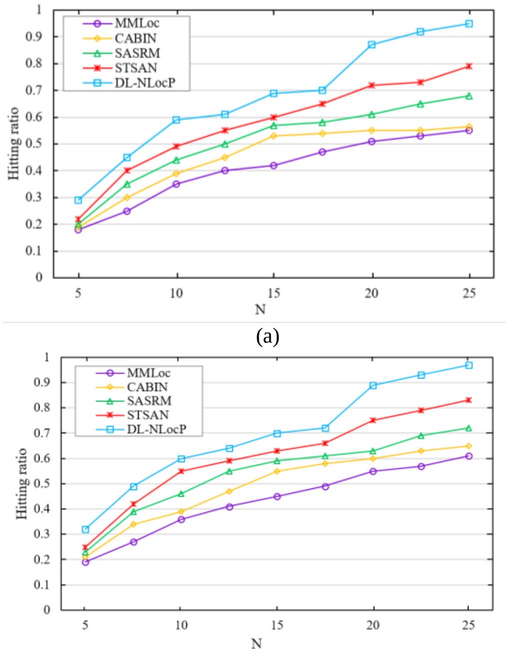
Mean reciprocal rank (MRR). It determines the prediction vector's average rank reciprocal or the point at which the first pertinent entry was found,

$$M_{RR} = \frac{1}{N}\sum_{a=1}^{N}\frac{1}{\text{rank}_a}, \tag{25}$$

where N stands for the number of test samples $\text{ran}k_a$ and denotes the rank of the ground truth location.

5.3. COMPARATIVE ANALYSIS

We evaluate the proposed DL-based NLocP by contrasting it with the MMLoc [12], CABIN [13], SASRM [14], and STSAN [15] techniques currently in use. Employing the Geolife and Ningbo AIS datasets, we evaluate the suggested method against other approaches with well-tuned parameters based on hitting ratio, precision, recall accuracy, and NDCN.



(a)

(b)

Fig 3– Comparison of hitting ratio: a) Geolife; b) Ningbo AIS.

The hitting ratios of the two datasets are compared in Fig. 3 between the proposed DL-NLocP method and the current MMLoc [12], CABIN [13], SASRM [14], and STSAN [15] methods using varying N values. As illustrated in Fig. 3a for the Geolife dataset, the proposed approach achieves a hitting ratio of 0.95 percent for $N = 20$ in contrast to the existing methods. Compared to existing approaches, the suggested method produces a 0.97 percent hitting ratio when $k = 25$, which is depicted in Fig. 3b, which displays the hitting ratio of the Ningbo AIS dataset. Of the two datasets, the Ningbo AIS dataset exhibits superior hitting.

Figure 4 compares the accuracy, recall, NDCG, and precision attained on the Geolife dataset using the suggested and current methods. The precision of the proposed DL-NLocP is compared with the existing MMLoc [12], CABIN [13], SASRM [14], and STSAN [15] methods in Fig. 4a. The recall of the proposed approach is shown in Fig. 4b. When $N\ = 20$, the recall of the existing technique is 0.59, 0.51, 0.75, and MMLoc, CABIN, SASRM, and STSAN methods are 0.52, 0.89, and 0.89 for the suggested method, which is comparatively higher than the prior studies. The comparison of NDCG between MMLoc, CABIN, SASRM, STSAN, and DL-NLocP is shown in Fig. 4c. The comparison of NDCG between MMLoc, CABIN, SASRM, STSAN, and DL-NLocP is demonstrated in Fig. 4c, and the accuracy for the proposed method is depicted in Fig. 4d.

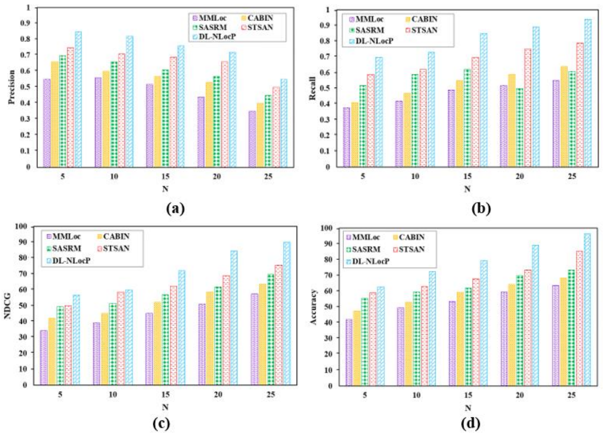


Fig. 4 – Comparative analysis of: a) precision; b) recall; c) NDGC; d) accuracy using the Geolife dataset.

Table 1 depicts the experimental findings on mean reciprocal rank (MRR) for the Geolife dataset. Compared to other current approaches, the suggested model performs better. Based on the analysis, the proposed approach achieves a relative increase in performance of 0.90 percent MRR@25 in the Geolife dataset, surpassing all other available methods.

Table 1

Performance evaluation of MRR for Geolife dataset

MRR	MMLoc [12]	CABIN [13]	SASRM [14]	STSAN [15]	Proposed DL-NLocP
MRR@5	0.37	0.44	0.46	0.501	0.56
MRR@10	0.408	0.49	0.51	0.58	0.68
MRR@15	0.46	0.56	0.59	0.63	0.75
MRR@20	0.53	0.61	0.65	0.74	0.83
MRR@25	0.56	0.67	0.76	0.85	0.90

Table 2

Performance evaluation of MRR for Ningbo AIS dataset

MRR	MMLoc [12]	CABIN [13]	SASRM [14]	STSAN [15]	Proposed DL-NLocP
MRR@5	0.52	0.59	0.63	0.66	0.75
MRR@10	0.56	0.61	0.69	0.72	0.79
MRR@15	0.57	0.65	0.74	0.77	0.86
MRR@20	0.65	0.67	0.79	0.85	0.90
MRR@25	0.69	0.73	0.86	0.89	0.93

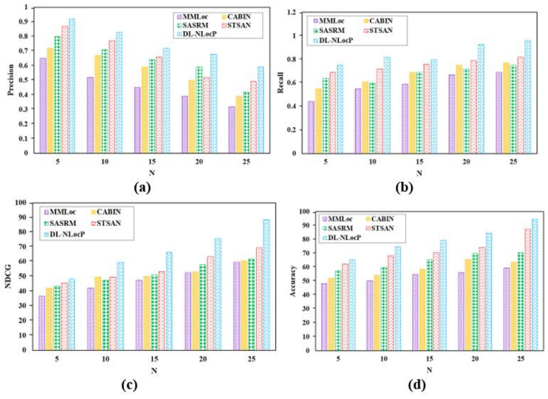


Fig 5 – Comparative analysis of: a) precision; b) recall; c) NDGC; d) accuracy using Ningbo AIS dataset.

Figure 5 a shows the precision comparison with different *N* sizes. The suggested method obtains a recall of 0.93% when *N* is 5, which is comparatively high compared to the current techniques. Figure 5b shows that, in comparison to current methodologies, the recall of the suggested approach grows as the N size increases.

Figure 5c shows that the suggested method's NDCG is higher than the existing approaches. Figure 5d shows that the proposed approach is more accurate than the current model, with 96.78% accuracy. The recommended method outperforms STSAN, SASRM, CABIN, and MMLoc by 1.25%, 3.12%, 5.66%, and 7.5%, respectively. Table 2 depicts the suggested method surpassing existing methods, achieving 0.93% on MRR@25 for the Ningbo AIS dataset.

Table 3

Prediction time comparison results

Techniques	Geolife	Ningbo AIS
	Time (s)	Time (s)
MMLoc	16.8	18.1
CABIN	14.8	17.6
SASRM	12.7	14.5
STSAN	10.74	12.1
Proposed DL-NLocP	7.2	8.15

The prediction time of the different datasets compared to the proposed and existing methods is displayed in Table 3. Regarding prediction time, the proposed DL-NLocP performs noticeably better than current techniques for both datasets. The proposed DL-NLocP is 19.5%, 14.5%, 10.8%, and 7.6% less effective than the current MMLoc, CABIN, STSAN, and SASRM, respectively, for the Geolife dataset. Compared to the MMLoc, CABIN, STSAN, and SASRM, the proposed method is 14.5%, 12.9%, 8.45%, and 5.4% lower for the Ningbo AIS dataset. According to the comparison, the Geolife dataset outperforms the Ningbo AIS dataset in terms of processing times.

6. **CONCLUSIONS**

This paper proposed a DL-NLocP Network to increase the accuracy of the next location prediction. Initially, the datasets are pre-processed to enhance the data quality and employ term frequency-inverse document frequency (TF-IDF) with Glove word embedding approaches to convert the textual data into real-valued vectors. Afterward, Multihead CNN extracts temporal, location, and user behavior features from the vector data. Finally, Squeeze and Excitation with the BiGRU network are developed to predict the following location in each trajectory with contextual information. The proposed DL-NLPN model was tested on the Ningbo AIS and Geolife dataset, and experimental results supported the model's validity. On average, the accuracy@20 of the proposed model is 93.75% higher for Geolife and 94.75% higher for Ningbo AIS than the state-of-the-art methods. The findings demonstrate that the proposed method outperforms the current techniques in predicting the following location. The advantage of the proposed method is the integration of Multihead CNN and Squeeze and Excitation BiGRU, which enables the model to extract and utilize temporal, spatial, and user-behavioral features effectively by enhancing prediction performance. The limitation of the proposed DL-NLocP method is that it does not integrate data from multiple modalities and is slightly computationally complex. These limitations will be considered in future work to extend our proposed system.

The significant contribution of the proposed method is:

- This research introduced a novel following next-location prediction network (DL-NLocP) for next-location prediction.
- The proposed framework devises the ideology from frequency-inverse document frequency (TF-IDF) with Glove word embedding approaches to convert the textual data into real-valued vectors.
- Moreover, the Multihead CNN technique improves the precision of location prediction by extracting temporal, location, user behavior characteristics, and user preferences into a unified framework.
- The Squeeze and Excitation with the BiGRU network are developed to predict the following location in each trajectory.
- According to experimental results, the proposed model's accuracy is 93.75% for Geolife and 94.75% for the Ningbo AIS dataset, which is higher than that of the existing techniques.

ACKNOWLEDGEMENTS

The author would like to express his heartfelt gratitude to the supervisor for his guidance and unwavering support during this research and his advice and support.

CREDIT AUTHORSHIP CONTRIBUTION

Uma Natarajan: study conception and design, data collection.

Anitha Ramachandran: analysis and interpretation of results, draft manuscript preparation

Received on 7 December 2023

REFERENCES

1. Y. Xiao, Q. Nian, *Vehicle location prediction based on spatiotemporal feature transformation and hybrid LSTM neural network*, Information, **11,** *2,* p. 84 (2020).
2. J.P. Singh, Y.K. Dwivedi, N.P. Rana, A. Kumar, K.K. Kapoor, *Event classification and location prediction from tweets during disasters*, Annals of Operations Research, **283**, pp. 737–757 (2019).
3. D. Yao, C. Zhang, J. Huang, J. Bi, *Serm: A recurrent model for next location prediction in semantic trajectories*, ACM on Conference on Information and Knowledge Management, pp. 2411–2414 (2017).
4. S. Kumar, M.I. Nezhurina, *An ensemble classification approach for prediction of user's next location based on Twitter data,* Journal of Ambient Intelligence and Humanized Computing, **10**, pp. 4503–4513 (2019).
5. K.R. Seelam, V.A. Reddy, N. Arpitha, *Location prediction on Twitter using machine learning,* Journal of Engineering Sciences, **14,** *01* **(**2023).
6. H. Li, B. Wang, F. Xia, X. Zhai, S. Zhu, Y. Xu, *PG^2 Net: personalized and group preferences guided network for next place prediction*, arXiv:2110.08266 (2021).
7. P.J. Shermila, A. Ahilan, A.J.G. Malar, R. Jothin, *MDEEPFIC: food item classification with calorie calculation using modified dragonfly deep learning network*., Journal of Intelligent & Fuzzy Systems, **17**, *7,* pp. 1–9 (2023).
8. J. Lv, Q. Sun, Q. Li, L. Moreira-Matias, *Multi-scale and multi-scope convolutional neural networks for destination prediction of trajectories,* IEEE Transactions on Intelligent Transportation Systems, **21,** *8,* pp. 3184–3195 (2019).
9. A. Agasthian, Rajendra Pamula, L.A. Kumaraswamidhas, *Integration of monitoring and security-based deep learning network for wind turbine system,* International Journal of System Design and Computing, **1,** *1,* pp. 11–17, (2023).
10. M. Luca, L. Pappalardo, B. Lepri, G. Barlacchi, *Trajectory test-train overlap in next-location prediction datasets,* Machine Learning, pp. 1–38 (2023).
11. A. Sassi, M. Brahimi, W. Bechkit, A. Bachir, *Location embedding and deep convolutional neural networks for next location prediction*, IEEE 44th LCN symposium on emerging topics in networking (LCN symposium), pp. 149–157 (2019).
12. R. Zhang, J. Guo, H. Jiang, P. Xie, C. Wang, *Multi-task learning for location prediction with deep multi-model ensembles*, IEEE 21st International Conference on High-Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), pp. 1093–1100 (2019).
13. T. Qian, F. Wang, Y. Xu, Y. Jiang, T. Sun, Y. Yu, *CABIN: a novel cooperative attention based location prediction network using internal-external trajectory dependencies*, Springer International Publishing, pp. 521–532 (2020).
14. X. Zhang, B. Li, C. Song, Z. Huang, Y. Li, *SASRM: a semantic and attention spatio-temporal recurrent model for next location prediction*, International Joint Conference on Neural Networks (IJCNN) IEEE, pp. 1–8 (2020).
15. S. Wang, A. Li, S. Xie, W. Li, B. Wang, S. Yao, M. Asif, *A spatial-temporal self-attention network (STSAN) for location prediction,* Complexity, pp. 1–13 (2021).
16. N. Rathore, N.L. Panwar, F. Yettou, A. Gama, *Estimation and mapping of linke turbidity factor from solar radiation measurement for Indian climatic conditions*, Rev. Roum. Scie. Techn. – Électrotechn. et Énerg., **66**, *4,* pp. 285–294 (2021).
17. A. Dobre, A.M. Ilie-Sandoiu, A.M. Morega, E. Gheorghiu, *Magnetic field control in an analytic platform for assessment of pathogenic bacteria*, Rev. Roum. Scie. Techn. – Électrotechn. et Énerg., **68,** *3,* pp. 317–322 (2023).