Ibn Al-Haitham Journal for Pure and Applied Sciences
Journal homepage: jih.uobaghdad.edu.iq
PISSN: 1609-4042, EISSN: 2521-3407
IHJPAS. 2024, 37(4)

Mobility Prediction Based on Deep Learning Approach Using GPS Phone

Nabaa Kareem Mhalhal[1] and Suhad Faisal Behadili[2,*]
[1,2] Computer Science Department, College of Sciences, University of Baghdad, Baghdad, Iraq.
*Corresponding Author.

Received: 28 February 2024 Accepted: 21 April 2024 Published: 20 October 2024
doi.org/10.30526/37.4.3916

Abstract

Accurate prediction of activity location is a crucial component in various mobility applications and is particularly vital for the creation of customized, environmentally friendly transport systems. Next-location prediction, which entails predicting a user's forthcoming place by analyzing their previous movement patterns, has substantial ramifications in diverse fields, such as urban planning, geo-marketing, disease transmission, wireless network performance, recommender systems, and numerous other sectors. Recently, researchers have proposed a variety of predictors, including cutting-edge ones that utilize advanced deep learning methods, to tackle this problem. This study introduces robust models for predicting a user's future location based on their previous location. It proposes a Recurrent Neural Networks (RNNs) prediction scheme and a Gated Recurrent Unit (GRU), which are well-suited for learning from sequential data. Additionally, the clustering technique Density-Based Clustering (DBSCAN) is implemented to extract the stay points. Furthermore, the suggested method is more accurate at predicting the future than the current method, showing improvements in loss mean square error of up to 0.0005 in the RNN model and 0.01 in the GUR model. So, the models that were used led to a decrease in loss MSE, which was shown in the real-world dataset (Geolife) in this paper. The results are also consistent with other similar works that look at the same issue, showing how well the models can predict mobility.

Keywords: Recurrent Neural Networks (RNNs), Gated Recurrent Unit (GRU), Density-Based Clustering (DBSCAN), Next Location Path, Deep Learning.

1. Introduction

User mobility pertains to the relocation of individual users from one location to another, usually within a defined geographical scope such as a city, region, or nation. The prediction of user movement has yielded instant advantages in several subsequent tasks, including recommendations for sites of interest, forecasting communication network traffic, optimizing road traffic, and more [1, 2].

Indeed, our current existence is characterized by a sophisticated technological landscape that includes state-of-the-art systems and equipment such as cellphones, computers, smartwatches, and other such devices. Individuals are unable to conceive of their existence without a smartphone or personal computer [3]. Technological advancements have yielded extensive trajectory data for studies and applications related to individual mobility [4]. Furthermore, examining the movement of individuals is essential for addressing other significant societal issues, such as urbanization, segregation, and the proliferation of epidemics, among others [5, 6]. Next-location prediction is the process of predicting the specific location that an individual will visit based on their previous movement pattern [7, 8].

The scope of user mobility research is largely unexplored, and accurately predicting individual user movement is a complex and diverse undertaking. Existing algorithms for predicting individual mobility often lag behind in terms of accuracy, typically obtaining only approximately 93%. Consequently, in order to fill this gap, it is necessary to provide a theoretical foundation for predicting user mobility [9], such as Artificial Neural Networks (ANN) [10]. Despite its crucial importance, location prediction remains an unresolved daunting task. Statistical distributions can describe the strong and predictable pattern that individual actions within a group exhibit. However, at the user level, mobility encompasses several unplanned decisions and shows complicated features that impact the accuracy of forecasts [11]. Various domains, including the following, consider predicting future locations or the next location as a viable method to deliver such a service.

- It is crucial in several applications, such as providing travel recommendations and optimizing routes, detecting potential public emergencies at an early stage, estimating urban emissions, delivering location-based adverts and geo-marketing, and suggesting friends on social network platforms [11].
- The prediction of the next location of a mobile user based on their trajectory information is a vital feature in the creation of smart applications that may aid users without the need for manual input [12].
- The estimation and Prediction of user mobility in an ad hoc wireless network will have an advantageous effect on ensuring the quality of service (QoS) and quality of experience (QoE) inside the network [13].
- The next 5G and B5G mobile communication technologies are anticipated to provide substantial enhancements in terms of decreased latency and delay [13].
- The quality of telecommunications services has a substantial influence on the mobility of consumers in their everyday activities, particularly in vehicular ad hoc networks (VANETs). This is crucial for addressing issues such as traffic congestion, handovers, and activity routing protocol [2].
- In the domain of urban science, analyzing the trajectory data of mobility users assists researchers in understanding urban dynamics, identifying regions with high levels of activity, and improving the allocation of resources and public services [14].

To enhance accuracy and reliability, it is necessary to employ innovative methods to predict user movement [15]. To anticipate the future trajectories of users in word prediction, we rely on their previous behavior [16]. However, various researchers have demonstrated the need to improve the current prediction models to provide a reliable and comprehensive forecast for determining future locations. We can highlight the key contributions of this research as follows: The research effectively creates two models that forecast the user's future path. The suggested methods give numerical and visual results that show how well they work for predicting user mobility, finding stay-points, and extracting significant locations using DBSCAN.

This work is organized into the following sections: Section 2 presents the literature review. Section 3 offers a concise overview of the research methodology and outlines the proposed strategy. Section 4 presents the results and discussions, while Section 5 presents the conclusion and future work of our study.

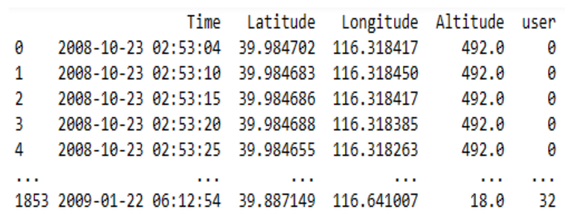
2. Literature Review

Human mobility prediction entails the analysis and prediction of human movements, whether on an individual or group level, using a variety of data sources and advanced predictive modeling approaches [5]. This section examines research endeavors, specifically those that pertain to the prediction of future locations and the identification of points of interest (POI). For example, Dongliang Liao et al. [17] introduced spatial-activity topics as the underlying factor that represents both the user's activity and location preferences. Their proposal involves the utilization of a multi-task Context Aware RNN to exploit the spatial-activity theme in order to predict both activity and location. Meanwhile, Qiang Gao et al. [18] suggested a new model called Variational Attention-Based Next (VANext) for predicting Points of Interest (POI). In particular, they used the Singapore dataset by 31.52%, 36.81%, and 32.68%, respectively, in terms of Top@1, Top@5, and Top@10. Additionally, Katharina Prinz in [19] employed a Hidden Markov Model (HMM) to forecast future locations, utilizing the clustering technique. Density-based clustering (DBSCAN) is used to group several stay-points together, resulting in the formation of notable places. Therefore, to address this issue, we introduced deep learning (DL) models, which have the ability to understand complex sequential relationships from large volumes of data. However, integration uses the Long Short Memory (LSTM) and DBSCAN algorithms in [20] because they can tackle individual mobility challenges like modeling large-scale sequential data and determining the number of clusters in arbitrary trajectories. The prediction error, or RMSE value, consequently reached a score of 3.551.

3. Research Methodology

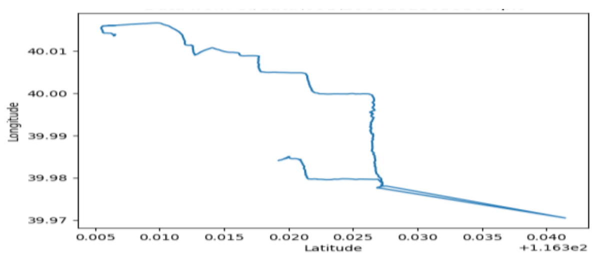
3.1 Dataset

The DL model was applied to the real-world GeoLife GPS Trajectories dataset. The dataset captured a diverse range of individuals participating in various outdoor activities, including everyday tasks like commuting to work as well as recreational and sporty pursuits such as shopping, sightseeing, dining, hiking, and cycling. Diverse study fields widely employ it, including mobility pattern analysis, user activity recognition, location-based social networks, location privacy, and location recommendation. The GPS trajectory data consists of a series of time-stamped points, denoted as $T_r = \{t_{r1}, t_{r2}, \ldots, t_{rn}\}$obtained from an urban area or city. Each trajectory comprises a sequence of m recordings. The sequence is denoted as $t = t_1, t_2, ..., t_m$. Each record t is a tuple that comprises t = {sequence number, time as string, latitude, longitude, altitude, and number of users}[21], as in **Figure 1**. where the raw data set is shown and some trajectories are plotted. Algorithm 1 explains how the dataset is preprocessed.



```
                     Time   Latitude   Longitude  Altitude  user
0     2008-10-23 02:53:04  39.984702  116.318417     492.0     0
1     2008-10-23 02:53:10  39.984683  116.318450     492.0     0
2     2008-10-23 02:53:15  39.984686  116.318417     492.0     0
3     2008-10-23 02:53:20  39.984688  116.318385     492.0     0
4     2008-10-23 02:53:25  39.984655  116.318263     492.0     0
...                   ...        ...         ...       ...   ...
1853  2009-01-22 06:12:54  39.887149  116.641007      18.0    32
```

(a)



(b)

Figure 1. (a) Sample of raw dataset (b) trajectories for one user

3.2 Preprocessing dataset

The dataset contains 182 folders. Each folder in this collection includes the PLT-formatted GPS log files that belong to a certain user. Every PLT file has a unique trajectory, with a starting time assigned to it. The raw data contains six columns of data. Once the raw data is extracted from the server, it will be analyzed, and the features required for the next location will be selected. Here, the first six lines in each folder, assumed to be headers, must be dropped and skipped; this requires manipulating and representing the acquired data through graphical visualization [21]. Given that each line in the data list consists of comma-separated values, with the first value denoting latitude and the second value denoting longitude, these lines extract and convert these values into floating-point data types, resulting in two lists: latitudes and longitudes, as shown in Algorithm 1:

```
        Algorithm 1: The Pre-processing dataset
        Input: Dataset
        Output: Obtain two NumPy arrays(X,Y)
1:      Initialize data path
2:      For loop to open the file for reading
3:              For loop to Skip the first six lines (assumed headers)
4:              Read the remaining lines, extracting latitude and longitude coordinates from each line.
5:              Scale the extracted coordinates using a MinMaxScaler.
6:              Create sequences of length sequence length from the scaled coordinates.
7:              End for
8:      End for
9:      For loop to each sequence, add the coordinate immediately following it as a target value to the targets list.
10:     Convert Lists to NumPy Arrays(X, Y)
12:     Return two NumPy Arrays(X, Y)
13:     End for
```

3.3 Locating fixed locations

During this phase, the GPS trajectory of a user over a period of more than 65 days identifies multiple stationary points. A stay point refers to a specific geographical location where an individual remains for a certain duration. The work employed the Density-Based Clustering (DBSCAN) algorithm to group together regions of interest based on stay points. In many use cases, the domain knowledge required to choose a fixed number of clusters is frequently lacking. To get around this, DBSCAN methods take into account the fact that there is a lot more density inside a cluster than outside of it [22]. DBSCAN's basic idea is to find groups of data points with a small number of nearby points within a certain radius. In its simplest form, a neighborhood density has to be higher than a certain threshold. With DBSCAN, you can control the neighborhood shape using a variety of distance functions [19]. The work uses this approach to calculate clusters of proximate stay points. The algorithm parameters consist of epsilon (EPS) and the minimum number of points in the neighborhood (MinPts). The epsilon parameter sets the maximum distance between points required to classify them as part of the same cluster. For this investigation, the work established a different EPS as a threshold for points to be considered part of a cluster, as shown in algorithm 2.

```
    Algorithm 2: DBSCAN [23]
    Input:dataset, EPS, MinPts
    Output: clusters C
1:  Leave all points as unmarked.
2:  Do
3:  From the collection of unmarked points, pick a random point X
4:  Indicate X as unmarked
5:  If the number of points within a radius epsilon of X is more than or equal to MinPts, then
6:      Produce a new cluster C and put P into C
7:      Let P for each point X do
8:      For P for each point X do
9:          If X is unmarked then
10:             Set X as marked
11:             If the number of points in X adjacent area EPS >MinPts then
12:                 Put them all to X
13:                 If X does not belong to any cluster then
14:                     Put X to C
15:                 End if
16:             End if
17:         End if
18:     End for
19:     Output C
20: Else
21:     Marks X as noise points
22: End if
23: Repeat the above procedure until there are no unmarked points
```

3.4 Modeling and predicting path trajectory

The first model, Recurrent Neural Networks (RNNs)[24], has recently demonstrated its effectiveness in modeling sequential data by capturing intricate long- and short-term relationships within input sequences. Existing strategies aim to incorporate contextual factors into the RNNs in order to handle sequences that are sparse and incomplete [25]. One can incorporate temporal considerations by dividing each sparse input sequence into multiple shorter sessions or by treating temporal elements as supplementary inputs to the RNN units. Research has demonstrated that spatiotemporal features are highly effective predictors for location prediction based on limited user movement data [26]. The second model, the Gated Recurrent Unit (GRU), is an alternative form of the RNN design that tackles the problem of limited short-term memory. The next section provides an overview of the various components of the suggested models and presents a comprehensive explanation of the proposed deep learning model architecture.

3.5 The fundamentals of the RNNs and GRU models

RNN is an extension of the standard feed-forward neural network that is capable of processing input sequences of varying lengths [9]. The RNN is capable of processing sequences of different lengths by utilizing a hidden state that is recurrent and whose activation at each time step depends on the activation of the preceding time step [27]. A generative RNN produces the probability distribution of the subsequent element in a sequence based on its presence [26]. RNN cells can be explained in Equations (1) and (2). The basic structure is shown in Figure 2 [26].

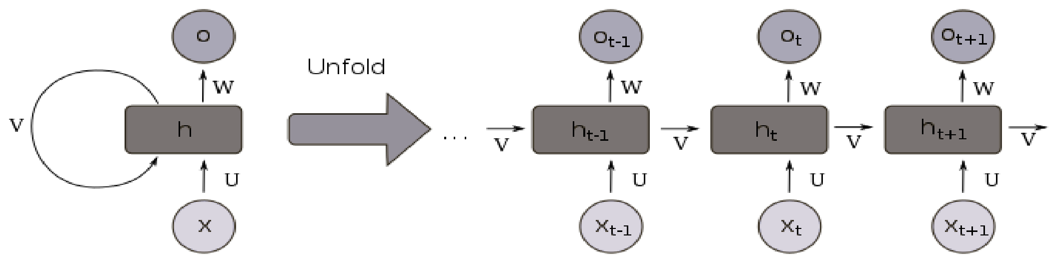


Figure 2. RNN neural network basic structure

$$h_\tau = \sigma_h(W_h x_\tau + U_h h_{t-1} + V) \tag{1}$$

$$O_t = \sigma_o(W_o h_t + V_o) \tag{2}$$

Where x_t *is* the input vector, h_t is the hidden layer vector, Ot is the output vector, W, U, V are Parameter matrices and vector, σ_x, σ_O are Activation Functions. A GRU unit comprises three primary components: an update gate, a reset gate, and the current memory content. The gates facilitate the GRU in selectively updating and utilizing information from preceding time steps, hence enabling it to capture long-term dependencies in sequences [26], as in **Figure 3**.

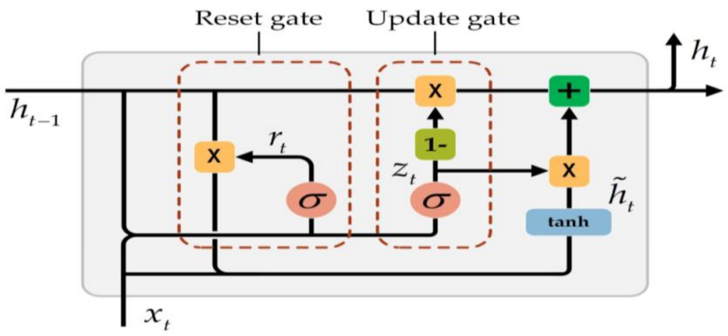


Figure 3. The basic structure of GRU

The update gate establishes the degree to which the current state retains state information from the previous instant. The update gate preserves more state information from the preceding moment, the greater its value. Equation (3) determines the update gate state at time t by the following formula:

$$\mathrm{z_t} = \sigma(\mathrm{W_z} \cdot [\mathrm{h_{t-1}}, \mathrm{x_t}]) \tag{3}$$

Where x_t is the input vector, z_t is the update gate vector, h_{t-1} is the output vector of the previous state, σ is the sigmoid function, and $\mathrm{W_z}$ is the weight of the update gate. The GRU calculates the reset gate state (rt) at a specific time (t) within the reset gate to indicate the degree of disregard for previous information. A smaller reset gate value indicates a greater amount of ignored information. The gate executes the subsequent Equation (4):

$$\mathrm{r_t} = \sigma(\mathrm{W_r} \cdot [\mathrm{h_{t-1}}, \mathrm{x_t}]) \tag{4}$$

Where $\mathrm{r_t}$ is the reset gate, x_t is the input vector, h_{t-1} is the output vector of the previous state, σ is a sigmoid function, and $\mathrm{W_r}$ is the weight of the reset gate. The current state $\tilde{\mathrm{h}}_\mathrm{t}$ of the GRU memory unit is determined by the following calculation in Equation (5):

$$\tilde{\mathrm{h}}_\mathrm{t} = \tanh(W \cdot [r_t \cdot h_{t-1}, x_t]) \tag{5}$$

Where *tanh* is the Hyperbolic Tangent function and *W* is weight. Then, finally, the output state h_t of the GRU at the current moment is determined by Equation (6):

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \quad (6)$$

3.6 The proposed models

In this work, two models were proposed. The first model is RNN, which was explored in Algorithm 3.

```
        Algorithm3: The training process of First model (RNN)
        Input: Dataset
        Output: The predicted path of the next location
1:      Initialize data path, sequence length
2:      For do
3:          Obtain two NumPy arrays  from the algorithm (1)
4:      End for
5:      For each epoch do
6:          For each batch do
7:              Simple RNN model forward pass;
8:              Dence layer;
9:              Dence layer;
10:         End for
11:     If the value of the Loss function is greater than the previous state, stop training
12:     else
13:     update weight vi Adam optimizer algorithm until end epoch
14:     End for
13:     Calculate evolution metrics MSE, RMSE, MAP
```

The RNN model depends on simple RNN units [28]. The models were trained by backpropagation through time using the Adam stochastic optimizer [29] with Mean Square Error (MSE) loss [30], whereas the second proposed model was explored in Algorithm 4, which depended on GRU units.

```
        Algorithm 4: The training process of the second model(GRU)
        Input: Dataset
        Output: The predicted path of the next location
1:      Initialize data path, sequence length,
2:      For do
3:          Obtain two NumPy arrays from the algorithm (1)
4:      End for
5:      For each epoch do
6:          For each batch do
7:              GRU model forward pass;
8:              Dence layer;
9:              Dence layer;
10:         End for
11:     If the value of the Loss function is greater than the previous state, stop training
12:     else
13:     update weight vi Adam optimizer algorithm until end epoch
14:     End for
13:     Calculate evaluation metrics MSE, RMSE, MAP
```

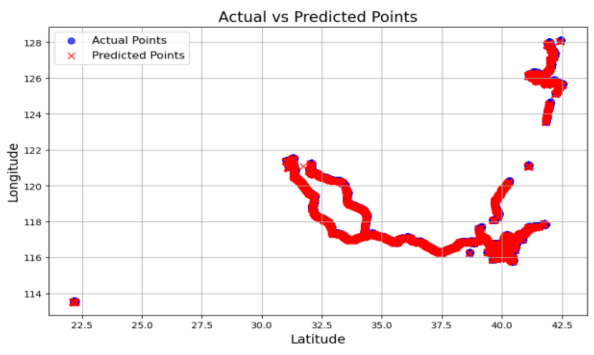
4. Results

This work implemented models on the TensorFlow platform [31]. This strategy typically yields greater prediction accuracy compared to the conventional approach of training the Neural Network (RNN and GRU) using the entire dataset (Geolife). Therefore, we employed several libraries to reduce the complexity of constructing the anticipated Neural Network. Therefore, we utilize TensorFlow, Keras, Sklearn, NumPy, and Pandas as the most significant libraries [11]. Thus, in order to optimize the proposed model, various settings need to be defined. Some of these settings are the loss function that MSE uses; the activation function called the Rectified Linear Unit (ReLU), which is the identity function for positive input and zero for negative input [32], and the adaptive optimization technique Adaptive Moment Estimation (Adam) [32]. Adam is a very good adaptive optimization algorithm that is often used instead of the traditional stochastic gradient reduction technique [29]. **Table 1.** shows the hyperparameters of the proposed models.

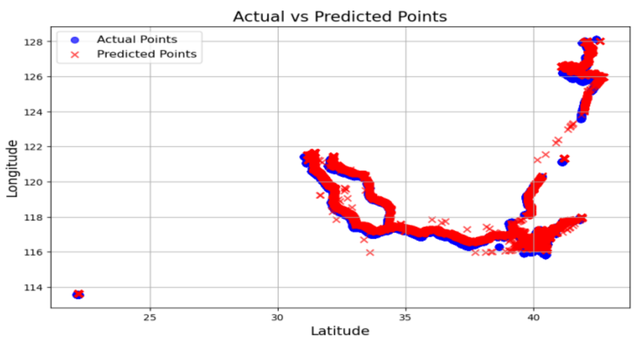
Table 1. The Hyperparameters of the proposed models

Epoch	Batch size	Iteration size	Optimization algorithm	Loss function	Number of layers	Number of neurons	Activation function
30	64	21612	Adam	MSE	3	152	ReLU

The goal of this work is to apply the proposed model network to predict individuals' next location path. DL was implemented in the Lenovo system, consisting of an Intel(R) CoreTM, an i5-8250U Gen processor @ 1.60 GHz and 1.80 GHz, and 16 GB of RAM. The system runs on the Windows 10 Pro 64-bit operating system. We perform the experiments using the Tensorflow deep learning package within the Jupyter Notebook environment. We employed various methodologies on the DL models, including early stopping, to enhance the model's accuracy. The primary objective during model training is to minimize the loss function, namely MSE. When the monitored parameter fails to show improvement, we use the early stopping technique to terminate the training process [9]. Consequently, the early stopping technique could have been more effective in consuming the required epoch training time. At the end of each epoch, the metric undergoes evaluation. This technique gives the best result for the model [31]. For all users, the work splits the trajectories into a training set and a test set. All sets include multiple user trajectories. Split the trajectory randomness for each user, putting the first 80% in the training set and the following 20% in the testing set. Then, using this data, we detect the user's path and trajectory. **Figure 4** displays the actual trajectory path and the predicted trajectory path, both of which closely match the actual path.



(a)



(b)

Figure 4. The actuals and the prediction path (a)1st model(RNN)(b)2nd model(GRU)

So, the proposed models can be improved to predict the next location path. In the final phase, the prediction model was evaluated using an evaluation score for the loss function, i.e., MSE, which is a statistical measure that quantifies the average difference between the predicted values and the actual values in a dataset. The quantity being referred to is the standard deviation of the projected errors. The concentration of information on the best-fit line is determined by this factor, which is defined in Equation (7)

$$MSE \frac{1}{m} \sum_{i=1}^{n} (y_i - \grave{y}_i)^2 \tag{7}$$

Root Mean Square Error (RMSE) refers to the standard deviation of the predicted errors, as shown in Equation 8 [33].

$$RMSE = \sqrt{MSE} \tag{8}$$

Mean Absolute Error (MAE) is calculated as the average of the absolute differences between the actual values and the anticipated values. It denotes the mean significance of the errors and is defined as in Equation 9[20].

$$MAE \frac{1}{m} \sum_{i=1}^{n} |y_i - \grave{y}_i| \tag{9}$$

where y_i is the actual value, $\grave{y}_i$ is the predicted value, and m is the total number of observations. MSE, MAE, and RMSE are suitable measurements for computing prediction sequential model accuracy and are the main criteria for determining whether the research is regarding the prediction model [34]. All those measurements fall within the optimal range, which is 0. The results in Table 2 below clearly indicate that they are nearly perfect. The performance of the suggested models was evaluated and compared with the work of the researcher Ida Nurhaida et al. [20], as well as with two traditional ML techniques, as follows:

- K-Nearest Neighbors (K-NN) refers to a straightforward method that has been successfully applied in different study domains, including financial modeling, image interpolation, and visual recognition, as shown in **Table 2**; for more details, see Ref [35].
- Random Forest Regression modelling (RFR) refers to a regression technique that utilizes machine learning. The basis of this is the utilization of bagging and random subspace approaches; for more details, see Ref [36].

Table 2. The results of the evaluation metrics compared with Ref [20] and with two ML models

Metrics	Proposed Models		ML models		Model of study[20]
	1st model(RNN)	2nd model(GUR)	K-NN	RFR	
MSE	0.0005522080	0.0146721750	1.1349027058	4.53745228884	
RMSE	0.0234991064	0.1211287538	1.0653181242	2.1301296413	3.551
MAE	0.9999998807	0.9999996423	3.69057880778	0.00046895485	

The findings in **Table 2** and the visualization results in **Figure 4** explore the actual and predicted path trajectory, which is clearly close to being ideal. Furthermore, **Figures 5** and **6** delve into the evaluation metrics of each model, focusing on each epoch specifically. This, in turn, confirmed the accuracy of the proposed models' predictions. The indicator's constant drop in **Figures 5** and **6** demonstrates how strong and consistent the models are.

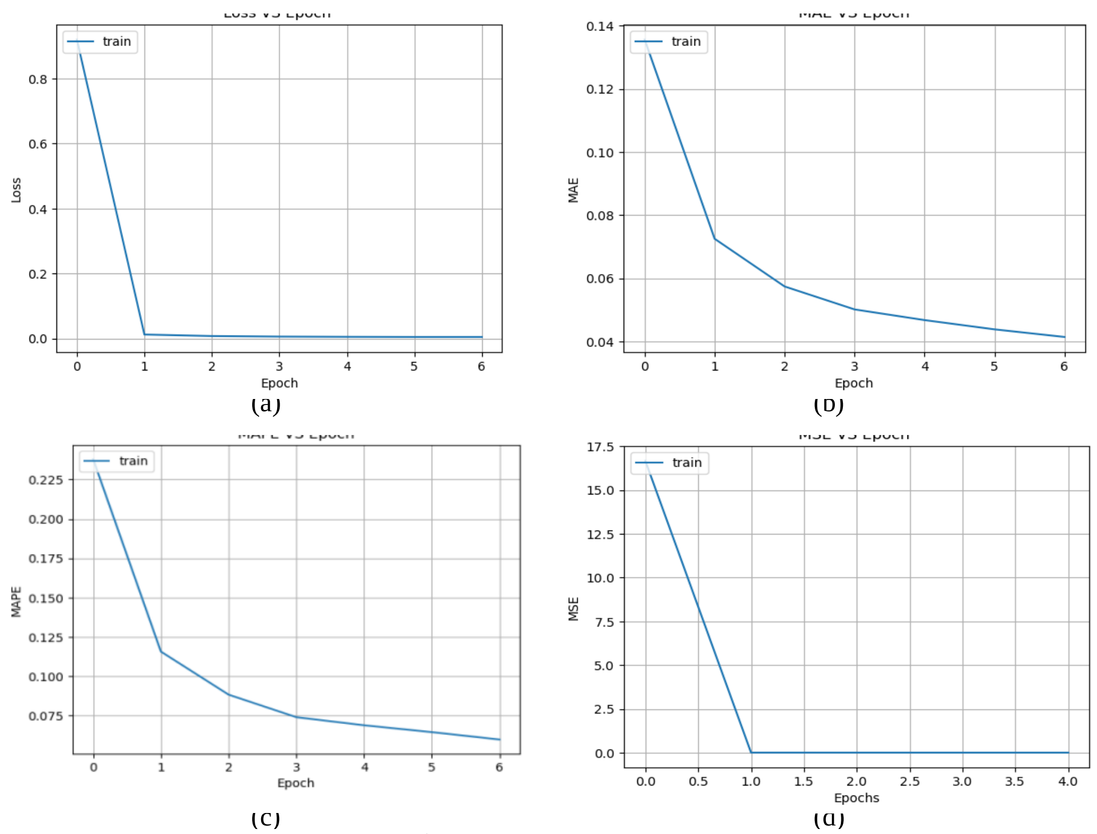


Figure 5. Visualization results of the 1st proposed model(RNN) for evaluation metrics where (a) MAE vs. epoch, (b) Loss vs. epoch, (c) MAPE vs. epoch, and (d)MSE vs. epoch

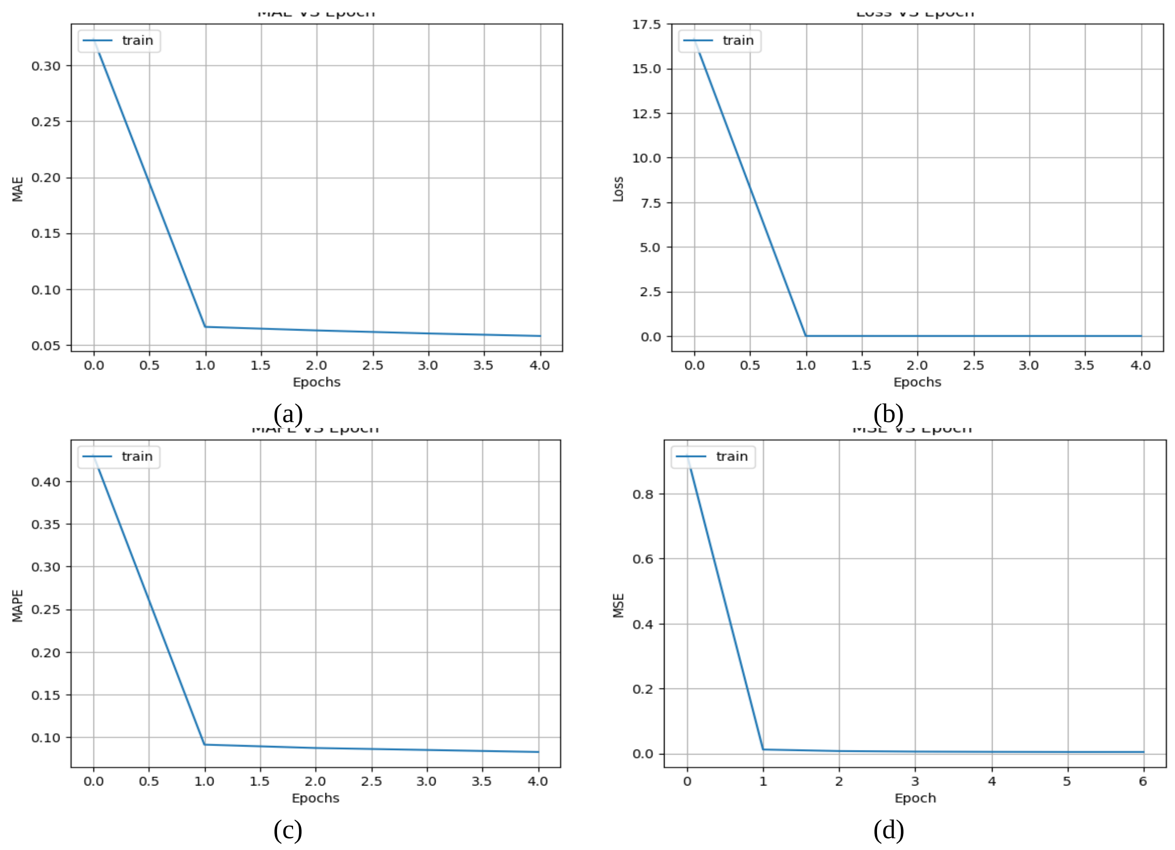


Figure 6. Visualization results of the 2nd proposed model(GRU) for evaluation metrics where (a) MAE vs. epoch, (b) Loss vs. epoch (c) MAPE vs. epoch, and (d)MSE vs. epoch

Finally, the algorithm (DBSCAN) outlines two primary considerations that control the extraction of stay points, as presented by Yang et al. By using DBSCAN on the stay-points of this trajectory, we may avoid the necessity of predefining the number of clusters to be extracted. Alternatively, the clustering technique requires us to specify a minimum number of points and a radius in order to identify the clusters. Figure 7 presents the stay points obtained for various settings, where the threshold primarily influences the number of extracted stay points and also causes variations in their precise locations. The visualizations depict the Geolife dataset's trajectory. We represent these points using their corresponding longitude and latitude values in radians.

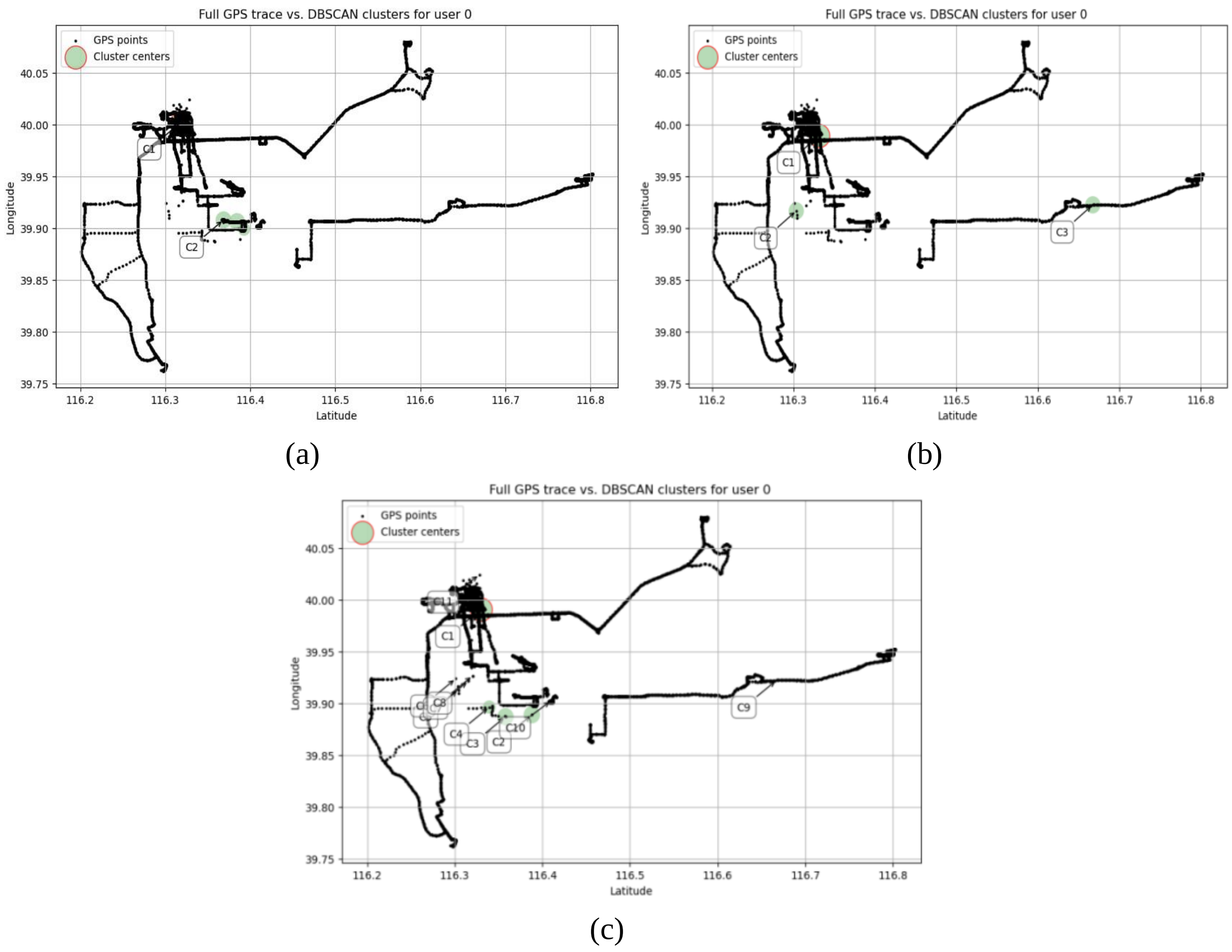


Figure 7. Identifying important locations using DBSCAN clustering algorithm ,where(a) EPS=1500 meter, MinPts=1(b) EPS=1000 meter, MinPts=1(c) EPS=500 meter, MinPts=1

Figure 7(a) illustrates the method of extracting stay points using a distance threshold of 1.5 km and a MinPts of 1. In the absence of longitude and latitude normalization, this entails identifying a stay-point when an individual remains within a 1.5 km radius for a minimum of 1 MinPts. The limited distance suggests a relatively large quantity of identified stay points, as this occurrence is likely to be common in a typical daily schedule. In contrast, **Figure 7(b)** shows the outcome of decreasing the EPS. By setting it to 1000 meters and using the same MinPts, we observe a substantial decrease in the number of identified stay-points in one cluster. **Figure 7(c)** depicts the outcome obtained by decreasing the distance threshold. When dealing with trajectories that are normalized, setting EPS = 500 meters and the same MinPts means that we identify a stay-point whenever a person remains within a 500 meter radius. We observe a substantial decrease in the number of identified stay-points in one cluster. Consequently, this leads to an increase in the number of clusters extracted compared to when the EPS is set to 1.5

KM and the EPS=1 KM. Put simply, a single stay-point records data for a greater number of longitude-latitude pairings in its raw form. The location points that contribute to a stay-point heavily influence its average position. This is why there is a big difference in the placement of stay-points in **Figure 7(c)** compared to **Figure 7(a)** and **Figure 7(b).**

Figure 8 below identifies important locations using the DBSCAN clustering algorithm, EPS= Kneedle algorithm, and the same MinPts, which determine the number of clusters and the noise. noise.

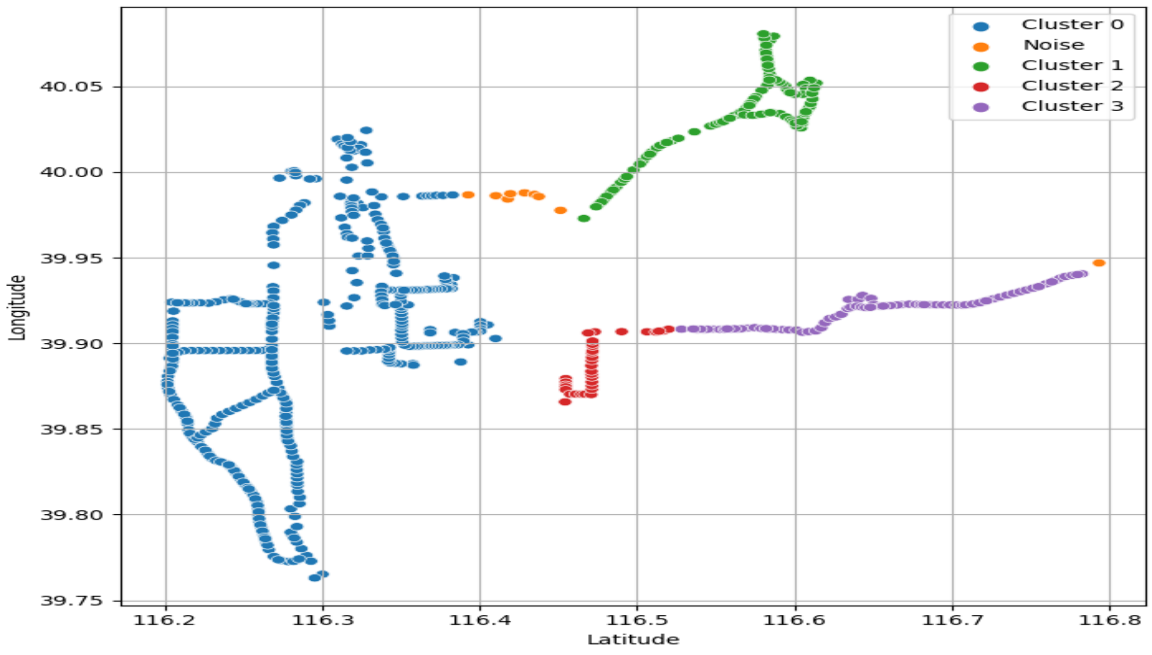


Figure 8. Identifying important locations using the DBSCAN clustering algorithm, EPS= Kneedle algorithm

Our goal is to develop a clustering algorithm that can autonomously determine an appropriate number of clusters without calculating distances and automatically excluding noise. An example of this could be going on an unusual trip only once, far from their daily routine. However, such events do not contribute to an important location and should be considered irrelevant data. Use EPS, an algorithm called Kneedle, instead of determining the distance, as shown in **Figure 8**. The algorithm "Kneedle" identifies advantageous data points that indicate optimal tradeoffs, referred to as "knees" (curves with negative concavity) or sometimes "elbows" (curves with positive concavity), in discrete data sets. We achieve this by applying the mathematical concept of curvature, typically used for continuous functions; for more details about this algorithm and its exact workings, refer to Ref [37].

Accordingly, previous analysis suggests that utilizing DBSCAN is appropriate for identifying relevant locations with the (EPS = Kneedle) algorithm. This technique's design successfully avoids the primary problems encountered by traditional clustering algorithms when dealing with spatial data. Furthermore, one can replace these EPS and MinPts parameters with various suitable methods.

5. Conclusion and Future Work

The study effectively utilized the GeoLife GPS Trajectories dataset to predict individual mobility. We then conducted a study on the paths of these locations, employing two distinct prediction models from this dataset. The prediction was evaluated using the RNN and GRU methods, resulting in a score of 30 for the epoch parameter and 64 for the batch size parameter. This demonstrates a high level of accuracy in extracting stay-points and significant location extraction using the DBSCAN algorithm. Additionally, the prediction error, measured by the mean square error (MSE), achieved a score of 0.0005 for the RNN model and 0.01 for the GRU model. Finally, dealing with diverse trajectory data is an important but difficult challenge for future research; additional efforts are required to address parameter optimization and feature weighting. Also, in order to improve location prediction, we must pay close attention to and investigate the role of social relationships and friendship. The next prediction path method described in this work continues to face numerous research constraints. Our primary goal is to anticipate a user's next location as a reference point for comparison with actual sensor data, which may have limited accuracy in determining the user's current location. This would allow us to precisely pinpoint a user's current location or the likelihood that they are currently attired. So far, we have only looked at an individual's daily routine. When this individual engages in unexpected or unusual actions, such as traveling to a foreign country during holidays, the research aims to ensure that a system remains operational. We have to determine the probability of new, unexpected transitions by examining things like the time it would take for the legitimate user to reach the new destination. In this sense, a person's significant places might change over time. For example, relocating to a new home necessitates a change in the major site associated with the home. To address this issue, it is best to restart the training process of the models, which involves reprocessing the user data. Future research could explore incremental learning, utilizing new observations not only for justification but also for additional training of the subsequent place prediction model. Future research could integrate additional metrics into the dataset used in our trials. For example, time could be considered in the future; this would be interesting, especially if a person has distinct locations that can only be distinguished by their time. In the future, it may be interesting to explore a wider variety of sensors capable of capturing position data and collecting our dataset. Additionally, we could consider using discrete observations in the future instead of continuous ones.

Acknowledgment

The authors would like to thank the University of Baghdad Department of Computer in the College of Sciences for their motivation and support.

Conflict of Interest

The authors declare that they have no conflicts of interest.

Funding

No funding**.**

References

1. Zaidi, S.M.A.; Chandola, V.; Yoo, E.-H. DST-Predict: Predicting Individual Mobility Patterns From Mobile Phone GPS Data. *IEEE Access* **2021**, *9*, 167592–167604.

https://doi.org/10.1109/ACCESS.2021.3134586

2. Mo, B.; Zhao, Z.; Koutsopoulos, H.N.; Zhao, J. Individual Mobility Prediction in Mass Transit Systems Using Smart Card Data: An Interpretable Activity-Based Hidden Markov Approach. *IEEE Transactions on Intelligent Transportation Systems* **2022**, *23(8)*, 12014–12026. https://doi.org/10.1109/TITS.2021.3109428
3. Aloubade, Y.K.A. Developing and Organizing an Easy Work Environment For Users of Computers Using Information Technology. *Ibn AL-Haitham Journal For Pure and Applied Sciences* **2022**, *35(2)*, 117–130. https://doi.org/10.30526/35.2.2808
4. Li, Q.; Zou, D.; Xu, Y. Combining Individual Travel Behaviour and Collective Preferences for next Location Prediction. *Transportmetrica A: Transport Science* **2022**, *18(3)*, 1754–1776, https://doi.org/10.1080/23249935.2021.1968066
5. Wang, X.; Fang, M.; Zeng, Z.; Cheng, T. Where Would I Go Next? Large Language Models as Human Mobility Predictors. *arXiv preprint arXiv* **2023**, arXiv:2308.15197. https://doi.org/10.48550/arXiv.2308.1519
6. AlQaysi, M.K.; Salam Ali, S.F.B. Spatiotemporal Modeling in Wireless Communication Networks. *Baghdad Science Journal* **2022**, *20(3)*, 0904-0904. https://dx.doi.org/10.21123/bsj.2022.6848
7. Hong, Y.; Martin, H.; Raubal, M. How do you go where? improving next location prediction by learning travel mode information using transformers. *In Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, Seattle, Washington, **2022**, Article No.: 61, 1-10. https://doi.org/10.1145/3557915.3560996
8. Alqaysi, M.K.; Behadili, S.F. Gravity Model for Flow Migration within Wireless Communication Networks. *Iraqi Journal Science* **2022**, *63(10)*, 4474–4487. https://doi.org/10.24996/ijs.2022.63.10.32
9. De Barrena, T.F.; Ferrando, J.L.; García, A.; Badiola, X.; de Buruaga, M.S.; Vicente, J. Tool Remaining Useful Life Prediction Using Bidirectional Recurrent Neural Networks (BRNN). *The International Journal of Advanced Manufacturing Technology* **2023**, *125(9)*, 4027–4045. https://doi.org/10.1007/s00170-023-10811-9
10. Mijwil, M.; Salman Shukur, B. A Scoping Review of Machine Learning Techniques and Their Utilisation in Predicting Heart Diseases. *Ibn AL-Haitham Journal For Pure and Applied Sciences* **2022**, *35(3)*, 175–189. https://doi.org/10.30526/35.3.2813
11. Mo, B.; Zhao, Z.; Koutsopoulos, H.N.; Zhao, J. Individual Mobility Prediction: An Interpretable Activity-Based Hidden Markov Approach. *arXiv preprint arXiv* **2021**, arXiv :2101.03996. https://doi.org/10.48550/arXiv.2101.03996
12. Chen, P.; Shi, W.; Zhou, X.; Liu, Z.; Fu, X. STLP-GSM: A Method to Predict Future Locations of Individuals Based on Geotagged Social Media Data. *International Journal of Geographical Information Science* **2019**, *33(12)*, 2337–2362. https://doi.org/10.1080/13658816.2019.1630630
13. Asad, S.M. User Mobility Prediction and Management Using Machine Learning, Doctoral dissertation, University of Glasgow, **2022**. https://doi.org/10.5525/gla.thesis.83181
14. Yan, M.; Li, S.; Chan, C.A.; Shen, Y.; Yu, Y. Mobility prediction using a weighted Markov model based on mobile user classification. *Sensors* **2021**, *21(5)*, 1740. https://doi.org/10.3390/s21051740
15. Luca, M.; Pappalardo, L.; Lepri, B.; Barlacchi, G. Trajectory Test-Train Overlap in next-Location Prediction Datasets. *Machine Learning* **2023**, *112(11)*, 4597–4634. https://doi.org/10.1007/s10994-023-06386-x
16. Graser, A.; Jalali, A.N.; Lampert, J.; Weißenfeld, A.; Janowicz, K. Deep Learning From Trajectory Data: a Review of Deep Neural Networks and the Trajectory Data Representations to Train Them. *In Proceedings of the EDBT/ICDT Workshops*, **2023**.
17. Liao, D.; Liu, W.; Zhong, Y.; Li, J.; Wang, G. Predicting Activity and Location with Multi-Task Context Aware Recurrent Neural Network. *IJCAI Int. Jt. Conf. Artif. Intell.* **2018**, 3435–3441. https://doi.org/10.24963/ijcai.2018/477
18. Gao, Q.; Zhou, F.; Trajcevski, G.; Zhang, K.; Zhong, T.; Zhang, F. Predicting human mobility via variational attention. *In Proceedings of the The world wide web conference.* **2019**, 2750-2756.

https://doi.org/10.1145/3308558.331361

19. K, Prinz. Next Place Prediction with Hidden Markov Models. Master's thesis, Johannes Kepler University Linz, Institute of Networks and Security, Linz, Austria, **2019**.
20. Nurhaida, I.; Noprisson, H.; Ayumi, V.; Wei, H.; Putra, E.D.; Utami, M.; Setiawan, H.; Dwika Putra, E.; Utami, M.; Setiawan, H. Implementation of Deep Learning Predictor (LSTM) Algorithm for Human Mobility Prediction. *International Journal of Interactive Mobile Technologies (iJIM)* ***2020****, 14(18)*, 132–144. https://doi.org/10.3991/ijim.v14i18.16867
21. Zheng, Y.; Wang, L.; Zhang, R.; Xie, X.; Ma, W.-Y. GeoLife: Managing and Understanding Your Past Life over Maps. *In Proceedings of the The Ninth International Conference on Mobile Data Management* , Beijing, China, **2008**, 211–212. https://doi.org/10.1109/MDM.2008.20
22. Faizan, M.; Zuhairi, M.F.; Ismail, S.; Sultan, S. Applications of Clustering Techniques in Data Mining: A Comparative Study. *International Journal of Advanced Computer Science and Applications* **2020**, *11(12)*, 146–153. https://doi.org/10.14569/IJACSA.2020.0111218
23. Yang, Y.; Qian, C.; Li, H.; Gao, Y.; Wu, J.; Liu, C.J.; Zhao, S. An Efficient DBSCAN Optimized by Arithmetic Optimization Algorithm with Opposition-Based Learning. *J. Supercomput.* **2022**, *78(18)*, 19566–19604. https://doi.org/10.1007/s11227-022-04634-w
24. Jiang, K.; Huang, Z.; Zhou, X.; Tong, C.; Zhu, M.; Wang, H. Deep Belief Improved Bidirectional LSTM for Multivariate Time Series Forecasting. *Math. Biosci. Eng.* **2023**, *20(9)*, 16596–16627. https://doi.org/10.3934/mbe.2023739
25. Z., Z.; E. A., P.A.; M. H., H. Predicting Machine Failure Using Recurrent Neural Network-Gated Recurrent Unit (RNN-GRU) through Time Series Data. *Bulletin of Electrical Engineering and Informatics* **2021**, *10(2)*, 870–878. https://doi.org/10.11591/eei.v10i2.2036
26. Shiri, F.M.; Perumal, T.; Mustapha, N.; Mohamed, R. A Comprehensive Overview and Comparative Analysis on Deep Learning Models: CNN, RNN, LSTM, GRU. *arXiv* **2023**, arXiv:2305.17473. https://doi.org/10.48550/arXiv.2305.17473
27. Ghojogh, B.; Ghodsi, A. Recurrent Neural Networks and Long Short-Term Memory Networks: Tutorial and Survey. *arXiv* **2023**, arXiv:2304.11461. https://doi.org/10.48550/arXiv.2304.11461
28. Luca, M.; Barlacchi, G.; Lepri, B.; Pappalardo, L. A Survey on Deep Learning for Human Mobility. *ACM Computing Surveys (CSUR)* **2021**, *55(1)*, 1 – 44. https://doi.org/10.1145/3485125
29. Javid, A.M.; Das, S.; Skoglund, M.; Chatterjee, S. A ReLU dense layer to improve the performance of neural networks. *In Proceedings of the ICASSP 2021—2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, ON, Canada, **2021**, 2810-2814. https://doi.org/10.1109/ICASSP39728.2021.9414269
30. Yang, J.; Cao, J.; Liu, Y. Deep Learning-Based Destination Prediction Scheme by Trajectory Prediction Framework. *Security and Communication Networks* **2022**, 8385854. https://doi.org/10.1155/2022/8385854
31. Joseph, F.J.J.; Nonsiri, S.; Monsakul, A. Keras and TensorFlow: A hands-on experience. *Advanced deep learning for engineers and scientists: A practical approach* **2021**, 85–111. https://doi.org/10.1007/978-3-030-66519-7_12
32. Reyad, M.; Sarhan, A.M.; Arafa, M. A Modified Adam Algorithm for Deep Neural Network Optimization. *Neural Computing and Applications* **2023**, *35(23)*, 17095–17112. https://doi.org/10.1007/s00521-023-08568-z
33. Bhimavarapu, U.; Battineni, G.; Chintalapudi, N. Improved Optimization Algorithm in LSTM to Predict Crop Yield. *Computers* **2023**, *12(1)*, 10. https://doi.org/10.3390/computers12010010
34. Guo, B.; Wang, K.; Yang, H.; Zhang, F.; Wang, P. A New Individual Mobility Prediction Model Applicable to Both Ordinary Conditions and Large Crowding Events. *Journal of advanced transportation* **2023**, *2023(1)*, 3463330, 1-14 . https://doi.org/10.1155/2023/3463330
35. Tajmouati, S.; Wahbi, B. El; Bedoui, A.; Abarda, A.; Dakkoun, M. Applying K-Nearest Neighbors to Time Series Forecasting : Two New Approaches. *Journal of Forecasting* **2024**, *43(5)*, 1559–1574. https://doi.org/10.1002/for.3093

36. Ouaret, R.; Birregah, B.; Soulier, E.; Auclair, S.; Boulahya, F. Random Forest Location Prediction from Social Networks during Disaster Events. *Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, Granada, Spain, **2019**, 535-540. https://doi.org/ 10.1109/SNAMS.2019.8931863.
37. Satopaa, V.; Albrecht, J.; Irwin, D.; Raghavan, B. Finding a "Kneedle" in a Haystack: Detecting Knee Points in System Behavior. In *2011 31st international conference on distributed computing systems workshops* **2011**, MN, USA, 166–171. IEEE https://doi.org/ 10.1109/ICDCSW.2011.20