J. Integr. Sci. Technol. 2024, 12(1), 701 Article

Journal of Integrated
SCIENCE & TECHNOLOGY

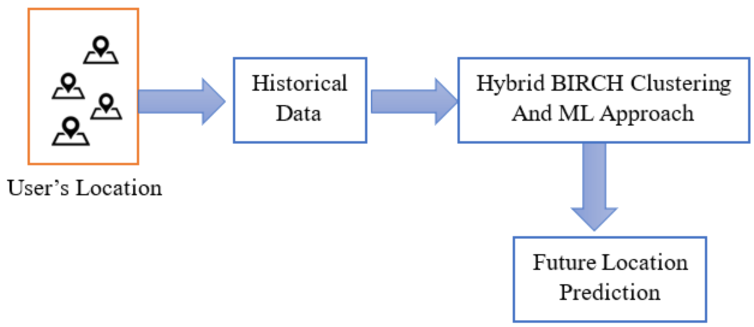
User Location prediction using Hybrid BIRCH clustering and Machine Learning approach

Madhur Arora[1*], Sanjay Agrawal[2], Ravindra Patel[1]

[1]*Department of Computer Application, UIT, Rajiv Gandhi Proudyogiki Vishwavidyalaya Bhopal, India.* [2]*Department of Computer Application, National Institute of Technical Teachers Training and Research Bhopal, India.*

Received on: 19-Jun-2023, Accepted and Published on: 03-Aug-2023

ABSTRACT



This paper discusses the importance of location prediction using machine learning methods and its potential applications in various fields. The accuracy of these models is high, but they face challenges such as data quality, model complexity, privacy concerns, and limited data availability. Despite these challenges, the future scope of location prediction is vast, and ML techniques play a crucial role in improving these models. The paper sheds light on modern techniques for better performance and highlights the difficulty of predicting a user's position in real-time, which limits the utility of location-based services. This study proposes a novel approach for predicting complete user trajectories using a Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH), based on a scalable architecture that uses clustering to reduce the search space. Bidirectional Long ShortTerm Memory (BiLSTM) with random forest classifier models are used for analyzing temporal data and predicting trajectories. The proposed approach, termed BIRCH-LSTM, outperforms other reported results on prediction accuracy. The result analysis of the proposed solution has potential applications in navigation systems, traffic management, and location-based recommendation systems.

Keywords: User Location, Trajectory, Clustering, Machine Learning

INTRODUCTION

The process of predicting the user's location on various sources of data such as Wi-Fi signals, cellular network data, user behavior patterns, and GPS data is known as user location prediction.[1,2] By advertising, notifications, and providing location-based recommendations the personalized services and user experiences will be improved. With the rise of location-based services and the growth of mobile devices location prediction has become increasingly important.[3,4] For instance, location-based advertising is on the rise because it allows businesses to reach specific audiences with appropriate ads. In addition, location-based services such as food delivery, online shopping, and ride-hailing are relying on precise location information. Using a variety of techniques, including machine learning algorithms,5 deep learning models,[6] and statistical methods the user's location can be predicted.[7,8,9] To predict the location of the user accurately, these methods use various sources of data. In location prediction, GPS data is among the most common sources of data used. By triangulating the signals from the GPS satellites, GPS sensors can accurately determine the user's location. However, to predict the user's location accurately, GPS data alone is not sufficient, especially in urban areas with

*Madhur Arora, Department of Computer Application, UIT, Rajiv Gandhi Proudyogiki Vishwavidyalaya Bhopal, India
Email: madhurarora1179@gmail.com

Cite as: *J. Integr. Sci. Technol., 2024, 12(1), 701. URN:NBN:sciencein.jist.2024.v12.701*

 ISSN: 2321-4635
http://pubs.thesciencein.org/jist

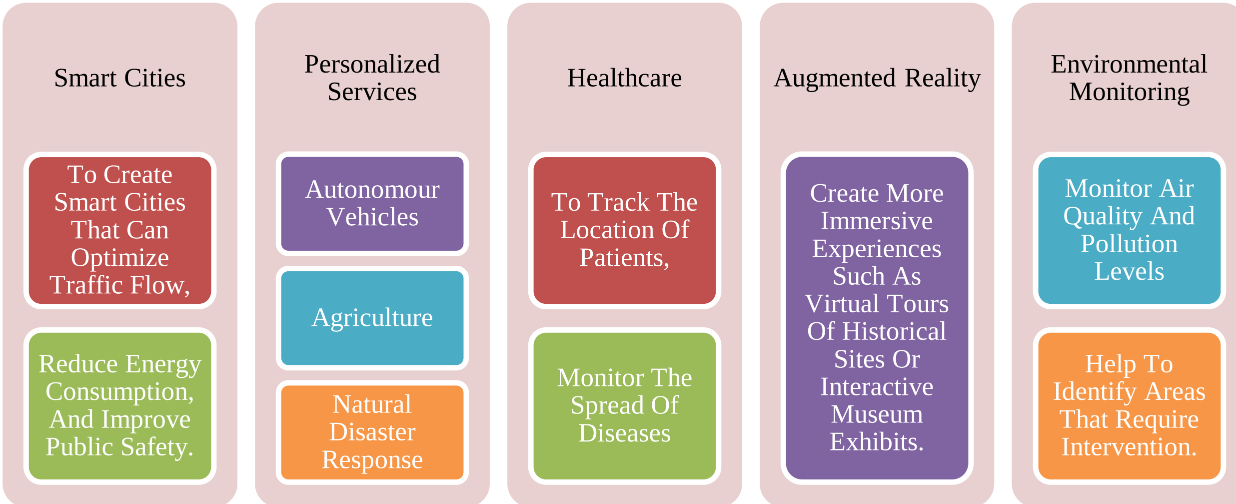


Figure 1. Application Areas Where Location Predictions are Used

indoor environments or tall buildings. In location prediction, Wi-Fi signals are another source of data used. To determine the distance from the access point and the user's location based on the signal strength, Wi-Fi access points can be used. To improve location accuracy, Wi-Fi data can be integrated with GPS data. To determine the user's location based on the cell tower signals, the cellular network can be used to provide any narrow location information.[10-17] To predict the user's location, user behavior patterns could be used. For instance, if the user frequently uses the same coffee shop or gym, that information can be used to infer where the user is.[18-22] Some of the application areas are illustrated in Fig 1 where location prediction can be applied.

In location prediction, ML algorithms are commonly used. To train a model that can forecast the user's future location these algorithms use historical location data. As new location data becomes available these models can be updated in real-time.[11,12] The collection of data is the first step in user prediction which used ML techniques. This data may include Wi-Fi signals, cell tower information, accelerometer readings, GPS coordinates, and other location-related data. To facilitate the analysis, the information is gathered in a structured format. The gathered information may contain missing values, irrelevant data, or errors. The data must be preprocessed to remove irrelevant data, remove these errors, and fill in the missing values. Normalization, transformation to prepare the data for analysis, and data cleaning are the crucial steps involved in the process.[13] The second step is to predict the user's location, selection of relevant features is used. To improve prediction accuracy, feature selection entails picking out the most relevant features. After settling on a set of features, picking an appropriate machine-learning method comes next. Numerous algorithms can be used for user location prediction, such as support vector machines, random forests, neural networks, and decision trees. After cleaning up the pre-processed data, it is used to teach the chosen machine-learning method. Prediction models are constructed with the help of the training data.[14] Using the input features, the model is trained to correctly forecast the user's location. It is assessed to assess its accuracy, once the model is trained. During the training phase, the evaluation is performed using a test dataset that has not been used. To assess the model's accuracy, including precision, recall, F1-score, and accuracy, the evaluation metrics are used. The model is tested for accuracy and then put into production to provide forecasts in real-time. The model can be used in conjunction with LBS or other apps to provide location-aware, user-specific experiences. The purpose of this paper is to explore and contrast existing ML models and methods for making reliable location predictions. Machine learning systems use statistical models to analyze this data and determine the user's most likely location. However, there are challenges to making accurate real-time predictions, including data quality, model complexity, lack of standardization in data gathering, privacy concerns, limited data availability, and changing environments.[15] Factors such as data quality, model complexity, lack of standardization in gathering geographical information, privacy concerns, limited data availability, changing environments, and user acceptance can all affect the accuracy and effectiveness of location prediction models. Despite these challenges, machine learning remains a key component in overcoming the difficulties inherent in making an accurate real-time prediction of a user's location for location-based services. Therefore, the paper provides an application of machine learning in location prediction.

The current study reports a predictive model that addresses the scaling issue in BiLSTM by applying a clustering technique to group large volumes of data into smaller clusters. This enables BiLSTM to capture local phenomena in the data without requiring high-end resources. The research objectives are to overcome scalability issues and conduct a comparative analysis of the proposed hybrid approach with traditional machine learning approaches such as SVM and k-NN. The proposed approach aims to improve the accuracy of trajectory prediction by using local clustering information using the BIRCH algorithm, and the results of the comparative analysis can be used to guide the selection of appropriate methods for future analyses.

Related Work

Jiang et al.[5] suggested a reliable approach to forecasting consumer spending on shopping. The authors used a machine learning algorithm called XGBoost to guess which stores consumers are near right now. We tested our approach using data from actual shoppers' purchases at malls, and the findings indicated that it was successful at the task at hand. There was a 92.40% rate of success. The predicted stores allow us to expand our service offerings to include more services connected to shopping, and they can be easily incorporated into our existing suite of mobile apps. The services mentioned quickly in this paper will be discussed in greater depth in our future work after we have obtained precise customer positioning. We also need to test our method's scalability on a variety of datasets to ensure its reliability.

Jiang et al.[6] examined the possibility of using machine learning methods on GPS tracking data to generate unique models characterizing patterns of movement. Dementia patients' normal whereabouts can be predicted using these patterns, and any unusual motions that may indicate aimless wandering can be identified. A total of 337 GPS devices' worth of data was analyzed. Following the completion of any necessary preprocessing, the data will be utilized for iterative clustering and then used for classification learning. The reliability of position prediction models was found to be dependent on the wearer's routine. Precision is at 0.662, recall is at 0.604, accuracy is at 0.631 and the mean AUC, or the area under the ROC, is 0.778. Secondary categorization learning was used to look for unexpected locations that might correspond to wandering episodes after filtering out data that corresponded to conventional mobility. This allowed for the identification of these unusual locations.

Long et al.[11] discussed that for location-based services, the ability to foresee the movements of specific vehicles is crucial. Two main characteristics of individual vehicles can be used to forecast their locations: regularity and preference. To combat the non-existence of sequential dependency in sparse scenarios, the suggested model augments LSTM with memory to hold all the output hidden states. To fully capture regularity and preference for prediction, a backtracking attention mechanism is created that gathers all of the crucial hidden states from the past in memory and gives each one a weight based on similarity in regularity and preference. The new hidden state generated by the weighted aggregation is then used to make the forecast. Our proposed model improves upon the state-of-the-art models in terms of prediction accuracy by a margin of 7% to 10%, as demonstrated by experiments on three real-world vehicle trajectory datasets comprising over 10,000 unique vehicles.

Sangaiah et al.[16] suggested a technique for using machine learning techniques to preserve the privacy of roaming PBS users' locations. To better accommodate roaming PBS viewers, we suggest a three-step process. combining decision trees and k-nearest neighbor, it determines the user's location and then uses hidden Markov models to predict the user's next move in conjunction with the position trace sequence. Additionally, the suggested paradigm includes the utilization of a mobile edge computing service policy, which will guarantee the on-schedule delivery of PBSs. Using networking and computing services close to roaming users, the mobile edge service policy provides low latency and position confidentiality. After extensive testing, it is determined that the proposed technique successfully protects the privacy of positions in PBSs to an extent of more than 90%.

Singh et al.[19] proposed an algorithm to find people who need assistance over Twitter. Tweets are processed by the algorithm and classified as either urgent or unimportant. The Markov model is used to infer the position of users who sent urgent but location-less tweets by looking at their past movements. Location prediction accuracy of 87% and Classification accuracy of 81% indicate that the algorithm is working well. The current system can be expanded for use in the event of both natural and man-made calamities, such as tsunamis, riots, terrorist attacks, and earthquakes. This is the first-of-its-kind system designed to use tweets to aid disaster victims.

Wojtusiak and Nia[21] aimed to predict the user's next position based on their spatial history. By contrasting the two, we can pick the most appropriate approach. This study makes use of information contributed by Beijing residents who are actively using the Geolife collection. With a 91.98 percent overall accuracy, weighted K-Nearest Neighbors (KNN) produces the best outcome. It also establishes the relationship between the model prediction and observed data. This study also presents the idea of routineness, which demonstrates the degree to which individual users' actions can be predicted and the extent to which they deviate from this prediction based on the unique spatial and temporal contexts in which they occur. To further assess the efficacy of the approaches, a comparison is made with other studies that have used the same data. Compared to other methods of its kind, the proposed technique is 2.72% more accurate in terms of predictions.

METHOD USED

Problem Identification

The notations for entities like trajectories and clusters used in the paper are shown in this section. G is initially the collection of randomized trajectory data shown as:

$$\boldsymbol{G = T_1, T_2, T_3 \dots\dots\dots\dots\dots\dots . T_N} \qquad (1)$$

where G is a set made up of all trajectories T, covering the network of the region, or N, for a specific amount of time. The sequence of the user's position coordinates is called a trajectory. The length l of each trajectory T may differ. A set of parts that are a time-ordered succession of coordinates can be used to define the trajectory T as follows:

$$\boldsymbol{T = C_1, C_2, C_3 \dots\dots\dots\dots\dots\dots . c_l} \qquad (2)$$

Clusters are created from the set of trajectories G as a whole. Each trajectory has a few clusters associated with it. $D_j$is the j_{th}cluster, T_i^jrepresents the i_{th}trajectory associated to the j_{th}cluster.

$$\boldsymbol{D_j = T_1^j, T_2^j, T_3^j \dots\dots\dots\dots\dots\dots\dots .. T_i^j} \qquad (3)$$

If a group of historical paths are given as$G = \{T_1, T_2, T_3, \dots, T_N\}$ then the objective is to foresee the place. C_{t+1}.

Methodology

The section describes the stages involved in the proposed framework for forecasting a mobile user's next location. The framework covers feature point extraction, identification of significant locations, and prediction. Figure 2 depicts the whole training procedure. This section provides a method for forecasting the upcoming position based on prior trajectories. The strategy entails grouping past trajectories based on similarities and building a deep learning model (BiLSTM) for each group to predict future destinations. To form clusters and pick cluster leaders, item-ploys the BIRCH (Balanced Iterative Reducing and Clustering Using Hierarchies) technique. Representative trajectories (RT) are created as cluster heads from the streamed data that is grouped using the BIRCH clustering approach. The BiLSTM architecture is developed separately for each cluster and trained with member trajectories. When a partial trajectory is queried, a classifier is used to evaluate which cluster's RT is the most similar and the trajectory is fed as input to the BiLSTM model of that cluster to predict the next location. This section outlines the methodology used in the paper. This is split into the training and prediction phases. The trajectory data is clustered based on geographic locations during the training phase, and the model is trained using the clustered data.

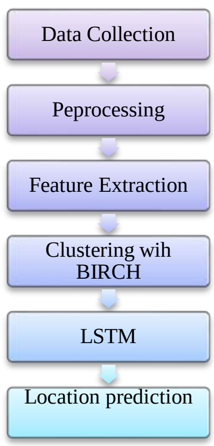


Figure 2. Proposed Flowchart for User Location Prediction System using Trajectories

Trajectory Clustering

K-means clustering, a common clustering approach, has drawbacks when processing huge datasets with few resources. This problem is addressed by the Balanced Iterative Reducing and Clustering utilizing Hierarchies (BIRCH) algorithm, which creates Clustering Feature (CF) entries, a condensed summary of the huge dataset. CF entries are dense regions defined by an ordered triple (N, LS, SS), N is the total number of data points, LS is their linear sum, and SS is their squared sum for the cluster. CF entries can also contain other CF entries. BIRCH algorithm clusters the CF entries instead of the larger dataset, and the resulting CF tree is a compact representation of the dataset. A sub-cluster, a pointer to a child node, and a CF entry made up of the total of CF entries in the child nodes are all present in each leaf node of the CF tree. BIRCH can be used in conjunction with other clustering algorithms. The parameters of the BIRCH algorithm include the threshold, which is the maximum number of entries in each leaf node, the branching factor, which is the maximum number of child nodes for each external node in the CF tree, and the radius, which determines the maximum diameter of the sub-clusters. These parameters need to be optimized based on the specific dataset being clustered.

The threshold, branching factor, and n_clusters are the three crucial parameters of the BIRCH algorithm. A sub-cluster in the leaf node of the CF tree can retain a certain number of data points at a time, which is specified by the threshold. The maximum number of CF sub-clusters that can be in each internal node of the CF tree is determined by the branching factor. The number of clusters the algorithm should return following the last clustering step is indicated by the n_clusters argument. The final clustering phase is skipped if this parameter is set to None; instead, intermediate clusters are returned.

Figure 3 gives a general summary of BIRCH. Global Clustering, Optional Refining, Optional Condensing, and Loading are the four stages of the BIRCH algorithm. By scanning all data that fits under the memory limit, the loading phase develops an initial in-memory CF-tree and generates a summary of the data with deleted outliers and grouped subclusters. During the global clustering phase, all of the leaf entries are grouped across the boundaries of various nodes using a global or semi-global clustering method. The second phase, Optional Condensing, rebuilds a more compact CF tree by deleting more outliers and consolidating more packed subclusters into bigger ones. The fourth phase, Optional Refining, refines the clusters even more by correcting minor errors and redistributing data points to their nearest seed. Phase 4 uses the centroids of the clusters created by Phase 3 as seeds to acquire a set of new clusters, allowing points belonging to a cluster to migrate and making sure that all copies of a given data point move to the same cluster. Phase 3 only scans the original data once. Algorithm 1 presents the BIRCH Clustering algorithm.

Algorithm 1: BIRCH Algorithm

Input
D= $\{t1, t2, t3 \ldots\ldots tn\}$ // Set of elements
T // Threshold for CF tree construction
Output
K // Set of clusters.
BIRCH clustering algorithm:
For each $ti \in D$ do
    Determine correct leaf node for ti insertion;
    If threshold condition is not violated then
    Add ti to cluster and update CH triples;
  else
      if room to insert ti then
      insert ti as singe cluster and update CF triples;
    else
      split leaf node and redistribute CF features;
end

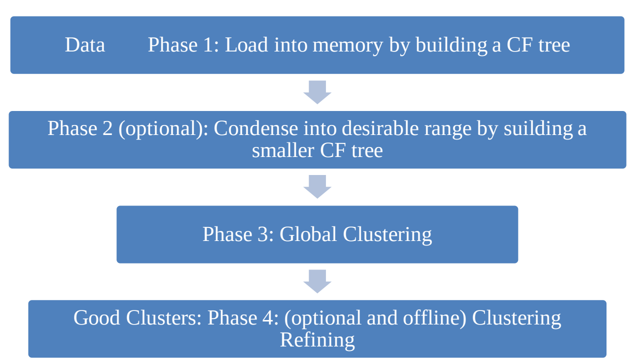


Figure 3. BIRCH Overview

Location Prediction

The paper proposes using separate BiLSTM networks for each cluster after clustering the trajectory data using the BIRCH algorithm. BiLSTM is chosen due to its ability to store long-term dependencies better than conventional prediction models like Markov Chains and basic RNNs. BiLSTM is an advanced version of LSTM based on RNN architecture, which uses "gate" devices such as input gate, output gate, forget gate, and memory unit to overcome the memory loss issues of RNN, as presented in figure 4.

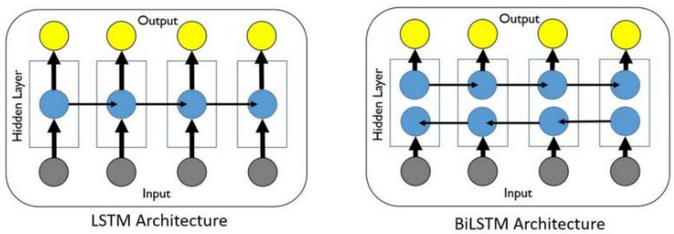


Figure 4. Network Architecture of LSTM and BiLSTM[24]

Backpropagation is used to train the weighted connections connected to these gates to understand how they function. The paper explains the use of the sigmoid function as an activation function in these units:

$$\boldsymbol{S(x) = \frac{1}{1+\ e^{x}}} \tag{4}$$

Input x_t is the input at t_{th} time instance, and the output is represented as eqn (9), where W_f is the weight factor and b_f is the deviations.

$$\boldsymbol{S(x) = \frac{1}{1+\ e^{x}}} \tag{5}$$

$$f_t = \sigma\,(W_f * [h_{t-1}, x_t] + b_f) \tag{6}$$

The input gate controls the addition of new data from $h_{t-1}$and $x_t$and remembers new data. The output i_t and C_t are formed from the inputs h_{t-1} and x_t, represented mathematically as:

$$\boldsymbol{i_t = \sigma\,(W_i * [h_{t-1}, x_t] + b_i)} \tag{7}$$

$$C_t = \tanh(W_c * [h_{t-1}, x_t] + b_c) \tag{8}$$

The previous unit information, C_t−1 is multiplied with $f_t$to evaluate the current information after passing from memory units and add it with $i_t$and $\hat{c}_t$ to identify the information in memory. It is represented mathematically as:

$$\boldsymbol{C_t = (f_t * C_{t-1} + i_t * \hat{c}_t)} \tag{9}$$

Finally, the output gate determines the $h_t$passed to the next unit through the sigmoid function and the tanh function as:

$$\boldsymbol{O_t = \sigma(W_o * [h_{t-1}, x_t] + b_o)} \tag{10}$$

$$h_t = o_t * \tanh(C_t) \tag{11}$$

During the prediction phase, the representative cluster is computed using the BIRCH algorithm for each trajectory. The subsequent position is then predicted by the associated LSTM model of that cluster and added to the trajectory. The process is continued until the entire trajectory can be predicted using this updated trajectory and the best-matching cluster estimate. This sequential technique enables the model to take into account long-term dependencies and more precisely estimate future positions.

Result and Discussions

The results for the suggested model are presented in this section. The Python platform is used to implement the model. In this study, TensorFlow was used to implement and train the models in the Keras framework. On Google Colab, the proposed model was trained using a GPU. The effectiveness of the model is assessed using the performance parameters below in the following ways:

$$\boldsymbol{Accuracy = \frac{TruePositive + TrueNegative}{TruePositive + TrueNegative + FalsePositive + FalseNegative}} \tag{12}$$

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{13}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{14}$$

$$F1\_score = \frac{2}{1/precision + 1/Recall} \tag{15}$$

For result analysis, we used the geolife dataset.[25,26] The dataset introduces a social networking service called GeoLife, which incorporates GPS trajectories and locations to build three graphs: location-location, user-location, and user-user. The user-location graph shows the times at which users have visited a specific location, the user-user graph shows users who have frequented the same location, and the location-location graph shows users who have often traveled between two locations. Figure 5 compares the accuracy of four machine learning models - LSTM, LSTM+RF, LSTM+kNN, and LSTM+SVM - for different users, represented by User IDs. The LSTM+RF and LSTM+kNN models perform the best overall, while the LSTM and LSTM+SVM models also perform well but with slightly lower accuracy. Figure 6 represents the precision comparison of machine learning models. LSTM+RF model appears to perform the best in terms of precision, achieving the highest precision for most users. The LSTM+kNN and LSTM+SVM models also perform well but with slightly lower precision. Figure 7 represents the recall comparison of machine learning models. LSTM+RF model appears to perform the best in terms of recall, achieving the highest recall for most users. The LSTM+kNN and LSTM+SVM models also perform well but with slightly lower recall. Figure 8 represents the f1_score comparison of machine learning models. In terms of F1_score, it appears that the LSTM+RF model performs the best, garnering the highest score from the majority of users. The F1 scores for the LSTM+kNN and

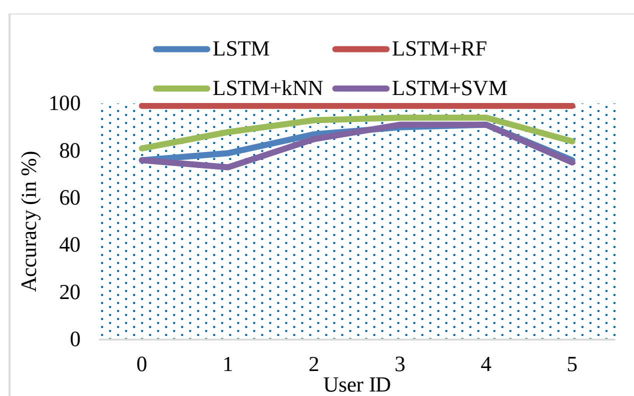


Figure 5. Accuracy Comparison of Models

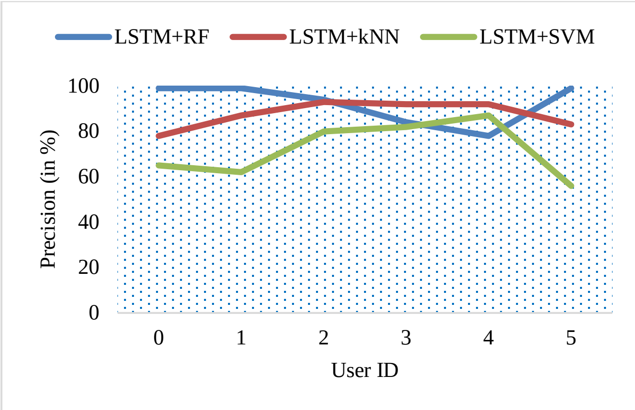


Figure 6. Precision Comparison of Models

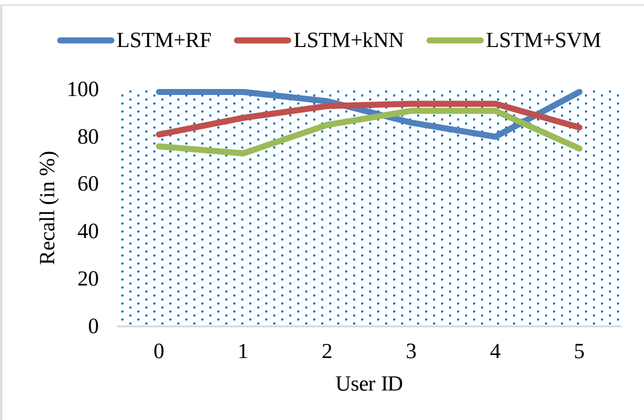


Figure 7. Recall Comparison of Models

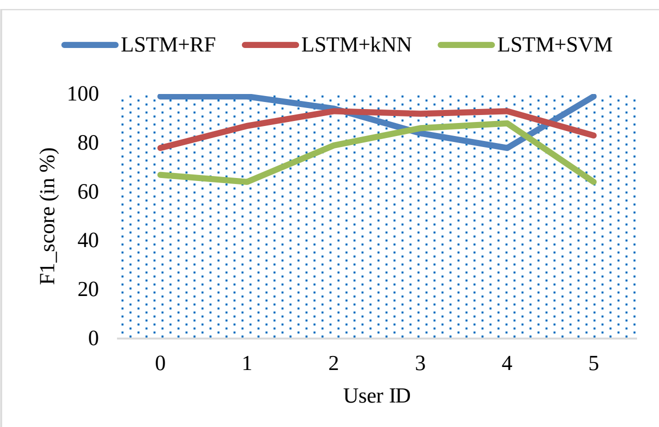


Figure 8. F1_score Comparison of Models

LSTM+SVM models are a little bit lower, but they still perform well.

Table 1 provides a comparison of the accuracy of prior art models and the proposed BIRCH-BiLSTM approach for trajectory prediction. Models compared in the table include XgBoost,[7] GP,[13] and weighted k-NN.[4] The proposed BIRCH-BiLSTM approach achieves 99% accuracy, which outperforms the other models listed in the table. This comparison demonstrates the effectiveness of the proposed approach for improving the accuracy of trajectory prediction and highlights its potential for practical applications such as autonomous vehicles and surveillance systems.

Table 1. Comparative State-of-Art Models

Models	Accuracy
XgBoost [7]	90%
GP [13]	84%
Weighted k-NN [4]	97%
BIRCH-BiLSTM	99%

Discussion

This paper discusses the importance and challenges of location prediction using machine learning methods in various fields. The research gaps identified the scalability issue while forecasting, lower reliability in predicting, and lower efficiency of the clustering approach.27-30 Addressing these gaps, the paper proposes a novel approach called BIRCH-LSTM for predicting complete user trajectories using a combination of BIRCH clustering and Bidirectional Long Short Term Memory (BiLSTM) models. The BIRCH-LSTM approach for location prediction offers advantages such as improved accuracy, scalability, complete trajectory prediction, temporal analysis, versatility, adaptability, and reduced search space, making it suitable for real-time applications and various domains.

Conclusion

This paper reports a method for predicting user location using historical trajectories. The proposed approach involves clustering the data into several groups using the BIRCH algorithm and then predicting using deep learning techniques, specifically BiLSTM, on each of the clusters. The clustered approach improves predictive performance by reducing data variance and overfitting, resulting in faster convergence in BiLSTM architecture. Therefore, the hybrid approach of BIRCH and BiLSTM helps to overcome the scalability issue and improve the accuracy of trajectory prediction. The results of the comparative analysis show that the proposed approach has the potential to enhance the accuracy of trajectory prediction and overcome scalability issues in BiLSTM-based models. The approach satisfies the requirement for real-time predictions and is scalable with the use of a big data environment framework. This paper shows the future opportunities to deploy this model over the cloud and use it as a base framework for the development of more modular and extensive trajectory prediction systems. The presented work is straightforward, intuitive, highly scalable, and robust. Future work could focus on cloud deployment, hyperparameter tuning, alternative clustering algorithms, additional feature integration, diverse dataset evaluations, model interpretability, real-time adaptation, privacy considerations, ensemble approaches, real-world applications, and trajectory generation.

Conflict of Interest: None

References

1. A.G. Chekol, M.S. Fufa. A survey on next location prediction techniques, applications, and challenges. EURASIP J. Wirel. Commun. Netw. **2022**, (1), 29.

2. S. Durga, R. Nag, E. Daniel. Survey on Machine Learning and Deep Learning Algorithms used in Internet of Things (IoT) Healthcare. In 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC); **2019**; pp 1018–1022.
3. R. Choudhary, H.K. Gianey. Comprehensive Review On Supervised Machine Learning Algorithms. In *2017 International Conference on Machine Learning and Data Science (MLDS)*; **2017**; pp 37–43.
4. A. Hamoudzadeh, S. Behzadi. Predicting user's next location using machine learning algorithms. *Spat. Inf. Res.* **2021**, 29 (3), 379–387.
5. H. Jiang, M. He, Y. Xi, J. Zeng. Machine-learning-based user position prediction and behavior analysis for location services. *Inf.* **2021**, 12 (5), 1–15.
6. W. Jiang. Bike sharing usage prediction with deep learning: a survey. *Neural Comput. Appl.* **2022**, 34 (18), 15369–15385.
7. H. Jin, H. Wu, Z. Xu, W. Huang, C. Liu. Travel-mode classification based on GPS-trajectory data and geographic information using an XGBoost classifier. *IOP Conf. Ser. Earth Environ. Sci.* **2022**, 1004 (1).
8. V. Koolwal, K.K. Mohbey. A comprehensive survey on trajectory-based location prediction. *Iran J. Comput. Sci.* **2020**, 3 (2), 65–91.
9. J. Latif, C. Xiao, A. Imran, S. Tu. Medical Imaging using Machine Learning and Deep Learning Algorithms: A Review. In *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*; **2019**; pp 1–5.
10. Y. Liu, R. Jia, J. Ye, X. Qu. How machine learning informs ride-hailing services: A survey. *Commun. Transp. Res.* **2022**, 2, 100075.
11. W. Long, T. Li, Z. Xiao, et al. Location Prediction for Individual Vehicles via Exploiting Travel Regularity and Preference. *IEEE Trans. Veh. Technol.* **2022**, 71 (5), 4718–4732.
12. M. Mohammadi, A. Al-Fuqaha. Enabling Cognitive Smart Cities Using Big Data and Machine Learning: Approaches and Challenges. *IEEE Commun. Mag.* **2018**, 56 (2), 94–101.
13. F. Namdarpour, M. Mesbah, A.H. Gandomi, B. Assemi. Using genetic programming on GPS trajectories for travel mode detection. *IET Intell. Transp. Syst.* **2022**, 16 (1), 99–113.
14. S. Olugbade, S. Ojo, A.L. Imoize, J. Isabona, M.O. Alaba. A Review of Artificial Intelligence and Machine Learning for Incident Detectors in Road Transport Systems. *Math. Comput. Appl.* **2022**, 27 (5), 77.
15. S. Ray. A Quick Review of Machine Learning Algorithms. In *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*; **2019**; pp 35–39.
16. A.K. Sangaiah, D.V. Medhane, T. Han, M.S. Hossain, G. Muhammad. Enforcing Position-Based Confidentiality With Machine Learning Paradigm Through Mobile Edge Computing in Real-Time Industrial Informatics. *IEEE Trans. Ind. Informatics* **2019**, 15 (7), 4189–4196.
17. R. Saravanan, P. Sujatha. A State of Art Techniques on Machine Learning Algorithms: A Perspective of Supervised Learning Approaches in Data Classification. In *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*; **2018**; pp 945–949.
18. A. Shrivastava, J.P. V Verma, S. Jain, S. Garg. A deep learning based approach for trajectory estimation using geographically clustered data. *SN Appl. Sci.* **2021**, 3 (6), 597.
19. J.P. Singh, Y.K. Dwivedi, N.P. Rana, A. Kumar, K.K. Kapoor. Event classification and location prediction from tweets during disasters. *Ann. Oper. Res.* **2019**, 283 (1), 737–757.
20. Z. Ullah, F. Al-Turjman, L. Mostarda, R. Gagliardi. Applications of Artificial Intelligence and Machine learning in smart cities. *Comput. Commun.* **2020**, 154, 313–323.
21. J. Wojtusiak, R. Mogharab Nia. Location prediction using GPS trackers: Can machine learning help locate the missing people with dementia? *Internet of Things* **2021**, 13, 100035.
22. R. Wu, G. Luo, J. Shao, L. Tian, C. Peng. Location prediction on trajectory data: A review. *Big Data Min. Anal.* **2018**, 1 (2), 108–127.
23. X. Zheng, J. Han, A. Sun. A Survey of Location Prediction on Twitter. *IEEE Trans. Knowl. Data Eng.* **2018**, 30 (9), 1652–1671.
24. Mohan, Arvind T., and Datta V. Gaitonde. A deep learning based approach to reduced order modeling for turbulent flow control using LSTM neural networks. *arXiv preprint arXiv*. **2018**, 1804.09269.
25. Y. Zheng, X. Xie, W.-Y. Ma. GeoLife: A Collaborative Social Networking Service among User, Location and Trajectory. *IEEE Data Eng. Bull.* **2010**, 33, 32–39.
26. Y. Zheng, L. Zhang, X. Xie, W.-Y. Ma. Mining Interesting Locations and Travel Sequences from GPS Trajectories. In *Proceedings of the 18th International Conference on World Wide Web*; WWW '09; Association for Computing Machinery, New York, NY, USA, **2009**; pp 791–800.
27. Kandhare, Pravinkumar Gangadharrao, Arie Nakhmani, and Nikolay Metodiev Sirakov. Deep learning for location prediction on noisy trajectories. *Pattern Analysis and Applications*. **2023**, 26(1), pp. 107-122.
28. Huang, Renhao, Hao Xue, Maurice Pagnucco, Flora Salim, and Yang Song. Multimodal trajectory prediction: A survey. *arXiv preprint arXiv*. **2023**, 2302.10463.
29. Qin, Yan, Yong Liang Guan, and Chau Yuen. Spatiotemporal capsule neural network for vehicle trajectory prediction. *IEEE Transactions on Vehicular Technology*. **2023**.
30. Hussain, Lana Alhaj, Shakti Singh, Rabeb Mizouni, Hadi Otrok, and Ernesto Damiani. A predictive target tracking framework for IoT using CNN–LSTM. *Internet of Things,* **2023,** 22, 100744.