FOREX Publication
Open Access | Rapid and quality publishing

International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 11, Issue 2 | Pages 639-645| e-ISSN: 2347-470X

Machine Learning Technique for Predicting Location

Madhur Arora[1], Sanjay Agrawal[2] and Ravindra Patel[3]

[1]*Department of Computer Application, UIT, Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal, India, madhurarora1179@gmail.com*
[2]*Department of Computer Application, National Institute of Technical Teachers Training and Research, Bhopal, India, sagrawal@nitttrbpl.ac.in*
[3]*Department of Computer Application, UIT, Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal, India, ravindra@rgtu.net*

***Correspondence:**Madhur Arora, Email: *madhurarora1179@gmail.com*

ABSTRACT- In the current era of internet and mobile phone usage, the prediction of a person's location at a specific moment has become a subject of great interest among researchers. As a result, there has been a growing focus on developing more effective techniques to accurately identify the precise location of a user at a given instant in time. The quality of GPS data plays a crucial role in obtaining high-quality results. Numerous algorithms are available that leverage user movement patterns and historical data for this purpose. This research presents a location prediction model that incorporates data from multiple users. To achieve the most accurate predictions, regression techniques are utilized for user trajectory prediction, and ensemble algorithmic procedures, such as the random forest approach, the Adaboost method, and the XGBoost method, are employed. The primary goal is to improve prediction accuracy. The improvement accuracy of proposed ensemble method is around 21.2% decrease in errors, which is much greater than earlier systems that are equivalent. Compared to previous comparable systems, the proposed system demonstrates an approximately 15% increase in accuracy when utilizing the ensemble methodology.

Keywords: Random Forest model, XGBoost model, Adaboost model, Ensemble Technique, encoder-decoder, Location prediction, Trajectory Prediction, GPS trajectory data, Geolife Dataset, LSTM.

ARTICLE INFORMATION

Author(s): Madhur Arora, Sanjay Agrawal, Ravindra Patel;

Received: 09/03/2023; **Accepted**: 13/06/2023; **Published**: 30/06/2023;
e-ISSN: 2347-470X;
Paper Id: IJEER-2023_189;
Citation: 10.37391/IJEER.110254
Webpage-link:
https://ijeer.forexjournal.co.in/archive/volume-11/ijeer-110254.html

crossref member
CROSSREF.ORG
THE CITATION LINKING BACKBONE

This article belongs to the Special Issue on **Mobile Computing assisted by Artificial Intelligent for 5G/ 6G/ Radio Communication**

Publisher's Note: FOREX Publication stays neutral with regard to Jurisdictional claims in Published maps and institutional affiliations.

1. INTRODUCTION

With the global proliferation of cellphones and location-based services, the volume of mobility data has increased tremendously. The enormous amount of mobility data provides new potential for analyzing and forecasting human movement [1]. People are increasingly utilizing mobile devices to acquire information on where they travel and how they get there (such as GPS data, Wi-Fi signals and bus-trip records, as well as credit card transactions and online social network check-in data). Due to the volume of mobility data. It is now feasible to recognize and predict the properties of human movement patterns. Human mobility forecasting is critical for a wide range of modern applications, including customized recommendation systems, intelligent transportation, urban planning, and mobility management in the fifth generation (5G) mobile communication system. [2].
A model for location prediction has been created as a consequence of the study mentioned that takes into account for social interactions [3]. These models are used for the regression of multiuser trajectories through the process of machine learning. The user's trajectory locations, as well as his location history, are significant in forecasting the user's next movement in this process. Our study will be based on a well-known dataset available from Kaggle or the UCI repository that provides information on Beijing City's latitude and longitude. Our study will center on the development of an algorithm for predicting the future position of the user by applying machine learning methods to a dataset that is comprised of trajectory data. This algorithm will be based on our work. Following this, will assess the accuracy of our model's predictions with regard to a variety of different prediction parameters. In addition to this, the suggested model will be evaluated alongside other models already in use. The primary focus would be on ensemble learning algorithms, which bring together a number of distinct bases to optimize regression performance and produce the most precise prediction possible, regression models are refined. The implications of given research will have highly positive implications for the successful functioning of context-aware mobile apps [4].

There are many more applications, such as when building a new restaurant and needing to predict hot spots where customers normally visit. Many e-commerce businesses and service providers like Uber, OLA, etc. depend on the idea of providing services on time at customers' doors [5]. An improved location prediction mechanism in devices would meet consumers' expectations and raise organization

FOREX Publication
Open Access | Rapid and quality publishing

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 11, Issue 2 | Pages 639-645| e-ISSN: 2347-470X

profitability. Here we are using ensemble technique to make trajectory dataset with highest accuracy and precision. By using methodology only needs a few easy steps.

- Data Preparation - Acquire, preprocess, and split the Dataset, predict the same thing, handling missing values, outliers, and performing feature selection and normalization.
- Model Development - Develop diverse predictive models using various machine learning algorithms, ensuring consistent input features and training them on the prepared dataset.
- Ensemble Technique - Apply ensemble methods (e.g., averaging, voting) to combine predictions from multiple models, leveraging their collective intelligence for improved accuracy.
- Result Analysis - Evaluate individual model performance and ensemble system effectiveness using metrics (e.g., mean absolute error, root mean square error, accuracy) to identify estimated outcomes.

For the proposed approach, it has to resolve the dataset with other predictive models like linear regression the assumption that the dependent variable and independent variables are linearly related. In the actual world, data separation is seldom linear [6]. It often incorrectly believes that there is a linear connection between the dependent and independent variables. Why we are not using decision tree because A little alteration to the data may result in a substantial alteration to the structure of the decision tree, resulting in instability; this is not the ideal approach for location prediction [7]. Since training the model takes more time. LSTMs need more time to train [8]. Training LSTMs requires greater memory. Dropout is far more difficult to implement in LSTMs. LSTMs are vulnerable to random initializations of weight. Disadvantages of encoder decoder for location prediction is that in semantic picture segmentation, encoder-decoder frameworks are popular [9-12]. However, encoder-decoder models are plagued by two major issues. The first is a structural stereotype that is anchored in imbalanced receptive fields within such frameworks. Deeper neural networks often confront the infamous issue of disappearing gradients due to inadequate learning. Structural stereotypes result in unequal learning and disparate thinking [13]. Predicting the future location of a person at a specific instant in time is an extremely difficult problem that is currently the subject of a significant amount of research. In machine learning there is conflict between minimizing bias and also reducing variation. A model is more likely to have a high degree of bias and a low degree of variation as its complexity increases. A model struggles to generalize outside of the initial training set while attempting to understand the perfect link between input and output. In the actual world, data separation is seldom linear .The following elements constitute a spatial representation of a geographic area: The focuses on defining the points of interest (POI), and its prediction algorithm attempts to anticipate the next possible point of interest. The location prediction algorithm takes into account the user's history of moving between cells to make its predictions about the user's subsequent location. Because of this, our approach work is based on an ensemble technique, within which theXGBoost method takes up the majority of our attention [14].

1. XGBoost Model

XGBoost is an ensemble machine learning method for decision trees that use a gradient boosting architecture. Artificial neural networks usually outperform all current algorithms and frameworks in situations involving unstructured input and prediction (such as photos, text, etc.).XGBoost and GBMs (Gradient Boosting Machines) are two ensemble tree approaches that boost weak learners by using the gradient descent architecture (CARTs) [15]. The GBM framework, on the other hand, benefits from system optimization and algorithmic developments in XGBoost. In comparison to other algorithms, the XGBoost model offers the greatest balance between prediction accuracy and processing speed.

2. AdaBoost Model

The AdaBoost algorithm, abbreviated as Adaptive Boosting, is a Boosting technique used in Machine Learning as a Group Method. Adaptive Boosting derives its name from the fact that weights are redistributed to each instance, with more weights being assigned to instances that were mistakenly identified. Boosting is a supervised learning approach used to minimise bias and variation. It relies on the principle of successive learning. Each following student, with the exception of the first, is produced from previously developed pupils [16]. In other words, poor learners become strong learners. The AdaBoost algorithm, with a small exception, operates on the same principles as boosting. The most common AdaBoost approach is one-level decision trees, which are decision trees with only one split. These trees are commonly known as Decision Stumps. [17] The word "adaptive boosting" refers to the process of reassigning weights to each instance, applying higher weights to instances that were incorrectly classified. In supervised learning, boosting is used to reduce bias and variance. Its foundation is the idea of consecutive learning. Each following student, each student, with the exception of the first, is built from previously-made learners. Alternatively, weak students become strong ones. With one exception, the AdaBoost algorithm behaves similarly to boosting. Decision trees with a single level or a single split are the most used AdaBoost approach. The popular name for these trees is "Decision Stumps."

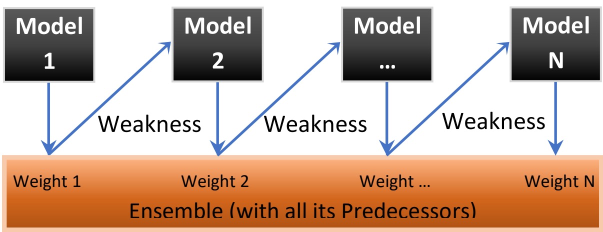

Figure 1: Understanding of Adaboost algorithm

3. Random Forest

Algorithms for supervised machine learning, such as random forest, are often used in classification and regression applications. It creates decision trees based on a variety of

FOREX Publication
Open Access | Rapid and quality publishing

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 11, Issue 2 | Pages 639-645| e-ISSN: 2347-470X

different samples and uses the results of the majority vote to classify and average the data. For combining multiple models into one, the ensemble learning method is used. It also integrates bagging and boosting methods, where bagging creates a different training subset from sample data with replacement and the final output is based on majority voting, and boosting turns weak learners into strong ones by creating sequential models, so that the final model's accuracy is at an all-time high [18].
This article is organised as follows. In *Section II*, we explain the existing system's associated efforts and a study of several technologies and methods for location prediction. The third section explains how to implement the proposed system model. Dataset collection and use, data pre-processing, problem characterization, and model training are all addressed. The results and analysis are reported in *Section IV*. Conclusions are presented in *Section V*.

2. LITERATURE REVIEW

H. Alereja et al. (2022) aim to predict the user's next position using machine learning techniques like Artificial Neural Networks and classification methods such as K-Nearest Neighbors (KNN), Support Vector Machine, and Decision Tree. The Weighted K-Nearest Neighbors (KNN) approach achieves the highest accuracy of 91.98%, outperforming similar methods by 2.72%.
V. Koolwal et al. (2020) provide a comprehensive introduction to location prediction, including terms, concepts, sources, techniques, and applications. They explore challenges, approaches, and future research directions, along with discussing applications and issues related to the user's expected future location. Their survey findings are pivotal for designing reliable location prediction systems.
Wu et al. (2018) highlighted the significance of location prediction in various applications like route navigation, dining location prediction, and traffic planning. They explored current methods of location prediction, evaluated their pros and cons, discussed related problems, and proposed future research directions.

Table1.Research Analysis of Related Literature Review for Individual User Location Prediction

Year	Author	Title	Dataset	Approaches	Results
2020	H. Alireja et al.	Predicting user's next location using machine learning algorithm	Geolife dataset (city of Beijing)	Artificial Neural Networks, K-Nearest Neighbors (KNN), Support Vector Machine, Decision Tree	Weighted K-Nearest Neighbors (KNN) achieves 91.98% accuracy. Routineness concept introduced for behavior predictability. 2.72% higher prediction accuracy compared to similar methods.
2020	P. Samuel	RNN-Based User Trajectory Prediction Using a Pre-processed dataset	Geolife (GPS-dataset)	Line simplification techniques, Recurrent Neural Networks (RNN) and its variants	Significant decrease in execution time from 4616s to 932s. Loss value of 0.10 achieved using LSTM-based model.
2020	A. Nawaz	GPS Trajectory Completion Using End-To-End Bidirectional Convolutional Recurrent Encoder-Decoder Architecture with Attention Mechanism	Microsoft Geolife trajectory dataset	Deep learning-based bidirectional convolutional recurrent encoder-decoder architecture, attention mechanism	Proposed model achieves better results in terms of average displacement error compared to state-of-the-art benchmark methods.
2018	H. Ning et al.	A Deep Learning Approach for Next Location Prediction	197 million vehicle license plate recognition (VLPR) records in Xiamen, China	Similarity mining, contextual feature modeling, CNN and bidirectional LSTM networks	Proposed method outperforms several existing methods in next location prediction using VLPR records in Xiamen, China.
2018	Ruizhi Wu et al.	Learning Individual Moving Preference and Social Interaction for Location Prediction	Two real GPS trajectory datasets: Porto taxi data and Geo-life data	Two-stage clustering, association rule mining, pair-wise ridge regression	Proposed PSI model achieves better prediction performance compared to state-of-the-art methods.
2018	Li, Lu et al.	Predicting future locations of moving objects with deep fuzzy-LSTM network	Real-world mobile phone dataset	Fuzzy-LSTM, explicit incorporation of periodic movement patterns	TrjPre-FLSTM outperforms comparative methods in prediction accuracy.
2015	M. Chen et al.	Predicting Next Locations with Object Clustering and Trajectory Clustering	Real data set	Object-clustered Markov model, Trajectory-clustered Markov model	Object Tra-MM shows significant increase in prediction accuracy compared to existing methods.
2018	Chen et al.	MPE: a mobility pattern embedding model for predicting next locations	Real-world datasets	Mobility pattern embedding model (MPE)	MPE outperforms state-of-the-art methods significantly in various tasks.

3. PROPOSED SYSTEM MODEL

The data obtained from human trajectories provide light on the movement habits and preferences of individuals in their daily lives. Mining trajectory data involves collecting, managing, and patterning data about previous travels. This is referred to as "trajectory data mining." The main objective of trajectory data mining is location prediction. Based on their present locations and past information, it analyses the movement patterns of moving objects and predicts where they will be in the future. [19] The framework for the location prediction process is shown in *figure 2*, and it consists of three stages.

(1) Because the data on trajectory was sampled from multiple positioning devices, its quality was poor; therefore, preprocessing of the data was required. In the preprocessing phase, there are several steps that include:

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 11, Issue 2 | Pages 639-645| e-ISSN: 2347-470X

(a) Noise filtering,
(b) Data cleaning, and
(c) Trajectory data compression.
(d) The segmentation of the trajectory, and
(e) The semantics of the trajectory.

The method of modelling known as learning from previous data on trajectories is an important component in the process of estimating the path that an object in motion will take as its primary focus of motion [20].

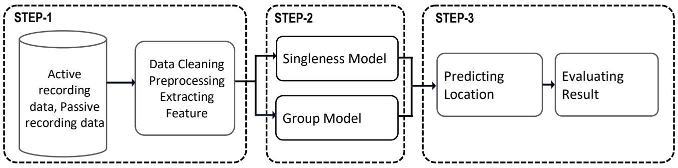

Figure 2: The merger of a new direction is the use of single object and group models in the widely used general framework of location prediction

3.1 Problem definition

As a result, we are dividing the trajectories into periods of 15 minutes; 10 minutes' worth of data will be used to predict the user's location every 30 seconds for the next 5 minutes. In contrast to the area-based strategy that was used in our study paper, we will be applying our idea throughout the whole city. The fact that we intended to extend the forecast time frame served as the impetus for all of our choices. Better practical uses may be found in having the user's position within the next five minutes. Since we are studying the trajectory over a range of 15 minutes, it would not make sense to construct a model based on an area, such as 5 kilometers by 5 km, since a user may easily change location during the course of 15 minutes. We used a sliding window strategy with a window size of 30 (*i.e.*, 30*0.5 = 15 minutes), where 20 points (10 minutes) are used for input sequence and the next 10 points (5 minutes) are target sequence, and we have used a stride of 3. As a result, 15-minute trajectories have been extracted with a regular interval of 30 seconds. After collecting trajectories, we then separated them into three parts: eighty percent Training data, ten percent Validation data, and ten percent Testing Data.

3.2 Framework

This article will examine a method known as Google Collaboratory, a free Jupyter notebook environment that operates totally in the cloud and requires no setup. Boosting techniques employ a sequential process where each model tries to fix the errors of previous models, in contrast to other ensemble approaches that rely on the simultaneous building of numerous isolated base learners that are subsequently used to produce the prediction. Due to the fact that each model is built separately and depends on the outcomes of earlier iterations, this has a significant disadvantage. [21] Google Collaborative uses the technique to reduce faults in sequential models to optimize this flaw. Therefore, it may be considered an optimization issue with the goal of minimizing the error (loss) function. Working locally or in the cloud are both options for Jupyter notebooks. Each document consists of several cells with script or markdown code in each cell, and the result is embedded in the content. Text, tables, charts, and images are common outputs.

3.3 Dataset

Dataset are using a large real-world GPS trajectory dataset collected by 182 users in the (Microsoft Research Asia) Geolife project from April 2007 to August 2012. The GPS trajectory of this data collection is represented by a series of time-stamped points, each of which comprises latitude and longitude information. There are 17,621 trajectories in this collection. These trajectories were captured by various GPS loggers and GPS phones at various sampling rates. The majority of the information was collected in Beijing by a large number of volunteer GPS loggers. As shown in the figure 4, the majority of the data is contained within Beijing's 5th Ring Road. As a result, we have chosen that data for our research. Data distribution of GPS points, with color representing point density. There are 17621 trajectories in the dataset. [22-24] the majority of the information was collected in Beijing by a large number of volunteer GPS loggers. As a result, this is selected data from Beijing, which contains complicated depicts traffic patterns and dense representations of GPS locations in *figure 3*. The crowd density ranges from 0 to 400 in the middle of Beijing.

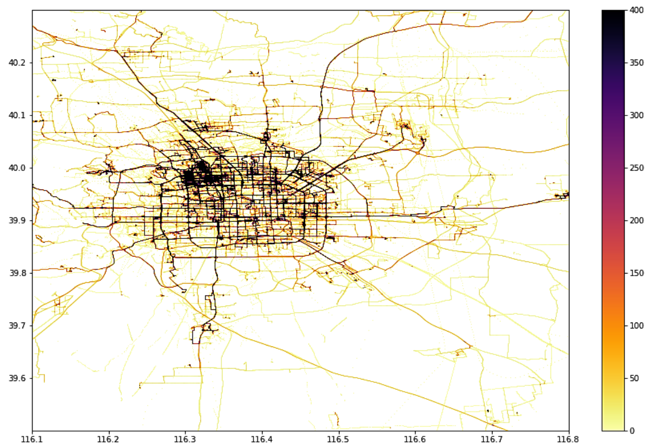

Figure 3: Data overview in Beijing

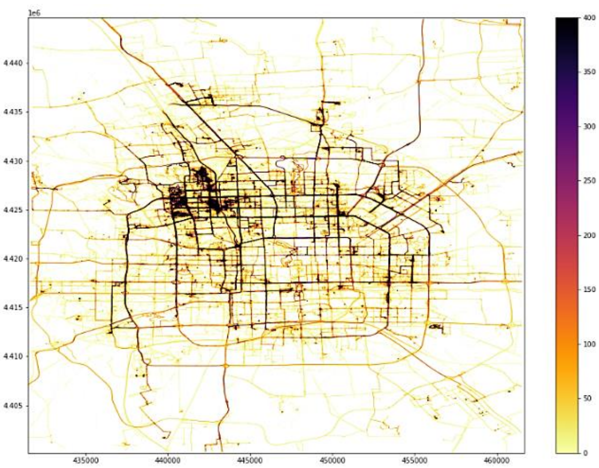

Figure 4: Data overview within the 5th ring road of Beijing

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 11, Issue 2 | Pages 639-645| e-ISSN: 2347-470X

3.4 Preprocessing

Using the framework provided by Python, as shown in figure 4, we are converting the geographic coordinates that are represented by latitude and longitude into the native map projection x, y coordinates (in meters). We are working with the UTM projection [25-26], and we are converting utilizing the UTM zone that Beijing belongs to. After the conversion, Eastings and Northings are used to measure coordinates, and the units used are meters. Therefore, we will be using them as x and y coordinates from this point on. Because the majority of the trajectories are captured at a relatively high sampling resolution (between 1 and 5 seconds), an average position reading will be taken once every 30 seconds. This will eliminate any redundant information in the data. We are only evaluating data with a speed of less than 60 kilometers per hour since the high-speed data was behaving as an outlier. Almost 90 percent of the data had a speed of less than 60 kilometers per hour.

3.5 Model Training

Then, using a tool called MinMaxScaler, which was trained on our own Training data, to standardize the data. Next, the Training data were applied to our Ensemble models so that they could be trained [27]. This can be tried out many different models, including the Random Forest model, the XGBoost model, the Adaboost model, and the Ensemble method. For the purpose of the comparison research, in addition to training the seq2seq Encoder decoder model described in this paper, this also trained several other models. Those present projections for the next five minutes at intervals of one minute and thirty seconds. In order to evaluate the model, approach uses the average distance Error that was provided in the previous research to compare models. The evaluation parameter is investigated even further in the section titled "Results and Discussion."
The instruction gave below can be summarized in the following math formula for percentage difference [28]:

*Percent decrease in Error = [(original error – newerror) / original error * 100]* (1)

Generally, decision tree, LSTM, encoder-decoder technique, and ensemble-based approach are distinct modeling frameworks with varying characteristics. Therefore, in the context of trajectory prediction, their advantages and disadvantages have been enumerated.

Table 2: Comparative Representation of Location Prediction Models with Previous Research Models

S. No.	Para-meters	Decision tree	Encoder-Decoder model	LSTM	Ensemble method
1	Model interoperability	High interoperability due to explicit rules	Work as sequence-to-sequence model & less interoperability due to its complex architecture	Less interoperability, involve multiple layers & intricate connections.	Moderate interoperability due to combination of multiple models
2	Handling Categorical& Numerical features	Handle both Categorical & Numerical features without explicit encoding	It can handle only categorical features	It requires numerical encoding for both categorical & numerical features.	Handle both categorical & numerical features withoutexplicit encoding
3	Handling sequence data	Not suitable for sequence data	Suitable for sequence-to-sequence tasks	Specifically designed for sequence data	Handle sequential data as well as capture different aspects of sequence.
4	Training time & scalability	Quick training, limited scalability	Variable training, scalable depending on architecture	Longer training, limited scalability	Training time varies, scalable with distributed computing
5	Performance and Accuracy	Accuracy can be limited by the depth of tree & the quality of features	Model perform well in capturing complex pattern but require sufficient data for training	Model perform well in sufficient training data and appropriate architecture design	Performance and accuracy improved by combining multiple models, reducing bias and variance, and capturing diverse patterns in the data

4. RESULTS AND DISCUSSION

In this paper, a strategy for applying ensemble machine learning to the regression and prediction of spatial data is suggested. The study's main goal is to merge many machine learning algorithms into a single model in order to get superior results. The data will be pre-processed before being input into the model for training, testing, and assessment, followed by the prediction of the outcome. Instead of the user to whom it belongs, the geographic information provided in the trajectory may help exploit and demonstrate the relationship between regional geographic variables and trajectories [29].

After going through the training process using linear regression and other classic machine learning techniques decision trees, LSTM, and encoder decoder approaches [30], the results were quite disappointing. Despite the fact that our ensemble techniques conducted across several users, in which the methods used include XGBoost, Adaboost, and random forest, the findings are quite accurate and exact. Every result will be presented for further consideration. *Figure 5* and *table 3* both show the distance error graph and the distance error forecast at regular intervals of 30 seconds. In a similar manner, the average distance error comparison for the various approaches is described by *figure 6* and *table 4*.

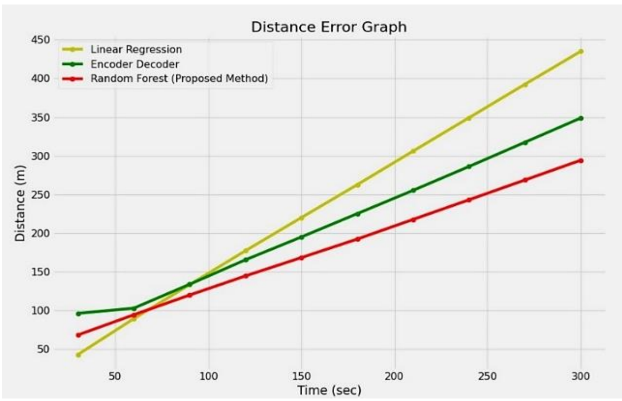

Figure 5: Distance error graph

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 11, Issue 2 | Pages 639-645| e-ISSN: 2347-470X

Finally, it suggests the result with ensemble method in which random forest, Adaboost, and XGBoost models are included that have much better error time accuracy as varied according to the distance increases and will show in *figure 5*, which has the least amount of distance error that occurs . When looking at the graph of linear regression, encoder-decoder, it is discovered that the distance error is quite high, which shows that a distance error is less significant.

To evaluate the effectiveness of our approach, a comparison with numerous cutting-edge methods for the trajectory completion problem is necessary. When compared to all benchmark models, *Table 3* shows that our XGBoost, Adaboost, and Random forest (proposed technique) design produces better outcomes in terms of the average distance error. This can be noticed by looking at the table.
According to *table 3*, our suggested method (random forest) performs better when attempting to anticipate the ten points that are necessary to finish the trajectory.

Table 3: Distance error of predictions at regular interval of 30s for different methods (Unit: m)

Methods	30s	60s	90s	120s	150s	180s	210s	240s	270s	300s
Linear Regression	43.21	89.56	133.54	177.39	220.16	262.69	306.07	349.17	392.34	434.95
Decision Tree	106.11	133.84	163.69	193.69	223.83	254.06	284.86	316.78	348.58	379.94
LSTM	98.11	124.98	156.88	187.26	214.35	243.32	271.99	303.09	335.26	368.65
Encoder Decoder	96.42	103.1	133.99	165.65	195.09	225.23	255.35	286.13	317.41	348.77
XGBoost	42.41	79.08	111.48	142.63	171.75	200.83	229.68	258.68	286.24	315.42
Adaboost	42.36	78.6	111.15	142.29	169.59	198.3	227.31	254.33	283.3	308.91
Random Forest	68.49	94.57	120.07	144.8	168.47	192.36	217.78	243.13	268.67	294.25

On a data set that is accurate environment, we will now assess the performance of a single-user prediction system that we have described. The trajectories are split into 30-point time series with a constant delta sampling time interval. The input sequence consists of the first 20 points, *fp1p2::: p10g* whereas the target sequence consists of the last 10 points. This is done in *figure 5*, for the purpose of simplicity.

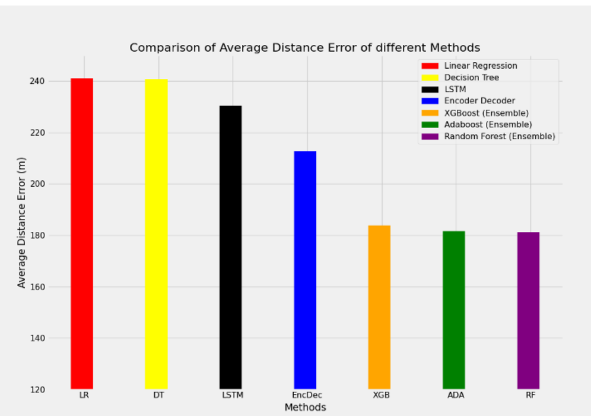

Figure 6: Comparison of average displacement error of different methods

Table 4: Average distance error of different methods (Unit: meters)

S. No.	Method	Average Distance Error
1	Linear Regression	240.91
2	Decision Tree	240.54
3	LSTM	230.39
4	Encoder Decoder	212.71
5	XGBoost	183.82
6	Adaboost	181.61
7	Random Forest	181.26

Table 4 provide a comparison of the average distance error for the different approaches and among all those methods the proposed technique (XGBoost, Adaboost, and Random forest) outperforms, demonstrating the efficacy and high reliability of our ensemble model.Our assessment metric was Average Distance Error (ADE), this is the Euclidean distance for all trajectories between the actual trajectory position and the expected trajectory location. The data utilized in this study was obtained from active users via the Geolife dataset from the city of Beijing. The XGBoost and Adaboost methods produce the greatest results, with an overall accuracy of 94.98 percent.

Where, (*x, y*) are anticipated trajectory coordinates and (*z*) are ground truth trajectory coordinates.

The decrease in error for XGBoost model is

*XGBoost = [(212.71 – 183.82)/212.71] *100 = 13.58%* (2)

The decrease in error for Adaboost model is-

*Adaboost = [(212.71 – 181.61)/212.71] *100 = 14.62%* (3)

The decrease in error for Random Forest model is-

*Random Forest = [(212.71–181.26)/212.71] *100 =14.785%* (4)

The difference in location between expected and actual placements. Particularly, the step-i error may be computed as:

$$\sqrt{\left(\overset{N}{P}_{T+i}.x - P_{T+i}.x\right)^2 + \left(\overset{N}{P}_{T+i}.y - P_{T+i}.y\right)^2} \quad (5)$$

The performance of the predictions is initially assessed for 30 seconds. *Table 3* lists the distance error for each step for several approaches, and it shows excellent consistency.

5. CONCLUSION

This study investigates the effectiveness of trajectory prediction methods for a single user and proposes practical solutions. We introduce an ensemble approach to establish a prediction system that minimizes average distance error for single-user predictions. Additionally, we propose a framework based on ensembles and region-based prediction technique for single-user prediction scenarios. Experimental results using real-world data demonstrate the superiority of our framework over competing approaches, showcasing its exceptional prediction performance, robustness, and stability for single-user prediction task. However, the study does not consider the

International Journal of
Electrical and Electronics Research (IJEER)
Open Access | Rapid and quality publishing
Research Article | Volume 11, Issue 2 | Pages 639-645| e-ISSN: 2347-470X

semantic context within the trajectory, such as points of interest. Future research could incorporate semantic information to further enhance prediction precision and performance by leveraging ensemble techniques and addressing data limitations. Notably, our recommended ensemble architecture significantly reduces average distance error compared to benchmark models.

REFERENCES

[1] Y. Zhang, X. Shi, Z. Sheng, A. Anuj, "A XGBoost based lane change prediction on time series data using feature engineering for autopilot vehicles", *In proceedings: IEEE, transaction on intelligent transportation systems,* pp: 1-14, march, 2022.

[2] R. L. Gokul, N. Revathy, S. Megha, M. Jeena, E.J. Nithaya, "Traffic flow prediction using random forest and bellman ford for best route detection", *International Journal of engineering research and technology (IJERT),* vol. 09, issue: 13, 2021.

[3] C. Zhen, F. Wei, "A freeway travel time prediction method based on an XGBoost model", *sustainability,* MDPI, 13, 2021.

[4] G. Sbeastian, M. Christoph, B. Klaus, "truck parking occupancy prediction XGBoost -LSTM model fusion", *original research article,* 2021.

[5] Chekol, A.G., Fufa, M.S. A survey on next location prediction techniques, applications, and challenges. J Wireless Com Network 2022, 29 (2022).

[6] B. Narsimha, Ch V Raghavendran, Pannangi Rajyalakshmi, G Kasi Reddy, M. Bhargavi and P. Naresh (2022), Cyber Defense in the Age of Artificial Intelligence and Machine Learning for Financial Fraud Detection Application. IJEER 10(2), 87-92. DOI: 10.37391/IJEER.100206.

[7] Manoj Kumar, Dr Pratiksha Gautam and Dr Vijay Bhaskar (2022), Effect of Machine Learning Techniques for Efficient Classification of EMG Patterns in Gait Disorders. IJEER 10(2), 117-121. DOI: 10.37391/IJEER.100211.

[8] Bahra, Nasrin, and Samuel Pierre. "RNN-based user trajectory prediction using a preprocessed dataset." *2020 16th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE, 2020.

[9] Nawaz, Asif, et al. "GPS trajectory completion using end-to-end bidirectional convolutional recurrent encoder-decoder architecture with attention mechanism." *Sensors* 20.18 (2020): 5143.

[10] G. Meena, D. Sharma and M. Mahrishi, "Traffic Prediction for Intelligent Transportation System using Machine Learning," 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE), pp. 145-148, 2020.

[11] Shah, Syed Aziz & Fan, Dou & Zhao, Nan & Yang, Xiao dong & Tanoli, Shujaat, "Seizure episodes detection via smart medical sensing system. *Journal of Ambient Intelligence and Humanized Computing",* vol. 11, 2020.

[12] O. Mbaabu, "Introduction to random forest in machine learning", *EngEd community*, Dec 2020.

[13] R.A. Asmara, B. Syahputro, D. Supriyanto and A.N. Handayani, "Prediction of Traffic Density Using YOLO Object Detection and Implemented in Raspberry Pi 3b + and Intel NCS 2," 2020 4^{th} *International Conference on Vocational Education and Training (ICOVET),* pp. 391-395, 2020.

[14] F. Xiaoliang, G. Lei, N. Han, Wang, S. Yujie, Y. Jia, Yongna, "A Deep Learning Approach for Next Location Prediction", pp: 69-74, 2018.

[15] R. Wu, G. Luo, J. Shao, Ling Tian, and C. Peng, "Location Prediction on Trajectory Data: A Review", Vol. 1, pp: 108–127, issue: 2, June 2018.

[16] R. Wu, G. Luo, Q. Yang and J. Shao, "Learning Individual Moving Preference and Social Interaction for Location Prediction", *date of current version March* 15, 2018.

[17] Fu, Zhongliang, et al. "Mining frequent route patterns based on personal trajectory abstraction." *IEEE Access* 5 (2017): 11352-11363.

[18] I. Hazan and A. Shabtai, "Improving Grid-Based Location Prediction Algorithms by Speed and Direction Based boosting", *date of current version February* 27, 2019.

[19] C. Jin, Z. Lin and M. Wu, "Augmented Intention Model for Next-Location Prediction from Graphical Trajectory Context, *Hindawi Wireless Communications and Mobile Computing*", Article ID 2860165, 2019.

[20] Rao, Kavikondala & Murugan, Tamilarasan, "An Efficient Routing Algorithm for Software Defined Networking using Bellman Ford Algorithm", *International Journal of Online and Biomedical Engineering (iJOE*), 2019.

[21] C. Zhang and P. Patras, "Long-term mobile traffic forecasting using deep spatiotemporal neural networks," In *Proceedings: MOBIHOC, Los Angeles, CA, USA,* pp. 231-240, 2018.

[22] Y. Liu and H. Wu, "Prediction of Road Traffic Congestion Based on Random Forest," *2017 10th International Symposium on Computational Intelligence and Design (ISCID),* pp. 361-364, 2017.

[23] Forsberg SKG, Bloom JS, Sadhu MJ, Kruglyak L, Carlborg Ö, "Accounting for genetic interactions improves modeling of individual quantitative trait phenotypes in yeast", *Nature Generation*. 2017.

[24] M. Pritesh, N. Daniel, T. Paul and D.V. Jonathan, "Using random forest and decision tree models for a new vehicle prediction approach in computational toxicology", *in soft computing, springer, August*, 2016.

[25] C. Serdar, L. Antonio, and M. C. González, "Understanding congested travel in urban areas", *Nature Communications,* vol. 7, 2016.

[26] Y. Han, G. Wang, and J. Yu. "A Service-Based Approach to Traffic Sensor data integration and analysis to support community-wide green commute in China," *IEEE Transactions on Intelligent Transportation Systems* 17.9, pp. 2648-2657, 2016.

[27] X. Fan and Y. Wang, "A Deep Learning Approach for Next Location Prediction", *http://xm.lanfw.com/0826/337970.html, 2015.*

[28] M. Chen, Y. Liu, and X. Yu, "NLPMM: A Next Location Predictor with Markov Modeling," *Pacific-Asia Conference on Knowledge Discovery and Data Mining Springer, Cham,* pp. 186-197, 2014.T. Minh Tri Do, O. Dousse, M. Miettinen, D. Gatica-Perez, "A probabilistic kernel method for human mobility prediction with smartphones, *Published by Elsevier* B.V., 2014.