Received: May 10, 2024. Revised: June 14, 2024. 

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

# A Hybrid Clustering Strategy for Recommending Pick-Up Locations to Cab Drivers in Cluster-Based Cab Recommender System (CBCRS)

**Supreet Kaur Mann[1]*** **Sonal Chawla[1]**

[1]*Department of Computer Science and Applications, Panjab University, Chandigarh, India*
* Corresponding author's Email: supreet@pu.ac.in

**Abstract:** The study presents a Cluster-Based Cab Recommender System (CBCRS) designed to optimize cab services by suggesting the nearest locations with a higher likelihood of finding passengers. To achieve this, the system employs advanced clustering techniques to cluster historical cab pickup locations, identifying areas with higher passenger possibilities at specific times and days. The research aims to develop an algorithmic framework for CBCRS based on a hybrid clustering technique. The objectives of the study are twofold: first, to identify current clustering techniques used in clustering cab pickup geo-points, and second, to propose a framework for CBCRS based on the most efficient clustering technique. This framework will accept the current location of the cab driver and recommend the next nearest passenger pickup location. Additionally, the study compares and contrasts the proposed system with other clustering techniques using three standard datasets, evaluating them based on intrinsic measures such as the Calinski-Harabasz Index and Silhouette-Score. The paper concludes by evaluating and contrasting the proposed CBCRS framework with different clustering techniques, analyzing the results using statistical parameters. The findings reveal that the proposed CBCRS system generates better recommendations for the cab drivers using CBCRS hybrid clustering technique as compared to K-Means, BIRCH, DBSCAN clustering algorithms.

**Keywords:** Recommender system, Clustering, Collaborative filtering, Unsupervised learning, Cab, Machine learning.

## 1. Introduction

In the realm of urban transportation, the dynamic nature of the industry, influenced by evolving consumer demands and technological advancements, has underscored the need for innovative solutions [1]. A critical challenge faced by cab driver pertains to resource allocation, specifically in optimizing pickup locations based on historical data [2]. Recommender systems deployed in cab services offer a dual perspective, benefiting both cab drivers and passengers[3]. These systems assist passengers in identifying the nearest available cab, thus enhancing convenience and efficiency [4].

The recommendations aid in making informed, value-based decisions [1]. There are three primary approaches to recommender systems[5]: Collaborative Filtering [6], Content-Based Filtering [7], and Hybrid Filtering techniques [8]. Content-based filtering utilizes item features to offer personalized recommendations to users based on their preferences, ensuring transparency and independence from user-item interactions. In collaborative filtering, the system predicts a user's preferences for items by analyzing the preferences and behaviors of other users. Hybrid filtering combines both content-based and collaborative filtering techniques, offering a comprehensive approach to recommendation systems.

Traditional methods often result in suboptimal routes, increased idle time, and higher fuel consumption for cab drivers [9]. This study addresses these issues by introducing a hybrid clustering-based approach. This approach recognizes that extracting meaningful clusters using collaborative filtering is essential for effective recommendations [6]. This paper proposes a novel Cluster-Based Cab Recommender System (CBCRS) that aims to recommend the next passenger location to cab drivers based on historical pickup data. By leveraging

clustering techniques, the system identifies high-density passenger pickup zones, thereby increasing the likelihood of finding passengers quickly and reducing idle times for drivers [10].

The remainder of this paper is organized as follows: section 2 examines existing research on cab recommender systems and clustering techniques, highlighting research gaps; section 3 outlines research objectives of the study; section 4 describes the research methodology carried to conduct the research section 5 presents and discusses experimental results and findings with performance metrics and comparative analysis; section 6 compares the research work with the state of the art and the Conclusion summarizes the findings and implications.

## 2. Literature review

Efficiently guiding cab drivers to their next pickup location is crucial for optimizing cab services in modern urban transportation. This optimization can lead to increased income for drivers and reduced fuel consumption [2]. Recent studies have investigated different methodologies and algorithms that utilize data such as passenger mobility patterns, cab driver pickup locations, spatiotemporal distribution of cab passenger demands, and cab GPS trajectories.

Yuan et al.[9] introduced a novel approach to delineate distinct functional regions within a city using human mobility patterns and Points of Interest (POI) categories. Their comprehensive framework integrated location-based services data with machine learning techniques to identify areas with diverse socio-economic activities and urban functions. The primary limitations of the taxi dispatch system research are the reliance on precise demand and destination predictions, where inaccuracies can lead to inefficiencies.

Yuan et al. [11] presented a data-driven approach to assist taxi drivers in predicting high-demand areas for passenger pickups using GPS trajectory data. Their technique involved preprocessing GPS traces to extract relevant features and applying machine learning algorithms like K-Means clustering and support vector regression to model spatial-temporal passenger demand. Major limitation of this research is the scalability of the proposed clustering methods, as their performance and effectiveness might degrade when applied to larger datasets or more complex urban environments.

A recommender system aiming to efficiently match passengers with vacant taxis was proposed by Yuan et al. [3] , leveraging GPS data to predict taxi and passenger trajectories for potential matches. Incorporating spatiotemporal features such as taxi availability and passenger demand patterns using collaborative filtering techniques, the system demonstrated effectiveness through empirical evaluation. The approach may have limited applicability to other types of location-based services or geographic regions that are not represented within the dataset, thereby limiting the generalizability of the findings.

Ma et al. [12] proposed a stop planning method for airport shuttle buses, addressing the challenge of optimizing routes based on big traffic data. Analyzing large-scale traffic data to identify optimal bus stop locations and schedules, considering factors like passenger demand and traffic flow patterns. The technique involves removing irrelevant records in the dataset and clustering relevant data points using K-Means clustering. However, other clustering methods are not implemented for a fair comparison.

Large-scale Taxi Trace mining addressed the issue of passenger wait times for vacant taxis in smart cities by Qi et al. [13]. This research paper involved analyzing extensive taxi trajectory data to study passenger waiting times and factors influencing them, such as time of day, location, and traffic conditions. The taxi spots were defined by clustering the pickup locations using Density-Based Spatial Clustering of Applications with Noise (DBSCAN) Clustering. The prediction accuracy is constrained by the variability and granularity of the collected GPS data, potentially leading to less reliable predictions in sparsely populated areas or during off-peak hours

The passenger pickup patterns were explored for taxi location recommendation by Lee et al. [14]. The researcher analyzed taxi GPS data to identify patterns in passenger pickup locations, aiding in the recommendation of optimal taxi locations for improved service efficiency using K-Means Clustering technique. The limited geographic scope may not generalize to other areas with different traffic patterns and taxi usage behaviors. Additionally, the paper does not provide a quantitative evaluation of the effectiveness of the location recommendation approach.

Mann et al. [15] addressed the challenge of sparse data in new areas or low-demand periods in taxi services. Traditional recommendation algorithms struggle with limited data, leading to inaccurate recommendations. The researcher proposed a hybrid clustering algorithm that groups similar pick-up locations, enabling the system to recommend drivers even with scarce data. However, the comparison of the proposed method is done with K-Means and BIRCH and rest of the techniques are not compared with.

As indicated by the reviewed literature, cab recommender systems are significantly influencing urban mobility. However, these systems predominantly rely on traditional clustering methods such as K-Means. DBSCAN etc. While the K-Means algorithm is efficient and intuitive, determining the appropriate number of clusters (K value) can be challenging and highly variable. Additionally, K-Means clustering results often contain noise. Density-based clustering algorithms are commonly used to mitigate noise interference, but controlling the density radius can be challenging, potentially leading to suboptimal solutions. Hierarchical clustering, particularly when based on a grid, can enforce various constraints on the clustering outcomes, but it is often too complex to implement effectively.

Traditional cab recommender systems frequently struggle to provide accurate and timely pickup location suggestions due to their reliance on real-time data, which may not capture the broader historical context of cab movements [16]. Therefore, there is a compelling need for a comprehensive framework that integrates hybrid clustering techniques to enhance recommendation accuracy for cab drivers.

## 3. Research objectives

After an extensive literature review, it was noted that there is a lack of a comprehensive framework for suggesting optimal pickup locations for cab drivers. This identified gap underscores the need for an efficient algorithmic framework for a Cluster-Based Cab Recommender System (CBCRS) to recommend cab drivers with their next pickup location.

Consequently, the research objective was defined to develop a framework for a recommender system that takes the current location of the cab driver and recommends the next nearest passenger pickup location within reachable limits of the driver, based on an efficient clustering technique. While existing research has explored standard clustering techniques individually, there is limited exploration of hybridizing clustering techniques in this context.

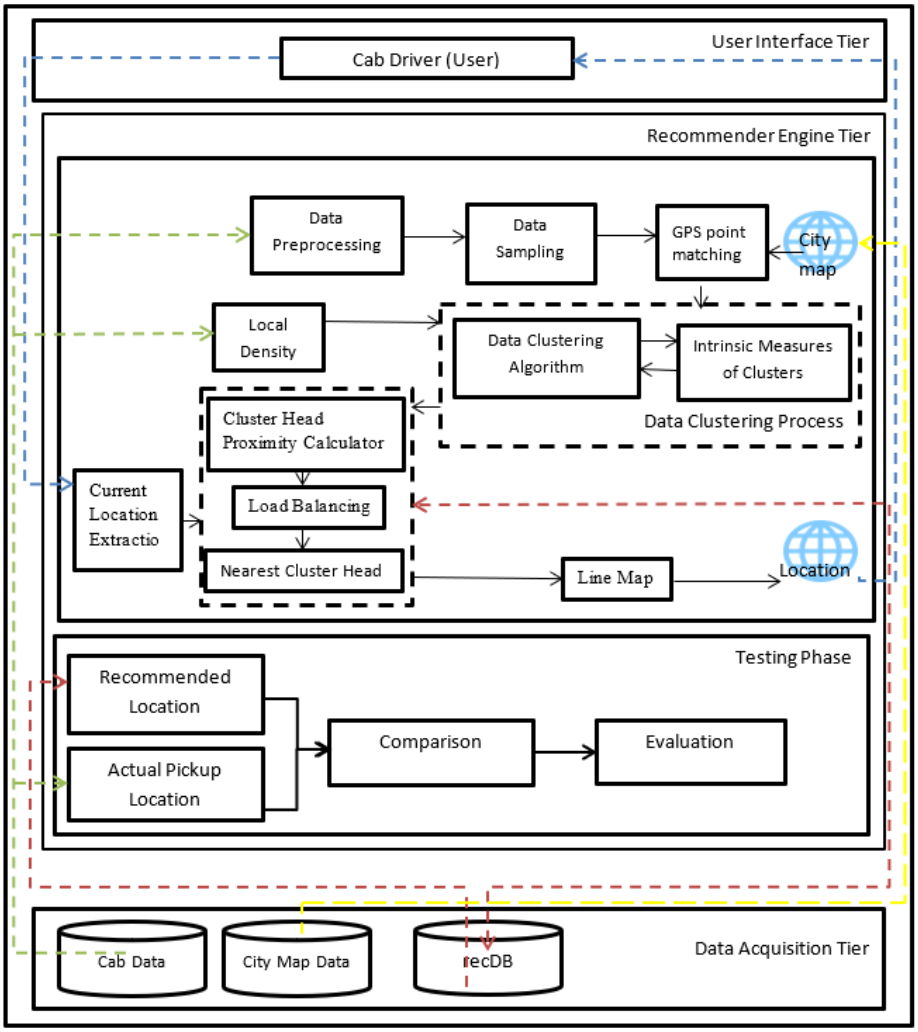



Figure. 1 Proposed framework for CBCRS

To validate the research objective, a proposed framework of CBCRS based on a hybrid clustering technique was compared against standard clustering techniques using three standard datasets, with evaluation based on intrinsic measures such as the Calinski-Harabasz Index and Silhouette Score. Furthermore, the CBCRS system was evaluated using the identified optimal clustering technique compared to traditional clustering methods.

## 4. Research methodology

The research methodology for achieving the stated research objectives comprised three phases:

Phase I: Development of a design framework for the Cluster-Based Cab Recommender System (CBCRS), which recommends the next pickup location for cab drivers.

Phase II: Identification of the most efficient clustering technique for generating optimal clusters of cab pickup locations using intrinsic measures for CBCRS.

Phase III: Implementation of CBCRS using the clustering technique identified as most suitable in Phase II.

**Phase I: Design of framework for CBCRS**

The proposed framework for the Cluster-Based Cab Recommender System (CBCRS) involved several key steps. Initially, cab pick-up data points were acquired from the dataset. Subsequently, historical pick-up points were clustered, and cab drivers were recommended the nearest locations with a higher probability of finding passengers.

To facilitate these steps, the CBCRS framework required a well-designed interface. Therefore, a three-tier approach was proposed for the CBCRS framework, as depicted in Fig. 1.

*Data Acquisition Tier:* The Data Acquisition Tier played a foundational role in the framework, providing essential support and functionality. It facilitated the retrieval of cab data files (.csv) containing historical pickup locations. This tier ensured the secure storage of Cab Data and Recommended Pickup Location Database (recDB) in separate folders within a database, maintaining the confidentiality and integrity of the data.

Additionally, it stored city map data, including road maps of the city where historical pickup locations are saved in the cab dataset. Furthermore, the data acquisition tier maintained a comprehensive database containing essential information about cab pickup locations, timestamps, dates, etc., as well as the city map points and previously recommended pickup points at the same spatial-temporal parameters. This approach enhanced the framework's overall usability. By performing these functions, the data acquisition tier ensured the availability and accessibility of essential resources for the framework's successful operation, contributing to a seamless and user-friendly experience.

*Recommendation Engine Tier:* The Recommendation Engine Tier served as the core component of the CBCRS framework, responsible for executing algorithms that drive various operations. Users requested the Recommendation Engine Tier to generate recommendations for the nearest pickup location with a higher probability of finding a passenger. It began by loading the cab dataset relevant to the city from which the user request originated. The loaded cab dataset underwent preprocessing, sampling, GPS point matching, and clustering to generate the recommendation.

Initially, the framework prepared the cab dataset for the proposed clustering algorithm. Subsequently, this layer generated the nearest cluster head, ensuring balanced load distribution among all cabs converging on the same location. The Recommendation Engine Tier also encompassed the logic used to recommend the cab driver's next passenger location and the logic for displaying a road map from the driver's current location to the recommended location.

Testing and evaluation activities were crucial to ensure the application's correct functionality and alignment with its goals. These activities involved assessing the performance of the algorithms based on statistical parameters, which were integral to the application layer's responsibilities. Actual testing utilized test data from selected datasets, comparing recommendations using various statistical parameters to analyze their similarity.

*User Interface Tier:* The User Interface Tier functioned as an essential component of the CBCRS application, serving as the presentation layer for user interaction. It encompassed all visual and interactive elements that users engaged with during their interaction with the system. The user interface featured text input fields allowing users to specify the city for which they sought recommendations and the clustering algorithm they intended to use. Users could select the city from a list of available cities with historical databases compatible with CBCRS. Furthermore, the user interface provided error feedback, alerting users to invalid inputs and ensuring a user-friendly experience.

**Phase II: Identify the efficient clustering technique for CBCRS**

In a study by Mann et al. [17], three standard datasets were used to evaluate clustering techniques for the Cluster-Based Cab Recommender System (CBCRS). The study found that K-Means, BIRCH,

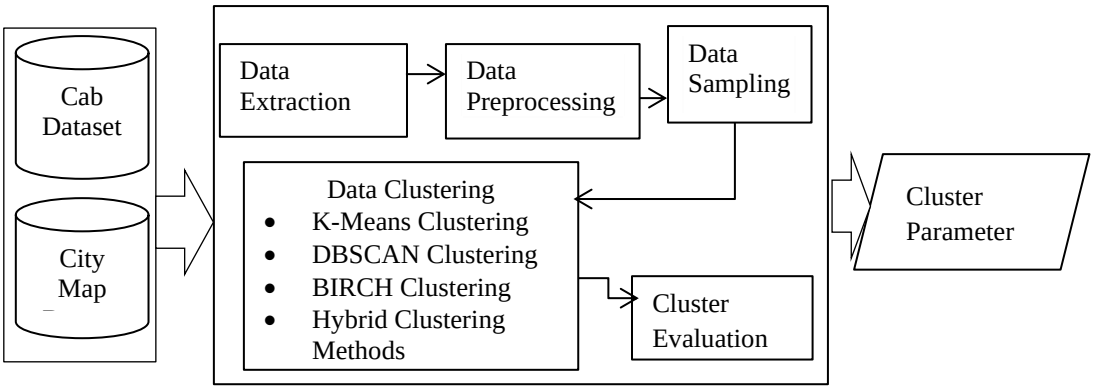



Figure. 2 Cluster pick-up geo-points

and DBSCAN were effective for Partition-based clustering, Hierarchical-based clustering, and Density-based clustering, respectively. Building on these findings, Mann et al. [15] proposed a hybrid clustering technique combining BIRCH and K-Means, comparing it with traditional methods such as K-Means and BIRCH clustering techniques using the Silhouette Score and Calinski-Harabasz Index.

To further investigate clustering techniques for cab pick-up locations, various methods including K-Means Clustering (KMC), DBSCAN Clustering (DC), BIRCH Clustering (BC) , and two hybrid approaches (BIRCH integrated with K-Means (BC+KMC) and DBSCAN combined with K-Means (DC+KMC)) were tested using public datasets from New York Dataset (NYD), Mexico Dataset (MD), and Porto Dataset (PD). As illustrated in Fig. 2, the cab pick-up locations underwent extraction, pre-processing, sampling, and clustering using these techniques.

The resulting clusters were then evaluated based on intrinsic parameters such as the Silhouette Score formulated as Eq. (1) and Calinski-Harabasz Index formulated as Eq. (2).

$$Silhouette\ Score = \frac{(b-a)}{max\ \ (a,b)} \tag{1}$$

Where ‘a’ is the average distance of a data point to all other data points in same cluster and ‘b’ is the average distance of sample to all the data points in the nearest cluster[18].

$$Calinski\text{-}Harabasz\ Index = \frac{Between\ cluster\ variance}{Within\ cluster\ variance} \tag{2}$$

Fig. 3(a-c) illustrates the performance of K-Means, DBSCAN, and BIRCH, along with two hybrid clustering techniques (BIRCH integrated with K-Means and DBSCAN combined with K-Means), across three datasets: New York, Porto, and Mexico cities, using the Silhouette Score as the evaluation metric.

Fig. 4(a-c) illustrates the performance of K-Means, DBSCAN, and BIRCH, along with two hybrid clustering approaches (BIRCH integrated with K-Means and DBSCAN combined with K-Means), across three datasets: New York, Porto, and Mexico cities, using the Calinski-Harabasz Index as the evaluation metric.

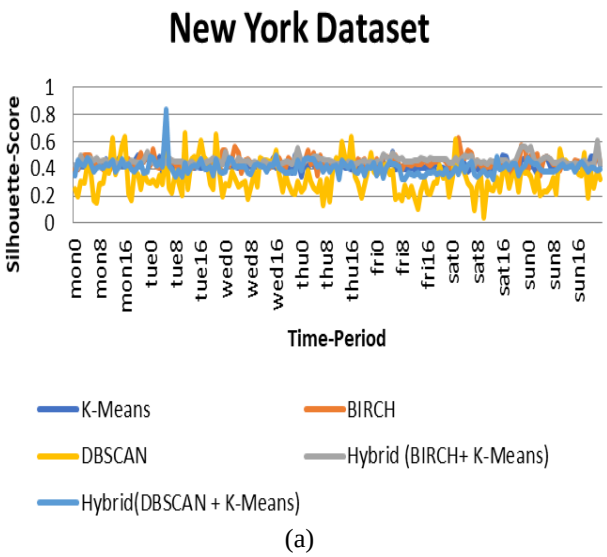
From the above experimental results, the average Silhouette-Score and average Calinski-Harabasz Score were tabulated in Table 1 and Table 2.

In Table 1, it is evident that the average Silhouette Score for the Hybrid Clustering Technique (BIRCH + K-Means) was higher as compared the average Silhouette Score of the Hybrid Clustering Technique (DBSCAN + K-Means), K-Means, BIRCH and DBSCAN for the New York, Porto, and Mexico datasets.

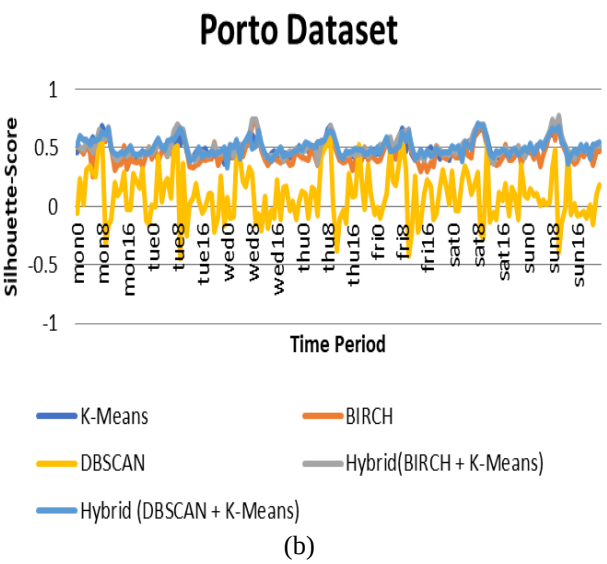
In Table 2, it is evident that the average Calinski-Harabasz Index for the Hybrid Clustering Technique (BIRCH + K-Means) was higher than that of the Hybrid Clustering Technique (DBSCAN + K-Means), K-Means, BIRCH and DBSCAN for the same datasets.

Based on the research findings and the tabulated results, it was concluded that the Hybrid Clustering Technique (BIRCH integrated with K-Means) outperforms the Hybrid Approach of DBSCAN combined with K-Means, as well as the traditional methods of K-Means, DBSCAN, and BIRCH. Therefore, the most suitable clustering technique for the Recommender Engine tier was determined to be the hybrid clustering technique based on BIRCH and K-Means. Subsequently, this identified technique needed to be integrated and implemented within the CBCRS for the proposed research framework.

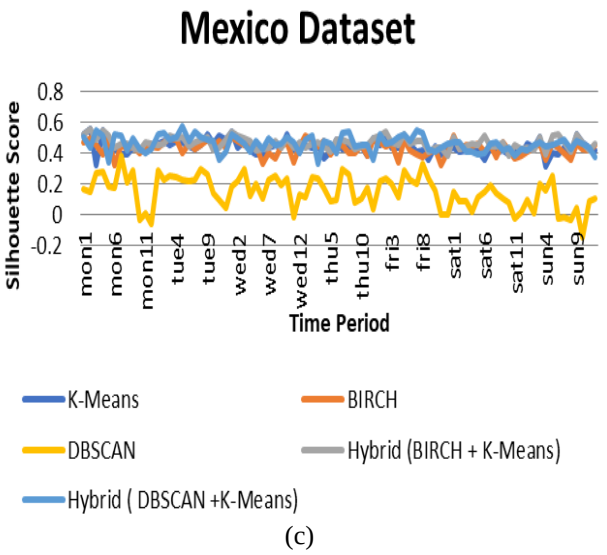
**Phase III: Implementation of identified Hybrid Clustering Method for CBCRS (BIRCH integrated with K-Means)**




(a)

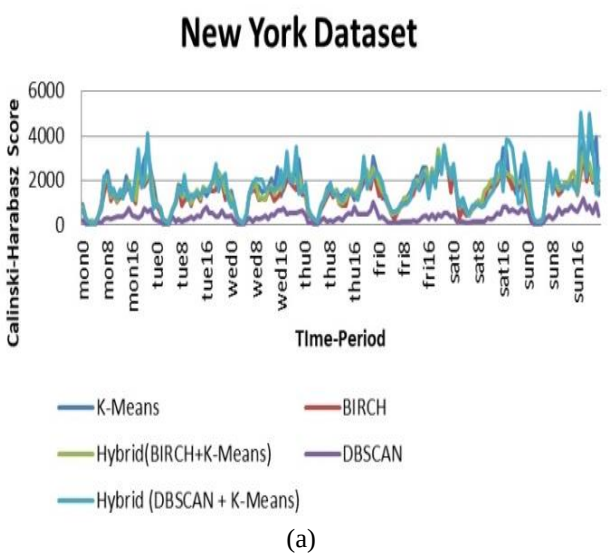



(b)

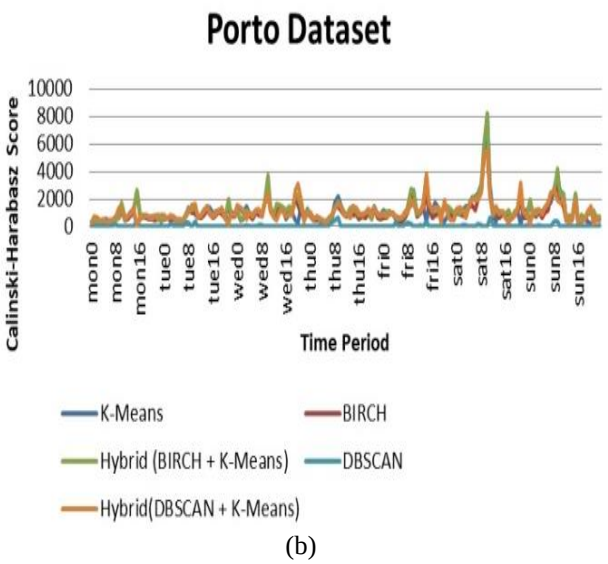



(c)

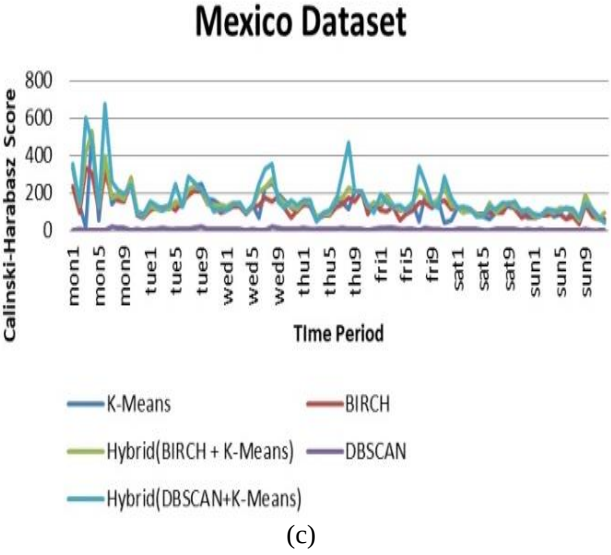
Figure. 3 Silhouette Score for: (a) New-York City, (b) Porto City, and (c) Mexico City




(a)




(b)




(c)

Figure. 4 Calinski-Harabasz Index for: (a) New-York City, (b) Porto City, and (c) Mexico City

Table 1. Average Silhouette Score for New-York, Porto and Mexico Dataset.

| | KMC | BC | DC | BC+KMC | DC+KMC |
|---|---|---|---|---|---|
| NYD | 0.4247 | 0.4486 | 0.3360 | 0.4559 | 0.4088 |
| PD | 0.4944 | 0.4596 | 0.0943 | 0.5065 | 0.5059 |
| MD | 0.4397 | 0.4333 | 0.1502 | 0.4655 | 0.4620 |

Table 2. Average Calinski-Harabasz Index for New-York, Porto and Mexico Dataset.

| | KMC | BC | DC | BC+KMC | DC+KMC |
|---|---|---|---|---|---|
| NYD | 1614.9 | 1347.9 | 384.6 | 1675.1 | 1605.4 |
| PD | 1080.6 | 987.9 | 60.9 | 1224.0 | 1148.9 |
| MD | 137.7 | 129.1 | 9.2 | 157.4 | 155.1 |

The implementation of the proposed Hybrid Clustering technique was carried out using Python programming language version 3.19.13. The implementation within the Recommender Engine tier involved two main steps:

1. Cluster Head Generation using the cab pick-up locations using hybrid clustering technique.
2. Finding the nearest cluster head from the user's current location.

Algorithm 1 was formulated to generate cluster heads utilizing the hybrid clustering approach combining BIRCH and K-Means methods. In algorithm 1, CF_Tree was the Clustering Feature Tree and X represented the pick-up geo-points. CF_Tree was initialized as an empty tree. Each geo-point $X_i$ in X, $X_i$ was inserted into the CF_Tree. While the number of Non Leaf Nodes in CF_Tree was greater than 1, the closest Non Leaf Node based on their distances was found. The two closest nodes were merged into a new node and the CF_Tree was updated. ClusterHeads were initialized as an empty set. Each non leaf node N in CF_Tree was extracted using the clustering feature of N as $CF_N$. The centroid of the node as $C_N$ was calculated. CN was added to ClusterHeads. Further clusters were refined

using K-Means. K was initialized to the number of ClusterHeads and initial centroids for K-Means were initialized to ClusterHeads. Each geo-point $X_i$ in X was assigned to the nearest centroid based on their Euclidean Distance and repeated until convergence, assuring further reassignment of geo-points for clustering do not occur. The centroids were recalculated as the mean of the geo-points assigned

to each cluster. $C_j$ represented the centroid of cluster j, and $ClusterAssignment^{-1}_{(j)}$ be the set of geo-points assigned to cluster j. The mean of the assigned geo-points was calculated as the sum of all the geo-points divided by the total number of geo-points in the cluster as in step 21.

**Algorithm 1. Proposed Hybrid Clustering Method for CBCRS**

**Input:** Bf (Branching Factor), Th (Threshold Limit) and X (Data Points)
**Output**: Centroids Cj
**Steps:**
**1.** CFTree = ∅, ClusterHeads={}
**2.** For each geopoint $X_i$ in X:
**3.** Insert $X_i$ into the CF_Tree
**4.** While | NonLeafNodes (CF_Tree) | > 1:
**5.** ($N_1$, $N_2$) = $\arg\min d(N_i, N_j)$ where $N_i, N_j \in$ NonLeafNodes (CF_Tree) and i ≠ j
**6.** $N_{new}$ = Merge($N_1$, $N_2$)
**7.** CF_Tree = Update ($N_1$, $N_2$, $N_{new}$)
**8.** ClusterHeads = {}
**9.** For each Non-Leaf Node N in CF_Tree:
**10.** $CF_N = ([\sum_{i=1}^{n} Latitude\ (X_i), \sum_{i=1}^{n} Longitude\ (X_i), n]$
**11.** $C_N = \left[\frac{\sum_{i=1}^{n} Latitude\ (X_i)}{n}, \frac{\sum_{i=1}^{n} Longitude\ (X_i)}{n}\right]$
**12.** ClusterHeads = ClusterHeads ∪ $C_N$
**13.** K ← |ClusterHeads|
**14.** Centroids = ClusterHeads
**15.** Repeat until convergence:
**16.** For each $X_i$ in X:
**17.** $C_i = argmin_j \|X_i - C_j\|$
**18.** ClusterAssignment($X_i$) = Ci
**19.** For each cluster j:
**20.** $C_j = \frac{1}{|ClusterAssignment^{-1}_{(j)}|} \sum_{x_i \in ClusterAssignment^{-1}_{(j)}} X_i$

The second step involved identifying the nearest cluster head to the generated clusters, regardless of the clustering technique used. A new algorithm was proposed to determine the closest cluster head based on two key factors: proximity and passenger-finding potential. The algorithm computed the distance between the cab's current location and each cluster head, ensuring that the recommended cluster head was the closest spatially. Simultaneously, it assessed the passenger-finding potential of each cluster head based on historical data.

Algorithm 2 was devised to recommend the nearest cluster head to the cab driver. The passenger-finding potential was determined by analyzing past passenger pickup patterns within each cluster. Clusters with a higher frequency of passenger pickups were deemed to have a higher potential for

finding passengers. This information is vital for cab drivers as it directs them to locations where they are more likely to receive ride requests, thus optimizing their income and reducing cruising time.

**Algorithm 2: Recommend nearest cluster head as Pickup Location**

**Input:** C = set of centroids {$C_1, C_2 \ldots C_k$ }, X (Data Points), CurrentLocation

**Output:** Recommended_Centroid

**Steps:**

1: For each $C_i$ in C:
2: Calculate $P(C_i)$
3: Highest_Prob = max($P(C_i)$) for all $C_i$ in C
4: Eligible_Centroids = $C_j$ where $P(C_j) \geq 0.7 \times$ Highest_Prob
5: Distances = {}
6: For each $C_i$ in Eligible_Centroids:
7: Distances($C_i$) = Distances(CurrentLocation , $C_i$)
8: Recommended_Centroid = $argmin_{C_j}$ (Distances($C_i$))
9: Return Recommended _Centroid

In algorithm 2, C represented the set of centroids generated by Algorithm1 and X was the data points which were clustered using algorithm 1. CurrentLocation denoted the current location of the cab driver in requesting the recommendation of next pick-up location. The probability to find the passenger was calculated as $P(C_i)$ for each $C_i$ of the set C. The cluster heads with at least 70% probability to find the passenger were marked as Eligible_Centroids as the data points beyond 70% probability were too less to form multiple clusters. From the set of Eligible_Centroids, the centroid closest to the current location of the cab driver was marked as the Recommended_Centroid.

## 5. Experimental results

CBCRS was evaluated using two basic research studies:

Research St*udy 1: The aim of this research study was to test CBCRS on the parameter of accuracy of recommendations to the cab driver.*

*Research Study 2: The aim of this research study was to test CBCRS with standard clustering techniques.*

The methodology employed in the aforementioned research studies followed a systematic approach. Initially, the cab driver (user) initiated a request to the Cluster-Based Cab Recommender System (CBCRS) to recommend the nearest location where the likelihood of finding a passenger would be higher. Subsequently, the Recommender Engine generated cluster heads by clustering the cab pick-up locations obtained from the cab dataset. Finally, the system recommended the nearest cluster head to the cab driver's current location as the next passenger pick-up location.

The recommendations made by CBCRS were evaluated using three standard datasets from New York, Mexico, and Porto cities, as well as one live dataset from Chandigarh (CD) city. The evaluation process focused on assessing the accuracy of the recommendations provided by the system. These recommendations were compared with those generated by traditional clustering methods such as K-Means, BIRCH, and DBSCAN. The evaluation criteria included Accuracy, Precision, Recall, and F1-Score. The accuracy, Precision, Recall and F1-Score for the system were calculated using Eq.(3),(4),(5) and (6) respectively.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (3)$$

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (6)$$

Where TP be the number of True Positives, TN be the number of True Negative, FP be the number of False Positive and FN be the number of False Negatives.

The experimental results were tabulated for comparison in Table 3. In the New York region, the Cluster-Based Cab Recommender System (CBCRS) exhibited superior accuracy compared to traditional clustering algorithms, achieving an accuracy rate of 87.1% compared to 75.4%, 81.5%, and 76.1% for K-Means, DBSCAN, and BIRCH recommendations, respectively. Additionally, CBCRS demonstrated a balanced performance between precision and recall, leading to a high F1-score. Similar trends were observed across other regions. In Mexico, CBCRS achieved an accuracy of 71.9%, surpassing K-Means, DBSCAN, and BIRCH with accuracies of 69.5%, 58.63%, and 70.9%, respectively. In Porto, CBCRS attained an accuracy of 72.1%, outperforming K-Means, DBSCAN, and BIRCH with accuracies of 70.1%, 67.1%, and 71.0%, respectively. In Chandigarh, CBCRS achieved an accuracy of 73.6%, compared to 72.8%, 68%, and 66.8% for K-Means, DBSCAN, and BIRCH, respectively.

Table 3. Accuracy, Precision, Recall and F1-Score of Recommendations

| | | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|---|
| NYD | K-Means | 0.754 | 0.9960 | 0.7555 | 0.8592 |
| | DBSCAN | 0.815 | 0.9827 | 0.8243 | 0.8966 |
| | BIRCH | 0.761 | 0.9960 | 0.7625 | 0.8638 |
| | CBCRS | 0.871 | 0.9977 | 0.8724 | 0.9309 |
| MD | K-Means | 0.695 | 0.9811 | 0.7011 | 0.8178 |
| | DBSCAN | 0.586 | 0.9717 | 0.5892 | 0.7336 |
| | BIRCH | 0.709 | 0.9797 | 0.7152 | 0.8268 |
| | CBCRS | 0.719 | 0.9809 | 0.7261 | 0.8345 |
| PD | K-Means | 0.701 | 0.9800 | 0.7069 | 0.8213 |
| | DBSCAN | 0.671 | 0.9579 | 0.6814 | 0.7964 |
| | BIRCH | 0.710 | 0.9895 | 0.7139 | 0.8294 |
| | CBCRS | 0.721 | 0.9780 | 0.7287 | 0.8352 |
| CD | K-Means | 0.728 | 0.9803 | 0.6976 | 0.8152 |
| | DBSCAN | 0.68 | 0.9673 | 0.6636 | 0.7872 |
| | BIRCH | 0.668 | 0.9930 | 0.6339 | 0.7738 |
| | CBCRS | 0.736 | 0.9806 | 0.7069 | 0.8216 |

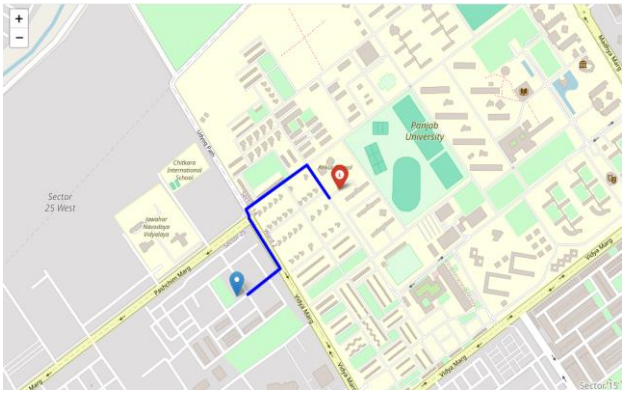



Figure. 5 Line map joining current location of cab driver to the recommended position marked for Chandigarh dataset.

The CBCRS recommended pickup location was communicated to the cab driver, as shown in Figure 1. Figure 5 displayed a line map connecting the driver's current location (blue) to the recommended pickup spot (red), helping the driver navigate

The results highlight the effectiveness of the Hybrid Clustering approach, showcasing its adaptability across various geographic datasets. Higher precision values indicate fewer false positives, while increased recall values suggest better cluster identification. This blend of strengths makes the hybrid approach a robust solution for spatial analysis and clustering tasks.

The proposed CBCRS framework outperforms state-of-the-art methods, with the hybrid clustering technique proving superior to K-Means, DBSCAN, and BIRCH. Tested on large datasets across four different regions, the system consistently performed better, demonstrating its reliability regardless of dataset size or geographic location.

## 6. Related work

Recent studies have focused on providing recommendations to taxi drivers. Yuan et al. [11] used GPS trajectory data and K-Means Clustering to predict high-demand areas for pickups, but performance may degrade with larger datasets. Yuan et al. [3] developed a recommender system for matching passengers with vacant taxis, though its applicability is limited to specific geographic regions. Ma et al. [12] proposed a stop planning method for airport shuttles using K-Means, but did not compare other clustering methods. Qi et al. [13] used DBSCAN to identify taxi spots, but prediction accuracy was limited by the variability of GPS data. Lee et al. [14] analyzed taxi GPS data with K-Means to identify pickup patterns, but the findings may not generalize to other regions with different traffic patterns and taxi behaviors.

Our approach differs from previous methods in several key aspects:

1. Large Dataset: Our research uses a dataset with millions of records, demonstrating robustness to dataset size, unlike [11].
2. Geographic Diversity: We include diverse geographic areas such as New York, Porto, Mexico, and Chandigarh, showing broader applicability compared to [3][13-14].
3. Public and Live Data: We test on publicly available datasets from New York, Porto, and Mexico, as well as live data from Chandigarh, unlike state-of-the-art methods which are limited to specific locations [3][13-14].
4. Comprehensive Comparison: Our proposed clustering technique is compared with multiple standard clustering methods like K-Means, DBSCAN, and BIRCH, unlike [12][15] which did not offer such comparisons.

## 7. Conclusion

In conclusion, the hybrid clustering technique utilizing BIRCH and K-Means has demonstrated superiority over K-Means, BIRCH, and DBSCAN based on intrinsic measures such as Silhouette-Score and Calinski-Harabasz Index over different datasets. The proposed Cluster-Based Cab Recommender System (CBCRS) employing this hybrid clustering approach offers a robust framework to tackle the challenges encountered by cab drivers in optimizing their routes and enhancing passenger-finding

efficiency. The proposed system gives higher accuracy, precision, recall and F1-Score as compared to standard clustering techniques such as K-Means, DBSCAN and BIRCH. Through the amalgamation of BIRCH and K-Means clustering methodologies, the system adeptly identifies spatial patterns within historical pickup data. Moreover, the recommendation mechanism, which considers both proximity and passenger-finding potential, augments the system's efficacy by directing cab drivers to locations with the highest probability of receiving ride requests. Consequently, CBCRS emerges as a promising solution poised to elevate the overall efficiency and income of cab drivers within contemporary urban transportation systems.

## Conflicts of Interest

The authors declare no conflict of interest.

## Author Contributions

Conceptualization, Supreet Kaur Mann and Sonal Chawla; Methodology, Supreet Kaur Mann and Sonal Chawla; Validation, Supreet Kaur Mann; Writing—original draft preparation, Supreet Kaur Mann; writing—review and editing, Sonal Chawla; visualization, Sonal Chawla; supervision, Sonal Chawla.

## References

[1] S. S. Choudhury, S. N. Mohanty, and A. K. Jagadev, "Multimodal trust based recommender system with machine learning approaches for movie recommendation", *International Journal of Information Technology*, Vol. 13, No. 2, pp. 475–482, 2021.

[2] T. Lyu, P. (Slade) Wang, Y. Gao, and Y. Wang, "Research on the big data of traditional taxi and online car-hailing: A systematic review", *Journal of Traffic and Transportation Engineering*, Vol. 8, No. 1, pp. 1–34, 2021.

[3] N. J. Yuan, Y. Zheng, L. Zhang, and X. Xie, "T-finder: A recommender system for finding passengers and vacant taxis", *IEEE Transaction on Knowledge and Data Engineering*, Vol. 25, No. 10, pp. 2390–2403, 2013.

[4] R. Wang et al., "TaxiRec: Recommending road clusters to taxi drivers using ranking-based extreme learning machines", *IEEE Transaction on Knowledge and Data Engineering*, Vol. 30, No. 3, pp. 585–598, 2018.

[5] D. Xu and Y. Tian, "A Comprehensive Survey of Clustering Algorithms", *Annals of Data Science*, Vol. 2, No. 2, pp. 165–193, 2015.

[6] Z. Jia, Y. Yang, W. Gao, and X. Chen, "User-based collaborative filtering for tourist attraction recommendations", In: *Proc*. of *IEEE International Conference Computational Intelligence and Communication Technology CICT* Ghaziabad, India, pp. 22–25, 2015.

[7] M. J. Pazzani and D. Billsus, "Content-Based Recommendation Systems", In: *The Adaptive Web: Methods and Strategies of Web Personalization*, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 325–341, 2007.

[8] J. Chen, Uliji, H. Wang, and Z. Yan, "Evolutionary heterogeneous clustering for rating prediction based on user collaborative filtering", *Swarm and Evolutionary Computation*, Vol. 38, pp. 35–41, 2018.

[9] J. Xu, R. Rahmatizadeh, L. Bölöni, and D. Turgut, "A taxi dispatch system based on prediction of demand and destination", *Journal of Parallel and Distributed Computing*, Vol. 157, pp. 269–279, 2021.

[10] D. Zhang, T. He, Y. Liu, S. Lin, and J. A. Stankovic, "A carpooling recommendation system for taxicab services", *IEEE Transactions on Emerging Topics in Computing.*, Vol. 2, No. 3, pp. 254–266, 2014.

[11] J. Yuan, Y. Zheng, L. Zhang, X. Xie, and G. Sun, "Where to find my next passenger?", In: *Proc*. of *ACM Conference on Ubiquitous Computing*, Beijing, China, pp. 109–118, 2011.

[12] J. Ma, X. Chen, Z. Xing, Y. Zhang, and L. Yu, "Improving the performance of airport shuttle through demand-responsive service with dynamic fare strategy considering mixed demand", *Journal of Air Transport Management*, Vol. 112, p. 102459, 2023.

[13] G. Qi et al., "How Long a Passenger Waits for a Vacant Taxi -- Large-Scale Taxi Trace Mining for Smart Cities", In: *Proc*. of *IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, pp. 1029–1036, 2013.

[14] J. Lee, I. Shin, and G.-L. Park, "Analysis of the Passenger Pick-Up Pattern for Taxi Location Recommendation", In: *Proc.* of *Fourth International Conference on Networked Computing and Advanced Information Management*, pp. 199–204, 2008.

[15] S. K. Mann and S. Chawla, "A proposed hybrid clustering algorithm using K-means and :BIRCH for cluster based cab recommender system (CBCRS)", *International Journal of*

*Information Technology*, Vol. 15, No. 1, pp. 219–227, 2023.

[16] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas, "Predicting Taxi–Passenger Demand Using Streaming Data", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 14, No. 3, pp. 1393–1402, 2013.

[17] S. K. Mann and S. Chawla, "Cluster-Based Cab Recommender System (CBCRS) for Solo Cab Drivers", *International Journal of Information Retrieval and Research,* Vol. 12, No. 1, pp. 1–15, 2022.

[18] M. Shutaywi and N. N. Kachouie, "Silhouette analysis for performance evaluation in machine learning with applications to clustering", *Entropy*, Vol. 23, No. 6, pp. 1–17, 2021.