Academic Journal of International University of Erbil

Journal Homepage: https://ojs.cihanrtv.com/index.php/public

PRINT-ISSN: 2519-6928

Online-ISSN: 3080-7174

A Hybrid Algorithm for Scalable Friend Recommendation in Large-Scale Social Networks

Karwan Mohammed Hamakarim[1] Govar Abubakr Omar[2]* Rukhsar Hatam Qadir[3]

[1]Department of Information Technology, University of human Development, Sulaymaniyah, IRAQ

[2] Department of Information Technology, Qala University College, Erbil, IRAQ

[3]Department of Statistics and Informatics, College of administration and economics, University of Sulaimani, Sulaymaniyah, IRAQ

*Corresponding Author: Govar Abubakr Omar[2]

DOI: https://doi.org/10.63841/iue23576

Received 02 March 2025; Accepted 18 June 2025; Available online 20 July 2025

ABSTRACT

The rapid growth of online social networks has led to a surge in user-generated data, resulting in significant information overload and making accurate friend recommendations increasingly difficult. Traditional recommendation methods—such as social graph analysis, collaborative filtering, and semantic similarity—struggle to scale effectively in large-scale environments due to high computational costs or limited precision. This research proposes a hybrid algorithm that combines location-based proximity detection and user classification to enhance the accuracy and scalability of friend recommendations. By identifying geographically or behaviorally close user clusters and separating distinct user classes, the system can efficiently uncover potential social connections. Experimental results demonstrate that the proposed method improves friend suggestion relevance while reducing computational overhead, making it suitable for large and dynamic social platforms.

Keywords: Social networks, Recommendation system, Social graph, Semantic similarity

1 INTRODUCTION

A friendship is a bond of affection shared between two individuals. It is a stronger type of interpersonal tie than an "acquaintance" or "association"—such as relationships with students, neighbors, coworkers, or colleagues.

In certain cultures—such as the U.S. and Canada—a person can have many friends and may develop closer bonds with one or two individuals who are referred to as best friends. In other cultures, however, friendship is limited to a small number of extremely deep relationships. Some popular expressions include "best friends" or "Best Friends Forever (BFFs)." While friendships take many different forms, most share basic characteristics such as the desire to spend time together, enjoyment of each other's company, and the capacity to provide constructive and encouraging support. Although the distinction can be unclear in cases of "friends with benefits," friends are generally distinguished from family members (as in the phrase "friends and family") and romantic partners. Similarly, someone who is unable to transition from friend to romantic partner is said to be in the "friend zone."

*Corresponding author: Govar.Abubakr@qala.edu.iq

Academic disciplines including communication, sociology, social psychology, anthropology, and philosophy have all examined friendship. Numerous scholarly theories have been proposed about friendship, including attachment theory, relational dialectics, social exchange theory, and equity theory. The digital revolution has affected many aspects of life, particularly social interaction. Social networks, which enable widespread communication and interaction among individuals who share similar interests, beliefs, and viewpoints, have become an integral part of modern life. Most of these changes are communication-related, as high-quality digital communication technologies have emerged through these platforms. In the digital age, social networks have fundamentally altered how individuals interact, connect, and communicate. These networks, which have complex topologies with nodes representing entities (people, groups, or organizations) and edges representing connections or interactions between them, can be found in both biological systems and online social platforms.

1.2 PROBLEM STATEMENT

The problem we are addressing involves identifying closest friends and grouping friends who are in the same geographical area to determine proximity relationships between them. Another aspect of our work focuses on finding the nearest locations and determining friends in close proximity to each other. By combining several algorithms, we can better identify locations and find friends in the same area. This approach aims to make our methods faster, more intelligent, and more effective at identifying similarities between users or locations on digital platforms.

1.3 AIMS AND OBJECTIVES

- Develop a fast similarity-based method for location-based classification using DBSCAN and K-means algorithms
- Identify nearby locations and group them into single classes
- Find groups of similar friends and organize them into cohesive groups
- Improve the Silhouette Score of our predictions by developing better methods to measure similarity between people in specific locations

1.4 CONTRIBUTION

First, we selected the optimal number of trees for the decision tree algorithm and combined two algorithms (DBSCAN and K-means). We achieved a better Silhouette Score with a more refined dataset that facilitated easier location clustering predictions.

2 LITERATURE REVIEW

In [6], the authors presented a novel approach to power service continuity issues related to fault location. They proposed a statistical approach utilizing finite mixtures. By extracting voltage sag magnitude recorded during fault events and incorporating network topology and parameters, they created a statistical model. The goal was to provide a cost-effective and easily implementable alternative for developing methods aimed at enhancing reliability through reduced restoration times in power distribution networks. The application case demonstrated successful identification of faulty zones with low error rates in a power distribution system.

In [7], the authors employed K-means clustering to enhance neighboring point selection, thereby improving the KNN algorithm. The K-means clustering algorithm in their proposed method groups nearby neighbors based on their distance from the mobile user. The mobile user's position is then determined using the group closest to them. The evaluation results showed strong correlations between the number of neighbors to be clustered, the number of clusters, the starting points of center points in the K-means algorithm, and the performance of the clustered KNN.

In [8], the authors introduced a heuristic for local improvement based on moving centers in and out. They demonstrated how this led to a (9+ε)-approximation technique. They proved that the approximation factor was nearly optimal by providing an example where the method achieved an approximation factor of (9-ε). To demonstrate the heuristic's effectiveness, they conducted empirical studies showing that the heuristic performed well in practice when combined with Lloyd's algorithm.

In [9], the authors deployed vertiport network architecture in a large metropolitan area using real-world data and established techniques. Vertiport locations were selected based on commuter density for urban air mobility (UAM) in the Seoul metropolitan area. To minimize noise impact on residents, they created noise-priority routes using the Aviation Environmental Design Tool (AEDT) software. They used MATLAB's built-in K-means algorithm to cluster and analyze demand statistics, presenting them on maps. Cluster centers were selected as vertiport locations. They evaluated clustering accuracy and reliability using silhouette analysis. The chosen vertiport sites were located using satellite imagery and relocated when necessary to ensure suitability for actual vertiport positions. Using helicopter models, they compared shortest-distance routes with noise-priority routes, demonstrating that noise-priority routes were more effective and generated less noise exposure.

In [10], the authors examined local search strategies for metric facility location problems, including uncapacitated variations of k-median, k-center, and k-means problems, as well as the uncapacitated facility location problem (UFL). Natural local search strategies are applicable to all these problems. For instance, obvious steps in solving the UFL problem include opening new facilities, closing existing ones, and replacing closed facilities with open ones. However, in k-median problems, only swap moves are permitted. Arya et al. (SIAM J. Comput. 33(3):544-562, 2004) used an innovative "coupling" argument to demonstrate that local optima had costs within constant factors of the global optimum in their analysis of the k-median local-search algorithm.

In [11], the authors proposed ZigBEACON, a ZigBee-based indoor localization system for Ambient Intelligence (AmI) applications. The recommended approach is based on the k-nearest neighbor algorithm. RSSI (Received Signal Strength Indication) values are categorized into four groups based on path loss distribution. After correction, signals categorized by various ratios are described using weighted RSSI. Weighted RSSI can efficiently select the p-nearest reference nodes. The mobile node's position is then calculated using coordinates of the nearest reference nodes. Compared to the ZigBee-based LANDMARC system, the proposed technique improved average error distance by 29%. The method also improved accuracy and reduced computational complexity compared to previous LANDMARC enhancement strategies. The ZigBEACON technique offers a viable indoor positioning system solution for AmI applications.

In [12], the authors proposed minimizing total costs by considering Joint Distribution Willingness (JDW) of restaurants and coverage areas of Joint Distribution Centers (JDC). These costs included penalty fees for missed deliveries, fixed charges, and transportation costs. They used an integer programming model to determine optimal JDC locations, opening only k-JDCs. They proposed an Improved K-means (I-K-means) algorithm combining local search with penalty functions. As a case study, they investigated delivery problems from a freight survey of 114 restaurants in Beijing, China. This study offered solutions that could reduce restaurant delivery costs, decrease the number of freight vehicles on roads, and promote cooperative distribution within urban freight systems.

In [13], the authors addressed single facility location problems using the center-of-gravity approach. They used the elbow method to determine the k-value after applying K-means clustering to divide demand points into k clusters. Subsequently, they treated the k clusters as single facility location problems and implemented the center-of-gravity approach. This discrete model analysis can be applied when determining optimal solutions in real-world situations.

In [14], the authors presented NicheClust, a novel sharing-based niche genetic algorithm (NGA). They used a unique hybrid K-means initial population method to select the best chromosome, followed by K-means clustering. The fitness functions employed by NGA included SSE, DB-index, PBM-index, and COSEC. Testing results demonstrated that NicheClust achieved high performance and efficiency across three GPS location datasets.

In [15], the authors developed GKA, which uses a single step of the K-means algorithm instead of crossover as a search operator. They also devised a biased mutation operator specific to clustering, called distance-based mutation. Using finite Markov chain theory, the authors proved that GKA converges to the global optimum. Simulations demonstrated that GKA converges to well-known optima matching the provided data, consistent with convergence results. GKA was also observed to search faster than several other evolutionary algorithms used for clustering.

In [16], the authors accomplished their work in two steps: first, they mapped preprocessed data with spatial characteristics for location prediction; second, they clustered spatial data to find optimal clusters according to proposed multi-objectives. Using the recommended methodology, they found that locations had good inbound water supply suitable for water treatment facilities.

In [17], the authors examined optimal sites considering rice field size and proximity to rice field clusters, as well as geographical coordinates and area (hectares). The goal was to adapt agricultural yields around clusters. After acquiring primary data from the Malang Regency Department of Agriculture, the authors processed it as input for analysis.

In [18], the authors used taxi GPS data collected in November 2012 to identify potential bus stops near Beijing's Capital International Airport. They applied a modified DBSCAN technique, called C-DBSCAN, to cluster taxi drop-off locations. The C-DBSCAN algorithm incorporated distance between drop-off locations and the airport, as well as drop-off location density.

In [19], the author developed an approach to handle GPS data without acceleration or speed information, which are frequently used as crucial components in rule-based approaches. The methodology uses density-based clustering in the first stage and support vector machines (SVMs) in the second stage. By eliminating incorrectly identified stopping points using entropy as an updated constraint, the methodology's overall accuracy improved by 1.5% compared to the previous version. Furthermore, the first phase's output enhanced SVM performance.

In [20], the author addressed road accident problems using data mining techniques on accident datasets. They developed an Android application using fragments to alert users with pop-up messages when they reach accident-prone areas (ARP), which are cluster regions established by the DBSCAN method. The Apriori algorithm was used to identify association rules.

In [21], the authors introduced an approach applicable to storing and organizing various types of spatiotemporal data. The paper demonstrates implementation of this approach and presents results from data mining a spatial-temporal data warehouse system.

In [22], the authors proposed a method for automatically determining DBSCAN parameters from each dataset's point distribution. They tested the algorithm's ability to locate points of interest (POIs) in areas with different distribution patterns using Bangkok's taxi pick-up and drop-off locations. The results demonstrated this capability.

In [23], the authors proposed an innovative and reliable method based on the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) approach for identifying acoustic emission (AE) sources. The methodology considered the significant influence of anomalies on positional accuracy.

In [24], the authors discovered location semantics using predefined daily behavior patterns and temporal characteristics of data in each category. When calculating distances between samples, the technique proved to be 8.8 times more efficient than conventional algorithms.

In [25], the authors examined and compared spatial distribution characteristics of art galleries, museums, theaters, libraries, and cultural institutions from micro-gathering center and macro-spatial form perspectives. Based on optimization research, the paper provided analysis of spatial arrangements of state and international public institutions. The study began with current configurations of urban public sports facilities and applied the Location Allocation (LA) model and Geographic Information System (GIS) technology to urban public sports service facility placement. They investigated local behavioral characteristics to determine their effects on cultural facility (CF) spatial forms and provided relevant recommendations for CF design and development in Zhengzhou. By optimizing layout and selecting ideal sites, the study took a novel approach to sports facility placement utilizing DBSCAN network analysis capabilities for urban public sports facilities. Furthermore, the case analysis served as a blueprint for future public service facility placement.

In [26], the authors compared velocity auto-picking techniques using K-means and DBSCAN. They found that DBSCAN outperformed K-means in velocity selection tasks. Furthermore, DBSCAN required significantly less user interaction than K-means. Computational cost differences between the approaches were negligible for the given situation. These findings have potential to significantly reduce costs of processing large seismic datasets by eliminating the need for manual velocity selection.

In [27], the authors addressed location privacy, a significant security threat in wireless sensor networks (WSNs), by proposing an IoT cluster-based location privacy (KCLP) defense mechanism built on k-means clusters. They used fake source nodes to replicate real source behavior for source location protection. To protect sink location privacy, they employed specific transmission patterns and fake sink nodes. K-means clusters were used to construct clusters and fake packets that must pass through designated areas to boost safety time.

In [28], the authors illustrated practical applications of location modeling and analysis. Comprehensive analysis using a village dataset from India including four well-known states showed that the proposed solution reduced average traveling costs for villagers by 33% compared to random CSC allocation in sorted order and by 11% compared to centroid allocation using FCM-based approaches alone. The recommended plan performed 31% and 14% better than traditional approaches such as P-Center and P-Median. Consequently, the proposed approach outperformed conventional FCM and other computational techniques like random search and linear programming.

In [29], the authors provided a novel cluster design for wireless sensor networks using K-means clustering. Sensor nodes are deployed in challenging environments, randomly scattered throughout the region of interest in flat configurations. Network lifetime is shortened by packet transfer, so clustering techniques are required to reduce network traffic and increase overall network lifespan. To cluster sensor nodes in the network, position information of each deployed sensor node must be known. Based on sensor node locations in wireless sensor networks, clusters are formed using maximum residual energy and shortest distance from the base station.

In [30], the authors aimed to determine effects of initial score-line, game location, and opponent quality on quarterfinal scores in basketball games. The sample included 504 game quarters from the Spanish Basketball Professional League using k-means cluster algorithms. These were categorized as balanced (score difference ≤ 8 points, $n = 194$) and unbalanced (score difference > 8 points, $n = 310$). They used linear regression analysis to examine predictor variable effects on game quarter outcomes (difference between points scored and points received) for entire games and for second, third, and fourth quarters.

In [31], the authors developed a revolutionary technique using artificial neural networks (ANN) as decision-making systems to process radiographic data and extract information for finding minor apical foramina (AF).

In [32], the authors identified popular pick-up and drop-off locations by enhancing the density-based spatial clustering of applications with noise (DBSCAN) technique using taxi GPS data. The demonstrated density-based hot spots are suitable options for bus stop locations. This work further employs the modified DBSCAN approach, called C-DBSCAN, to identify potential bus-stop sites near Beijing's Capital International Airport based on taxi GPS data acquired in November 2012. Finally, this study discusses the effects of significant C-DBSCAN algorithm parameters on clustering results.

In [33], the author suggested a two-step procedure for determining activity stop locations. An improved density-based spatial clustering of applications with noise (DBSCAN) algorithm is used in the first step to identify stop points and moving points. In the second step, support vector machines (SVMs) distinguish between activity stops and non-activity stops among identified stop points. Two constraints augment DBSCAN: a direction change constraint and a time sequence constraint (resulting in an upgraded algorithm called C-DBSCAN). Three main features are then extracted for application in the Support Vector Machines methodology: halt duration, shortest path between current location and home, and average distance to the centroid of a set of locations at a stop position or workplace.

3 METHODOLOGY

defines the study strategy, including systematic procedures, system design, and mathematical requirements.

3.1 ALGORITHM DESCRIPTION

3.1.1 K-MEANS

K-means clustering is a vector quantization method that separates n observations into k clusters, with origins in signal processing. Each observation belongs to the cluster with the nearest mean (called the cluster centroid or cluster center), making each observation a prototype of its cluster. This creates Voronoi cells within the data space. While the geometric median minimizes Euclidean distances (addressing the more complex Weber problem), the mean optimizes squared errors. K-means clustering minimizes within-cluster variances (squared Euclidean distances) rather than regular Euclidean distances. For better Euclidean solutions, alternatives like k-medians and k-medoids can be used.

Algorithm Steps:

1- Initialization: Select K initial cluster centroids using a heuristic approach or randomly. Each centroid represents one cluster's center.

2- Assignment Step: Assign each data point to the nearest centroid using a distance metric (typically Euclidean distance). This creates K clusters.

3- Update Step: Recalculate the centroids of the K clusters based on the mean of data points assigned to each cluster.

4- Repeat: Iteratively repeat the assignment and update steps until convergence conditions are met. Common convergence criteria include no change in cluster assignments or minimal centroid position shifts.

5- Convergence: The algorithm converges when centroids stabilize and each data point remains assigned to its nearest centroid. K-means has now discovered an optimal clustering of the data.

K-means is frequently used for various clustering applications, including customer segmentation, image compression, and document clustering, due to its efficiency and simplicity. However, it can be influenced by initial centroid selection and may converge to local rather than global optima. This can be mitigated by running multiple iterations with different initializations.

K-means Algorithm: Step-by-Step Process

Algorithm Steps:

1- Start: Randomly select K points in your dataset as initial estimates for cluster centers.

2- Assignment Step: For each point in your dataset, find the closest cluster center and assign that point to the cluster represented by that center.

3- Update Step: Recalculate the center of each cluster by computing the average of all points assigned to that cluster. Move the cluster center to this new average position.

4- Repeat: Continue repeating the assignment and update steps until the cluster centers stop moving significantly (until they converge).

5- Finish: Once the cluster centers stop moving significantly, you have successfully identified your clusters.

Mathematical Formulation:

1- Distance Calculation

K-means typically uses Euclidean distance to measure the distance between data points and cluster centroids. The Euclidean distance between two points *xi* and *cj* in a *d*-dimensional space is calculated as:

Distance (xi, cj) = $\sqrt{[\Sigma(xik - cjk)^2]}$ .... eq 3.1.1.1

where the summation is over all dimensions k from 1 to d.

2- Assignment Step:

- For each data point xi, calculate the distance to each centroid cj using the distance formula
- Assign the data point xi to the cluster represented by the nearest centroid cj

3- Update Step:

- Recalculate the centroid of each cluster j by computing the mean of all data points xi assigned to that cluster
- The new centroid cj is calculated as:

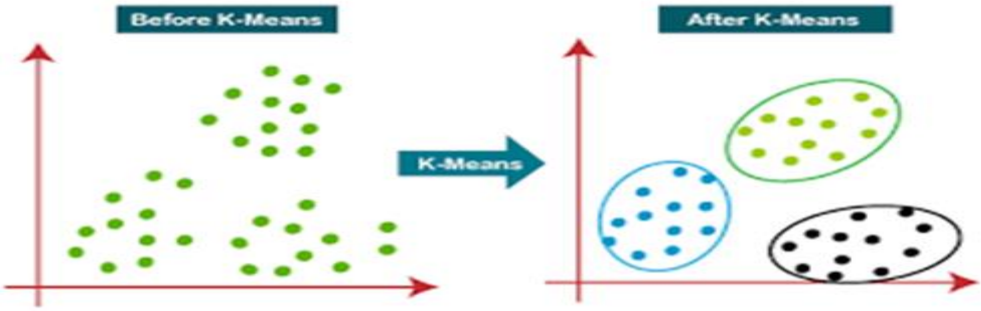
cj = (1/nj) × Σxi .... eq 3.1.1.2

where nj represents the number of data points assigned to cluster j, and the summation is over all data points xi assigned to cluster j.

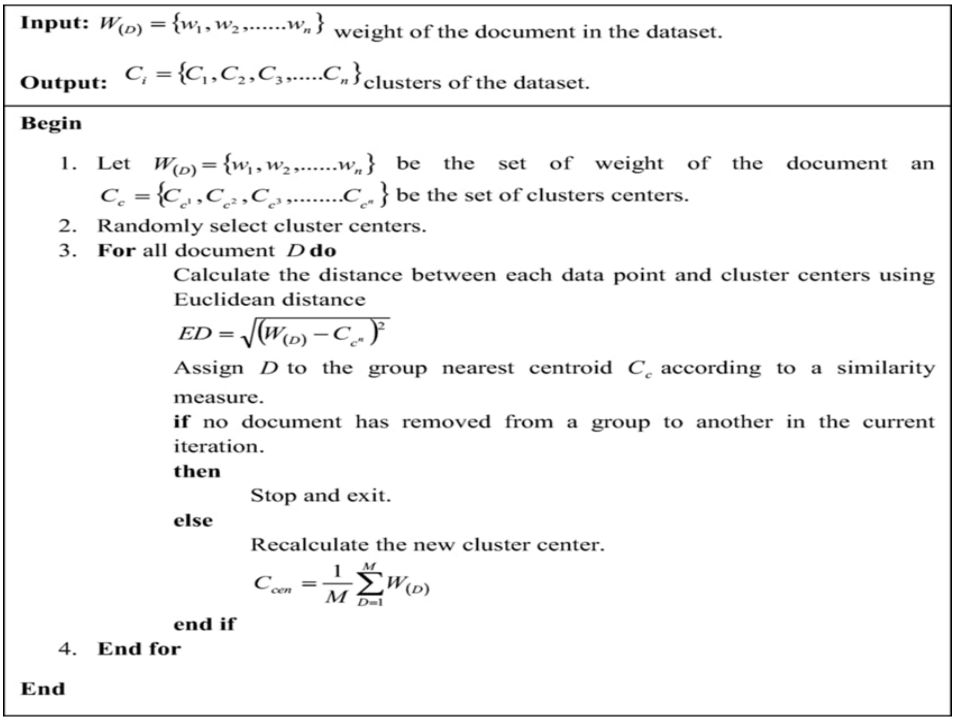
4- Convergence and Completion

The algorithm iterates between the assignment step and the update step until convergence occurs, meaning that the centroids stop moving significantly, or until a maximum number of iterations is reached.

In essence, K-means attempts to find K clusters by repeatedly adjusting the positions of cluster centers to minimize the distances between data points and their assigned centers. It is analogous to grouping similar items together by moving reference points until they are positioned at the center of their respective groups.



The following is the pseudocode for the K-Means algorithm.



```
Input: $W_{(D)} = \{w_1, w_2, ......w_n\}$ weight of the document in the dataset.
Output: $C_i = \{C_1, C_2, C_3, .....C_n\}$ clusters of the dataset.

Begin
    1. Let $W_{(D)} = \{w_1, w_2, ......w_n\}$ be the set of weight of the document an
       $C_c = \{C_{c^1}, C_{c^2}, C_{c^3}, ........C_{c^n}\}$ be the set of clusters centers.
    2. Randomly select cluster centers.
    3. For all document $D$ do
            Calculate the distance between each data point and cluster centers using
            Euclidean distance
            $ED = \sqrt{(W_{(D)} - C_{c^n})^2}$
            Assign $D$ to the group nearest centroid $C_c$ according to a similarity
            measure.
            if no document has removed from a group to another in the current
            iteration.
            then
                Stop and exit.
            else
                Recalculate the new cluster center.
                $C_{cen} = \frac{1}{M}\sum_{D=1}^{M} W_{(D)}$
            end if
    4. End for
End
```

Figure1. Explains the changes observed before and after applying the K-Means algorithm.

3.1.2 DBSCAN

DBSCAN: Density-Based Spatial Clustering of Applications with Noise

Introduction and History

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) was proposed by Xiaowei Xu, Jörg Sander, Martin Ester, and Hans-Peter Kriegel in 1996 as a data clustering technique. This non-parametric method for density-based clustering groups points that are closely spaced (i.e., have many nearby neighbors) using a set of points in space. Points that are isolated in low-density areas—those whose nearest neighbors are too far away—are labeled as outliers. DBSCAN is one of the most popular and frequently cited clustering algorithms.

The algorithm received the 2014 Test of Time Award at the ACM SIGKDD conference, an award given to algorithms that have gained significant attention in both theory and practice. The follow-up work "DBSCAN Revisited, revisited: Why and How You Should (Still) Use DBSCAN" ranks among the top 8 most downloaded publications as of July 2020 in the prestigious ACM Transactions on Database Systems (TODS) journal. Arthur Zimek further improved HDBSCAN* in 2015, producing hierarchical rather than flat results and modifying some original design choices, such as the treatment of border points.

Core Concept

The main idea of DBSCAN is to identify clusters using two parameters:

- **Epsilon (ε)**: establishes the radius of the neighborhood around a data point
- **MinPts**: indicates the minimum number of points needed to form a dense region (cluster)

Points that have at least minPts neighbors within ε distance of a core point are grouped together as members of the same cluster.

Advantages

DBSCAN is valuable because it is resistant to noise and outliers and can identify clusters of any shape. Its main advantage over other clustering algorithms is that it does not require the user to specify the number of clusters in advance.

Point Classification

For DBSCAN to function, each data point's neighborhood is iteratively examined and classified into three categories:

Core Points

Data points that have at least "minPts" neighbors within "epsilon" (ε) distance. Clusters are formed around core points. A core point has a circle with radius ε surrounding it, and if this circle contains at least "minPts" points (including itself), it is considered a core point.

Border Points

Data points that fall within the ε-distance of a core point but lack sufficient neighbors within their own ε-radius to be classified as core points themselves. Border points exist at cluster boundaries.

Noise Points

Data points that are neither core nor border points. They do not belong to any cluster and are regarded as noise or outliers.

Algorithm Process

Beginning with any unvisited data point, the algorithm:

1- Determines whether the initial point is a core point

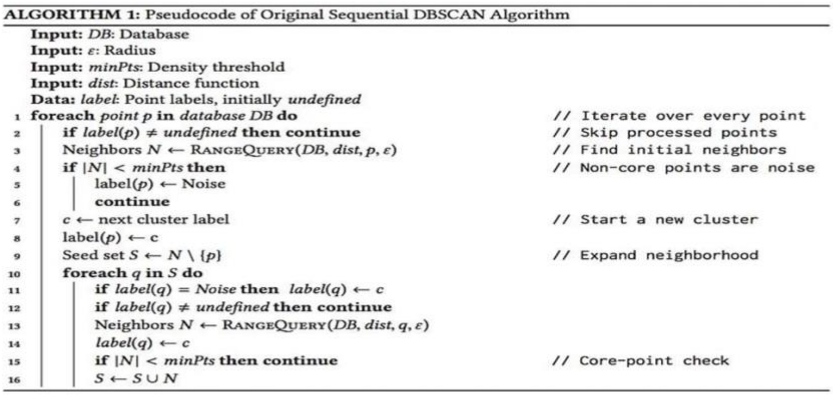
2- If it is a core point, recursively expands the cluster by adding nearby core and border points that are reachable from the starting point

3- Continues to an unvisited point and repeats the process until all points have been visited or no more points can be added. This process continues until every point has been processed.

Results

This technique efficiently delineates clusters in the data based on point density, with high-density regions forming clusters and sparse regions designated as noise. The algorithm successfully identifies clusters of varying shapes while maintaining robustness against outliers and noise.

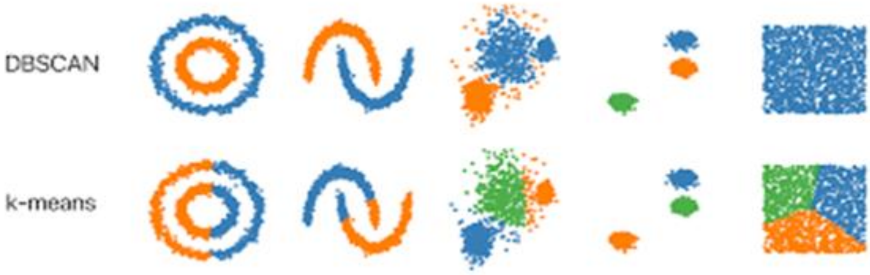
- Graphical Representation:
- Core Points: Represented by filled circles.
- Border Points: Represented by unfilled circles.
- Noise Points: Represented by points with an 'X'.
- This simplification should make the concept easier to grasp!



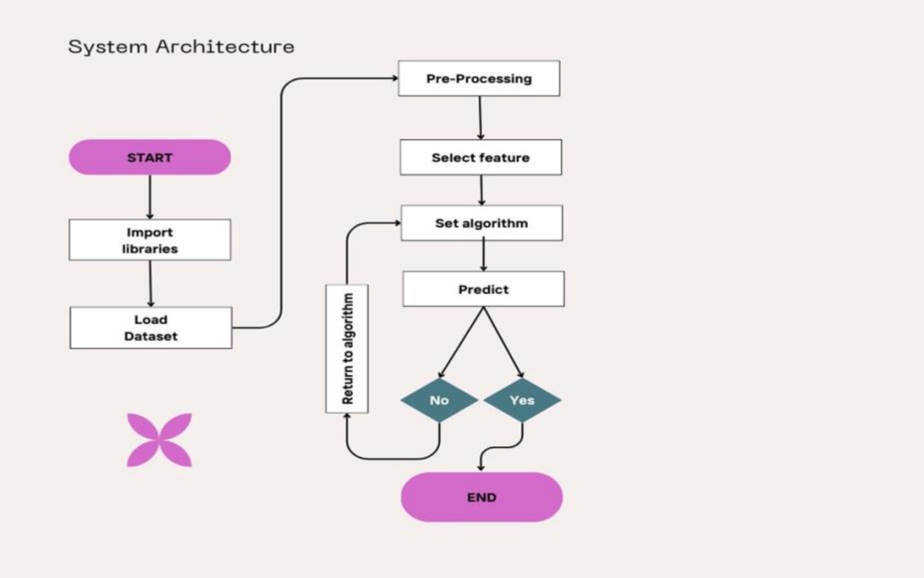
```
ALGORITHM 1: Pseudocode of Original Sequential DBSCAN Algorithm
Input: $DB$: Database
Input: $\varepsilon$: Radius
Input: $minPts$: Density threshold
Input: $dist$: Distance function
Data: $label$: Point labels, initially $undefined$
foreach $point\ p$ in $database\ DB$ do                       // Iterate over every point
    if $label(p) \neq undefined$ then continue                // Skip processed points
    Neighbors $N \leftarrow$ RangeQuery($DB, dist, p, \varepsilon$)   // Find initial neighbors
    if $|N| < minPts$ then                                    // Non-core points are noise
        label($p$) $\leftarrow$ Noise
        continue
    $c \leftarrow$ next cluster label                         // Start a new cluster
    label($p$) $\leftarrow$ c
    Seed set $S \leftarrow N \setminus \{p\}$                 // Expand neighborhood
    foreach $q$ in $S$ do
        if $label(q) = Noise$ then  $label(q) \leftarrow c$
        if $label(q) \neq undefined$ then continue
        Neighbors $N \leftarrow$ RangeQuery($DB, dist, q, \varepsilon$)
        $label(q) \leftarrow c$
        if $|N| < minPts$ then continue                       // Core-point check
        $S \leftarrow S \cup N$
```

Figure 2. DBSCAN pseudo code

Comparison DBSCAN and Kmean to find between friends



3.2 SYSTEM ARCHITECTURE



3.3 SYSTEM PROCESS

3.3.1 GATHERING DATA

- This part covers the importance of collecting data for the research. alternative ways to Gathering data from “kaggle” website or making a synthetic dataset for the research.

Importing the dataset (IPEDS_data.csv) into pandas data-frame.

3.3.2 IMPORT LIBRARIES

This part covers all the necessary libraries that should be imported for the tasks:

- Pandas: Python library used for working with data sets.

- Sklearn: Scikit-learn, also known as sklearn, is a Python machine learning library. It offers easy to use and effective tools for data analysis and mining.

- Matplotlib: Is a powerful Python library used to make various types of plots and charts, like histograms, scatterplots, and more.

- K-means: Is a machine learning algorithm available in Python's scikit-learn library. It's a simple and intuitive method used for regression tasks.

- DBSCAN: Is a machine learning algorithm available in Python's scikit-learn library. It's a simple and intuitive method used for classification and regression tasks.

3.3.3 DATA PREPROCESSING

This part covers preprocessing the dataset:

- Dataset (IPEDS_data.csv) is loaded into pandas Data Frame.
- applying label encoding for column such as “OPEID ” with label encoder() method for categorical purpose.
- applying future scaling for columns such as “LONGITUD, LATITUDE” with Standard Scaler () method for categorical purpose.

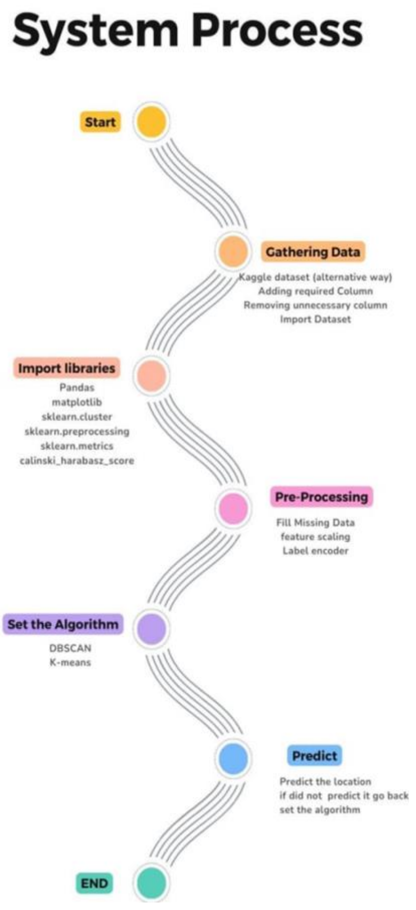


Figure 3. system process structure

4 RESULTS

4.1 PERFORMANCE COMPARISON

In this section presents a comparative analysis of our preposed models against existing algorithms.

Table 1. Silhouette Score

Algorithms	Silhouette Score
Density-Based Spatial Clustering of Applications with Noise (DBSCAN)	0.7800
K-means	0.5033
k-mean and DBSCAN labeled (KDB)	0.8880
Agglomerative Clustering	0.5690
Ordering points to identify the clustering structure (OPTICS)	0.2510

Table 2. Accuracy

Algorithms	Accuracy
Decision Tree Classifier	0.090
Support vector machine (SVM)	0.070

Table 3. Mixed supervised and unsupervised

Algorithms	Accuracy
Decision Tree Classifier and Density-Based Spatial Clustering of Applications with Noise (DBSCAN)(TBD)	0.097

4.2 EVALUATION METRIC AND MODEL PERFORMANCE

As assessment metrics, we used accuracy, f1-score, precision, and recall to gauge how well our suggested model performed.

Accuracy & Memory: Metrics like precision and recall are frequently used to assess how well categorization models work, particularly in the domains of information retrieval and machine learning.

Precision: Precision is the ratio of true positive predictions to the total number of positive predictions made by the model. In other words, it measures the accuracy of positive predictions made by the model. It is calculated using the formula:

$$\textbf{Precision} = \frac{\textit{True positive}}{\textit{True positive}+\textit{False positive}} \quad \text{eq 4.2.1}$$

Role: The precision indicates the percentage of the model's positive predictions that come true. It aids in comprehending the model's capacity to prevent false positives, or, to put it another way, to ensure that a given class prediction is almost certainly accurate.

Recall: Recall is the ratio of true positive predictions to the total number of actual positive cases in the dataset. It is sometimes referred to as sensitivity or true positive rate. It assesses the model's accuracy in identifying every good case. It is computed with the following formula:

$$\textbf{Recall=} \frac{True\ positive}{True\ positive+False\ negative} \quad \text{eq 4.2.2}$$

Role: Recall aids in our comprehension of the model's capacity to record every good experience. When there is a significant risk of overlooking a positive occurrence (false negative), it becomes crucial. For instance, even if it results in some false alarms (poor accuracy), we desire high recall in medical diagnosis to make sure we identify as many cases of a disease as feasible.

F1-score: A statistic called the F1 score aggregates recall and accuracy into a single number. It offers a balance between recall and accuracy as the harmonic mean of these two measurements. The following formula is used to determine the F1 score:

$$\textbf{F1-score=2*} \frac{precision*Recall}{precision*Recall} \quad \text{eq 4.2.3}$$

The F1 score ranges from 0 to 1, where 1 indicates perfect precision and recall, and 0 indicates either one of them being 0.

Role: When it's important to strike a balance between recall and precision, the F1 score is frequently employed. It offers a single figure that encapsulates a categorization model's effectiveness. When there is an unequal distribution of classes or when the costs associated with false positives and false negatives are significant, it is very helpful.

Accuracy: It gauges how accurate the model is overall in all of its predictions across all classes.

$$\textbf{Accuracy=} \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} \quad \text{eq 4.2.4}$$

NOTE:

True positive (TP) correctly identified

False positive (FP) = incorrectly identified.

True negative (TN) correctly rejected.

False negative (EN) = incorrectly rejected.

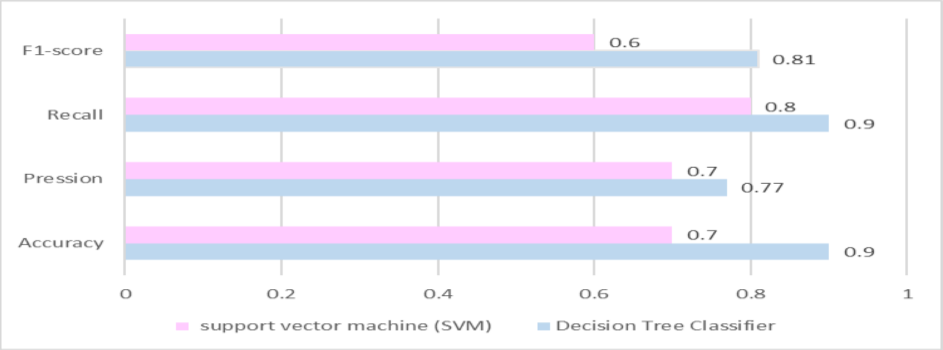


Figure 4. **Comparative between the Decision Tree Classifier and support vector machine (SVM)**

Table 4. Table of Evaluation Metrics

Algorithms	Accuracy	Pression	Recall	F1-score
Decision Tree Classifier	0.090	0.077	0.090	0.081
support vector machine (SVM)	0.070	0.070	0.080	0.060

4.3 PLOT OF THE ALGORITHMS

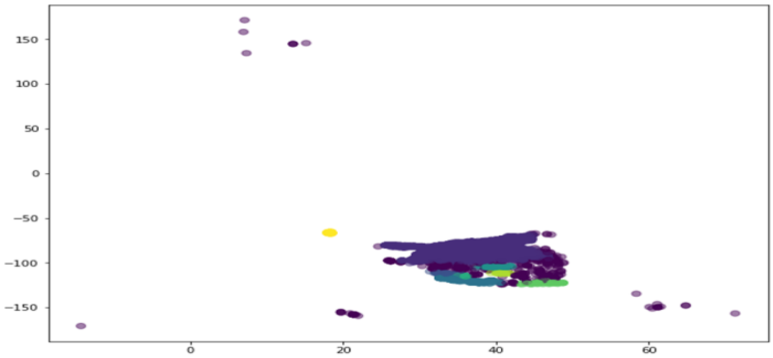


Figure 5. The plot of the TDB

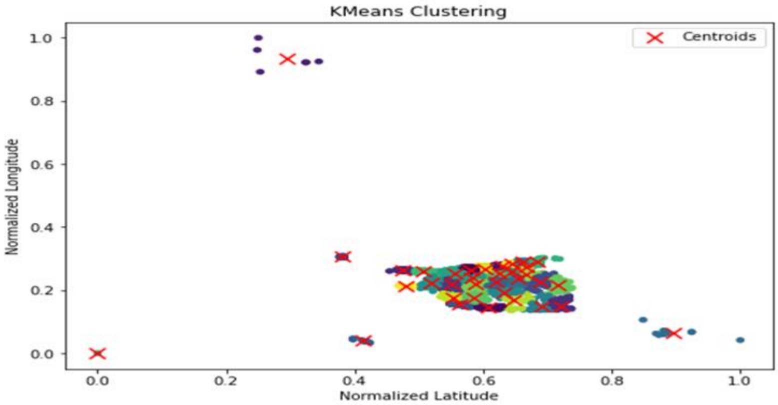


Figure 6. The plot of the K-mean

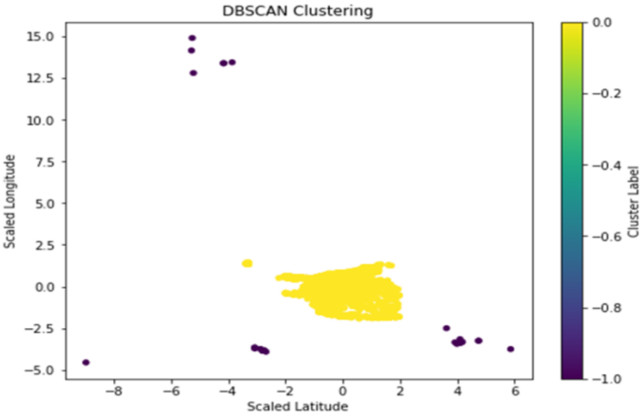


Figure 7. The plot of the DBSCAN

CONCLUSION

Based on our comprehensive analysis, we recommend the Decision Tree-DBSCAN hybrid approach for organizations seeking to implement effective colleague location prediction systems. The methodology should be implemented with careful attention to privacy considerations, data quality requirements, and computational resource availability. Organizations should also consider pilot implementations to validate the approach within their specific operational contexts before full-scale deployment.

The success of this hybrid approach opens new possibilities for advanced workplace analytics and demonstrates the value of combining different machine learning paradigms to address complex organizational challenges. As remote and hybrid work models continue to evolve, such location-based prediction systems will become increasingly valuable for maintaining effective collaboration and organizational cohesion.

REFERENCES

[1] P. Zhang, S. Yu, Y. Zeng, & M. Li, “Analysis of a New Kind of Memcapacitor,” Physica A: Statistical Mechanics and its Applications, vol.387, no.12, pp.2975-2982, 2008.

[2] M. Kiełbasiński, A. Pacut, & T. Skotnicki, “Optimization of Power Parameters of an Energy Storage System Based on a Supercapacitor in an Electric Vehicle,” in 2011 International Conference on Electrical Machines and Systems (ICEMS), Beijing, China, pp.1-5, 2011.

[3] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, & A. Y. Wu, “A local search approximation algorithm for k-means clustering,” Proceedings of the eighteenth annual symposium on Computational geometry - SCG ’02, 2002, doi: https://doi.org/10.1145/513400.513402.

[4] J. Jeong, M. So, & H.-Y. Hwang, “Selection of Vertiports Using K-Means Algorithm and Noise Analyses for Urban Air Mobility (UAM) in the Seoul Metropolitan Area,” Applied Sciences, vol.11, no.12, pp. 5729, 2021, doi: https://doi.org/10.3390/app11125729.

[5] A. Gupta & K. Tangwongsan, “Simpler Analyses of Local Search Algorithms for Facility Location,” arXiv.org, Sep. 15, 2008. https://arxiv.org/abs/0809.2554 (accessed May 03, 2024).

[6]C.-N. Huang, & C.-T. Chan, “ZigBee-based indoor location system by k-nearest neighbor algorithm with weighted RSSI,” Procedia Computer Science, vol. 5, pp. 58–65, 2011, doi: https://doi.org/10.1016/j.procs.2011.07.010.

[7] G. Zhang, Y. Li, C. Yan, & S. Li, "Energy-Efficient Scheduling for Wireless-Powered Internet of Things with Energy Harvesting," in IEEE Internet of Things Journal, vol.7, no.4, pp. 3385-3395, 2020.

[8] R. Adam, & P. Chi, "Deep Learning-Based Automatic Modulation Classification for Cognitive Radio Networks," in IOP Conference Series: Earth and Environmental Science, vol.587, no.1, pp. 012120, 2021.

[9] H. Ma, & X. Zhou, “A GPS location data clustering approach based on a niche genetic algorithm and hybrid K-means,” Intelligent Data Analysis, vol.23, pp.175-198, Jun. 2019, doi: https://doi.org/10.3233/ida-192791.

[10] K. Krishna, & M. Narasimha Murty, “Genetic K-means algorithm,” IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics), vol. 29, no.3, pp.433-439, 1999, doi: https://doi.org/10.1109/3477.764879.

[11] R. Mishra, V. K. Jain, & A. K. Verma, “Performance Analysis of Multi-Hop Cellular Networks with Mixed RF/FSO Links,” in IEEE Transactions on Communications, vol.67, no.9, pp.6461-6475, 2019.

[12] M. M. Abdoel Wahid, “Determining The Location Of RMU, Using K-Means Clustering, Evaluate The Location Of Existing RMU, Using R-Programming,” Journal of Informatics and Telecommunication Engineering, vol.6, no.1, pp. 10-17, 2022, doi: https://doi.org/10.31289/jite.v6i1.6126.

[13] W. Wang, L. Tao, C. Gao, B. Wang, H. Yang, & Z. Zhang, “A C-DBSCAN Algorithm for Determining Bus-Stop Locations Based on Taxi GPS Data,” Lecture notes in computer science, pp.293-304, 2014, doi: https://doi.org/10.1007/978-3-319-14717-8_23.

[14] Y. Wen, B. Wang, & H. Yuan, “A multi-objective optimization algorithm based on particle swarm optimization for RF coverage and throughput in heterogeneous networks,” Journal of Network and Computer Applications, vol.143, pp. 37-45, 2019.

[15] K. B. Tayyaba, T. R. Ahmed, S. Z. Zahra, & M. S. Ali, “Analysis of Road Accident Locations Using Geographic Information System (GIS) Techniques: A Case Study of Karachi, Pakistan,” in International Journal of Transportation Engineering and Technology (IJTET), vol.4, no.1, pp.1-11, Jan. 2018. Available: https://www.academia.edu/download/57082314/4599.pdf.

[16] D. Birant, & A. Kut, “ST-DBSCAN: An algorithm for clustering spatial-temporal data,” Data & Knowledge Engineering, vol. 60, no.1, pp.208-221, 2007, doi: https://doi.org/10.1016/j.datak.2006.01.013.

[17] S. R. Gubbala, S. A. Alshehri, A. S. Almansouri, & M. K. Hasan, “Performance analysis of massive MIMO systems with ZF precoding in the presence of transceiver impairments,” in 7th International Conference on Computer and Communication Engineering (ICCCE), Kuala Lumpur, Malaysia, pp.308-313, 2018

[18] Z. Wen, Z. Wang, & X. Xu, “Prediction of the collapse height of rubble piles using a hybrid neural network,” Advances in Engineering Software, vol.164, pp.102992, 2022.

[19] M. Z. Hossain, N. Nahar, M. T. Haque, & M. S. Islam, “Robust Iterative Learning Control for Uncertain Nonlinear Systems,” in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol.51, no.2, pp.1132-1143, 2021.

[20] H. A. Abbas, S. A. Mohammed, & A. A. Hassan, “Design and Implementation of a Smart Traffic Light Control System Based on Image Processing Techniques,” International Journal of Advanced Computer Science and Applications (IJACSA), Accessed on: May 3, 2024.

[21] K. M. Hama Karim, “Link Prediction in Dynamic Networks Based on the Selection of Similarity Criteria and Machine Learning,” UHD Journal of Science and Technology, vol.7, no.2, pp.32-39, 2023.

[22] A. El-Safty, M. El-Sahn, & M. K. El-Sherbeni, “Miniaturized Planar Antennas with Stable Radiation Characteristics Over Wide Bandwidths,” IEEE Transactions on Antennas and Propagation, vol.67, no.4, pp.2466-2474, 2019.

[23] N. E. Elhoseny, S. M. Riad, M. K. Hassan, A. S. El-rabaie, & M. Abdelrazek, “Efficient technique for MRI brain tumor detection using neural network,” Applied Intelligence, vol.48, no.9, pp.2595-2605, 2018.

[24] K. Kanagasabai, P. T. Vanathi, “Location-aware cluster-based routing in wireless sensor networks,” International Journal of Communication Networks and Distributed Systems, vol.26, no.2, pp.123-135,2021.

[25] J. Sampaio, C. Lago, L. Casais, & N. Leite, “Effects of starting score-line, game location, and quality of opposition in basketball quarter score,” European Journal of Sport Science, vol.10, no.6, pp.391-396, 2010, doi: https://doi.org/10.1080/17461391003699104.

[26] M. G. Ferrari, F. Mannocci, & S. A. Vichi, “A clustering procedure for the estimation of fibre direction in three-dimensional images from X-ray microtomography,” Journal of Microscopy, vol.244, no.1, pp.1-10, 2011.

[27] S. Xie, Y. Yao, & Y. Zhang, “Research on Location-Based Access Control Model for Web Services,” in Proceedings of the 7th International Conference on Intelligent Information Processing, IIP 2014, Hangzhou, China, pp. 239-248, 2014.

[28] Y. A. Siddiqi, M. Gondal, & R. A. Khan, “An efficient secure authentication protocol for wireless body area sensor networks,” Wireless Personal Communications, vol.82, no.4, pp.2677-2696, 2015.

[29] Kakarash, Z.A., Karim, S.H.T., Ahmed, N.F. & Omar, G.A., “New topology control base on ant colony algorithm in optimization of wireless sensor network,” Passer Journal of Basic and Applied Sciences, vol.3, no.2, pp.123-129. 2021.