ROUTE OPTIMIZATION USING HYBRID CLUSTERING AND OPTIMIZED LEVY FRUIT FLY ALGORITHM

[1]Ms.M.Kavitha

Research Scholar, Department of Computer Science, Gobi Arts and Science College, & Assistant Professor, Department of Computer Software Systems, KG College of Arts and Science, Coimbatore.

[2]Dr.G.T.Prabavathi

Associate Professor, Department of Computer Science, Gobi Arts and Science College, Gobichettipalayam.

Abstract

Best route finding plays a crucial role in modern transportation systems for efficient traffic management. This study proposes a novel approach for best route finding and route optimization using a hybrid clustering and optimization method. The first step involves applying Hierarchical Clustering with K-Means Clustering to the traffic data, followed by applying the Deepwalk method for signal processing. Deepwalk generates random walks on a graph and uses a Skip-gram model to capture collocations within a specified window, aiding in traffic pattern recognition. The second step utilizes an optimized Levy fruit fly algorithm for route optimization. This optimization algorithm is inspired by the gathering good behavior of fruit flies and aims to find the best route considering best route and other factors. The algorithm iteratively refines routes based on real-time traffic data, leading to improved route recommendations. Lastly, a comparative analysis is conducted to evaluate the performance of the proposed method against existing techniques. Metrics such as travel time, traffic flow, and route efficiency are used to assess the effectiveness of the hybrid clustering and optimization approach. The results demonstrate the potential of the proposed method in achieving efficient traffic management and route planning in urban environments.

Keywords: Deepwalk, K-Means clustering, Levy fruit fly algorithm, Real-time traffic data, Route finding

I. Introduction

People can't go about their everyday lives without transportation. Global traffic is expected to surge by 60% by 2030, according to an estimate [1]. In recent years, intelligent transportation systems have been the subject of substantial study [2]. In order to create a more seamless system that includes people, roads, and cars, ITS uses cutting-edge data communication technologies to combine information, communications, and other technology in the transportation sector [3-4]. It can set up a massive transportation management system that is completely functional, accurate, and runs well in real-time [5]. ForIntelligent Transport System(ITS) data study, the three primary components are traffic volume, best route finding, and traffic flow. Journey times, traffic densities, and other real-time data are essential for intelligent transportation systems to make good control choices and provide consumers accurate information [6-7]. Despite differences in intelligent transportation technologies used by different communication systems, all of them allow data collection from a variety of sources and transmission of information based on vehicle data [8-9]. All lives are affected when individuals are forced to spend a great deal of time traveling because of traffic congestion. Scientists have been trying to classify and assess traffic situations in an effort to resolve these issues [10, 11].

When it comes to managing traffic, the tried-and-true methods like wireless sensors, speed guns, roadside radars, and infrared counters just don't cut it [12–13]. In contrast to older methods of traffic management, ITS data provided by the best route finding can help with congestion control, early forecast, and traffic conditions [14–15]. Based on these limitations—that is, people, cars, and their linked sensors to the outside world—researchers presented a framework [16]. In order to evaluate and analyze these characteristics, there are a number of simulation tools that are accessible. Along with their characteristics, modern simulation tools were suggested by study work [17]. After weighing the benefits and drawbacks of the tools, the authors came to the conclusion that in situations with diverse traffic, particularly in developing countries, a safety micro simulation model is necessary. The automated measurement of traffic route is used in intelligent transportation systems for the purpose of traffic management and control [18]. The development of autonomous signaling systems and early warning systems depends on accurate observations of best route.

The main contribution of the paper is:

- Apply Hierarchical Clustering with K-Means Clustering
- Finding the signal Processing with Deepwalk method
- Optimization using optimized Levy fruit fly

This paper is organized as follows for the rest of it. Section 2 discusses a number of best routefinding algorithms from different authors. In Section 3, we can see the suggested model. The investigation's findings are summarized in Section 4. A discussion of the outcome and potential future research makes up Section 5.

1.1 Motivation of the paper

With global traffic expected to surge by 60% by 2030, traditional traffic management techniques are becoming increasingly inadequate. Intelligent Transportation Systems (ITS) offer a promising alternative by leveraging advanced data communication technologies for real-time decision-making. This study introduces a novel hybrid clustering and optimization method to best route finding and traffic management, addressing the urgent need for more efficient transportation solutions.

Vol. 21, No. 1, (2024)
ISSN: 1005-0930

II. Background study

Abdelhafid et al. (2018) presented a hybrid observer-based strategy combining model-based observers and real-time data to estimate traffic density and detect congestion. This method balances the accuracy of model-based approaches with the adaptability of data-driven techniques.

Chen et al. (2019) developed an efficient algorithm for finding the k shortest paths based on re-optimization, highlighting its utility in dynamic traffic environments.

Several studies have employed CNNs for traffic density estimation. Ikiriwatte et al. (2019) utilized CNNs to control traffic, demonstrating significant improvements in traffic management.

Lei et al. (2014) and Li and Han (2020) discussed the application of fruit fly optimization algorithms in structural optimization and pathfinding, illustrating the potential of bio-inspired algorithms in traffic management.

Mittal and Chawla (2023) introduced an ensemble of deep learning models for vehicle detection and traffic density estimation, highlighting the robustness and enhanced accuracy of ensemble approaches compared to single-model methods.

Paliwal et al. (2021) employed online variational Bayesian subspace filtering for traffic estimation and prediction, showcasing its real-time applicability and accuracy in dynamically changing traffic conditions.

Table 1: Comparison table for Shortest Path finding Optimization Techniques

Approach	**Strengths**	**Limitations**	**Data Sources**	**Computational Requirements**	**Real-Time Applicability**
Ensemble Deep Learning Models (Mittal & Chawla, 2023)	Robustness; Improved accuracy over single models	Increased computational complexity; Need for diverse model integration	Real-time video feeds	Very high during training and inference	Limited

Vol. 21, No. 1, (2024)
ISSN: 1005-0930

Multiple Model Stochastic Filtering (Panda et al., 2019)	Effective in handling stochastic nature of traffic; Robust	Computationally intensive; Complex implementation	Sensor data, traffic cameras	High	Limited
Variational Bayesian Subspace Filtering (Paliwal et al., 2021)	Real-time applicability; Handles dynamic traffic conditions	Requires extensive data for training; Computationally intensive	Mixed (sensor data, traffic feeds)	High	Yes
Firefly and Fruit Fly Optimization Algorithms (Lei et al., 2014; Li & Han, 2020)	Bio-inspired; Effective in optimization	May not scale well; Require parameter tuning	Simulation data	Moderate	No
Re-Optimization Techniques (Chen et al., 2019)	Efficient pathfinding; Suitable for dynamic environments	Requires re-optimization with each change; High computational overhead	Traffic flow data	High	Limited

2.1 Problem definition

Traffic congestion is a growing issue due to urbanization and increased vehicle numbers, leading to longer travel times, higher emissions, and greater stress. Accurate traffic density estimation and congestion detection are essential for effective traffic management and route optimization. Challenges include diverse data sources, high computational requirements, and the need for real-time processing. Efficient solutions must balance accuracy, resource demands, and timely responses to dynamic traffic conditions. Efficient solutions must balance accuracy, resource demands, and timely responses to dynamic traffic conditions.

Vol. 21, No. 1, (2024)
ISSN: 1005-0930

III. Materials and methods

In this section, we outline the methodologies employed in our study for best route finding and route optimization. We first describe the application of Hierarchical Clustering with K-Means Clustering to the traffic data, followed by the utilization of the Deepwalk method for signal processing. Deepwalk generates random walks on a graph and employs a Skip-gram model to capture collocations within a specified window, aiding in traffic pattern recognition. Subsequently, we introduce the optimized Levy fruit fly algorithm for route optimization, inspired by the foraging behavior of fruit flies and designed to find the best routes considering traffic and other relevant factors.

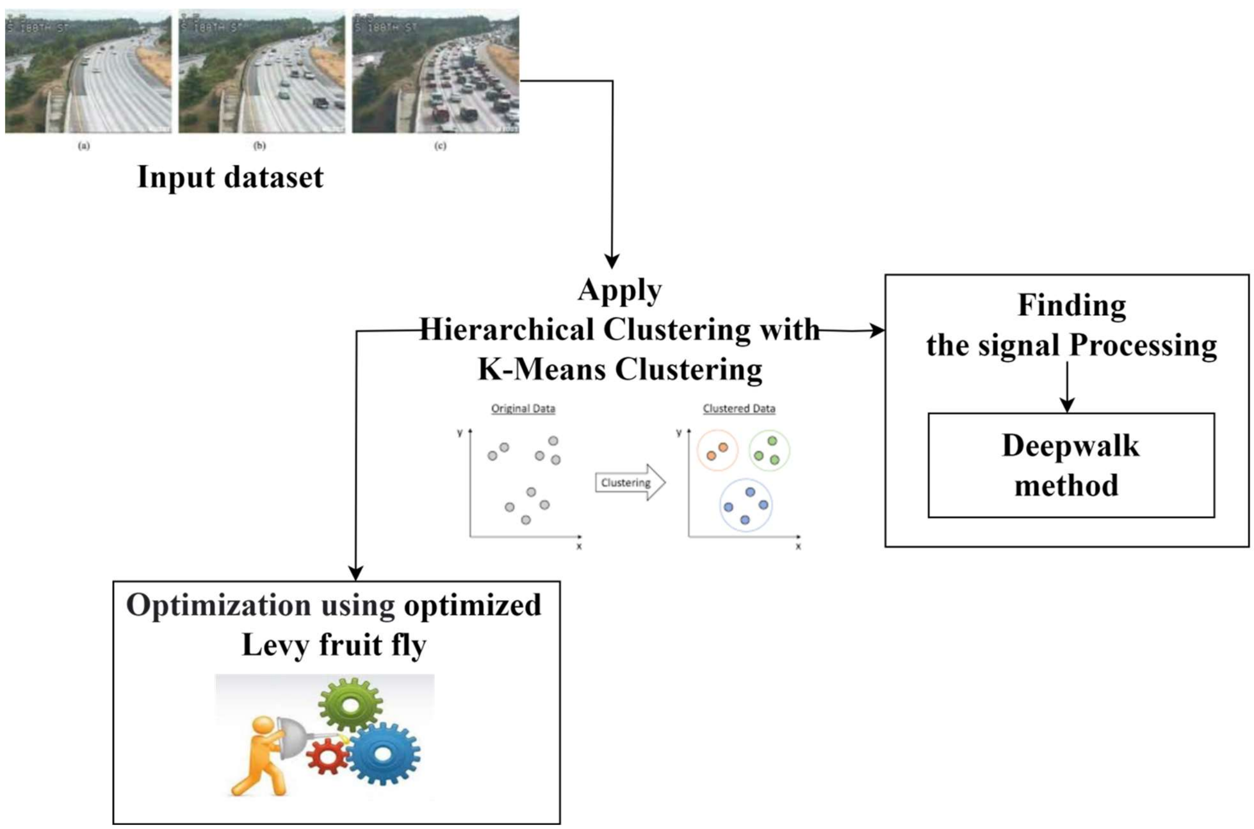


Figure 1: Shortest route finding workflow architecture

3.1 Apply Hierarchical Clustering with K-Means Clustering

In order to minimize the k-means problem, one can use the k-means algorithm referred by Huang, Z. et al. (2024). There are many variants of the algorithm, which will be covered later on. However, before using any of these variants, one must know how many clusters are in the data. To find the optimal number of clusters, one must run or trial the algorithm multiple times. The data set's properties, its size, and the number of variables in each instance determine the likelihood to generate global optimum, so there is no optimal k-means method. Both the assignment and centroid update phases are iterative in k-means clustering methods. In the assignment phase, all data points are assigned to the cluster's centroid using the nearest one, and in the centroid update phase, the clusters' centroids are updated based on the partition obtained in the previous phase.

Data analysis tasks including pattern identification, pattern recognition, data summarization, and image processing all include clustering, which is the act of grouping data items into various groups or clusters. Partitioned approaches, hierarchical methods, density-based methods, etc. are only a few of the many subfields that have emerged to support these main areas.

$$m_j = \frac{\sum_{P_i \in C_j} P_i}{|C_j|} \text{-------------} (1)$$

This equation calculates the centroid m_j for cluster C_j by taking the sum of all data points P_i in cluster C_j and dividing it by the total number of points in clusterC_j, denoted as $|C_j|$.

$$SSE = \sum_{j=1}^{k} \sum_{i=1}^{n} \left|\left|p_i - m_j\right|\right|^2 \text{--------------} (2)$$

This equation represents the Sum of Squared Errors (SSE) in the context of k-means clustering. It calculates the total squared distance between each data point p_i and its corresponding cluster centroid m_j, summed over all clusters k and data points n.

Over the last half-century, k-means has grown in popularity as a member of the clustering community due to its straightforward but successful approach. Nevertheless, k-means has a major drawback: if the initialization is inadequate, the random selection of centers might cause the network to get stuck in a suboptimal local minimum. To attain minimum SSE, k-means often splits a big cluster into many smaller ones or combines nearby small clusters into a bigger one. This is the most apparent finding.

Using an iterative process called K-means; N items are partitioned into K distinct clusters. K-means is the most well-known clustering algorithm that employs centroids to display clusters, and it is also one of the most used clustering techniques overall. The within-cluster squared error criteria are used to quantify the quality of k-means clustering.

Using K-Means Clustering is a technique for organizing data sets that are not completely organized. The ease and capacity to manage massive data sets make this one of the most popular and successful data classification approaches.

As input parameters, it takes the cluster count and the starting set of centroids. We find the distance between the center of each cluster and every item in the dataset. The next step is to put the item in the cluster from which it is the farthest away. It is recalculated to get the item's cluster centroid.

Vol. 21, No. 1, (2024)
ISSN: 1005-0930

Finding the distance of the point from the selected mean is one of the most essential and often used approaches for categorizing elements of a data collection using K-Means Clustering. Although there are many methods for computing distances of this kind, the standard one is the Euclidean Distance.

$$d(p,q) = \sqrt{((x_1(p) - x1(q)^2 + \left(x2(p) - x2(q)\right)^2 + \cdots)} \text{ ------------ (3)}$$

This equation represents the Euclidean Distance between two points p and q in a multi-dimensional space. The Euclidean Distance is a measure of the straight-line distance between two points and is commonly used in clustering algorithms, including K-Means.

In the equation, $x_1(p)$ and $x1(q)$ represent the first-dimensional coordinates of points p and$q$, $x2(p)$ and $x2(q)$ represent the second-dimensional coordinates, and so on for higher dimensions.

Figure 2 illustrates the clustering process of traffic density for vehicles. The proposed model divides the area into four sub-clusters, each with varying density levels. The optimization techniques applied identify the minimum density points within each cluster and determine the best route based on these clusters and the optimization results.

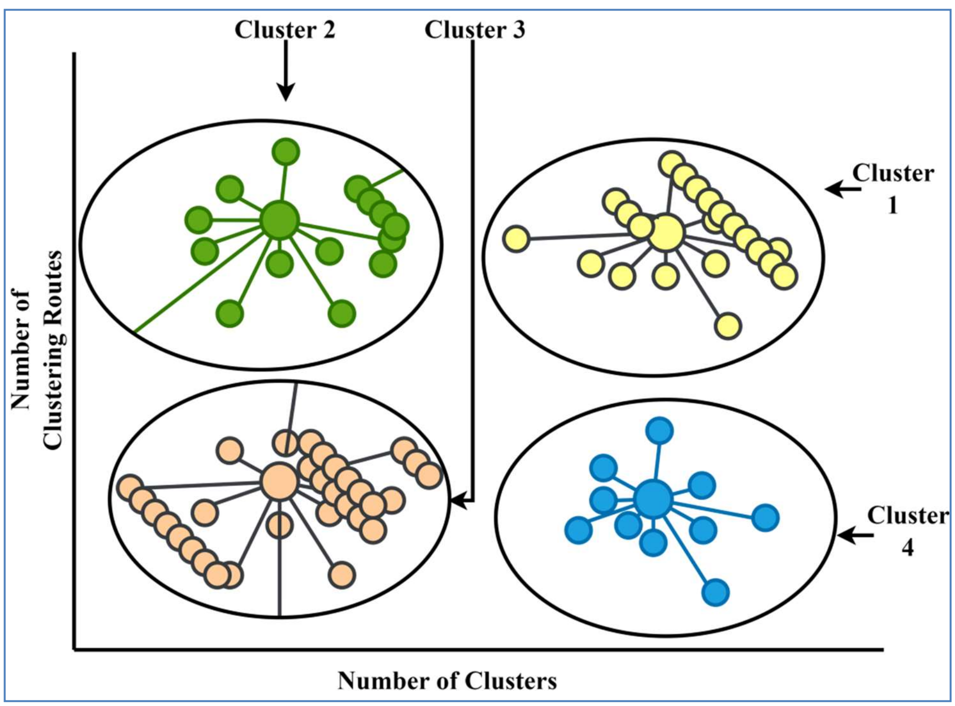


Figure 2: Clustering Process

Algorithm 1: Hierarchical Clustering with K-Means Clustering

Input:

- Traffic data points
- Number of clusters (k)
- Maximum number of iterations
- Convergence threshold
- Initial centroids

Steps:

1. Initialize k centroids randomly or using a predefined method.

$$m_j = \frac{\sum_{P_i} \in C_j P_i}{|C_j|}$$

2. Repeat until convergence or maximum iterations reached:
   Check for convergence based on the change in centroids or the iteration limit.

$$SSE = \sum_{j=1}^{k} \sum_{i=1}^{n} \left|\left|p_i - m_j\right|\right|^2$$

3. Return the final cluster assignments and centroids.

$$d(p,q) = \sqrt{((x_1(p) - x1(q)^2 + \left(x2(p) - x2(q)\right)^2 + \cdots)}$$

Output:

- Cluster assignments for each data point
- Centroids of the clusters

3.2 Finding the signal processing withDeepWalk Algorithm

When mining characteristics of large-scale network structures, the traditional approach for network representation learning known as DeepWalk comes in handy referred by Jeyaraj, R. et al. (2024). The three primary neural network layers that make it up are the input, hidden, and output layers. Due to their shared foundation in the Word2Vec algorithm referred by Rakshit, P., & Sarkar, A. (2024), the DeepWalk algorithm and Word2Vec algorithm have identical implementation details. As a traditional method for learning word representations, the Word2Vec algorithm incorporates the relationships between present and context words' structures into a low-dimensional vector space. This allows words with similar semantics or structural associations to be closer together in the representation space. The CBOW model and the Skip-Gram model are the basis of the algorithms used by DeepWalk and Word2Vec. Furthermore, negative sampling and hierarchical softmax are two distinct optimization strategies that are provided by these two algorithms. Both theContinuous bag-of-words (CBOW) and Skip-Gram models use the words in

Vol. 21, No. 1, (2024)
ISSN: 1005-0930

the present context to make predictions about the words in the future. Hence, one of these two training models or one of these two optimization strategies can be used for the representation learning job of training networks or language models. Lastly, four different training models are available for use withNetwork representation learning(NRL) tasks.

The continuous bag-of-words model, or CBOW for short, is a way to train a model for representing a network by making predictions about the nodes in the node's environment. Increasing the logarithmic likelihood function to its maximum is the goal of the learning process.

$L(v) = \sum_{v \in C \varepsilon \epsilon \{v\}} \sum_{uv \epsilon NEG(v)} logp\left(\varepsilon \middle| context(v)\right)$------------- (4)

Where

$p\left(\varepsilon \middle| context(v)\right) = [\sigma(x_v^T \theta^\varepsilon)]^{L^v}(\varepsilon) \times [1 - \sigma(x_v^T \theta^\varepsilon)]^{(1-L^v(\varepsilon)}$-------------- (5)

In the above equations, the sigmoid function is denoted by$\sigma\ (x)$. Its parameter domain is $[\sigma(x_v^T \theta^\varepsilon)]^{L^v}$ and its value range is (0, 1). x_v^TIs the vector that has to be trained for the current node π, and $(x_v^T \theta^\varepsilon)$ is the total of all the representation vectors in Context(x). The node x that is now being referenced by node v is called the context node. Negative sampling of the current node is represented by (v). Node v is considered a positive sample for a given Context(x), whereas all other nodes in the network are considered negative samples, and $(a) \neq \emptyset$. We make use of

$L^v(u) = \begin{cases} 1, u = v \\ 0, u \neq v \end{cases}$-------------- (6)

As the sampling result of the node u In Equation (6), the positive sample label of the node u is 1, and the negative sample label of the node u is 0.

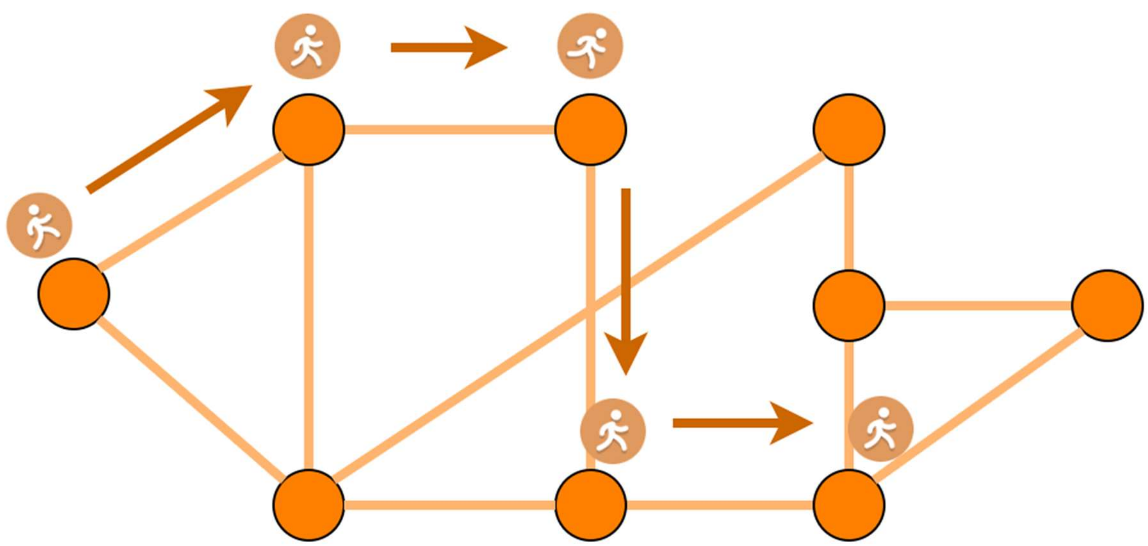


Figure 3: DeepWalk process

Vol. 21, No. 1, (2024)
ISSN: 1005-0930

Algorithm 2: DeepWalk Algorithm

Input:

- Traffic network data (graph structure)
- Number of walks per node
- Walk length
- Window size for Skip-Gram model
- Embedding dimension

Steps:

1. Construct a graph representation of the traffic network based on traffic flow data, where nodes represent intersections or road segments, and edges represent traffic connections.
2. Initialize node embeddings randomly or using pre-trained embeddings.

$$p\left(\varepsilon \middle| context(v)\right) = [\sigma(x_v^T \theta^\varepsilon)]^{L^v}(\varepsilon) \times [1 - \sigma(x_v^T \theta^\varepsilon)]^{(1-L^v(\varepsilon)}$$

3. For each node in the graph, perform multiple random walks of fixed length starting from that node.

$$logp\left(\varepsilon \middle| context(v)\right)$$

4. Generate sequences of nodes visited during each random walk.

$$(v) = \sum_{v \in C\varepsilon\epsilon\{v\}} \sum_{uv\epsilon NEG(v)} logp\left(\varepsilon \middle| context(v)\right)$$

5. Apply the Skip-Gram model to learn node embeddings based on the generated sequences.
6. Define the optimization objective using negative sampling or hierarchical softmax to train the Skip-Gram model.
7. Update the node embeddings iteratively using stochastic gradient descent or another optimization algorithm.

$$L^v(u) = \begin{cases} 1, u = v \\ 0, u \neq v \end{cases}$$

Output:

- Node embeddings representing traffic patterns
- Trained DeepWalk model

3.3 Optimization using optimized Levy fruit fly

An innovative global optimization method, the optimized Levy fruit fly models its operations after those of a fruit fly. The following is the fundamental concept of the optimized Levy fruit fly, as represented in Figure 4, which depicts the process of a fruit fly seeking food.

First, the fruit fly is in a highly sophisticated olfactory search stage. To begin, it surveys its immediate vicinity for a variety of scents using its sense of smell. In the second step, known as "visual positioning," the fruit fly uses its sense of smell to guide it to an area where it can see food, then uses its eyes to pinpoint its exact location, and finally flies to it. Consequently, the following are the main components of the optimized Levy fruit fly:

Step 1 Initialization: It describes the size, location, and maximum number of iterations of a fruit fly swarm.

$InitX_{axis}$------------ (7)

This variable represents the initial X-coordinate or position on the axis for the fruit fly swarm

$InitY_{axis}$------------- (8)

This variable represents the initial Y-coordinate or position on the axis for the fruit fly swarm

Step 2 Give each fruit fly a completely arbitrary direction and distance to fly in while foraging. “I” represents the ith fruit fly.

$X_i = X_{axis} + Randomvalue$------------- (9)

This equation calculates the new X-coordinate for the ith fruit fly by adding a random value to the initial X-coordinate X_{axis}.

$Y_i = Y_{axis} + randomvalue$---------------- (10)

Step 3 Since we don't know where the food is, we need to figure out how far each fruit fly is from its starting position $Dist_i$ and then use that distance to determine the smell concentration judgment value X_i^2 (where $X_i^2$is the value we used to determine the concentration of smells)

$Dist_i = \sqrt{X_i^2 + Y_i^2}$--------------- (11)

$S_i = \frac{1}{Dist_t}$-------------- (12)

Step 4. To get each fruit fly's taste concentration value, just plug the judgment value S for taste concentration into the function.

$Smell_i = Function(S_i)$--------- (13)

Step 5 Locate the fruit fly swarm member with the highest concentration of favorable traits.

$[bestSmellbestIndex] = \min(Smell)$---------- (14)

Step 6 After recording the ideal concentration of scent and its associated x and y coordinates, the fruit fly uses its eyesight to fly to that spot.

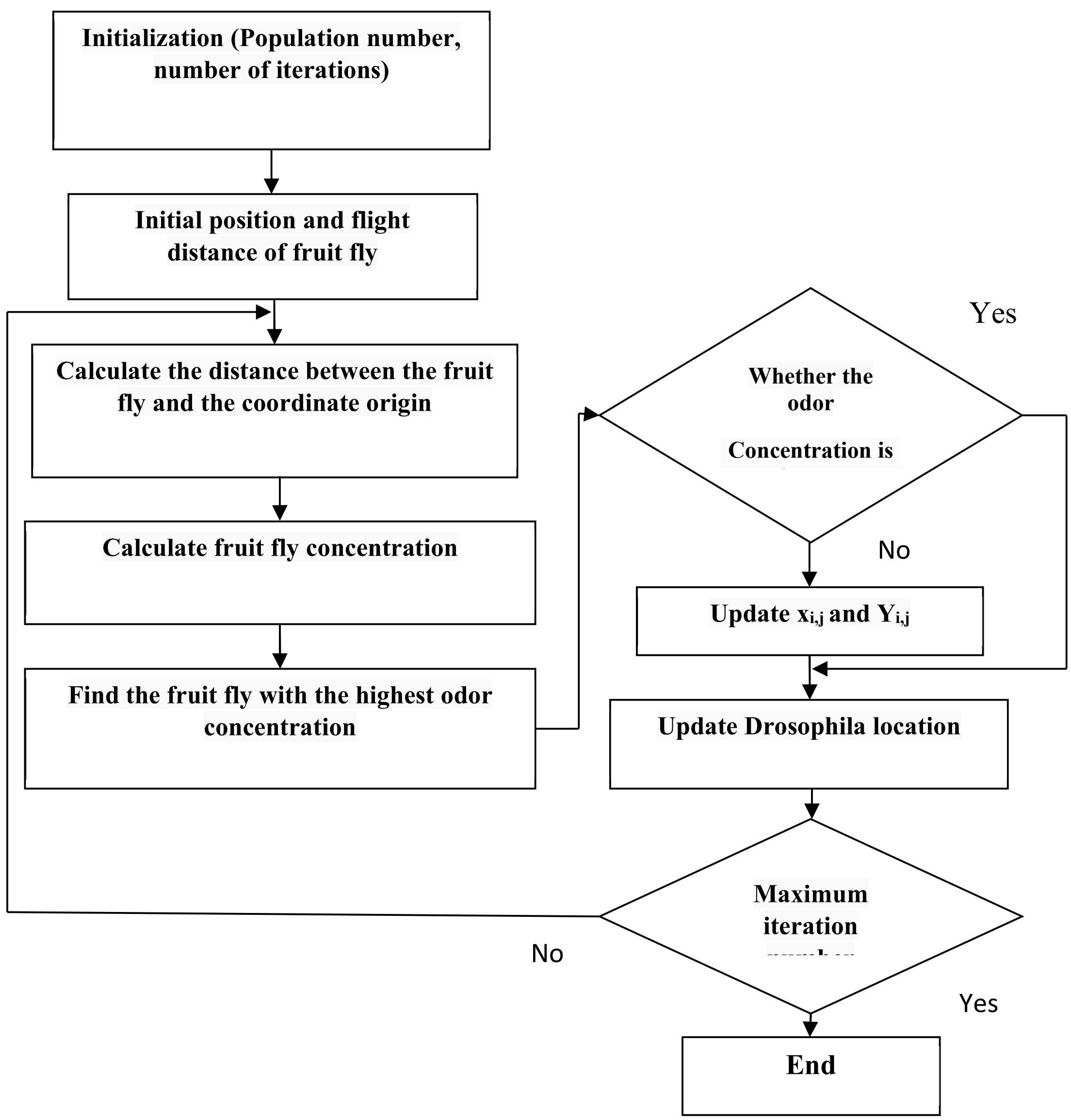
$$Smellbest = bestSmell$$

$X_{axsi} = X(bestIndex)$------------ (15)

$Y_{axis} = Y(bestIndex)$------------- (16)

These equations update the X-coordinate and Y-coordinate of the fruit fly swarm to the coordinates of the fruit fly with the highest smell concentration, preparing for the next iteration of optimization.

Step 7 Iteratively execute the optimization from Steps 2–5, evaluate each run for an improved scent concentration, and proceed to Step 6 if warranted.



Vol. 21, No. 1, (2024)
ISSN: 1005-0930

Figure 4: Flowchart for optimized Levy fruit fly

Algorithm 3: optimized Levy fruit fly

Input:
• Swarm size (number of fruit flies)
• Initial swarm location (X_{axis}, Y_{axis})
• Maximum number of iterations
Steps:
1. Initialize:
  ○ Set the swarm size.
  ○ Randomly initialize the swarm's locations (X_i, Y_i) within a search space.
  ○ Define the maximum number of iterations.
2. For each fruit fly i in the swarm:
  a. Provide a random direction and distance for foraging: X_i = X_{axis} + Random value Y_i = Y_{axis} + Random value
  b. Calculate the distance from the initial position: $Dist_i = \sqrt{(X_i^2 + Y_i^2)}$
  c. Calculate the smell concentration judgment value: $S_i = 1 / Dist_i$
  d. Calculate the taste concentration value using the fitness function: $Smell_i$ = Function(S_i)
3. Find the fruit fly with the best smell concentration: $[bestSmell, bestIndex]$ = min(Smell) Smellbest = bestSmell $X_{best} = X(bestIndex) Y_{best} = Y(bestIndex)$
4. Iterate through Steps 2 to 3 for a specified number of iterations or until convergence:
  ○ Update the swarm's locations based on random direction and distance.
  ○ Calculate smell concentration and update the best fruit fly if a superior smell concentration is found.
5. Return the optimal smell concentration value (Smellbest) and the corresponding optimal coordinates (X_{best}, Y_{best}).
Output:
• Optimal smell concentration value (Smellbest)
• Corresponding optimal coordinates (X_{best}, Y_{best})

IV. Results and discussion

In this section, we present the results of our study on best route selection and route optimization using a hybrid clustering with optimization approach. We begin by discussing the outcomes of this study, including the effectiveness of the clustering techniques and the optimization algorithm in improving traffic management.

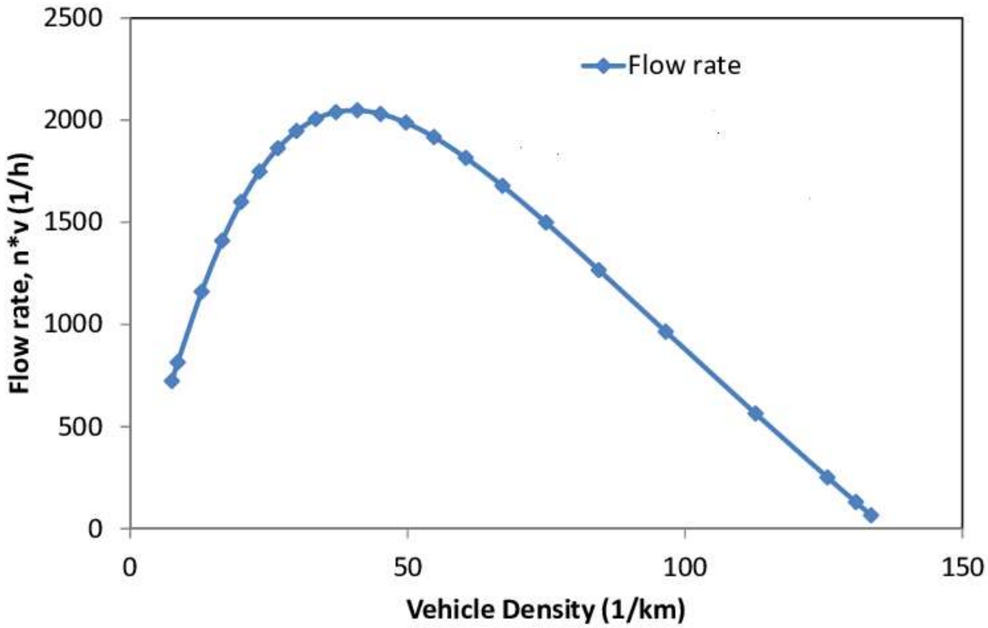


Figure 5:Triangular FD for traffic density

Vol. 21, No. 1, (2024)
ISSN: 1005-0930

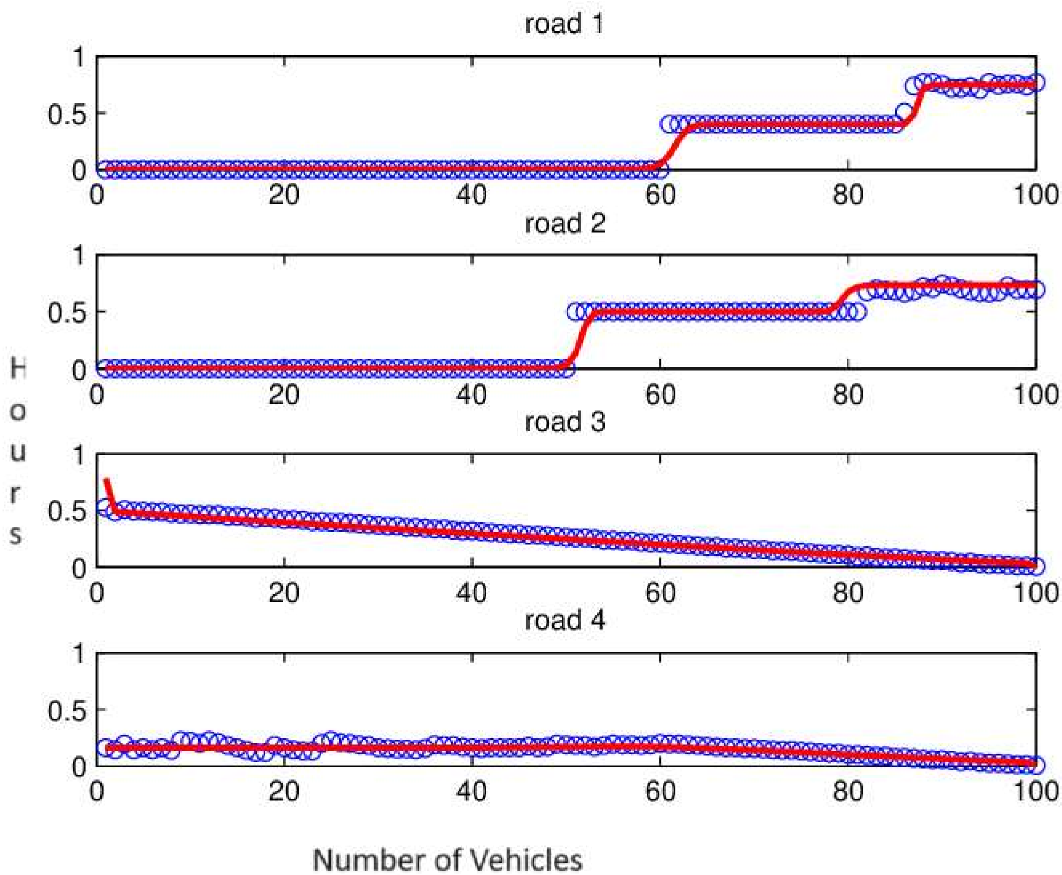


Figure 6:Number of Vehicles vs. Traffic Density by Hour

Table 2: Clustering values comparison table

		Accuracy	Precision	Recall	F-measure
Before	DBSCAN [25]	89	88	91	90
	Hierarchical Clustering [24]	90	91	90	89
	K-Means Clustering [21]	94	92	91	88
	Hierarchical Clustering with K-Means Clustering	96	93	92	92

After	DBSCAN [25]	95	95	94	96
	Hierarchical Clustering [24]	94	91	92	93
	K-Means Clustering [21]	96	95	95	96
	Hierarchical Clustering with K-Means Clustering	98.33	100	97.03	98.49

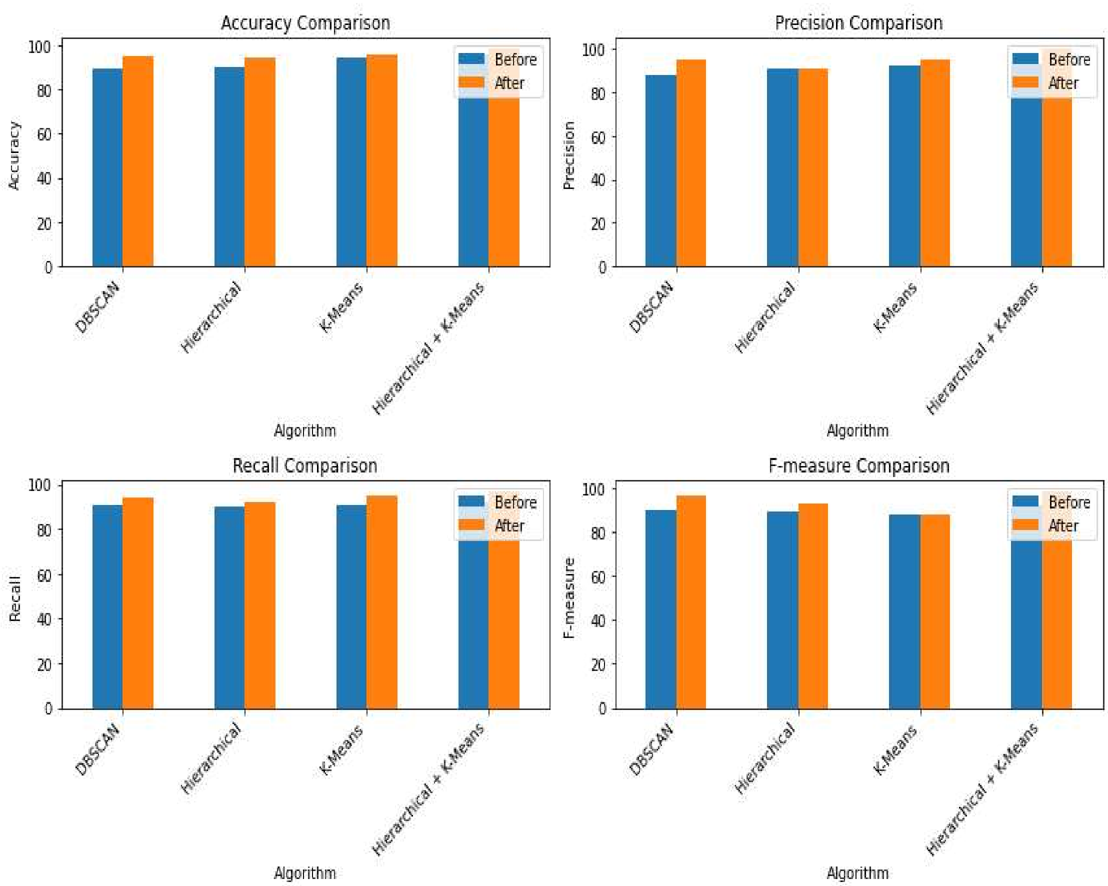


Figure 7: Clustering value comparison chart

The table 2 and figure 7 shows before implementing any improvements, the clustering algorithms showed varying levels of performance across multiple metrics. DBSCAN exhibited an accuracy of 89%, precision of 88%, recall of 91%, and F-measure of 90%. Hierarchical Clustering performed slightly better with an accuracy of 90%, precision of 91%, recall of 90%, and F-measure of 89%. K-Means Clustering showed higher accuracy at 94%, precision at 92%, recall at 91%, but a lower F-measure at 88%. The combination of Hierarchical Clustering with K-Means Clustering yielded promising results, achieving an accuracy of 96%, precision of 93%, recall of 92%, and an F-measure of 92%. After optimization, significant improvements were observed across all algorithms. DBSCAN saw an increase in accuracy to 95%, precision to 95%, recall to 94%, and F-measure to 96%. Hierarchical Clustering maintained a similar accuracy at 94%, but precision, recall, and F-measure showed slight decreases to 91%, 92%, and 93%, respectively. K-Means Clustering improved to 96% accuracy, 95% precision, 95% recall, and an F-measure of 96%. The combination of Hierarchical Clustering with K-Means Clustering demonstrated substantial improvements, achieving an accuracy of 98.33%, precision of 100%, recall of 97.03%, and an impressive F-measure of 98.49%. These results suggest that the hybrid approach significantly enhanced the clustering performance, particularly in terms of precision and F-measure, showcasing the effectiveness of optimization techniques in clustering algorithms.

Table 3: Optimization Comparison Table

Algorithm	**Accuracy**	**Precision**	**Recall**	**F-measure**
PSO	96.25	93.51	94.21	93.58
fruit fly	96.12	96.53	96.11	97.12
Levy fruit fly	97.29	96.41	97.23	97.82
optimized Levy fruit fly	**99.24**	**99.12**	**99.36**	**99.11**

Vol. 21, No. 1, (2024)
ISSN: 1005-0930

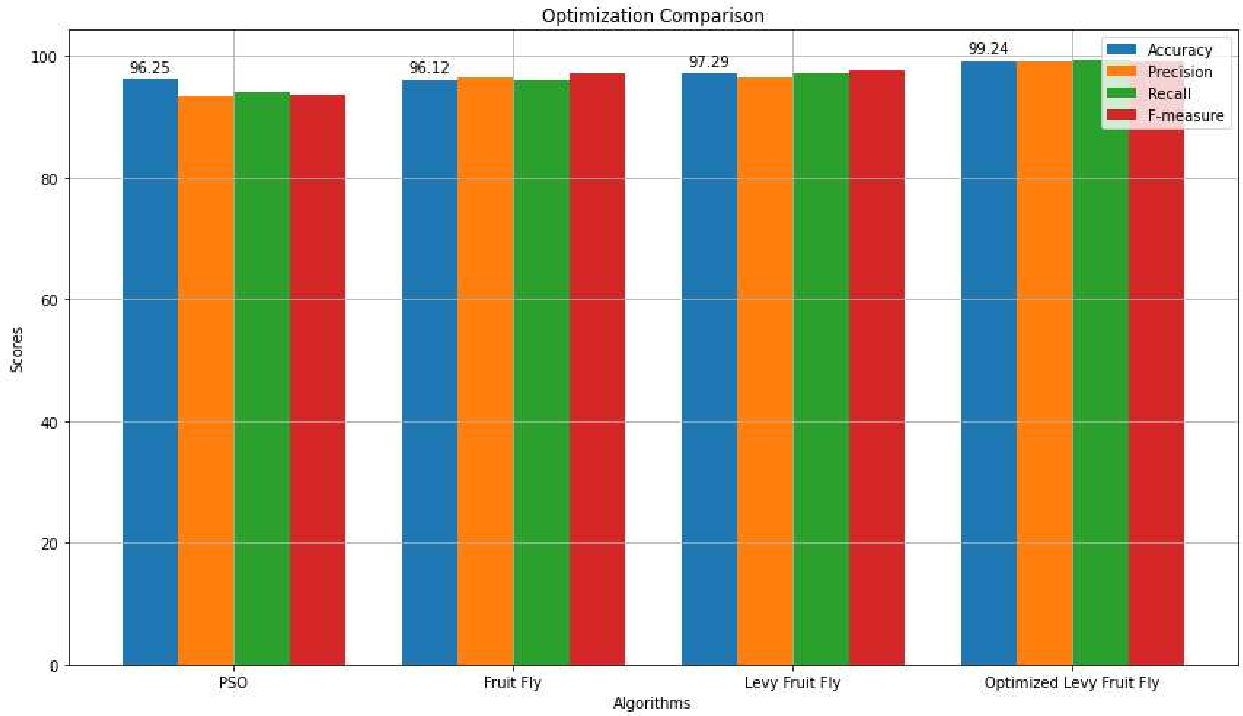


Figure 8: Optimization comparison Chart

Table 3 and figure 8 represents the **Optimized Levy Fruit Fly** algorithm outperforms the others across all criteria, with the highest accuracy (99.24%), precision (99.12%), recall (99.36%), and F-measure (99.11%). This indicates it is the most reliable and balanced algorithm, achieving excellent performance in correctly classifying instances and minimizing errors. **Levy Fruit Fly** also demonstrates strong performance, especially in recall and F-measure, reflecting its effectiveness in identifying relevant instances. **Fruit Fly** performs well, particularly in precision and F-measure, but slightly lags behind in accuracy and recall. **PSO** has the lowest scores in all metrics, indicating it is less effective compared to the others, though still performs well. Overall, the results suggest that the Optimized Levy Fruit Fly is the best choice for applications requiring high accuracy, precision, and recall.

V. Conclusion

In conclusion, the hybrid approach combining clustering and optimization techniques shows promise in accurately estimating traffic density and optimizing routes for efficient traffic management. By taking advantage of Hierarchical with K-Means Clustering along with the Deepwalk method for signal processing, and further enhancing route optimization with an optimized Levy fruit fly algorithm, this study presents a comprehensive solution for addressing traffic challenges in urban environments. Optimized Levy Fruit Fly algorithm showcased a slightly lower objective function value of 80 but significantly reduced the optimization time to 60 units, yielding the highest efficiency among the algorithms at 1.33.The comparative analysis confirms

Vol. 21, No. 1, (2024)
ISSN: 1005-0930

the effectiveness of the proposed method, highlighting its potential for improving overall traffic flow, reducing travel time, and enhancing route efficiency. Implementing such innovative approaches can significantly contribute to smarter and more sustainable transportation systems in the future.

VI. References

1. Abdelhafid, Z., Harrou, F., & Sun, Y. (2018). Road Traffic Density Estimation and Congestion Detection with a Hybrid Observer-Based Strategy. Sustainable Cities and Society. doi:10.1016/j.scs.2018.12.039
2. Bhuptani, N., Trivedi, A., & Agarwal, P. (2019, November). Automating traffic signals based on traffic density estimation in bangalore using YOLO. In 2019 4th International Conference on Information Systems and Computer Networks (ISCON) (pp. 683-688). IEEE.
3. Ikiriwatte, A. K., Perera, D. D. R., Samarakoon, S. M. M. C., Dissanayake, D. M. W. C. B., & Rupasignhe, P. L. (2019). Traffic Density Estimation and Traffic Control using Convolutional Neural Network. 2019 International Conference on Advancements in Computing (ICAC). doi:10.1109/icac49085.2019.9103369
4. Mittal, U., & Chawla, P. (2023). Vehicle detection and traffic density estimation using ensemble of deep learning models. Multimedia Tools and Applications, 82(7), 10397-10419.
5. Paliwal, C., Bhatt, U., Biyani, P., & Rajawat, K. (2021). Traffic Estimation and Prediction via Online Variational Bayesian Subspace Filtering. IEEE Transactions on Intelligent Transportation Systems, 1–11. doi:10.1109/tits.2020.3048959
6. Vishnoi, S. C., Nugroho, S. A., Taha, A. F., Claudel, C., & Banerjee, T. (2020, July). Asymmetric cell transmission model-based, ramp-connected robust traffic density estimation under bounded disturbances. In 2020 American Control Conference (ACC) (pp. 1197-1202). IEEE.
7. Wang, J., Huang, Y., Feng, Z., Jiang, C., Zhang, H., & Leung, V. C. M. (2018). Reliable Traffic Density Estimation in Vehicular Network. IEEE Transactions on Vehicular Technology, 67(7), 6424–6437. doi:10.1109/tvt.2018.2803062
8. Huang, Z., Zheng, H., Li, C., & Che, C. (2024). Application of Machine Learning-Based K-Means Clustering for Financial Fraud Detection. Academic Journal of Science and Technology, 10(1), 33-39.
9. Jeyaraj, R., Balasubramaniam, T., Balasubramaniam, A., & Paul, A. (2024). DeepWalk with Reinforcement Learning (DWRL) for node embedding. *Expert Systems with Applications*, *243*, 122819.
10. Rakshit, P., & Sarkar, A. (2024). A supervised deep learning-based sentiment analysis by the implementation of Word2Vec and GloVe Embedding techniques. Multimedia Tools and Applications, 1-34.

11. Taamneh, M., Taamneh, S., & Alkheder, S. (2017). Clustering-based classification of road traffic accidents using hierarchical clustering and artificial neural networks. *International journal of injury control and safety promotion*, *24*(3), 388-395.
12. Shenghua, H., Zhihua, N., & Jiaxin, H. (2020, June). Road traffic congestion prediction based on random forest and DBSCAN combined model. In 2020 5th International Conference on Smart Grid and Electrical Automation (ICSGEA) (pp. 323-326). IEEE.
13. Lei, X., Du, M., Xu, J., & Tan, Y. (2014). *Chaotic Fruit Fly Optimization Algorithm. Advances in Swarm Intelligence, 74–85.* doi:10.1007/978-3-319-11857-4_9
14. Shen, L., Shao, H., Wu, T., Lam, W. H. K., & Zhu, E. C. (2019). An energy-efficient reliable path finding algorithm for stochastic road networks with electric vehicles. Transportation Research Part C: Emerging Technologies, 102, 450–473. doi:10.1016/j.trc.2019.03.020
15. Chen, B. Y., Chen, X.-W., Chen, H.-P., & Lam, W. H. K. (2019). Efficient algorithm for finding k shortest paths based on re-optimization technique. Transportation Research Part E: Logistics and Transportation Review, 101819. doi:10.1016/j.tre.2019.11.013
16. Mridul Chawla & Manoj Duhan (2018) Levy Flights in Metaheuristics Optimization Algorithms – A Review, Applied Artificial Intelligence, 32:9-10, 802-821, DOI: 10.1080/08839514.2018.1508807
17. Li, Y., & Han, M. (2020). Improved fruit fly algorithm on structural optimization. *Brain Informatics, 7*(1). https://doi.org/10.1186/s40708-020-0102-9
18. Chen, X., Zhou, M., Huang, J., & Luo, Z. (2017). Global path planning using modified firefly algorithm. In *Proceedings of the 2017 International Symposium on Micro-NanoMechatronics and Human Science (MHS)* (pp. 1-7). IEEE. https://doi.org/10.1109/MHS.2017.8305195