UDC 303.732.4, 519.816
DOI: 10.20535/SRIT.2308-8893.2023.1.07

OPTIMIZATION OF ROUTE DISTANCE USING K-NN ALGORITHM FOR ON-DEMAND FOOD DELIVERY

PRADIP M. PAITHANE, SARITA JIBHAU WAGH, SANGEETA N. KAKARWAL

Abstract. Customers are now more able to purchase goods over the phone or the Internet, and the ability for those purchases to be delivered safely to the customer's location is proliferating. On-request meal delivery, where customers submit their food orders online, and riders deliver them, is growing in popularity. The cutting-edge urban food application necessitates incredibly efficient and adaptable continuous delivery administrations toward quick delivery with the shortest route. However, signing up enough food parcels and training them to use such food-seeking frameworks is challenging. This article describes a publicly supported web-based food delivery system. IoT (Internet of Things) and 3G, 4G, or 5G developments can attract public riders to act as publicly sponsored riders delivering meals using shared bikes or electric vehicles. The publicly funded riders are gradually distributed among several food suppliers for food delivery. This investigation promotes an online food ordering system and uses K-Nearest Neighbor calculations to address the Traveling Salesman Problem (TSP) in directing progress. The framework also uses the Global Positioning System (GPS) on Android-compatible mobile devices and the TOM-TOM Routing API to obtain coordinates for planning purposes. To evaluate the presentation of the proposed approach, recreated limited scope and certifiable enormous scope on-request food delivery occurrences are used. Compared to the conventional methodology, the proposed strategy reduces the delay time. Each rider will receive the most direct route to the order delivery address. The delivery delay time is reduced by approximately 10–15 minutes for every order. The food supplier can determine whether an item is available to the rider; thus, the food supplier can add an order to the rider having the shortest way.

Keywords: crowd-sourcing, hybrid optimization on-demand food-delivery, *k*-nearest neighbor algorithm (KNN), route optimization, traditional search (ts), urban logistic.

INTRODUCTION

The development of mobile Internet over the past few decades has made it possible to use smartphones for online ordering and delivery (like Domino's deliveries). By enhancing client happiness, a service provider can grow their customer base [1]. To make it better, you'll have a food ordering system that enables consumers to purchase products without physically going to the store, but also possibly using a phone or the internet, and after delivering them safely and in good condition to the specific customer's home. Users of the OCD service should place online orders for delivery of takeout food from crowd-sourced riders.

OCD framework enables food supply shops to accept food orders booked by customers through their computers or smartphones and then prepare personalized food and drinks[2]. With the help of on-demand couriers, customers of small catering businesses and food businesses can receive omnipresent scalable and affordable food service.

Deliveries make sure that the appropriate consumers receive the requested meal as quickly as feasible. The online-food in a sector where numerous small food suppliers often enter and exit. For these little food merchants, hiring a fleet of shipment boys would be quite expensive [3]. Additionally, because catering orders vary, it is challenging to modify the delivery team accordingly. However, some customers may be impatient and unable to wait many days for their food [4].

This suggested framework has encouraged a creative publically supported strategy framework to deliver parcels using already-existing taxicabs, hence reducing the task's time cost. This pioneering study demonstrated the viability of public funding in the last-mile transportation sector [5]. Food delivery is contingent on public support when requested. An online business known as "crowd-sourced delivery" for food is made up of riders, restaurants, customers, and publicly backed cargo[6]. Customers can order meals from catering shops via FoodNet. Cooking establishments accept customer orders to produce personalized food packages that are subsequently delivered by volunteers in place of the cafés' delivery personnel. The publicly supported riders register with the Online Crowd Sourced Delivery system, accept their assigned delivery commitments, visit the cafés, purchase the food packages, and deliver them to the relevant customers [7].

The food cloud attracts members of society at large to travel as delivery riders. Such crowded drivers can complete delivery tasks using shared bikes or electric motor bicycles using IoT and 3G, 4G, and 5G advancements[8]. In this way, the recently created online crowd-sourced delivery system can adapt to meet changing client requests [9]. The online crowd-sourced delivery method can reduce the cost of hiring delivery staff, in the opinion of the food suppliers[10]. The OCD method will aid vulnerable people, alleviate traffic congestion, and reduce emissions from fossil fuels. To reduce the absolute trip expense and delivery delay, a numerical model of the widely accepted delivery issue is developed [11]. The planning system makes use of Google Maps, an innovation of the Global Positioning System (GPS) [12]. By using one of the TSP solutions, heuristics calculation, the framework may speed up the delivery direction cycle. The matching of riders and suppliers is done via an expense-based coordination formula. Every rider's and supplier's respective moving costs are calculated. The problem is solved using a crossover meta-heuristic calculation that combines the Tabu Search and Adaptive Large Neighborhood Search methodologies for flexible large-area searches. Systems in Shenzhen that are both copied and truly demonstrate the validity, effectiveness, and distinctive advancement structure of the system that is being presented.

MOTIVATION

Today's unemployment is a growing issue, and this approach is suggested as a solution. The delivery wait time is longer than with more modern route optimization techniques. According to reports, it will lessen traffic problems during the delivery. To satisfy customer demands for prompt and accessible meal delivery [13]. A quick and comparatively simple way to get lunch delivered on demand

and on the same day. Since not every restaurant can afford to pay someone to deliver food, this approach will also solve that issue. A crowdsourced online delivery method that organizes between restaurants, clients, and crowdsourced riders can travel to the food supplier using this system while minimizing their overall travel costs and optimizing their route [10]. To distribute food delivery jobs and create high-quality delivery training in real-time. Overcome both the carbon emissions that result from transportation congestion. The on-demand food vendor is provided with a crowdsourced shipping method. Food delivery services that are quick to deliver food are essential, as demonstrated by the meals on call for the transport industry. Through an online dynamic optimization framework [14], the time options of the customers and crowd-sourced riders are addressed. The neighbor customer order details and the quickest route are managed using the *K*-Nearest Neighbor algorithm in this study. Following each food delivery, the rider and supplier can calculate the shortest route and closest location for the food order because they are exchanging information with one another.

METHODOLOGY

We conducted the research, and the authors followed a set of processes to determine the optimal path for the rider to deliver the food on time.

In each step, the activities are:

1. Observe the business process of different meal delivery services in the local area near the city. The proposed On-Demand Food Delivery via Online Crowdsourced in this study is based on these observed business processes.

2. Study of GPS technology to gain information, data, and expertise for the system's development. The developed application in this study makes use of Android handsets' GPS capability.

3. Examine the Tom-Tom API. This procedure seeks to investigate the characteristics and capabilities of Tom-Tom API for Maps. The research makes use of the internet and university library resources.

4. The author conducts analysis using the results of phases (1), (2), and (3) to create comprehensive knowledge for designing the system.

5. The author designs the system in the design phase utilizing the information gathered during the analysis phase. The blueprint for the created system is provided in this phase.

6. The application is developed utilizing the specified technology and tools during the implementation phase.

7. A testing phase is initiated to confirm the application's functionalities, during which the program is tested using many test cases

Food Ordering System via Online Crowdsourcing

Figure 1 depicts the architecture of the created online food ordering system. The main applications of the food ordering system are:

On-Demand Online Crowdsourcing application on the web. A rider has access to this application and can deliver orders after logging in. For the admin, this app is used to log in a rider who has placed an online purchase, input the order into the program, and create information about the outlet that will handle the request and the route that Delivery Staff should follow. The application is also used by the administrator to receive orders and prepare food. Customers can also use this web-based tool to check the status of their orders.

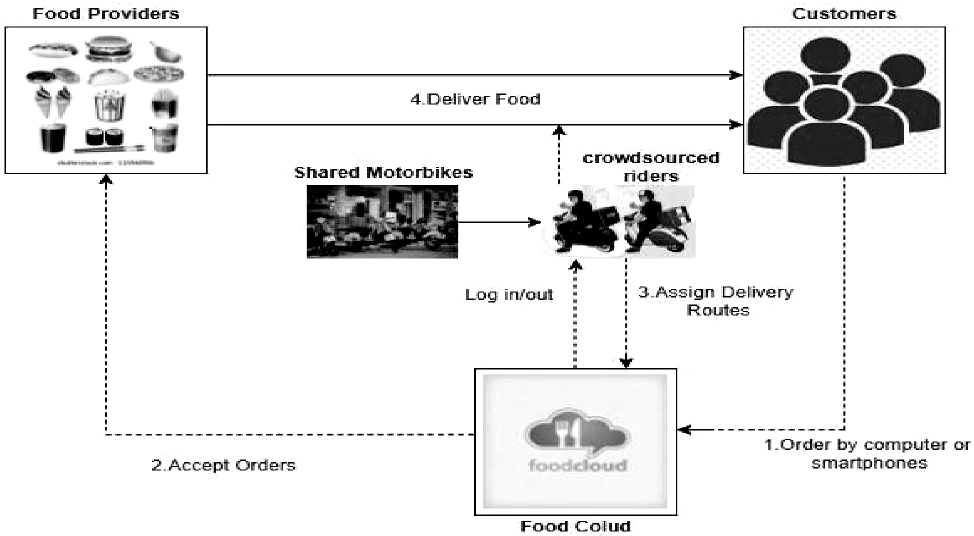

Fig. 1. The Crowdsourced on-demand Food Delivery System

K-Nearest Neighbor Algorithm for Routing Optimization

Steps:

Algorithm 1: K-Nearest Neighbor Algorithm

Input: *I*: The order set, *J*: The provider set, *K*: The rider set, $C_{a,b}$: Travel Cost, $T_{a,b}$: Travel Time, *i*: order, $a =$ placed by provider

Output: The initial routes *K*

1. sort_orders(I)
2. inserted node set $D = \text{empty}$
3. for all $i \in I$ do
4. $f_{\min} \leftarrow \infty, i_{best} = -1$
5. for all $a \in D$ Do
6. initialize route for $a : k \leftarrow (a)$
7. if $f_{\min} > insert\_\cos t(a,i)$ and $Q_k < q_k$ Then
8. $f_{\min} = insert\_\cos t(a,i)$
9. $i_{best} = a$
10. end if
11. insert_node (i_{best}, i)
12. $D \leftarrow D \cup i$
13. end for
14. end for

The below algorithm is used for demand match between rider and provider for a new shortest path for the next location address.

Algorithm 2: Rider Provider Matching

Input: J: The provider set, K: the rider set

Output: The initial routes $k(p)$

$C_{kj} \leftarrow calculate\_travel\_\cos t(K, J)$, for all K and all J

2. Store step 1 cost in a queue L,
3. Initialize rider status for route: $s_k =$ false
4. Initialize current total load of route: $q_j = 0$
5. for all $c_{kj} \in L$ do
6. if $s_k =$ false and $q_j + q_k < Q_j$, then apply K-Nearest Neighbor algorithm
7. $s_k =$ True, $q_j^+ = q_k$

Above equation is used for update shortest path with rider data.

8. update the rider's provider: $k_j = j$
9. end if
10. end for

The Q_j is total number of provider orders. The above algorithm is used for the determination of the shortest path for new orders as per the current status of the order and the location of a rider.

The Rider Provider Matching algorithm is working on the total calculated travel cost and a queue of riders available for the same route. In the first step, the initial route will be calculated as per the travelsellsman algorithm. In the second step, collect the status of all riders and the status of available food items in the basket. The new cost of distance will be calculated from the current status of riders. The proposed algorithm is used to get an updated route with the help of step number 7 and 10.

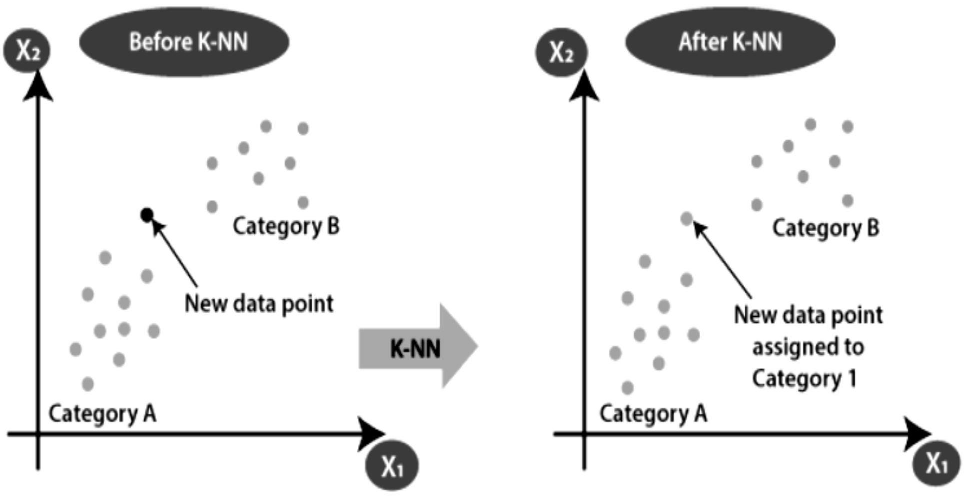

Fig. 2. K-Nearest Neighbor Graph

The output of this algorithm is the sequence of steps as the TSP solution. A simple web-based application was developed to test the heuristic algorithm for routing optimization. This testing application also utilizes TOM-TOM API for

Map [15]. In this application, the user can input several locations/addresses, and then Maps API will return the coordinate of those locations [16]. This function can be seen in Fig. 3, which shows an example of the optimized route for four addresses. In this application, point 1 is the initial and end point. The table with blue cells contains the data about the distance between two points that are generated using Google Maps API. Meanwhile, the table with red and yellow cells contains the solution for optimized routes [17].

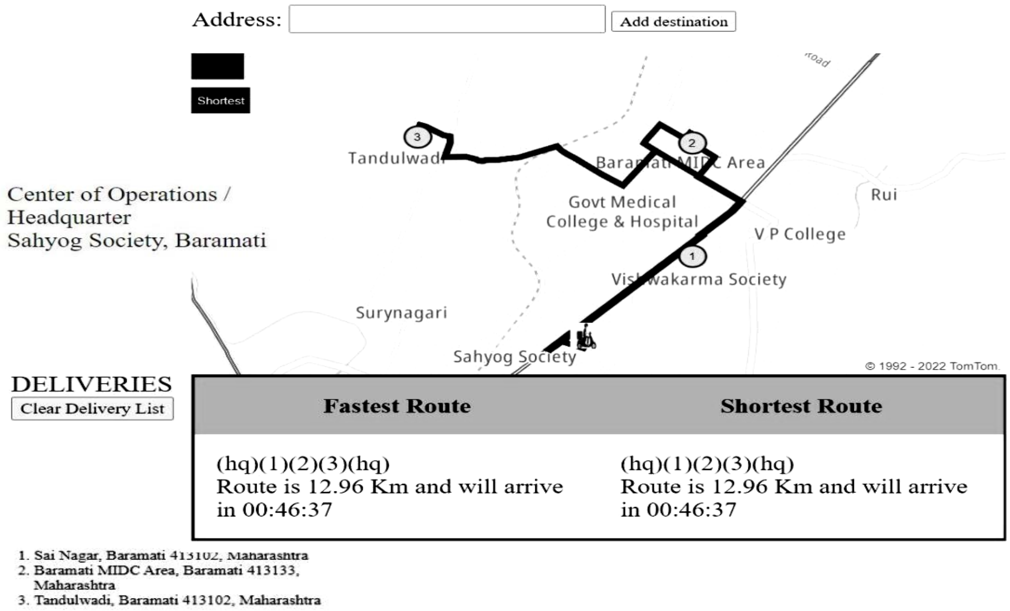

Fig. 3. Shortest Route for Food Delivery

As the Customer and Admin want to know the real-time location of the Rider for tracking purposes as shown in Fig. 4.

Fig. 4. Location of Rider

Mathematical Model

$$F = \beta_1 \sum_{k\in K} \sum_{a\in A} \sum_{b\in A} x_{ab}^k * c_{ab} + \beta_2 \sum_{a\in I} \max(0, T_a^k - t_a^e) . \quad (1)$$

The objective of OCD is the weighted travel cost and delivery delay, as shown in Eq. 1, where $\beta 1, \beta 2 \in [0,1]$ are weight parameters, a, b — source delivery and destination delivery location respectively, k — rider's index, T_a^k — arrival time of rider for delivery at node a , t_a^e — arrival time of rider from node a to provider location e, *e*- provider location, $C_{a,b}$ — weighted traffic cost from *a* to *b* delivery station

$$\sum_{a\in I\cup J} \sum_{k\in K} x_{ab}^k = 1 \quad \forall b \in I , \quad (2)$$

where I-the set of users, b-next destination location, x-travelling distance cost, J: The provider set, K: the rider set

$$\sum_{b\in I\cup J} \sum_{k\in K} x_{ab}^k = 1 \quad \forall a \in I , \quad (3)$$

The certain restrictions defined in Eq. 2 and Eq. 3 require that all accepted user orders must be provided and delivered.

$$\sum_{a\in I\cup J} x_{ab}^k = \sum_{a\in I\cup J} x_{ba}^k \quad \forall b \in I, \ \forall k \in K . \quad (4)$$

Eq. 4 requires a crowd-sourced rider to leave a user's location after delivering that user's delivery order.

$$T_a^k + t_{ab} = T_b^k \leftarrow x_{abk} = 1 \qquad \forall a \in I . \quad (5)$$

Eq. 5 states that the arrival time at user "*b*" is equal to the arrival time at the previous user "*a*" plus the travel time from node "*a*" to node "*b*", $t_{a,b}$. Note that the travel time $t_{a,b}$ can be updated using dynamic transportation analysis techniques, "x_{abk}" is travelling distance cost by *k* rider from "*a*" to "*b*".

$$Q_a^k \le q_k \leftarrow \sum_{b\in I} x_{ab}^k = 1 \quad \forall k \in K . \quad (6)$$

Eq. 6 ensures that the total amount delivered by crowd-sourced rider *R* does not exceed its capacity q_K.

$$x_{ab}^k \in \{0,1\} \quad \forall a, b \in I \cup J, \ \forall k \in K . \quad (7)$$

Eq. 7 defines the decision-variables.

In the above diagram, the *K*-Nearest Neighbor algorithm is used for the identification of the shortest path for the initial stage as well as the new order launch in the system. After improving the shortest path, the system is going to check the availability of riders for the next delivery order location and transfers or updated route share with an available rider. The *K*-Nearest Neighbor algorithm is focusing on neighbor values with the help of distance values. The closer neighbor distance value is performing a vital role in the identification of the shortest path route only [18].

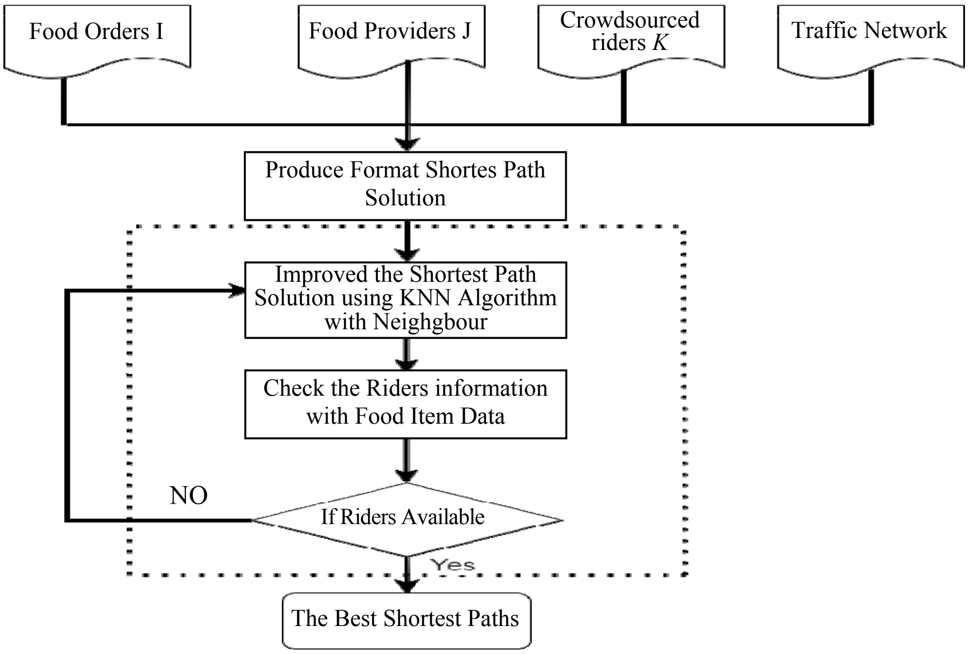

Fig. 5. Shortest Path Identification

EXPERIMENTAL RESULT

Result Analysis for Online Crowd-sourced Delivery vs. Traditional Approach

The proposed method is compared with a traditional method. The traditional method is used for food delivery with additional time for searching for the correct destination. In the OCD method, riders are getting updated information on their destination with location and delay time[19]. Based on the above information, Rider can transfer the same order to the nearest rider for delivery of the order so the waiting period is reduced.

The real-world OCD dataset consists of 10 instances from Baramati, Pune. The dataset covers the central business area as well citywide area of Baramati. Each instance consists of 435–1250 orders along with a set of providers and crowdsourced riders. The customers' preferred delivery times span the lunch period, from 11:00 AM to 2:00 PM, and another slot from 6.00 PM to 9.00 PM.

The travel speed is set to the average of the speeds of a bicycle and a motorcycle. The population of Baramati city is near about 200.000 and many people from Baramati MIDC are using food delivery applications for fast delivery. Heavy traffic and a limited number of a rider are major constraints to delivering food items at a scheduled time. The proposed system overcomes this drawback with updated delivery address information and updated rider details[20].

The presented dynamic crowdsourced food delivery framework was implemented in Python. The experiment was conducted on a personal computer equipped with an Intel(R) Core(TM) i7-4790 CPU @3.60 GHz and 16 GB of RAM. Two baseline algorithms were employed to solve the OCD instances for comparison with the presented hybrid solution method. The computing time for CPLEX was limited to 7200 seconds. The second algorithm used for comparison was the normal TS metaheuristic implemented in Python [21].

T a b l e 1 . Detail Comparison of Proposed Method with Traditional Method

Sr.No	Provider Location	User Location	Rider Number	Time (Min)	Distance by Proposed KNN Method(Km)	Distance by Proposed Tra-ditional Search(Km)	Waiting Time using Proposed Method(Min)	Waiting Time using Traditional Method (Min)
1	Saily Nagar	Rui Hospital	R1	8	4	4	2	6
2		Tandulwadi	R2	10	12	18	4	8
3		Muktai Lawns	R3	6	3	13	3	9
4		Reliance Mart	R1	12	20	25	5	12
5	Vivekanad Nagar	Vidyanagari	R1	11	18	22	7	20
6		Suryanagari	R2	5	3	8	2	4
7		Shriram Nagar	R2	12	5	5	4	22
8	Kasaba	Baramati Bus Stand	R1	13	3	3	5	8
9		Malegoan	R3	8	10	10	3	16
10		Desai Estate	R1	18	5	5	2	19

T a b l e 2 . Performance analysis of OCD with the tradition method according to distance and time parameter

Sr.No	Distance OCD	Distance Travel by Traditional Way	Extra Distance	Time OCD	Time Travel(Min) Traditional Way	Delay(min)
1	3.7	3.7	0	0+8	8	0
2	4	5+4	5	0+10	13+10	13
3	2.9	4.8+2.9	4.8	R1=8+6	9+6	9
4	6.2	1.2+6.2	1.2	R3=0+11	4+11	4
5	1.1	5.1+1.1	5.1	R2=10+5	12+5	12
Total	17.9	34	16.1	58	78	38

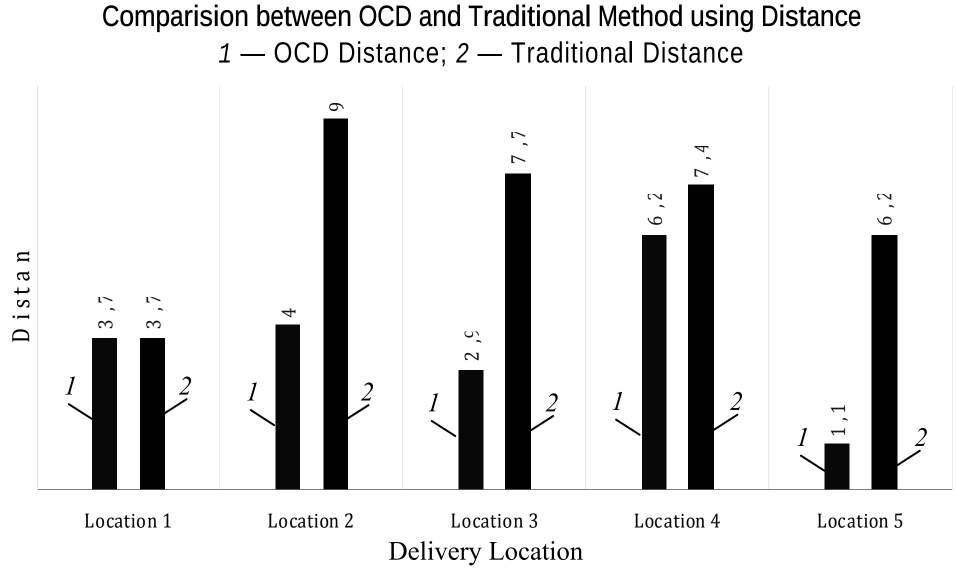

Fig. 6. Comparison between OCD and Traditional Method for Food delivery

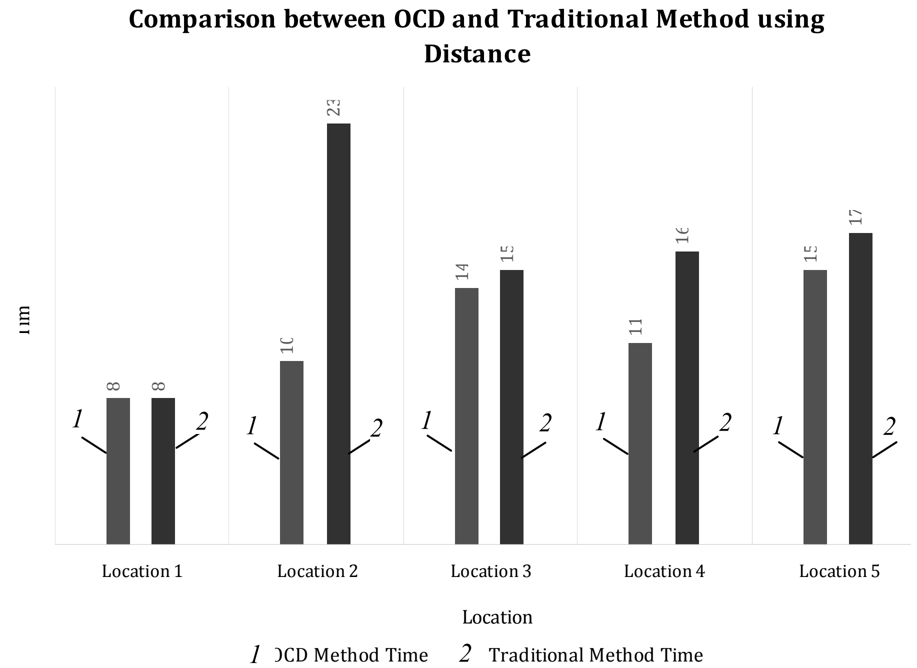

Fig. 7. Comparison between OCD and Traditional Method for Food delivery using Time Parameter

- A key feature of the proposed OCD method is that crowd-sourced riders are not tied to a single supplier, unlike the traditional approach, and can move between food suppliers. By limiting each passenger to a single vendor, the system simulated traditional food delivery logistics and compared the results.
- Crowd-Sourced food delivery goes beyond traditional logistics. As riders are spread across food vendors, which results in a reduction in median-total-travel-distance, and at the same time, the median total delivery delay is improved.

Above Table 3 depicted the detail comparison of traditional and K-Nearest Neighbor approach performance using time, distance and delay parameter.

Table 3. Performance Analysis of the Traditional Approach and KNN approach

Loca-tion	Rider No	Traditional Approach				KNN Approach			
		Obj	Time (Min)	Dist (Km)	Delay (Min)	Obj	Time (Min)	Dist (Km)	Delay (Min)
L1	R1	32.52	8	4	6	30.21	2	4	2
L2	R2	56.24	10	12	8	54.34	4	3	4
L3	R3	51.33	6	3	9	48.53	3	1	3
L4	R1	68.22	12	20	12	65.20	5	12	5
L5	R1	28.4	11	18	20	25.64	7	12	7
L6	R2	49.25	5	3	4	46.35	2	1	2
L7	R2	65.33	12	5	22	62.53	4	2	4
L8	R1	74.44	13	3	8	72.47	5	1	5
L9	R3	67.22	8	10	16	64.24	3	2	3
L10	R1	33.4	18	5	19	32.4	2	1	2
Average		52.64	10.3	8.3	12.4	50.20	3.7	3.9	3.7

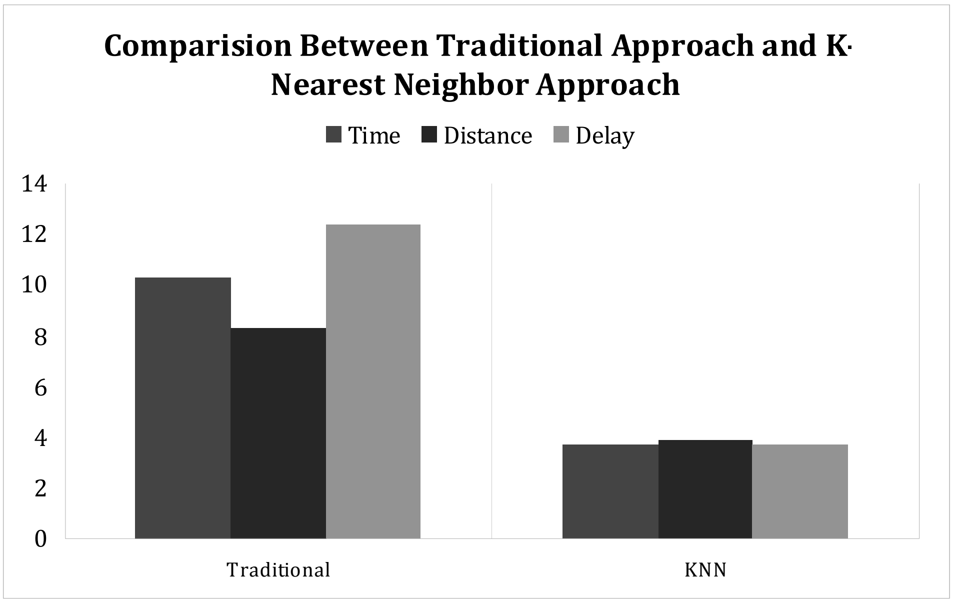

Fig. 8. Comparison between Traditional Approach and K-Nearest Neighbor Approachs

The K-Nearest Neighbor approach is reduced the time, delay and distance value for online food delivery process.

In above Fig. 9, the food delivery process is performed by the traditional method among all food suppliers. In this 3 food suppliers, 3 riders, and 6 customer locations are mentioned. The red line indicates the route path of rider 1 for provider A, the green line indicates the route of rider 2 for provider B and the blue line indicates the route path of rider 3 for provider C. Traditionally, the total 78 distance is covered by providers with 38 min minimum delay delivery time. The shortest path is not identified by Google Maps for the same process so an additional delay has been added to this method. Google map fails to detect the next order location with the shortest path because of the unavailability of food and other rider location data.

The proposed OCD system consists of information available for food item information, nearest neighbor rider information, and updated order with shortest path information. The proposed OCD method reduced the delivery delay time and identified the shortest path for the next order location.

From Fig. 9 and Fig. 10, we can claim that the KNN method is reduced the distance for food delivery location.

In Fig. 9, the traditional search algorithm has been used and the total distance covered by Rider is near about 56 km in figure 10, the K-Nearest Neighbor approach has been used and the total distance covered by the rider is 33 km only.

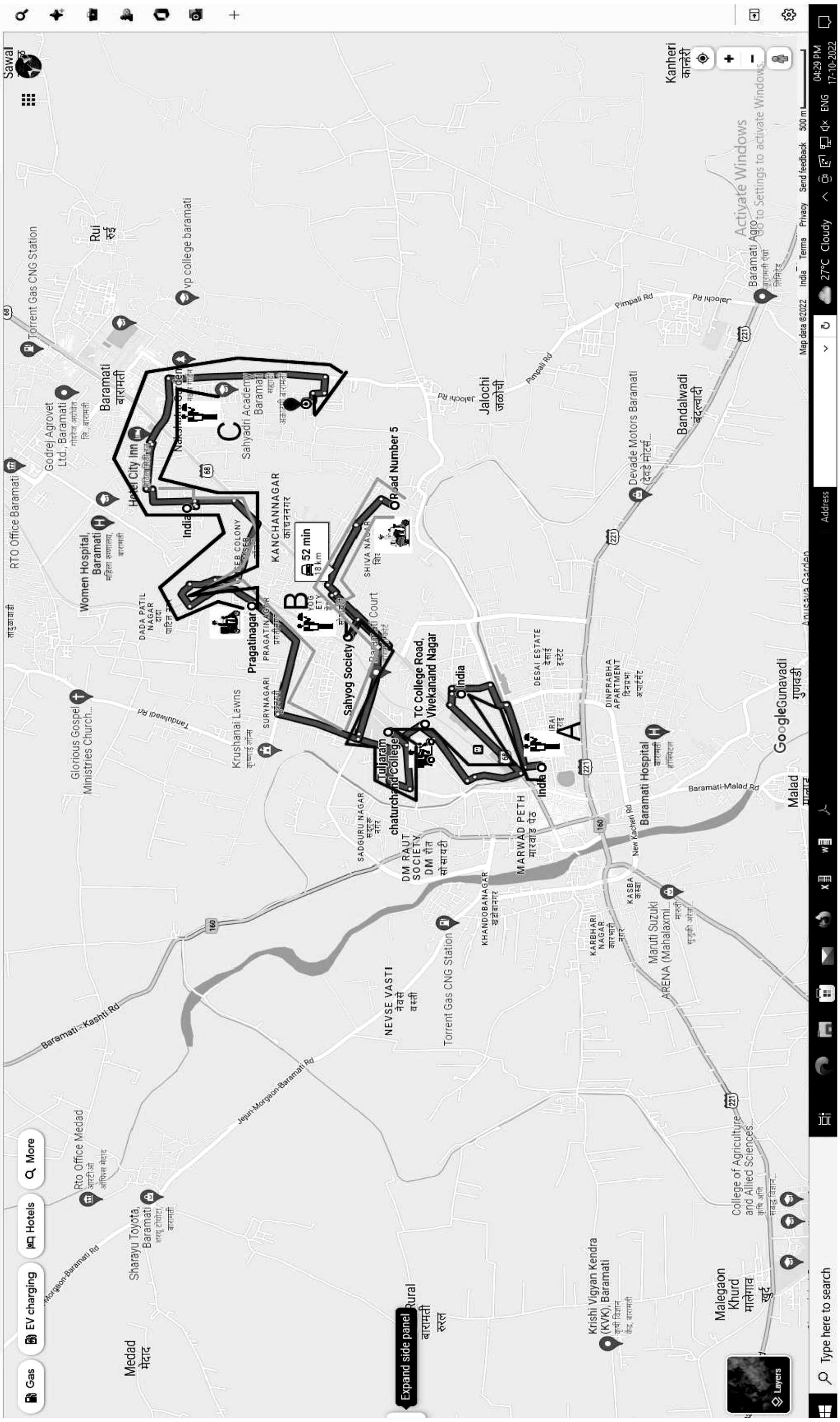

Fig. 9. Traditional Method for Food Delivery

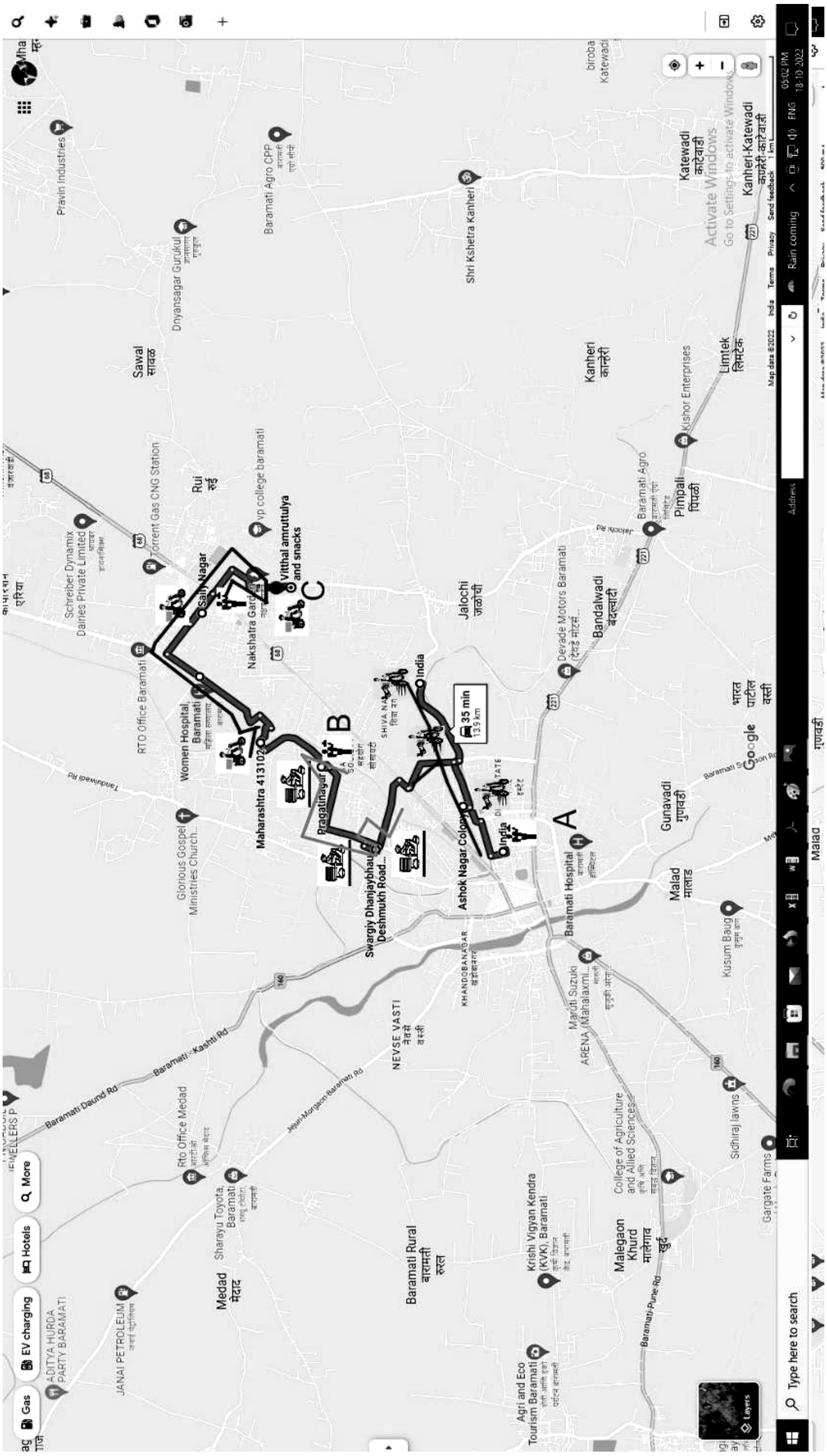

Fig. 10. K-Nearest Neighbor Shortest Path for Food Delivery Application

User Characteristics and Main Functions

The following are the major functions of the delivery system:

1. The registration of customers. This allows customers to save personal information for future orders. Customers must also register before they can place an order in the system.

2. The ability to delegate outlets. This function selects the most appropriate outlet to handle the order. The distance between the outlet and the customer's address is taken into account.

3. Create a delivery address routing that is optimum. The system generates optimized routing to provide detailed information about the sequence of delivery routes.

4. Management of orders. Order management allows the restaurant's workers to get order information and track the progress of each step.

5. Verification of order status. This feature allows customers to keep track of the status of their orders.

6. Order closure is a feature that allows you to close an order. This function is used to close an order by changing the order's status to "handled" and reporting the payment.

Aside from that, the application should be able to configure the following:

1. The outlet's address and exact coordinates.

2. An outlet's opening and closing times. This setting ensures that the order is handled by the only accessible outlet.

3. A delivery fee is a sum of money that the customer must pay to receive delivery service. This fee will be applied to the order bill automatically.

4. A food menu from which the customer can order.

5. The number of vehicles and delivery personnel available.

Web Service

A web service mediates data transmission between web-based applications. Users can obtain order information generated by the web-based application through the web service. The web services additionally provide a function that allows the Rider to update his current location.

The development tools for developing Online Crowd-Sourced Food Delivery can be seen in Table 4.

T a b l e 4 Development Tools

USER	ACTIVITIES
Platform	Microsoft Windows
Application Server	XAMP Server ,MySQL Server
IDE	Sublime Text
Technology	Web Service, CSS, Apache HTTP Server
Database	MySQL
Programming Languages	PHP, Java, Javascript, HTML
API	TOM TOM

After coding the application, the screenshot of the web-based food ordering system homepage can be seen in Fig. 11

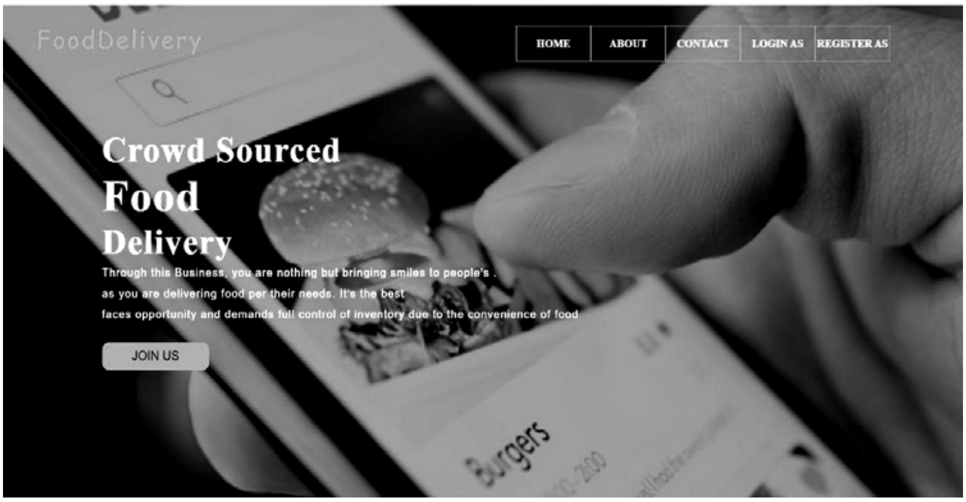

Fig. 11. Screenshot of the web-based food ordering system

CONCLUSION

This study suggested and created a web-based application for On-Demand Food Ordering using Online Crowdsourcing. In this research work, the utilization of riders with proper order delivery is performed. The traditional shortest method takes more time to deliver an order as compared to the OCD method. The OCD method minimizes the delay time and identifies the shortest path with the help of available Rider information. In future research, the author advises including more variables in the route optimization process, such as vehicle type, food package size, holiday season, Delivery Service's driver license type, and the maximum capacity of a vehicle type. These additional factors will increase the routing optimization's complexity.

Conflict of Interest

No conflict of Interest

REFERENCES

1. R.D.H. Tobing, "A food ordering system with delivery routing optimization using global positioning system (GPS) technology and google maps," *Internetworking Indonesia Journal*, 8(1), pp.17–21, 2016.
2. O.F. Aydin, I. Gokasar, and O. Kalan, "Matching algorithm for improving ride-sharing by incorporating route splits and social factors," *PloS one*, 15(3), p. e0229674, 2020.
3. D. Chen, Y. Yuan, W. Du, Y. Cheng, and G. Wang, "Online route planning over time-dependent road networks," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pp. 325–335.
4. Y. Liu et al., "FooDNet: Toward an optimized food delivery network based on spatial crowdsourcing," *IEEE Transactions on Mobile Computing*, 18(6), pp. 1288–1301, 2018.

5. W. Tu, T. Zhao, B. Zhou, J. Jiang, J. Xia and Q. Li, “OCD: Online crowdsourced delivery for on-demand food,” *IEEE Internet of Things Journal*, 7(8), pp. 6842–6854, 2019.
6. Pradip M. Paithane, S.N. Kakarwal, and D.V. Kurmude, “Top-down method used for pancreas segmentation,” *Int. J. Innov. Exploring Eng.(IJITEE)*, vol. 9, iss. 3, pp. 1790–1793, 2020.
7. J. Wang et al., “Multi-task allocation in mobile crowd sensing with individual task quality assurance,” *IEEE Transactions on Mobile Computing*, 17(9), pp. 2101–2113, 2018.
8. P. Li, J. Lai, and Y. Wu, “Accountable attribute-based authentication with fine-grained access control and its application to crowdsourcing,” *Front. Comput. Sci.*, **17**, 171802, 2023. Available: https://doi.org/10.1007/s11704-021-0593-4
9. C. Chen et al., “Crowddeliver: Planning city-wide package delivery paths leveraging the crowd of taxis,” *IEEE Transactions on Intelligent Transportation Systems*, 18(6), pp. 1478–1496, 2016.
10. A.M. Arslan, N. Agatz, L. Kroon, and R. Zuidwijk, “Crowdsourced delivery—a dynamic pickup and delivery problem with ad hoc drivers,” *Transportation Science*, 53(1), pp. 222–235, 2019.
11. Y. Ding et al., “A City-Wide Crowdsourcing Delivery System with Reinforcement Learning,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(3), pp. 1–22, 2021.
12. P.M. Paithane and S.N. Kakarwal, “Automatic determination number of cluster for multi kernel NMKFCM algorithm on image segmentation,” in *International Conference on Intelligent Systems Design and Applications. Springer, Cham*, *2018*, pp. 870–879.
13. P.M. Paithane, S.N. Kakarwal, and D.V. Kurmude, “Automatic Seeded Region Growing with Level Set Technique Used for Segmentation of Pancreas,” in *International Conference on Soft Computing and Pattern Recognition. Springer, Cham, 2020*, pp. 374–382.
14. P.M. Paithane and S.N. Kakarwal, “Automatic Pancreas Segmentation using A Novel Modified Semantic Deep Learning Bottom-Up Approach,” *International Journal of Intelligent Systems and Applications in Engineering*, 10(1), pp. 98–104, 2022.
15. Pradip Mukundrao Paithane, “Yoga Posture Detection Using Machine Learning,” *Artificial Intelligence in Information and Communication Technologies, Healthcare and Education: A Roadmap Ahead*, 2022: 27.
16. S.N. Kakarwal and P.M. Paithane, “Automatic pancreas segmentation using ResNet-18 deep learning approach,” *System Research and Information Technologies*, no. 2, pp.104–116, 2022.
17. N. Wang et al., “Collecting and analyzing key-value data under shuffled differential privacy,” *Front. Comput. Sci.*, **17**, 172606, 2023. Available: https://doi.org/10.1007/s11704-022-1572-0
18. A. Maatoug, G. Belalem, and S. Mahmoudi, “A location-based fog computing optimization of energy management in smart buildings: DEVS modeling and design of connected objects,” *Front. Comput. Sci.*, **17**, 172501, 2023. Available: https://doi.org/10.1007/s11704-021-0375-z
19. A. Biswas et al., “OCSO-CA: opposition based competitive swarm optimizer in energy efficient IoT clustering,” *Front. Comput. Sci.*, **16**, 161501, 2022. Available: https://doi.org/10.1007/s11704-021-0163-9
20. C. Zhu, B. Han, and Y. Zhao, “A bi-metric autoscaling approach for *n*-tier web applications on kubernetes,” *Front. Comput. Sci.*, **16**, 163101, 2022. Available: https://doi.org/10.1007/s11704-021-0118-1

Received 14.11.2022

INFORMATION ON THE ARTICLE

Dr. Pradip M. Paithane, ORCID: 0000-0002-4473-7544, Vidya Pratishthan's Kamalnayan Bajaj Institute of Engineering and Technology, India, e-mail: paithanepradip@gmail.com

Sarita Jibhau Wagh, ORCID: 0000-0003-4798-2147, T.C. College Baramati, Maharashtra, India

Dr. Sangeeta N. Kakarwal, ORCID: 0000-0003-4828-5247, ICEEM College, India

ОПТИМІЗАЦІЯ МАРШРУТНОЇ ДИСТАНЦІЇ З ВИКОРИСТАННЯМ АЛГОРИТМУ *K*-NN ДЛЯ ДОСТАВКИ ЇЖІ НА ВИМОГУ / Прадіп М. Пайтане, Саріта Джібхау Ваг, Сангіта Н. Какарвал

Анотація. Сьогодні можливість клієнтів купувати товари в Інтернеті чи по телефону і безпечно транспортувати до місця розташування клієнта швидко зростає. Стає поширеною послуга доставки їжі за запитом, коли клієнти розміщують запити на їжу в Інтернеті, а пасажири передають ці замовлення. Нове застосування столичних харчових продуктів вимагає продуктивних і універсальних безперервних адміністрацій транспортування для швидкої доставки найкоротшим шляхом. Складно зареєструвати достатню кількість пакетів їжі та навчити їх працювати з такими структурами запиту їжі. У цій роботі подано загальнодоступний веб-підхід до транспортування їжі за запитом. У співпраці з IOT (Інтернет речей) і досягненнями 3G, 4G або 5G громадські гонщики можуть бути залучені до подорожей як громадські гонщики, які перевозять їжу за допомогою спільних велосипедів або електромобілів. Підтримувані громадськістю райдери поступово розподіляються між різними постачальниками їжі для її транспортування. У дослідженні створено систему запитів на їжу в Інтернеті та застосовано обчислення KNN для вирішення проблеми комівояжера (TSP). Платформа додатково використовує технологію глобальної системи позиціонування (GPS) у мобільних пристроях, що підтримують Android, і використовує API маршрутизації TOM-TOM для координат для планування розташування. Для оцінювання запропонованого підходу використовуються відтворені події обмеженого обсягу та сертифікованого великого обсягу за запитом. Такий підхід зменшує час затримки доставки (до 10–15 хв). Кожен пасажир отримає оновлене місце призначення доставки замовлення найкоротшим маршрутом. Постачальник продуктів харчування може отримати статус харчового продукту, доступного на райдері.

Ключові слова: краудсорсинг, гібридна оптимізація, доставка їжі на вимогу, алгоритм *k*-найближчих сусідів, оптимізація маршруту, традиційний пошук, міська логістика.