International Journal of Internet, Broadcasting and Communication Vol.14 No.2 109-118 (2022)
http://dx.doi.org/10.7236/IJIBC.2022.14.2.109

IJIBC 22-4-15

Detecting Anomalous Trajectories of Workers using Density Method

Doi Thi Lan and Seokhoon Yoon*

Ph.D. Student, Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Korea
Professor, Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Korea
doilan151188@gmail.com, seokhoonyoon@ulsan.ac.kr

Abstract

Workers' anomalous trajectories allow us to detect emergency situations in the workplace, such as accidents of workers, security threats, and fire. In this work, we develop a scheme to detect abnormal trajectories of workers using the edit distance on real sequence (EDR) and density method. Our anomaly detection scheme consists of two phases: offline phase and online phase. In the offline phase, we design a method to determine the algorithm parameters: distance threshold and density threshold using accumulated trajectories. In the online phase, an input trajectory is detected as normal or abnormal. To achieve this objective, neighbor density of the input trajectory is calculated using the distance threshold. Then, the input trajectory is marked as an anomaly if its density is less than the density threshold. We also evaluate performance of the proposed scheme based on the MIT Badge dataset in this work. The experimental results show that over 80 % of anomalous trajectories are detected with a precision of about 70 %, and F1-score achieves 74.68 %.

Keywords: *Anomalous trajectory detection of worker, density method, EDR, distance threshold, density threshold.*

1. Introduction

A data pattern is known as an anomaly if it deviates from the remaining normal patterns. Anomaly detection aims to seek the previously unobserved patterns, such as detecting malicious activity, e.g., credit card fraud, cyber-intrusion, terrorist activity [1]. In the working environments, anomalies can occur in a variety of ways due to the complex nature of human behavior. A worker may, for example, be involved in an accident or visit a prohibited area. The safety and security of the working areas will be significantly improved if we can correctly detect anomalous behaviors of workers. Since workers' movement trajectories provide rich spatiotemporal information about their activities and behaviors, abnormal behaviors of workers can be found through detecting their abnormal trajectories [2-3].

There are several existing trajectory outlier detection algorithms in various industrial domains and applications, including maritime, smart transportation, aircraft security [4]. A trajectory outlier detection

Manuscript Received: March. 17, 2022 / Revised: March. 19, 2022 / Accepted: March. 24, 2022
Corresponding Author: seokhoonyoon@ulsan.ac.kr
Tel: +82-52-259-1403, Fax: +82-52-259-1687
Professor, Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Korea

algorithm based on distance function and density method is introduced in [5-6]. The authors proposed different distance measurements that are applied to calculate the similarity between two trajectories. In their algorithms, two parameters: distance threshold and neighbor density threshold are used to detect trajectory anomalies. However, they have not provided a method for choosing the parameter values that should be determined based on the dataset's distribution characteristics. Therefore, in our work, a method for determining the parameters based on trajectory data is proposed.

Besides, the study in [7] suggested an isolation-based anomalous trajectory (*iBAT*) detection method. This work aims to find abnormal driving patterns from taxi's GPS traces, with applications such as automatically detecting fraud of drivers or new roads in cities. Clustering method is also applied for detecting abnormal trajectories in [8-9]. In these works, the authors detected anomalous trajectories based on distance functions and the hierarchical clustering method. However, the approaches in [5-9] mainly detected anomalous trajectories of objects moving in outdoor space. They have paid no attention to detecting human's anomalous trajectories in indoor space, which is crucial for workplace safety. As a result, the focus of this paper is on identifying workers' abnormal trajectory patterns in the workplace.

In our work, trajectory anomaly detection is based on density method. This method is unsupervised in the sense that it does not require labelled patterns in training dataset. In some cases, labeling data patterns is difficult and requires the expertise of experts in the fields. Besides, density-based anomaly detection techniques are quite straightforward. However, these techniques require a distance function or a similarity measurement between two data instances. A lot of distance functions are used to determine the level of similarity between two trajectories [10]. Euclidean distance measures the total distance between all pairs of corresponding points between two trajectories. However, Euclidean distance requires the same length of two considering trajectories, and it is sensitive to noises or equipment recording errors. Several other distance measurements have been proposed to improve Euclidean distance. For instance, longest common subsequence (LCSS) can find the similar common subsequences between the two trajectories of different lengths. This measurement is also robust to noise by giving the space and time thresholds to find similar points of two trajectories. However, a disadvantage of LCSS is that it ignores unmatched points between the two trajectories [11]. Meanwhile, DTW (Dynamic time warping) has been successfully applied to time series data and trajectories. DTW is based on defining a cost for aligning two data points, and it finds the minimum cost to align all points between two trajectories. However, this measurement is sensitive to noises because the aligning cost is based on real distance between two points [12]. In our work, edit distance on real sequence (EDR) is used to calculate the distance between two trajectories. EDR can remove noise effects by quantizing the distance between a pair of elements to two values, 0 and 1. This measurement does not require the same lengths of two considering trajectories by seeking the minimum number of edit operations to change one trajectory to another [13].

Our trajectory anomaly detection framework consists of two phases: offline phase and online phase. The offline phase aims to determine the values for parameters, including distance and density thresholds. The distance threshold is used to determine whether two trajectories are neighbors or not. We propose a novel method to determine the distance threshold based on mean value and standard deviation value of pairwise distances between trajectories. Meanwhile, the density threshold is also chosen based on mean and standard deviation values of the trajectories' neighbor densities. The density threshold determines whether a trajectory is abnormal or normal.

The main contributions in this work can be summarized as follows:

- ✓ We design an anomalous trajectory detection framework based on EDR and density method.

- ✓ A novel method for determining the algorithm's parameters is proposed. We use the mean and standard deviation values of trajectories' pairwise distances and densities to determine distance and density thresholds, respectively.
- ✓ We validate the proposed method using a real trajectory dataset collected from 36 workers for one month. It achieves detection performance with recall of 80.87% and precision of 70.09%.

The rest of this paper is organized as follows. First, Section 2 introduces about the definitions. Then, the methods are presented in Section 3. The performance evaluation is discussed in Section 4. Finally, Section 5 consists of the conclusion of the paper.

2. Definitions

First, three obvious definitions related to trajectory are provided. Then, EDR is introduced in detail that is used in this work.

2.1. Definitions Related to Trajectory

Definition 1: A trajectory T is a sequence of multi-dimensional points and denoted by $T = \{p_1, p_2, \dots, p_i \dots p_n\}$. In our work, p_i is a location point in 2-dimensional space with latitude $p_{i,x}$ and longitude $p_{i,y}$. n is the length of trajectory T and can differ from the lengths of other trajectories.

Definition 2: Two trajectories *S* and *R* are neighbors to each other if $Dist_{S,R} \leq Thres_{Dist}$. Here $Dist_{S,R}$ is the distance between *S* and *R*. $Thres_{Dist}$ denotes the distance threshold used to determine whether two trajectories are neighbors or not.

Definition 3: A trajectory *S* is an anomaly if and only if $Den_S \leq Thres_{Den}$, where Den_S is the neighbor number of *S* and $Thres_{Den}$ is a given density threshold.

2.2. Definition of EDR

EDR is used to measure the difference between two trajectories using three edit operations: insert, delete, and replace [13]. EDR reduces noise by assigning the cost of the edit operations between two points to one of two values, 0 or 1. Due to finding the minimum number of edit operations to replace one trajectory to other, EDR can be applied to two trajectories of varying lengths.

Given two trajectories $S = \{s_1, s_2, s_3 \dots s_n\}$ and $R = \{r_1, r_2, r_3 \dots r_m\}$ of lengths n and m, respectively. EDR between *S* and *R* is defined as follows:

$$EDR(S,R) = \begin{cases} n & if\ m = 0 \\ m & if\ \ n = 0 \\ \min\{EDR(Rest(R), Rest(S)) + \text{Cost}[\text{replace}(s_n, r_m)], \\ EDR(Rest(S), R) + \text{Cost}[delete(s_n)], EDR(S, Rest(R)) + \text{Cost}[insert(r_m)]\} & otherwise \end{cases} \quad (1)$$

where Cost $[delete(s_n)] =$ Cost $[insert(r_m)]) = 1$. $Cost[replace(s_n, r_m)] = 0$ if and only if $|s_{n,x} - r_{m,x}| \leq \alpha$ and $|s_{n,y} - r_{m,y}| \leq \alpha$ else $Cost[replace(s_n, r_m)] = 1$. α is a predefined matching threshold. The cost of insert or delete operations is always 1. Meanwhile, the cost of replace operation between s_n and r_m is 0 if two points matches. In contrast, the cost of replace operation is 1.

3. Methods

First, an anomalous trajectory detection framework is introduced. Then, a new method is proposed to determine distance and density thresholds.

3.1 Anomalous Trajectory Detection Framework

In this work, we want to design an abnormal detection approach in real time. This means that the trajectories of workers are checked in real time. The abnormal trajectory detection framework is divided into two phases: offline and online as shown in Figure 1.

In during the offline phase, the parameters $Thres_{Dist}$ and $Thres_{Den}$ are determined based on collected trajectories in dataset. The dataset contains both normal and abnormal unlabeled trajectories. Before proceeding, these raw trajectories must be preprocessed. Since the objective is to detect the workers' abnormal behaviors as soon as possible, their trajectories are checked within short periods of time, which is known as timeslots. Hence, during the data preprocessing step, we divide the original trajectory into sub-trajectories based on timeslots. From now on, each worker's location trace within a single timeslot will be referred to as a trajectory.

In the online phase, each input trajectory is checked. First, we use EDR to calculate the distances between the input trajectory and all existing trajectories in the dataset. Then, the neighbor density of this trajectory is determined based on $Thres_{Dist}$. Finally, anomaly detector shows that the input trajectory is abnormal if its density is not higher than $Thres_{Den}$.

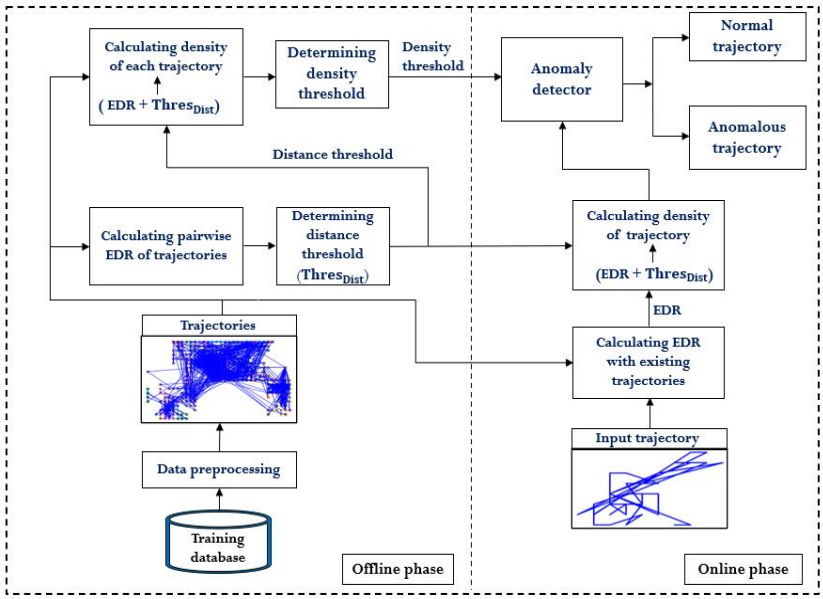


Figure 1. Anomalous Trajectory Detection Framework

3.2 Determining Distance Threshold

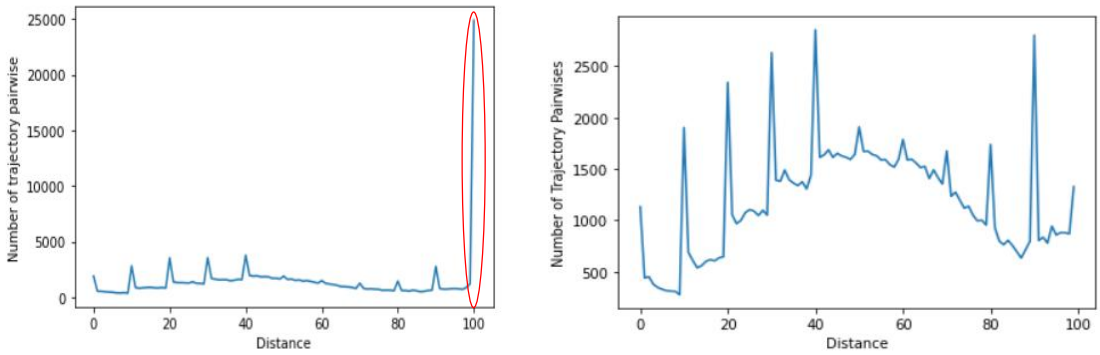
The distance threshold $Thres_{Dist}$ is used to determine whether two trajectories are neighbor or not. Two trajectories are neighbors if their distance is no larger than $Thres_{Dist}$. This parameter has a direct impact on the algorithm performance. If $Thres_{Dist}$ is small, many trajectories are detected as anomalies, and the false alarm rate is high. In contrast, if $Thres_{Dist}$ is large, there are very few anomalous trajectories found. Hence, choosing an appropriate value of $Thres_{Dist}$ is a crucial task in the density-based anomaly detection method.

From the dataset, the pairwise distances between trajectories are calculated. Then $Thres_{Dist}$ is chosen based on the mean value μ_{Dist} and the standard deviation σ_{Dist} of the pairwise distances. Since the distance

threshold is used to determine whether two trajectories are neighbors or not, only trajectories related to each other are considered. From this angle, the distance value that does not represent the relationship between trajectories is removed before calculating $Thres_{Dist}$. As shown in Figure 2(a), there is a peak at the value of 100. This is the maximum EDR value of two trajectories in the dataset, and it represents those two completely different trajectories. In the preprocessing step, we remove this peak, and receive the result as shown in Figure 2(b). Before choosing the distance threshold, the trajectories' pairwise distances at all timeslots are collected. Therefore, all timeslots have the same distance threshold $Thresh_{Dist}$. After calculating the mean value and the standard deviation of pairwise distances, we obtain μ_{Dist} and σ_{Dist}. The distance threshold is defined as follows:

$$Thresh_{Dist} = \mu_{Dist} + \beta_1 * \sigma_{Dist} \quad (2)$$

where β_1 is a predefined parameter by user.



a) Before preprocessing **b) After preprocessing**

Figure 2. Distribution of pairwise distances

3.2 Determining Density Threshold

The density threshold is used to determine whether a trajectory is detected as normal or abnormal. If density of a trajectory is no larger than $Thres_{Den}$, it is marked as an anomaly. The density threshold must also be chosen before using to detect anomaly.

The neighbor density of workers depends on the working hour. For example, the number of workers in during the first and last hours of a day often is less than during other hours. If we only use a $Thres_{Den}$ for all hours of the day, the method's performance may be affected. From this viewpoint, the density threshold is determined in accordance with timeslots. This means that each timeslot will receive an appropriate density threshold. The density threshold at timeslot T is denoted as $Thres_{Den}^{T}$.

To determine $Thres_{Den}^{T}$, we calculate the density of each trajectory in the dataset at timeslot T. The value of $Thres_{Den}^{T}$ is also selected based on the mean value μ_{Den}^{T} and the standard deviation value σ_{Den}^{T} of trajectories' densities. $Thres_{Den}^{T}$ is defined as follows:

$$Thres_{Den}^{T} = \mu_{Den}^{T} + \beta_2 * \sigma_{Den}^{T} \quad (3)$$

where β_2 is a predefined parameter by user. Since the anomalous trajectory has little in common with the other trajectories, the density threshold $Thres_{Den}^{T}$ for detecting anomaly is smaller than the mean value μ_{Den}^{T} of trajectories' densities $(\beta_2 < 0)$.

4. Performance Evaluation

In this work, we use a real dataset (MIT Badge dataset) to evaluate the performance of algorithm. First, the introduction of dataset is presented. Next, we present the setup of experiment, and then the results are presented.

4.1 Dataset

4.1.1 Data Description. MIT Badge dataset contains time-stamped geographic locations for workers at an IT Call Center in Chicago over a one-month period. Each employee is provided a unique badge that allows radio signals to be used to locate them. The in-house positioning system works by measuring the radio signal strength (RSSI) of each employee's badge at various base stations. These base stations, which are stationed throughout the office, are used to determine the instantaneous position of a badge [14].

Location data is sampled at the rate of 10 measurements/minute. There are 3 groups with 36 workers participated in the data collection. Table 1 summarizes the number of department groups.

Table 1. The number of workers for each type in subareas

Group	User ID	Number of users
Configuration	294, 265, 253, 292, 99, 104, 251, 107, 298, 256, 56, 109, 281, 82, 267, 273, 290, 280, 258, 101, 278, 264, 105, 276, 106	25
Pricing	263, 266, 268, 272, 288, 293, 297	7
Coordinator	285, 257, 261, 291	4

Each worker log file contains a few fields, which are listed in Table 2.

Table 2. Some samples of worker log file with user ID 257

X - Coordinate	Y - Coordinate	Date as a string	Time as a string
2800	2000	2007-03-28	09:38:00
2800	1800	2007-03-28	09:38:00
2600	2200	2007-03-28	09:38:00

4.1.2 Data Preparation. The purpose of this work is to detect abnormal trajectories of workers when comparing them with all remaining trajectories at the same time. Since data of each worker are collected at the different times of day, it is difficult for detecting anomalies using the raw data. Hence, we need to preprocess data.

To do this, we consider a working day is from 9:00 am to 6:00 pm, with each day divided into 54 timeslots. Each timeslot is 10 minutes. The timestamp of each data sample will then be assigned to one of the pre-defined 54 timeslots. Some data samples are assigned with timestamps as Table 3.

Table 3. Some preprocessed samples of log file of worker with ID 257

X - Coordinate	Y - Coordinate	Date as a string	Time as a string	Timestamp
2800	2000	2007-03-28	09:38:00	4
2800	1800	2007-03-28	09:38:00	4
2600	2200	2007-03-28	09:38:00	4

4.2 Experiment Setup

To evaluate the method, we need the labelled anomalies in the test dataset. However, since the anomalies are not labelled in the MIT Badge dataset, some assumptions are made for evaluation. As described in the section 4.1.1, all employees in the dataset are divided into three groups. The employee behavior in different groups is not similarity. Therefore, a hypothesis is given: if one group occurs more frequently than the others, this group can be considered as normal, while others are abnormal.

In the MIT Badge dataset, the worker number of Configuration group is 25 users while the Pricing group is only 7 users. In the case, the Configuration group accounts for approximately 78 % of the total, while the Pricing group accounts for only 22 %. As a result, we can assume that employees in the Pricing group are anomalies, while employees in the Configuration group are normal.

As described in the section 4.1, there are 10 location points collected within one minute for each worker, and each timeslot contains 10 minutes. This means that, in each timeslot, the maximum length of a trajectory is 100. Because data may be missing, this value may be less than 100.

The original dataset is divided into two parts. The first part accounts for 70% of the total and is used to determine two parameters: distance threshold and density threshold. The second part is 30% and is used to estimate the algorithm. The second part is called as the test dataset. Since the data is collected within 17 days, the test dataset is chosen randomly with 5 days. The original dataset contains both the normal and abnormal labeled trajectories. However, as described in Section 3.1, in the step of determining the parameters, the trajectories are unlabeled. Therefore, in this step, we use the trajectories without knowing their labels. Only in the test dataset, trajectories are labeled to evaluate the algorithm's performance. Furthermore, the test dataset should contain the same number of abnormal and normal trajectories. This ensures that the assessment is correct.

The test days should be randomly chosen from the list of data collection days. The algorithm will be implemented again with different sets of test days. Then the output of the algorithm is the average of all run times. In our work, we validate the algorithm over 5 times.

4.3 Results

To estimate the algorithm performance, we use three metrics: recall, precision, and F1-score. The recall is denoted as the equation (4). This value represents the algorithm's ability to detect anomalies. Meanwhile, the precision represents the algorithm's correction and is denoted by the equation (5). Finally, F1-score is the harmonic mean of precision and recall. It combines precision and recall into a single number as in the formulation (6).

$$Recall = \frac{Number\ of\ correctly\ detected\ anomalies}{Number\ of\ actual\ anomalies} \tag{4}$$

$$Precision = \frac{Number\ of\ correctly\ detected\ anomalies}{Number\ of\ detected\ anomalies} \quad (5)$$

$$F1 - score = 2 * \frac{Recall * Precision}{Recall + Precision} \quad (6)$$

There are three parameters determined for the algorithm: α in the EDR measurement, $Thresh_{Dist}$ and $Thresh_{Den}$. α is a threshold for determining whether two points of two trajectories are matched or not. If they are matched, the cost of replacement is 0; otherwise, it is 1. Based on the floor plan of the MIT Badge dataset in Figure 2, we recognize that two workers are close to each other if the Euclidean distance between them is less than 10 meters. The parameter α is then set to 1000 according to the scale in the floor plan of MIT Badge dataset.

In order to determine $Thres_{Dist}$ and $Thres_{Den}$, we must first define the values of β_1 and β_2, respectively. The precision and recall of algorithm depend on the values of β_1 and β_2. To ensure the anomaly detection ability as well as the precision of the algorithm, we choose: $\beta_1 = -0.4$ and $\beta_2 = -0.6$ in this work. Then we obtain the specific values of $Thres_{Dist}$ and $Thres_{Den}$. The value of $Thres_{Dist}$ is the same for all timeslots, and it is shown as in Table 4 over 5 run times.

Table 4. The value the distance threshold over 5 run times

Run Time	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}
$Thres_{Dist}$	57.13	57.25	55.6	56.69	55.64

The density threshold $Thres_{Den}^{T}$ is determined for each timeslot T, and distribution of the density threshold over all timeslots is shown in Figure 3. As can be seen, the density threshold is low during the first hours and quickly reduces during lunch and last hours. Meanwhile, this value is the highest from 1.00 pm to 3.00 pm in the range of red ellipse.

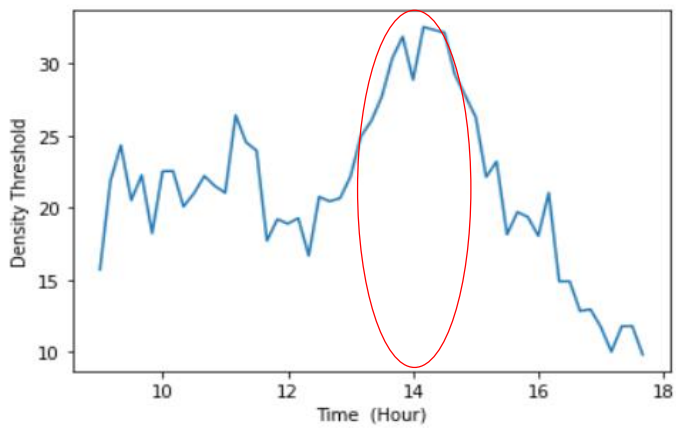


Figure 3. Distribution of density threshold over time

We also compare the algorithm's performance between using EDR and using Euclidean distance. Since Euclidean distance requires the same length of trajectories, the trajectories that missed some points are

interpolated. The linear interpolation method was chosen due to its simplicity [15]. Results are presented as Table 5 and figure 4. As can be seen, the recall and precision of algorithm using EDR measurement is better than using Euclidean distance function. In the case using EDR, the anomaly detection ability of algorithm is higher than 8.95% when using Euclidean distance. Besides, the precision is also improved from 62.96 % for Euclidean to 70.09% for EDR. Finally, the value of F1-score increases from 66.52% with Euclidean to 74.68% with EDR.

Table 5. Results

	Recall (%)	Precision (%)	F1-score (%)
Euclidean	71.92	62.96	66.52
EDR	80.87	70.09	74.68

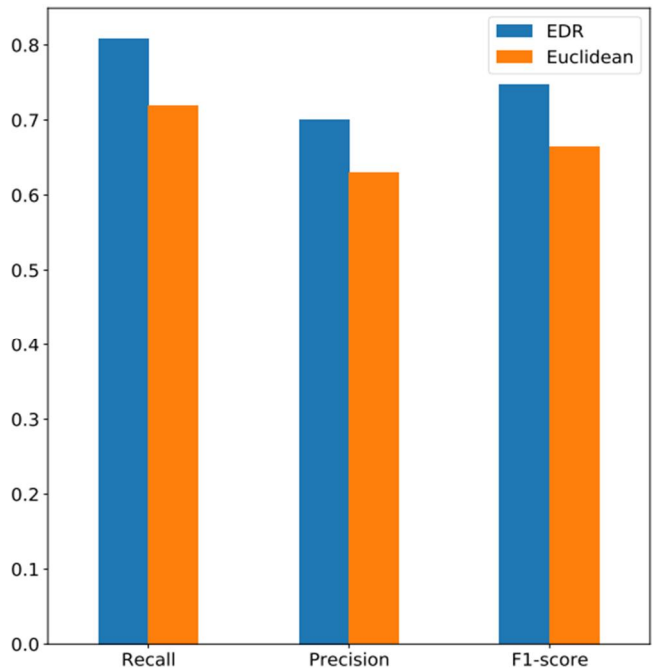


Figure 4. Performance of algorithm

5. Conclusion

In this work, we proposed a workers' anomalous trajectory detection framework. The framework is divided into two phases: offline and online. Task of the first phase is to determine the values for parameters: distance threshold and density threshold. Meanwhile, in the second phase, a new trajectory is checked that it is an anomaly or not. We also validate the algorithm using MIT Badge dataset and receive the better results with using EDR than with using Euclidean distance. As future work, we plan to improve the distance function by combining the semantic information and the space information for determining the distance of two trajectories.

Acknowledgement

This research was supported in part by Institute of Information & communication Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (2020-0-00869, Development of 5G-based

Shipbuilding & Marine Smart Communication Platform and Convergence Service), and in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) by the Ministry of Education under Grant 2021R1I1A3051364.

References

[1] Chandola, Varun, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey." *ACM computing surveys (CSUR)* 41.3 (2009): 1-58.
DOI: doi.org/10.1145/1541880.1541882

[2] Hsieh, Cheng-Ta, et al. "Abnormal event detection using trajectory features." *Journal of Information Technology and Applications* 5.1 (2011): 22-27.
DOI: 10.6302/JITA.201103_5(1).0003

[3] Morris, Brendan Tran, and Mohan Manubhai Trivedi. "A survey of vision-based trajectory learning and analysis for surveillance." *IEEE transactions on circuits and systems for video technology* 18.8 (2008): 1114-1127.
DOI: 10.1109/TCSVT.2008.927109

[4] Belhadi, Asma, et al. "Trajectory outlier detection: algorithms, taxonomies, evaluation, and open challenges." *ACM Transactions on Management Information Systems (TMIS)* 11.3 (2020): 1-29.
DOI: doi.org/10.1145/3399631

[5] Lee, Jae-Gil, Jiawei Han, and Xiaolei Li. "Trajectory outlier detection: A partition-and-detect framework." *2008 IEEE 24th International Conference on Data Engineering*. IEEE, 2008.
DOI: 10.1109/ICDE.2008.4497422

[6] Zhu, Zhihua, et al. "Sub-trajectory-and trajectory-neighbor-based outlier detection over trajectory streams." *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, Cham, 2018.
DOI: 10.1007/978-3-319-93034-3_44

[7] Zhang, Daqing, et al. "iBAT: detecting anomalous taxi trajectories from GPS traces." *Proceedings of the 13th international conference on Ubiquitous computing*. 2011.
DOI: doi.org/10.1145/2030112.2030127

[8] Ghrab, Najla Bouarada, Emna Fendri, and Mohamed Hammami. "Abnormal events detection based on trajectory clustering." *2016 13th International Conference on Computer Graphics, Imaging and Visualization (CGiV)*. IEEE, 2016. DOI: 10.1109/CGiV.2016.65

[9] Wang, Yulong, et al. "Detecting anomalous trajectories and behavior patterns using hierarchical clustering from taxi GPS data." *ISPRS International Journal of Geo-Information* 7.1 (2018): 25.
DOI: doi.org/10.3390/ijgi7010025

[10] Tao, Yaguang, et al. "A comparative analysis of trajectory similarity measures." *GIScience & Remote Sensing* 58.5 (2021): 643-669.
DOI: doi.org/10.1080/15481603.2021.1908927

[11] Vlachos, Michail, George Kollios, and Dimitrios Gunopulos. "Discovering similar multidimensional trajectories." *Proceedings 18th international conference on data engineering*. IEEE, 2002.
DOI: 10.1109/ICDE.2002.994784

[12] Berndt, D. J., and J. Clifford. "Using Dynamic Time Warping to Find Patterns in Time Series." *In KDD Workshop*, 1994, 359–370.

[13] Chen, L. M., T. Özsu, and V. Oria. "Robust and Fast Similarity Search for Moving Object Trajectories." *In Proc. ACM SIGMOD International Conference on Management of Data*, 2005, 491–502.
DOI:10.1145/1066157.1066213

[14] Olguín, Daniel Olguín, et al. "Sensible organizations: Technology and methodology for automatically measuring organizational behavior." *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39.1 (2008): 43-55.
DOI: 10.1109/TSMCB.2008.2006638

[15] Usman, Koredianto, and Mohammad Ramdhani. "Comparison of Classical Interpolation Methods and Compressive Sensing for Missing Data Reconstruction." *2019 IEEE International Conference on Signals and Systems (ICSigSys)*. IEEE, 2019.
DOI: 10.1109/ICSIGSYS.2019.8811057