Comparison of Similarity Measures for Trajectory Clustering - Aviation Use Case

Marija Todoric, and Toni Mastelic

***Abstract*—Various distance-based clustering algorithms have been reported, but the core component of all of them is a similarity or distance measure for classification of data. Rather than setting the priority to comparison of the performance of different clustering algorithms, it may be worthy to analyze the influence of different similarity measures on the results of clustering algorithms. The main contribution of this work is a comparative study of the impact of 9 similarity measures on similarity-based trajectory clustering using DBSCAN algorithm for commercial flight dataset. The novelty in this comparison is exploring the robustness of the clustering algorithm with respect to algorithm parameter. We evaluate the accuracy of clustering, accuracy of anomaly detection, algorithmic efficiency, and we determine the behavior profile for each measure. We show that DTW and Frechet distance lead to the best clustering results, while LCSS and Hausdorff Cosine should be avoided for this task.**

***Index Terms*—similarity measure; clustering; comparison; aviation.**

I. Introduction

Clustering is a common data analysis method in the field of statistics. Clustering algorithms identify distinct groups of data, by grouping similar data sets into clusters depending on the definition of a similarity function, or a pairwise distance.

In statistical data analysis, clustering is an essential tool broadly implemented in different scientific areas such as data mining [1], [2], [3], pattern recognition [4], geographic information systems [5], information retrieval [6] or microbiology analysis [7]. Moreover, it can be used as a form of feature engineering [8], where one can map together existing and new data points and relate with an already identified cluster. Some specific applications of cluster analysis in machine learning include market segmentation, image segmentation, medical image processing and building a recommendation system. An important application of clustering is anomaly detection, i.e. by using some clustering methods, one may detect outliers [9]. Relevant representatives of algorithms for anomaly detection include DBSCAN [10], HDBSCAN [11], OPTICS [12] and IMS [13].

Classification of data into groups requires certain methods for computing the distance between objects. Techniques providing distance information are known as similarity or distance measure - a measure of the distance between points in multidimensional space [14]. The outcome of the computation of (dis)similarity between each pair of objects is a distance(similarity) matrix - the greater (dis)similarity of two objects, the greater the value of the measure.

Distance-based clustering algorithms use distance measures to cluster similar data points into the same clusters, while distant data points are located in different clusters. The most commonly used measures are based on distance functions such as Euclidean distance, Manhattan distance, Cosine similarity, Minkowski distance, Dynamic Time Warping (DTW). Several studies were conducted on comparing and evaluating the similarity functions.

There are numerous algorithms proposed for clustering of data, but fundamentally they all depend on a similarity metric for categorizing individual data.

Therefore, instead of focusing on comparison of the performance of various clustering algorithms, it is interesting to compare different similarity metrics and their influence on data clustering.

Few studies have been reported that compared different similarity metrics in various machine learning and data mining applications [15], [16], [17], [18], [19]. In [20] six density functions including Euclidean, DTW, PDTW, EDR, EPR, and LCSS distances were analyzed for the measurement of trajectory similarity by applying different transformations.

More recently, in [21] strategies of clustering algorithms and similarity measures for general trajectories have been discussed. Focusing on particular domain of vessel trajectories [22], a brief review on topic has been provided. Regarding the performance of density functions in spatial trajectory clustering, Besse et al. [23] evaluated the performance of their suggested density function and some known density functions in two clustering methods of hierarchical and affinity propagation. In [24] the authors have compared the efficiency of eight similarity functions in density-based trajectory clustering on three different datasets and proposed two modified validation measures for density-based trajectory clustering. Here, we focus on aviation trajectory dataset and compare the impact of a set of similarity metrics on the robustness of clustering task.

In order to obtain the accurate clustering results, it is of interest to explore the robustness of the clustering methods. In particular, we focus on DBSCAN (density-based spatial clustering of applications with noise) [10] algorithm. In DBSCAN-like clustering methods the number of clusters nor

Manuscript received November 21, 2022; revised December 7, 2022. Date of publication June 30, 2023. Date of current version June 30, 2023.

M. Todoric and T. Mastelic were with the Research Department of Ericsson Nikola Tesla, Croatia, e-mails: marija.todoric@ericsson.com and toni.mastelic@ericsson.com.

Digital Object Identifier (DOI): 10.24138/jcomss-2022-0116

the initial cluster centers are not required, and these types of algorithms are efficient in detecting clusters of any shape. At the same time, in this type of algorithms, problems may arise in adjusting the parameters that determine neighborhood radius and neighborhood density according to the density of clusters.

In datasets with clusters of different levels of density, if the parameters are set up for clusters with low point density, clusters with high point density might merge with one another. On the other hand, when the parameters are adjusted for clusters with high density, the clusters with low density could be recognized as noise. From this perspective, one may conclude that algorithms that are able to run correctly in a broad range of change interval could be more powerful. Therefore, the robustness of algorithm with respect to parameters provides accurate classification for datasets with different densities.

The main contribution of this work is exploring the impact of different similarity measures on similarity-based trajectory clustering on the basis of commercial flight dataset. Moreover, as the novelty of the research, we investigate the robustness of clustering algorithm, evaluate the accuracy of clustering, accuracy of anomaly detection, computational cost, and we determine the behavior profile for each similarity measure. By analyzing the experimental results, we provide an overview of the validity of each measure in trajectory clustering task in aviation domain.

The evaluation is performed on a dataset from Flightradar24 (www.flightradar24.com) which contains 32,459 flights over Europe in a single day. A dataset is created by taking flights of 10 most frequent routes in Europe, where we also took into account the diversity of routes in order to cover different cases of incorrect clustering.

The rest of the paper is organized as follows. Section II describes published matter of this research area. Section III gives an overview of similarity measures. The methodology used in this work is explained in Section IV, with results presented in Section V. Finally, a discussion and conclusion are presented in Sections VI and VII, respectively.

II. Related Work

Different types of clustering algorithms classified as hierarchical or non-hierarchical clustering have been reported in the literature, and they are mainly designed for clustering of point data. Hierarchical clustering creates tree of clusters by decomposing the given dataset on the basis of hierarchy. The bottom-up decomposition approach is defined as the agglomerative hierarchical clustering algorithm, while the top-down decomposition method as the split hierarchical clustering algorithm. Some important representative algorithms are DIANA [25], BIRCH [26], and CURE [27].

In non-hierarchical clustering approach, the relationship between clusters is undetermined. It encompasses partition based method, density based method, grid based method, and model based method. In partition based methods one determines the count of clusters before processing. Common partition based algorithms are k-means [28], [29] and k-medoids [30]. The concept of density based methods is adding the area to the cluster which is nearer to it, as long as the density of points in the area is greater than the threshold. The representative algorithms include DBSCAN [10] and OPTICS [12] algorithm. Grid based methods are based on a multi-resolution grid data structure where the data space is quantized to a limited number of units, and one performs all clustering operations on the grid. The typical algorithms are STING [31] and CLIQUE [32]. Finally, model based clustering algorithms assume a model for each cluster, and tie to find the best fitting data for the given model. Such algorithm locates clusters by building density functions which indicate the spatial distribution of the data points. They include neural network method and statistical method, where the typical algorithm is COBWEB [33].

Regarding the anomaly detection, there are three different categories of clustering-based anomaly detection techniques [9]. In the first category, one assumes that normal data instances are a part of cluster while anomalies are not. Anomalies are considered as clustering outliers or noise. The most important representatives are DBSCAN [10], HDBSCAN [11], and OPTICS [12]. In the second category, it is assumed that normal instances lie near their closest cluster centroid, while anomalies lie far away from them. First, the algorithm clusters the data and then one computes an anomaly score for each data instance depending on the distance to its nearest cluster centroid. An important example of this is IMS [13] algorithm. The third category of algorithms deals with the problem when clusters of anomalies are formed. In order to determine the anomalies, the threshold is deduced by the cluster size or density [9].

There are various methods in the literature to cluster flight trajectories [34], [35], [36], [37], and the core of most of them are density-based clustering algorithms such as DBSCAN [10]. For example, DBSCAN in [37], gives satisfactory results in characterisation of traffic flows on the basis of the recorded radar tracks. Moreover, [38] reported another framework based on DBSCAN and k-means to examine the patterns of traffic.

In this work, in the context of trajectory clustering, we focus on the particular algorithm DBSCAN among previously explained clustering algorithms that use distance metrics. Next subsection provides detailed explanation of distance metrics we used, what is a foundation for correct clustering.

III. Overview of Distance Metrics used for Clustering

Trajectories are mathematical objects used to characterize the evolution of a moving object. They are described by the state vector with parameters $(x(t), y(t), \ldots)$ evolving in time. In practice, this state vector is only known at some sampled times.

The measurement of trajectory similarity is one of the key elements in trajectory clustering. Depending on the purpose of clustering, different comparison strategies should be selected. In this section, we give an overview of relevant distance measures in order to provide deeper understanding of research process.

A. Frechet Distance

A metric called Frechet distance [39], [40] takes into account both the location and sequential relationship of the points along the trajectories. Frechet distance between two curves corresponds to the maximum distance between two point objects that traverse the trajectories with arbitrary non-negative speeds.

B. Dynamic Time Warping Distance

Dynamic time warping [41], [42] (DTW) is one of well-known distance measures between a pairwise of time series, but any data that can be turned into a linear sequence can be analyzed with DTW. It calculates an optimal match between two given sequences with certain restrictions and rules. When applied to compute the similarity between two trajectories, its goal is to find the warping path between two trajectories with the smallest warping cost, by using dynamic programming technique.

C. Longest Common Sub-Sequence

Longest Common Subsequence [43], [44] (LCSS) denotes the longest common subsequence existing in two trajectory sequences. A subsequence is a sequence that appears in the same relative order, but not necessarily contiguous. On one hand, LCSS permits certain deviation existing in sampling data, and consequently it is effective and efficient in practical application. On the other hand, it uses two parameters - if the location of moving object is recorded in 2 dimensions, there is a distance threshold for each dimension. Therefore, the determination of two optimal parameters might be a complex problem.

D. Partial Curve Mapping

The partial curve mapping [45] method uses a combination of arc-length and area to determine the best match between two arbitrary curves. In original paper the steps of the algorithm for computing the curve mismatch error are described on test curve and computed curve. This methodology involves a curve matching metric which is computed using the volume between the test curve and the computed curve section.

E. Area between Two Curves

This curve similarity measure represents the area between two curves [46], i.e. the amount of mismatch between the two curves. If two curves would appear directly on top of each other, this measure of similarity would return a zero distance.

F. Hausdorff Distance

The Hausdorff distance measures the degree of mismatch between two sets by measuring the distance of the point of the first set that is farthest from any point of the second set and vice versa [47]. Intuitively, if the Hausdorff distance is d, then every point of the first set must be within a distance d of some point of the other set and vice versa. It is important to note that Hausdorff distance is related to spatial point sets where the distances between points are calculated, and temporal component is not included in calculation. Therefore, reversing the path does not change the Hausdorff distance [48]. The Hausdorff distance can be used in problems related to image comparison, contour fitting, pattern recognition, computer vision and many various fields, with the problems of shape matching and comparison. Moreover, clustering of trajectories based on Hausdorff distance has been reported in [49], [50]. Distance function used to calculate the distance can be Manhattan, Euclidean, Chebyshev, or Cosine similarity.

IV. TRAJECTORY CLUSTERING

This section describes the flight dataset and the methodology used for trajectory clustering.

A. Dataset

For the sake of clarity, we point out that, in this work, trajectory refers to a sequence of recording related to an aircraft. The dataset contains positions and altitudes of commercial flights over Europe, including also additional metadata information for more thorough analysis. In the following, we describe the steps of data preprocessing. We use data contributed by Flightradar24 AB, a global flight tracking service which provides real-time information about aircraft around the world. One obtains the data from a network of receivers that capture ADS-B (Automatic Dependent Surveillance-Broadcast) or mode-S (Selective) transponder signals from aircraft. We combined transponder data with Flightradar24's reference database in order to obtain a complete dataset with all important information. The dataset consists of two parts, flight data and trajectory data, as shown in Tables I and II.

We used the equipment and aircraft ID to remove the airport ground vehicles and private aircraft, while the flight number and call sign were used to find commercial flights. We used the flight ID to identify flights uniquely and to map trajectory data to them from Table II.

The update of aircraft positions was every 5 seconds during take-off and landing. During steady flight the update was increased to a maximum of 60 seconds. The initial dataset used in this research consisted of 47,126 trajectories recorded during a single day, namely 31 January 2018 over Europe. In this dataset, 32,459 flights were identified as commercial flights. Other non-commercial flights consist of ground vehicles, private aircraft, flights without a call sign, UFOs, and grounded flights.

In the next step, we eliminated flights where the flight took off or landed outside the recording period. We grouped the dataset by routes, i.e. flights which have the same combination of *'scheduled from'* and *'scheduled to'*, and sorted them by the count as shown in Table III.

Finally, we chose the flights of 10 most frequent routes, where we also took into account variety of routes in order to describe different cases of incorrect clustering. Table IV shows selected routes and associated number of flights.

Final dataset consists of 262 commercial flights, with a total of 57,187 positions and their altitudes, which were used for interpolation of flight trajectories. Figure 1 depicts the trajectories of commercial flights from the described dataset. There are occasionally minor inaccuracies presenting the noise in Flightradar24 data. For example, transponders can generate

TABLE I: METADATA OF THE DATASET SEGMENT CONTAINING THE FLIGHT DATA. ICAO, INTERNATIONAL CIVIL AVIATION ORGANIZATION.

Data Field	Description	Example
Flight ID	Unique Identifier for the flight	246716779
Aircraft ID	24 bit mode-S identifier in hexadecimals	7538182
Registration	Aircraft registration matched from the aircraft address	EPAPF
Equipment	ICAO aircsraft designator, mapped from the address	A320
Call Sign	Up to 8 characters as sent from the aircraft transponder	IRC511
Flight number	Commercial flight number, interpreted from the call sign	EP511
Schd_from	IATA code for scheduled departure airport	IST
Schd_to	IATA code for scheduled arrival airport	IKA

TABLE II: METADATA OF THE DATASET SEGMENT CONTAINING THE TRAJECTORY DATA.

Data Field	Description	Example
Snapshot ID	Time of position update in seconds since 1 Jan 1970 00:00:00 UTC	1504289296
Altitude	Height above sea level, in feet	5040
Latitude	Floating point format	60.39691
Longitude	Floating point format	5.19971

TABLE III: ROUTES SORTED BY THE COUNT.

Scheduled from	Scheduled to	Number of flights
BGO	OSL	32
DME	SIP	30
SIP	DME	30
OSL	BGO	29
TRD	OSL	29
OSL	TRD	29
MAD	PMI	27
⋮	⋮	⋮

TABLE IV: 10 SELECTED ROUTES.

Scheduled from	Scheduled to	Number of flights
BGO	OSL	32
DME	SIP	30
SIP	DME	30
OSL	BGO	29
TRD	OSL	29
MAD	PMI	27
SAW	AYT	27
IST	ADB	26
SVG	OSL	26
SAW	ADB	24

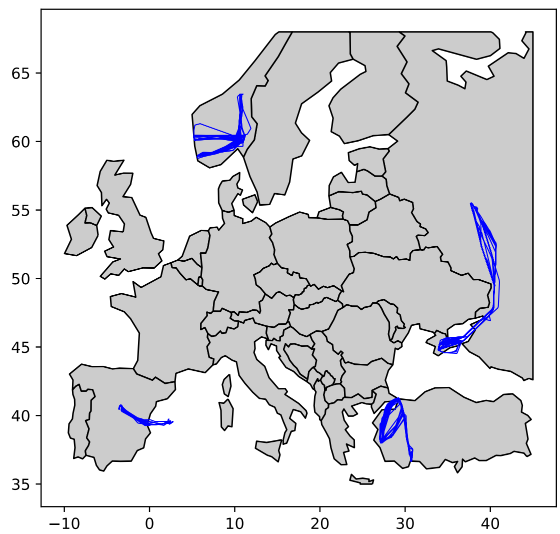

Fig. 1. Trajectories of selected commercial flights.

errors by transmitting random or incorrect position. However, our dataset is small and specific, and we assume that for our analysis, the noise can be neglected.

B. Methodology

Density-Based Clustering is an unsupervised machine learning method which determines distinctive clusters in the data, grounded on the concept that a cluster in data space is a dense region, divided from other clusters by regions of lower density of points. Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [10], a basic algorithm for density-based clustering, can discover clusters of different shapes and sizes from a large amount of data, containing noise and outliers. There are two parameters required for this algorithm, $eps(\epsilon)$, the maximum distance between two samples for them to be considered as in the same neighborhood, and $minPts$, the minimum number of points clustered together within ϵ radius. Depending on these parameters, the algorithm determines whether particular values in the dataset are outliers or not.

In the first step, for each similarity measure, one constructs similarity matrix from the trajectory dataset generated in the preprocessing phase. Then, we perform the DBSCAN algorithm that divides the dataset into clusters and determines the outliers. The output of the algorithm is a set of trajectory clusters and the set of outliers, trajectories which could not be allocated to a cluster and therefore demonstrate some degree of abnormality. DBSCAN algorithm is applied for a range of the density parameter ϵ, and the minimum sample size $minPts$, with the aim to examine the behavior of trajectory clustering

TABLE V: RELATIONSHIP BETWEEN FLIGHTS.

Name	Description
anomaly	outlier flight
younger brother	flight belonging to smaller cluster when flights of one route are divided in few clusters
step brother	outnumbered flights in a group of flights where one point is the same
twin	outnumbered flights when cluster contains flights of opposite routes
wrongly clustered	outnumbered flights in a cluster which contains flights of completely different routes

task. The ground truth set has 10 clusters, i.e. there are 10 different relations.

Anomalies are outlier flights detected as noise in DBSCAN algorithm. In the case where the flights of one route are divided in more clusters, *younger brother* denotes flights which ended up in smaller cluster. When a cluster contains flights with different routes where one point is the same, outnumbered flights are called *step brothers*. In the case where cluster contains flights of opposite routes, the outnumbered flights are denoted as *twins*. If a cluster contains flights of completely different routes, outnumbered flights are *wrongly clustered*. In Table V these relations are presented systematically.

We want to emphasize that we are looking at the complete flight paths, rather than the first and last lat/long/altitude points in paths. Namely, our goal is to compare complete paths and find a similarity between them in order to cluster them correctly and group those belonging to the same route. As a consequence, this approach can serve as anomaly detection because it can potentially detect unusual paths in common routes.

V. EXPERIMENTAL RESULTS

In this section, we apply the methodology described in the previous section on our dataset and present the results. The following framework is implemented in Python 3 with function DBSCAN from Python package 'sklearn.cluster', distance functions from packages 'hausdorff' and 'SimilarityMeasures', and module metrics from package 'tslearn'.

A. Similarity Matrices

Figure 2 shows the similarity matrices computed for 9 distance measures quantifying the similarity between the trajectories. The following methods are used: Frechet distance, Dynamic Time Warping, Longest Common Sub-Sequence, Partial Curve Mapping, Area between two curves, Hausdorff Manhattan Distance, Hausdorff Euclidean distance, Hausdorff Chebyshev distance and Hausdorff Cosine distance. Moreover, the time cost for each measure is shown in Table VI. According to similar research and their analysis [24], [17], we considered the time cost as a computational cost, i.e. the parameter to compare algorithms. Time cost, i.e. computational cost, denotes the time needed to execute the algorithm.

Distance matrix is a 262x262 image from a 2-dimensional numpy array, where one square represents the value of similarity between two specific trajectories. The color of each square is determined by the value of the corresponding distance matrix element, ranging from dark blue to yellow. The bottom of the color map corresponds to very similar trajectories, and ascends to top denoting very different trajectories. Since the dataset is sorted according to the frequency of the routes, category axis labels show the names of the routes. We expect that distance matrix is divided in squares of specific colors. Namely, flights belonging to certain route should have comparable similarity with flights belonging to some other specific route, resulting in a square of specific color. Moreover, as the flights of the same route are expected to be very similar, a diagonal of the distance matrix should be placed on the bottom of the color map. To summarize, different colors of similarity matrix display relationships between flight trajectories.

In view of this, Figure 2 demonstrates that Frechet method gives the most distinct squares, indicating that this measure finely recognizes different routes, and one expects that clustering algorithm will cluster the flights correctly. Similar results come from Hausdorff Manhattan, Hausdorff Euclidean, and Hausdorff Chebyshev metrics. Hausdorff Cosine metric generally follows this behavior, but gives greater similarity between trajectories. Yellow parts of the distance matrix correspond to most distant routes. It is predicted that clustering will give comparable outcome for these metrics.

According to results, DTW metric gives the clearest distinction between flights of the same route. LCSS, PCM and Area between two curves also show clear difference between trajectories belonging to the same route. It is also important to note that LCSS shows greater relative distance measure between trajectories belonging to different routes. On the contrary, PCM displays smaller relative distance measure for such trajectories.

Time cost during one run is shown in Table VI for all similarity measures. Frechet measure clearly obtains the highest time cost. Dynamic Time Warping and Area between two curves also have high time cost, Partial Curve Mapping is slightly easier to run. Computation times of Longest common Sub-Sequence Hausdorff Manhattan Distance, Hausdorff Euclidean Distance, Hausdorff Chebyshev distance and Hausdorff Cosine distance are of the same order and have the lowest values.

B. Correctness of Clustering

Here we show the results of the DBSCAN clustering algorithm which divides the original trajectory dataset into clusters for the range of ϵ values. The result is presented in a form of a stack plot in order to show how each part makes up the whole. The ϵ range is taken so that it presents characteristic behavior of the stack plot for each particular measure. Constituents of the stack plot are flights classified as anomalies (blue), step brother (orange), twin (green), younger brother (purple), wrongly clustered (red), or correctly clustered (dotted). The blue line shows the number of clusters for each measure for the range of ϵ values. Transparent yellow area shows the range

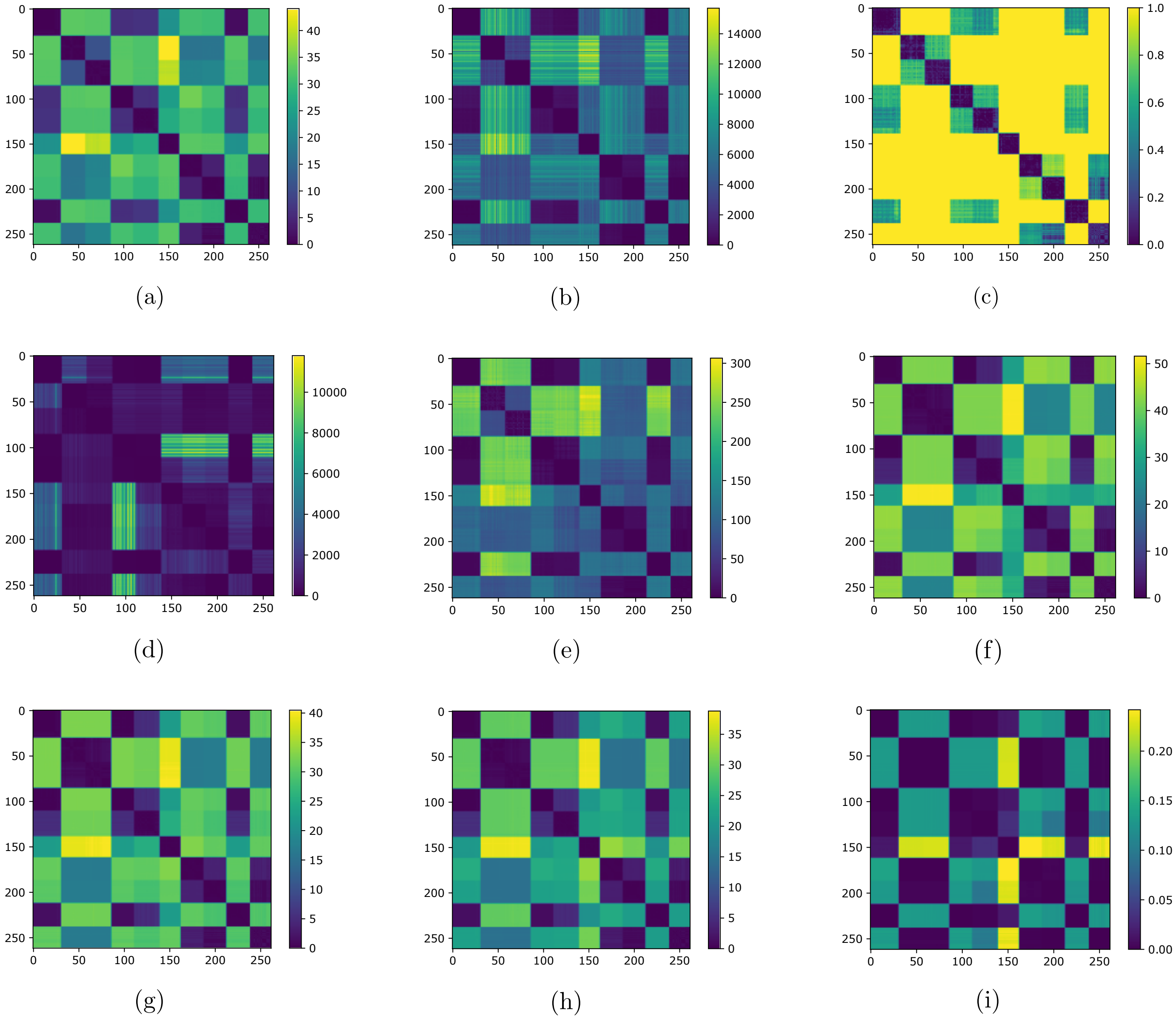

Fig. 2. Similarity matrices for different similarity measures: (a) Frechet distance (b) DTW (c) LCSS (d) PCM (e) Area between two curves (f) Hausdorff Manhattan distance (g) Hausdorff Euclidean distance (h) Hausdorff Chebyshev distance (i) Hausdorff Cosine distance.

of 10-11 clusters, corresponding to the ground-truth number of clusters for this dataset. Namely, there are 10 different routes and one cluster of anomalies. One has to determine the appropriate ϵ value in order to obtain good clustering results. We adopted the standard approach [51] for determination of optimal ϵ value for DBSCAN algorithm. Dashed black line represents the calculated result.

Figure 3 shows the stack plots computed for 9 similarity measures.

We can observe the general behavior of the stack plot. Anomalies (blue) are present for smaller ϵ values. For very small ϵ, the number of anomalies is very large. This is expected since ϵ represents the radius of neighborhood around a point. Since the radius is very small, distance between most of the points is larger than this threshold, and they are considered to be outliers, i.e. anomalies. As ϵ increases, the number of anomalies rapidly decreases and there are no anomalies above optimal ϵ value.

Flights incorrectly clustered as younger brothers appear for low ϵ values since the radius of formed clusters is small. Consequently, flights belonging to the same route may be separated into more clusters leading to the significant number of younger brothers. One should expect that twin flights emerge easily by increase of ϵ values, i.e. the radius of clusters, considering that they correspond to the reverse routes and are expected to be similar. Step brothers should also manifest with the rise of ϵ. Namely, for larger radius, flights with one coinciding point may be located into the same cluster. Obviously, if ϵ is significantly high, flights with completely different routes may arrive at the same cluster, manifesting in large portion of wrongly clustered (red) flights.

Let us consider stack plots from points of view of cor-

TABLE VI: TIME COST AND USED METHODS FOR DIFFERENT SIMILARITY MEASURES.

Similarity Measure	Method	Time Cost (s)
Frechet distance	package 'SimilarityMeasures'	29599.660
Dynamic Time Warping	package 'SimilarityMeasures'	2181.287
Longest Common Sub-Sequence	module tslearn.metrics	26.719
Partial Curve Mapping	package 'SimilarityMeasures'	316.308
Area between two curves	package 'SimilarityMeasures'	3414.910
Hausdorff Manhattan Distance	package 'hausdorff'	16.701
Hausdorff Euclidean distance	package 'hausdorff'	16.585
Hausdorff Chebyshev distance	package 'hausdorff'	16.407
Hausdorff Cosine distance	package 'hausdorff'	42.545

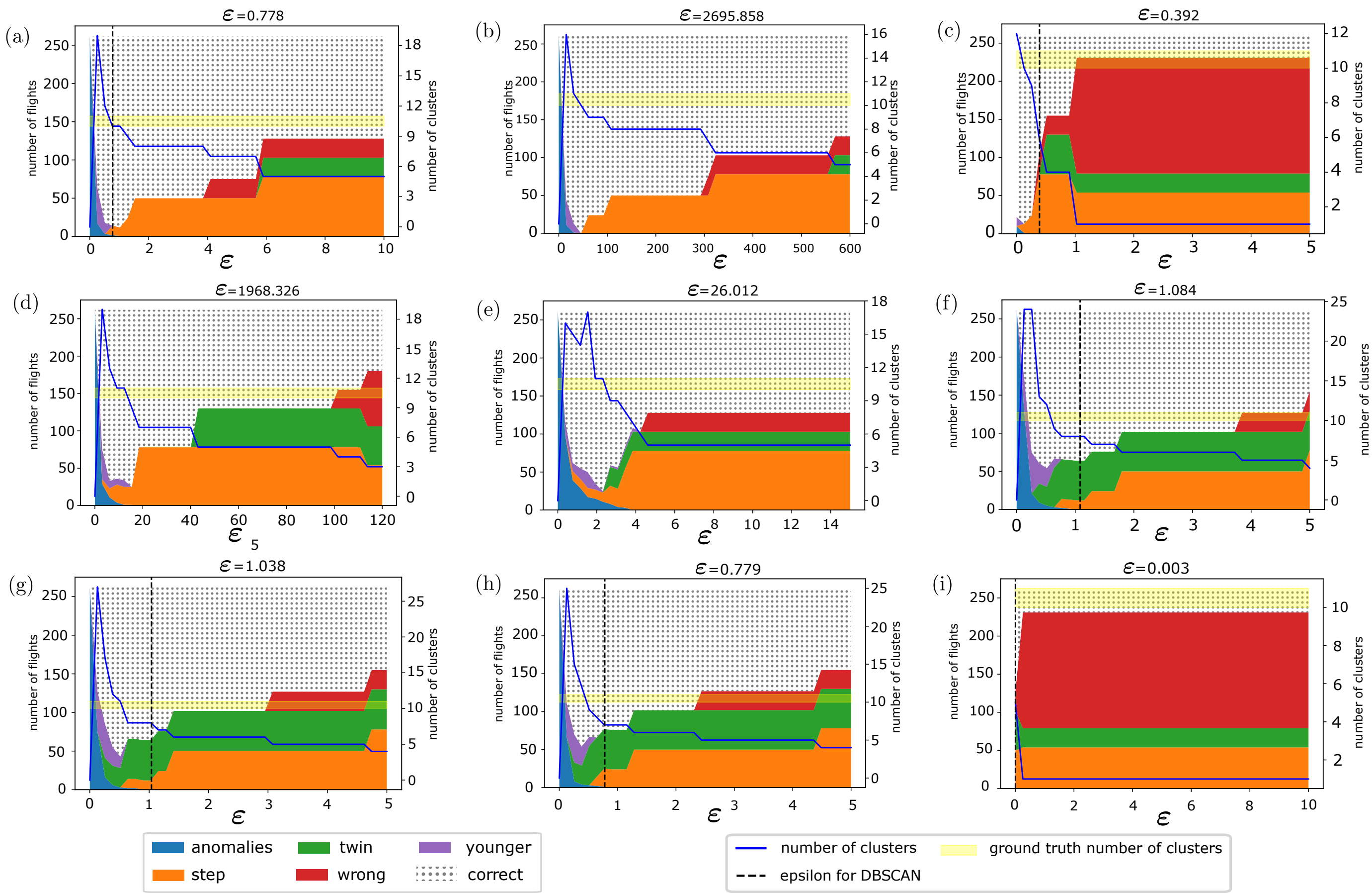

Fig. 3. Stack plots for different similarity measures: (a) Frechet distance (b) DTW (c) LCSS (d) PCM (e) Area between two curves (f) Hausdorff Manhattan distance (g) Hausdorff Euclidean distance (h) Hausdorff Chebyshev distance (i) Hausdorff Cosine distance.

rectness of the clustering and robustness with respect to parameter ϵ. The correctness of the algorithm is measured by the proportion of the dotted part of the stack plot, while the robustness is measured by the constancy of the result without substantial change.

We can observe that clustering using Frechet metric is very robust with respect to parameter ϵ. Moreover, we notice that it is also the most accurate measure according to the portion of the dotted part. Dashed line falls in the range of great accuracy. Number of clusters is in the yellow range around the optimal ϵ value.

DTW is another metric manifesting good clustering results. It is very robust for a large range of epsilon values, namely ϵ values can be taken within the range up to 300, still providing a good clustering result. However, calculated optimal ϵ value is very high, and falls deep within wrongly clustered trajectories. It is interesting to note that DTW similarity is a 'non-metric' distance as shown in [52], where Frechet distance has been preferred since it is a metric distance. However, despite of this fact, here we still find DTW useful as it achieves a good result for clustering task.

We can conclude that Frechet and DTW metrics give both the most accurate and most robust clustering results for DBSCAN algorithm. DTW provides the largest range of robustness. Moreover, there is a narrow range with completely correct result. Frechet metrics also produces a very satisfactory result, however, the range of robustness is significantly smaller than the range for DTW. On the other hand, here the method for determination of optimal ϵ can be safely applied since it produces the best clustering result. Nevertheless, when applying DTW, one cannot use this standard method for optimal ϵ since it gives value much larger than the range of interest. Considering the execution time, Frechet distance has convincingly largest execution time among all measures, while the execution time for Dynamic Time Warping is around 10 times smaller.

Hausdorff Manhattan, Hausdorff Euclidean and Hausdorff Chebyshev measures provide similar results. First, we notice that the shapes of the curves for number of clusters are comparable. The clustering results are good for a narrow ϵ range around optimal ϵ value. Correct number of clusters also appears for a narrow ϵ range. These metrics can be used for a small range around optimal ϵ value, but these methods do not produce very precise results. Robustness of the results is also limited. However, the method for determination of optimal ϵ value points to the value with satisfactory clustering result for all three metrics. Moreover, these metrics have lowest execution time among all measures. Therefore, we may suggest that they can be used for a quick result providing an approximate insight into division of flights into clusters.

PCM metric gives quite good clustering result for a significant ϵ range, namely for ϵ range up to 20. The correctness is reduced for larger ϵ values. Since our method for calculation of optimal ϵ gives very high value, this method is not applicable for PCM measure. The result for this metric is not robust, and it should not be used for larger ϵ values due to the increase of incorrectly clustered flights. Regarding the time cost, it is of the middle value.

Area between two curves produces a satisfactory clustering result for a very narrow ϵ range. Incorrectly clustered flights emerge easily for higher ϵ values. The calculated value of optimal ϵ is much higher than the ϵ values of interest, meaning that this method does not work for this metric. Moreover, time cost is high, and standard method for determination of ϵ does not produce correct result. Therefore, this metric is not recommended.

Next, longest common subsequence gives good clustering result for a very small ϵ range around calculated optimal ϵ value. However, the clustering result is not robust with respect to ϵ and wrongly clustered flights appear even for low ϵ values. Compared to other metrics, the portion of incorrectly clustered flights is extremely large for most of ϵ values. Small execution time cost does not justify the usage of this metric for DBSCAN algorithm. Finally, Hausdorff Cosine metric gives the worst clustering result. Namely, correctly clustered flights exist for extremely narrow ϵ range and wrongly clustered flights are present for a complete ϵ range, taking over the largest part of the clustered flights. Moreover, the calculated value of optimal ϵ is almost zero. We can conclude that this metric should be avoided for DBSCAN algorithm in case of trajectory clustering.

VI. Discussion

The dataset used in this research included 10 different routes for a single day, which were chosen so that various relationships between them are included, i.e. anomalies, younger brothers, step brothers, twins. Our aim was to explore the impact of different similarity measures on correctness of clustering, with the emphasis on fineness of clustering task. Obviously, different result would be obtained for another choice of routes having different relationships between them. There would be different percentages of anomalies, younger brothers, step brothers and twins, according to the chosen set. However, our results still give valuable insights into influence of each measure on clustering task with DBSCAN algorithm which can be mapped onto another dataset. For a larger flight dataset, we assume that the result would be more precise.

Since this dataset includes only one day, we could improve this analysis by including more flights for another days in a week. Generally, this methodology can be extended over air traffic to different types of trajectories where the spatial similarity is examined. For example, it can be used to analyze maritime traffic or animal movement trajectories with different similarity measures.

This study can be used to find the correct similarity metrics when using DBSCAN or similar density-based algorithms for trajectory clustering. It gives an overview of the robustness of clustering result for particular metrics, and therefore enables one to choose the appropriate ϵ. For example, if the analysis is aimed to determine trajectory anomalies, one should select ϵ which is small enough so that anomalies are present, but big enough so that anomalies indeed represent outliers. This analysis also points out which metric supports the standard method for determination of optimal ϵ value, and where one should look for another approach.

Next step would be to investigate another densely clustering models, which classify trajectories by distance metrics. For example, one could use some modifications of DBSCAN algorithm such as HDBSCAN [11], or TRACLUS [53] or autoencoders [54]. We could also extend this comparative study by including other similarity measures which take into account only spatial dimension, such as SSPD [23].

VII. Conclusion

In this paper, we have conducted a comparative study of impact of 9 similarity measures on trajectory clustering task in air traffic, by using DBSCAN algorithm. We examined the correctness of the clustering and robustness of the result with respect to ϵ parameter of the algorithm.

This research has shown that DTW and Frechet distance give the best results. The advantage of DTW is reasonable time cost, since high computational cost of Frechet distance weakens its ability compared to other measures. Hausdorff Manhattan, Hausdorff Euclidean, and Hausdorff Chebyshev measures produce worse results, but computational time is significantly smaller, suggesting that these measures can be used as a first step in a clustering task. PCM metrics produces to some extent acceptable result for moderate range. Area between two curves gives satisfactory result in the very narrow range, but has high computational cost. LCSS and Hausdorff Cosine are convincingly not competent for the task.

For future work, we aim at applying this methodology for larger aviation dataset obtained for more days or more routes. Furthermore, this methodology can be extended over air traffic to some other trajectory types such as ship trajectories. It would be of interest to examine other similarity measures, as well as different methods for optimal ϵ estimation.

REFERENCES

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

[2] S. Agarwal, "Data mining: Data mining concepts and techniques," in *2013 International Conference on Machine Intelligence and Research Advancement*, 2013, pp. 203–207. [Online]. Available: https://doi.org/10.1109/ICMIRA.2013.45

[3] J. Han, M. Kamber, and A. K. H. Tung, "Spatial clustering methods in data mining," in *Geographic Data Mining and Knowledge Discovery*. Taylor & Francis, 2001, pp. 188–217. [Online]. Available: https://doi.org/10.1201/b12382

[4] E. Diday, G. Govaert, Y. Lechevallier, and J. Sidi, "Clustering in pattern recognition," in *Digital Image Processing*. Springer Netherlands, 1981, pp. 19–58. [Online]. Available: https://doi.org/10.1007/978-94-009-8543-8_2

[5] T. Abraham and J. F. Roddick, "Survey of spatio-temporal databases," *GeoInformatica*, vol. 3, pp. 61–99, 1999. [Online]. Available: https://doi.org/10.1023/A:1009800916313

[6] S. Miyamoto, Y. Endo, S. Hayakawa, and E. Kataoka, "Classification and clustering of information objects based on fuzzy neighborhood system," vol. 4, 2005, pp. 3210–3215. [Online]. Available: https://doi.org/10.1109/icsmc.2005.1571640

[7] R. Nugent and M. Meila, "An overview of clustering applied to molecular biology," in *Methods in Molecular Biology*. Humana Press, 2009, pp. 369–404. [Online]. Available: https://doi.org/10.1007/978-1-60761-580-4_12

[8] M. Piernik and T. Morzy, "A study on using data clustering for feature extraction to improve the quality of classification," *Knowledge and Information Systems*, vol. 63, no. 7, pp. 1771–1805, 2021. [Online]. Available: https://doi.org/10.1007/s10115-021-01572-6

[9] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009. [Online]. Available: https://doi.org/10.1145/1541880.1541882

[10] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. of 2nd International Conference on Knowledge Discovery*, 1996, pp. 226–231. [Online]. Available: https://dl.acm.org/doi/10.5555/3001460.3001507

[11] R. J. G. B. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," in *Advances in Knowledge Discovery and Data Mining*. Springer Berlin Heidelberg, 2013, pp. 160–172. [Online]. Available: https://doi.org/10.1007/978-3-642-37456-2_14

[12] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS:Ordering Points To Identify the Clustering Structure," *ACM SIGMOD Record*, vol. 28, no. 2, pp. 49–60, 1999. [Online]. Available: https://doi.org/10.1145/304181.304187

[13] D. L. Iverson, "System health monitoring for space mission operations," in *2008 IEEE Aerospace Conference*. IEEE, 2008. [Online]. Available: https://doi.org/10.1109/aero.2008.4526646

[14] G. Upton and I. Cook, *A Dictionary of statistics*. Oxford University Press, 2008. [Online]. Available: https://doi.org/10.1093/acref/9780199541454.001.0001

[15] A. S. Shirkhorshidi, S. Aghabozorgi, and T. Y. Wah, "A comparison study on similarity and dissimilarity measures in clustering continuous data," *PLOS ONE*, vol. 10, no. 12, 2015. [Online]. Available: https://doi.org/10.1371/journal.pone.0144059

[16] S. Boriah, V. Chandola, and V. Kumar, "Similarity measures for categorical data: A comparative evaluation," in *Proceedings of the 2008 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, 2008. [Online]. Available: https://doi.org/10.1137/1.9781611972788.22

[17] Z. Zhang, K. Huang, and T. Tan, "Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes," in *18th International Conference on Pattern Recognition (ICPR'06)*. IEEE, 2006. [Online]. Available: https://doi.org/10.1109/icpr.2006.392

[18] D. J. Weller-Fahy, B. J. Borghetti, and A. A. Sodemann, "A survey of distance and similarity measures used within network intrusion anomaly detection," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 70–91, 2015. [Online]. Available: https://doi.org/10.1109/comst.2014.2336610

[19] J. Irani, N. Pise, and M. Phatak, "Clustering techniques and the similarity measures used in clustering: A survey," *International Journal of Computer Applications*, vol. 134, no. 7, pp. 9–14, 2016. [Online]. Available: https://doi.org/10.5120/ijca2016907841

[20] H. Wang, H. Su, K. Zheng, S. W. Sadiq, and X. Zhou, "An effectiveness study on trajectory similarity measures," in *Proceedings of the Twenty-Fourth Australasian Database Conference - Volume 137*, 2013, p. 13–22. [Online]. Available: https://dl.acm.org/doi/10.5555/2525416.2525418

[21] G. Yuan, P. Sun, J. Zhao, D. Li, and C. Wang, "A review of moving object trajectory clustering algorithms," *Artificial Intelligence Review*, vol. 47, no. 1, pp. 123–144, 2016. [Online]. Available: https://doi.org/10.1007/s10462-016-9477-7

[22] N. Lu, M. Liang, L. Yang, Y. Wang, N. Xiong, and R. W. Liu, "Shape-based vessel trajectory similarity computing and clustering: A brief review," in *2020 5th IEEE International Conference on Big Data Analytics (ICBDA)*. IEEE, 2020. [Online]. Available: https://doi.org/10.1109/icbda49040.2020.9101322

[23] P. C. Besse, B. Guillouet, J.-M. Loubes, and F. Royer, "Review and perspective for distance-based clustering of vehicle trajectories," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 11, pp. 3306–3317, 2016. [Online]. Available: https://doi.org/10.1109/tits.2016.2547641

[24] A. Moayedi, R. A. Abbaspour, and A. Chehreghan, "An evaluation of the efficiency of similarity functions in density-based clustering of spatial trajectories," *Annals of GIS*, vol. 25, no. 4, pp. 313–327, 2019. [Online]. Available: https://doi.org/10.1080/19475683.2019.1679254

[25] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley, 1990. [Online]. Available: https://onlinelibrary.wiley.com/doi/book/10.1002/9780470316801

[26] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: an efficient data clustering method for very large databases," pp. 103–114, 1996. [Online]. Available: https://doi.org/10.1145/235968.233324

[27] S. Guha, R. Rastogi, and K. Shim, "Cure: an efficient clustering algorithm for large databases," *Information Systems*, vol. 26, no. 1, pp. 35–58, 2001. [Online]. Available: https://doi.org/10.1016/s0306-4379(01)00008-4

[28] S. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982. [Online]. Available: https://doi.org/10.1109/tit.1982.1056489

[29] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, 1967, pp. 281–297.

[30] L. Kaufmann and P. J. Rousseeuw, "Clustering by means of medoids," *Data Analysis based on the L1-Norm and Related Methods*, pp. 405–416, 1987.

[31] W. Wei, Y. Jiong, and M. Richard, "Sting: A statistical information grid approach to spatial data mining," in *Proceedings of the International Conference on VLDB*, vol. 97, 1997, pp. 186–195. [Online]. Available: https://dl.acm.org/doi/10.5555/645923.758369

[32] H. Nagesh, S. Goil, and A. Choudhary, "Adaptive grids for clustering massive data sets," in *Proceedings of the 2001 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, 2001. [Online]. Available: https://doi.org/10.1137/1.9781611972719.7

[33] D. H. Fisher, "Knowledge acquisition via incremental conceptual clustering," *Machine Learning*, vol. 2, no. 2, pp. 139–172, 1987. [Online]. Available: https://doi.org/10.1007/bf00114265

[34] A. T. Nguyen, "Identification of air traffic flow segments via incremental deterministic annealing clustering, Ph. D. dissertation." University of Maryland, 2012.

[35] G. Andrienko, N. Andrienko, G. Fuchs, and J. M. C. Garcia, "Clustering trajectories by relevant parts for air traffic analysis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 34–44, 2018. [Online]. Available: https://doi.org/10.1109/tvcg.2017.2744322

[36] S. Rinzivillo, D. Pedreschi, M. Nanni, F. Giannotti, N. Andrienko, and G. Andrienko, "Visually driven analysis of movement data by progressive clustering," *Information Visualization*, vol. 7, no. 3-4, pp. 225–239, 2008. [Online]. Available: https://doi.org/10.1057/palgrave.ivs.9500183

[37] M. C. R. Murca, R. DeLaura, R. J. Hansman, R. Jordan, T. Reynolds, and H. Balakrishnan, "Trajectory clustering and classification for characterization of air traffic flows," in *16th*

AIAA Aviation Technology, Integration, and Operations Conference. American Institute of Aeronautics and Astronautics, 2016. [Online]. Available: https://doi.org/10.2514/6.2016-3760

[38] M. Gariel, A. N. Srivastava, and E. Feron, "Trajectory clustering and an application to airspace monitoring," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1511–1524, 2011. [Online]. Available: https://doi.org/10.1109/tits.2011.2160628

[39] T. Eiter and H. Mannila, "Computing discrete frechet distance," Technische Universitaet Wien, Tech. Rep., 1994.

[40] V. Khoshaein, "Trajectory clustering using a variation of Frechet distance, Ph. D. dissertation." University of Ottawa, 2014.

[41] D. Sankoff and J. Kruskal, *Time warps, string edits, and macromolecules: the theory and practice of sequence comparison*. Addison-Wesley, 1983.

[42] L. Chen, M. T. Özsu, and V. Oria, "Robust and fast similarity search for moving object trajectories," in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. ACM, 2005, pp. 491–502. [Online]. Available: https://doi.org/10.1145/1066157.1066213

[43] C. Rick, "Efficient computation of all longest common subsequences," in *Algorithm Theory - SWAT 2000*. Springer Berlin Heidelberg, 2000, pp. 407–418. [Online]. Available: https://doi.org/10.1007/3-540-44985-x_35

[44] M. Vlachos, M. Hadjieleftheriou, and D. Gunopulos, "Indexing multidimensional time-series," vol. 15. The VLDB Journal, 2006, pp. 1–20. [Online]. Available: https://doi.org/10.1007/s00778-004-0144-2

[45] K. Witowski and N. Stander, "Parameter identification of hysteretic models using partial curve mapping," in *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. American Institute of Aeronautics and Astronautics, 2012. [Online]. Available: https://doi.org/10.2514/6.2012-5580

[46] C. F. Jekel, G. Venter, M. P. Venter, N. Stander, and R. T. Haftka, "Similarity measures for identifying material parameters from hysteresis loops using inverse analysis," *International Journal of Material Forming*, vol. 12, no. 3, pp. 355–378, 2018. [Online]. Available: https://doi.org/10.1007/s12289-018-1421-8

[47] E. Deza and M. M. Deza, "Encyclopedia of distances," in *Encyclopedia of Distances*. Springer Berlin Heidelberg, 2009, pp. 1–583. [Online]. Available: https://doi.org/10.1007/978-3-642-00234-2_1

[48] A. A. Taha and A. Hanbury, "An efficient algorithm for calculating the exact hausdorff distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 11, pp. 2153–2163, 2015. [Online]. Available: https://doi.org/10.1109/tpami.2015.2408351

[49] J. Chen, R. Wang, L. Liu, and J. Song, "Clustering of trajectories based on hausdorff distance," in *2011 International Conference on Electronics, Communications and Control (ICECC)*. IEEE, 2011. [Online]. Available: https://doi.org/10.1109/icecc.2011.6066483

[50] L.-X. Wei, X.-H. He, Q.-Z. Teng, and M.-L. Gao, "Trajectory classification based on hausdorff distance and longest common SubSequence," *Journal of Electronics & Information Technology*, vol. 35, no. 4, pp. 784–790, 2014. [Online]. Available: https://doi.org/10.3724/sp.j.1146.2012.01078

[51] N. Rahmah and I. S. Sitanggang, "Determination of optimal epsilon (eps) value on DBSCAN algorithm to clustering data on peatland hotspots in sumatra," *IOP Conference Series: Earth and Environmental Science*, vol. 31, p. 012012, 2016. [Online]. Available: https://doi.org/10.1088/1755-1315/31/1/012012

[52] S. L. Seyler, A. Kumar, M. F. Thorpe, and O. Beckstein, "Path similarity analysis: A method for quantifying macromolecular pathways," *PLOS Computational Biology*, vol. 11, no. 10, 2015. [Online]. Available: https://doi.org/10.1371/journal.pcbi.1004568

[53] J.-G. Lee, J. Han, and K.-Y. Whang, "Trajectory clustering," in *Proceedings of the 2007 ACM SIGMOD international conference on Management of data - SIGMOD '07*. ACM Press, 2007. [Online]. Available: https://doi.org/10.1145/1247480.1247546

[54] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015. [Online]. Available: https://doi.org/10.1016/j.neunet.2014.09.003

Marija Todoric worked as a researcher at Ericsson Nikola Tesla d.d., Research department. She obtained Master degree in Physics at Faculty of Science in Zagreb in 2016. Afterwards, she worked as a research and teaching assistant at Department of Physics at Faculty of Science in Zagreb, where she received her PhD Degree in Physics in 2021. Her scientific interests include condensed matter physics, artificial inteligence, and data analysis.

Toni Mastelic worked as a researcher at Ericsson Nikola Tesla d.d., Research department. He did his bachelor and masters studies in Computer Science at the University of Split, FESB, Croatia, where he received his Bachelor degree in 2009, and Master degree in 2011. Afterwards, he worked as a research and later on as university assistant at Vienna University of Technology, where he pursued his PhD. Finally, he received his PhD degree in 2015 at the Institute of Software Technology and Interactive Systems, Vienna University of Technology.