Paper ET06

Cluster analysis vs. graph-based machine learning: the battle of the strongest in pattern recognition

Kostiantyn Drach, IST Austria / Intego Group, Klosterneuburg, AUSTRIA

Iryna Kotenko, Intego Group, Kharkiv, UKRAINE

ABSTRACT: Unsupervised learning algorithms allow researchers to discover hidden patterns within data. Algorithms of this nature independently extract and present hidden insights in complex datasets with no prior knowledge of the expected outcome. This paper discusses two categories of unsupervised learning algorithms: cluster analysis and graph-based machine learning. Clustering methods divide data into smaller subgroups and subsequently unveil any patterns hidden in the data. Graph-based machine learning, and other advanced methods based on dimensionality reduction, help researchers visualize hidden patterns by reducing the number of dimensions required to describe the original dataset and revealing robust geometric structures within the data. This paper describes a computational experiment performed on a clinical study involving 839 subjects. The authors described and ran algorithms of both categories to identify the pros and cons of each category while discovering patterns in the clinical study dataset, comparing results, and evaluating how two categories of algorithms can complement each other.

KEYWORDS: clustering, community detection, topological data analysis, graph-based machine learning

CONTENTS

INTRODUCTION

One of the most widely used tools in unsupervised machine learning designed for discovering hidden patterns and groupings in data is clustering. It can reveal similarities and differences in data and therefore is one of the main solutions for exploratory data analysis. Another widely used tool for exploratory analysis is community detection on networks (graphs) which allows to unveil the meaningful groupings of network nodes as well. Community detection algorithms can be used to analyze the data only when the graph model of the data has been built. Topological Data Analysis (TDA) – a novel approach to build a graph representation of a complex dataset – allows to construct such a model. TDA produces a robust topological model of the data.

In this paper, we run an extensive comparison between the methods of cluster analysis and community detection algorithms on topological models of the data. After a brief survey on TDA, clustering and community detection methods, we introduce several theoretical scores that are used in the comparison. As our main result, we show that the community detection algorithms applied to the topological model of the data can reveal statistically significant and clearer patterns different from those revealed by clustering. Many of the patters that we find with community detection methods were undetected by the clustering algorithms.

The dataset to perform the experiment is taken from NIDA research study of nicotine smokers (CENIC). The study included 839 participants, measured the average number of cigarettes smoked per day dependent on nicotine content over the span of 6 weeks. The original study managed to show that smoking low nicotine cigarettes decreases the average number of cigarettes smoked per day. In this paper, the results of the original study are confirmed. Furthermore, our methods resulted in additional, previously undiscovered findings.

Finally, we suggest a way to preprocess the data with the help of community detection algorithms on topological models of the data in order to improve the performance of clustering.

1. TOPOLOGICAL DATA ANALYSIS

Topological data analysis (TDA) is a novel approach of building visual representations of complex datasets. This analysis allows the extraction of comprehensive graphs from a dataset to provide a compressed graphical representation of a multidimensional set of interrelated outcomes. When applied to clinical data, this graph consists of nodes corresponding to patients participating in a study and edges connecting those who share similarities in terms of study outcomes. A similar approach was used in the current experiment, namely, TDA was applied to build a graph in which every node corresponds to a participant of the CENIC study (see Section 3.1), while two such nodes that exhibit a similar smoking behavior over the duration of the study are connected by an edge. We discuss this construction in more details in Section 3.2. In this section, we present a general introduction to TDA in clinical trials. This general approach will be specified, as indicated above, in the follow-up sections.

1.1. Topology and data mining

Topology is a field of mathematics that deals with the properties of objects that remain invariant under continuous deformation. Imagine a surface that is made of a very thin and elastic material. The surface can be bent, stretched, or crumpled in any way; however, it cannot be torn and its parts cannot be glued together. As the surface is deformed, it changes in many ways, but some properties remain the same. The idea underpinning topology is that some geometric properties depend not on the exact shape of an object but, rather, on how its parts are combined.

As a simple example, consider geometric figures on the plane representing the numerical digits 0, 1, 2, … 9. For a topologist, various representations of the digit 0 are equivalent since they can all be continuously transformed into each other without cutting or gluing (see Fig. 1 a-d). It is possible to change the size, thickness, or slope of the digit 0 through continuous deformation; however, one property remains invariant: the object separates the plane into two regions, namely the interior and the exterior. At the same time, 0 is not topologically equivalent to 1 or 8: 1 does not encircle a region and 8 contains two holes (see Fig. 1e). The topological classification of the digits 0, 1, 2, … 9 results in the following five classes:

$$\{0\}, \{1, 2, 3, 5, 7\}, \{4\}, \{6, 9\}, \{8\}.$$

The digits in any of the classes are topologically identical, but no two digits taken from distinct classes are equivalent from the topological point of view.

The number of holes in a geometric object is a basic topological property. Another significant property is connectedness. Intuitively, an object is connected if it consists of a single piece. For example, the curve representing 0 is connected; if any two points are removed from it, it will become disconnected. Pieces of a disconnected object that are themselves connected are referred to as connected components. In the mathematical study of topology, all of these intuitive concepts are examined on a rigorous basis and generalized to higher dimensions.

a) b) c) d) e)

Figure 1. Different representations of the digit 0 (a-d) are topologically equivalent. All share a common topological property: they divide the plane into an interior region and an exterior region. The digit e) is not equivalent to 0 since it encloses two internal regions

Topology deals with abstract mathematical entities, such as curves and surfaces, that consist of an infinite number of points. In practice, however, all datasets are necessarily finite. Recently, a new field has emerged at the crossroads of topology and data science. TDA aims to extract topological data, that is, qualitative information, from finite sets of data points. It involves exploring datasets (viewed as finite clouds of points in multidimensional space) at multiple scales or resolutions, from fine- to coarse-grained. Given a complex dataset, TDA can be used to extrapolate the underlying topology and build a compressed yet comprehensive topological summary of the dataset. TDA exploits various methods and algorithms stemming from computational topology and geometry, statistics, and data mining. For detailed expositions of the mathematical theories that underpin TDA and certain applications in biology, see [1, 2] and the references therein.

1.2. Understanding clinical data using topology

Topology was originally developed to distinguish between the qualitative properties of geometric objects. It can be used in conjunction with the usual data-analytic tools for the following tasks:

1. **Characterization and classification.** Topological features succinctly express qualitative characteristics. In particular, the number of connected components of an object is of importance for classification.
2. **Integration and simplification.** Topology is focused on global properties. From the topological perspective, a straight line and a circle are locally indistinguishable; however, they are not equivalent if they are considered as a whole. Topology offers a toolbox to integrate local information about an object into a global summary. Thus, topology can provide the researcher with a natural “big-picture” view of complex, multidimensional data.
3. **Features extraction.** Topological properties are stable. The number of components or holes is likely to persist under small perturbations or measurement errors. This is essential in data mining applications because real data are always noisy.

In the context of clinical research, the dataset under study is typically a table of outcomes in a particular clinical trial or study. The table rows correspond to individual participants in the clinical trial, and the columns contain information on specific outcome measures of interest, such as lab tests, vitals, questionnaires, etc. Given a table of clinical outcomes, two types of parameters are required to generate a graph using TDA. The first of these is a projection, a function that is used to stratify patients into subpopulations (bins). The second is a distance function, which measures the proximity between patients. The distance function makes it possible to connect the related patients with similar outcomes within each population.
To be considered for further analysis, a graph extracted from the dataset using TDA algorithms should meet certain requirements. Namely, it should:

- accurately represent the original dataset;
- eliminate the features of the dataset that are not relevant to the purpose of the study;
- reduce the complexity of the features that are shown on the data map; and
- be insensitive to small noise, such as errors of measurement.

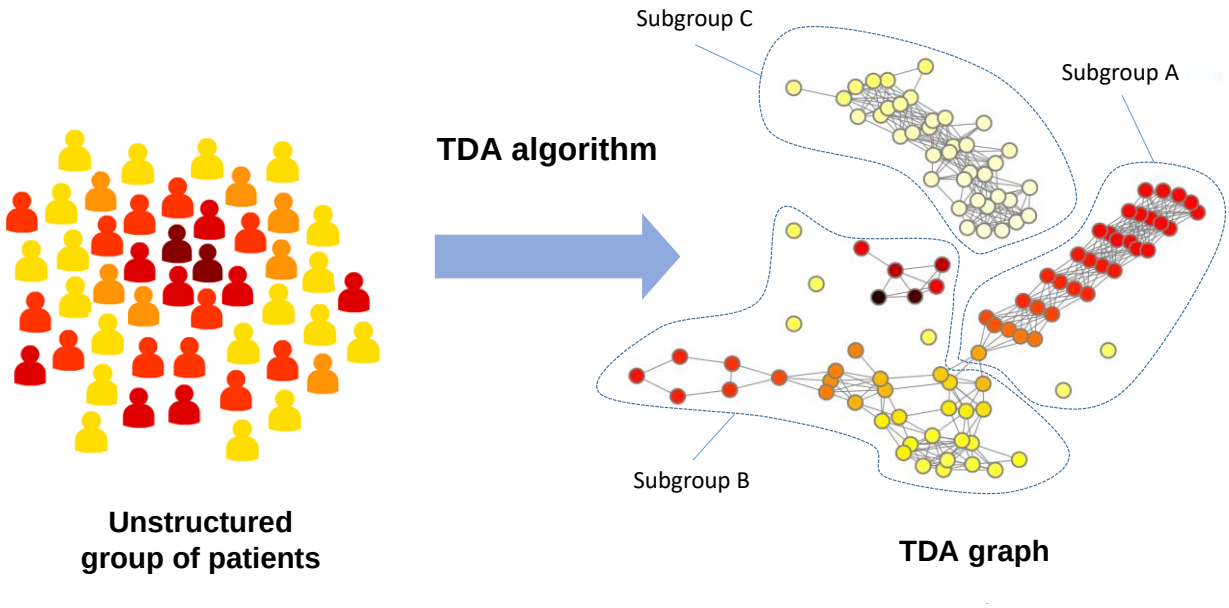


Figure 2. Discovery of multivariate patterns in clinical trial outcomes. The graph represents groups of patients structured according to the similarity of their outcomes

The core idea of clinical data mining using TDA relies on the visual discovery of subgroups of related patients in a graph (see Fig. 2) that presents the relevant information about the dataset in a compact and efficient manner. For the clinical dataset, the following criteria have to be met to perform the analysis:

- **Each node represents a patient**: a graph extracted from a clinical dataset is actually a graphical representation of the dataset in which each node represents an individual (trial subject).
- **Similar nodes are connected**: two nodes representing similar patients (in terms of a predefined set of clinical outcomes) are connected with an edge.
- **Coloring focused on specific outcomes**: the color of the nodes helps to highlight emerging patterns in the data and identify subgroups of patients related to the distribution of a variable of interest.
- **Visual discovery of subgroups**: clusters or "communities" of nodes on a graph reflect a segmentation of patients that may indicate robust patterns within the data.

TDA was successfully applied in the context of clinical studies, e.g., in the analysis of real-world data and understanding the spread of COVID-19 pandemic (see [3, 4]).

2. CLUSTERING AND COMMUNITY DETECTION AS PATTERN RECOGNITION

In many problems, it is convenient to represent data as a point cloud in a multidimensional space. The points in the point cloud often tend to form dense subgroups. In these subgroups the points are closer to each other than to the points from the rest of the point cloud. We call these dense subgroups *patterns* as they help to identify and exploit relationships of interest in the dataset. Searching for robust geometric patterns becomes one of the most important tasks in data analysis. In this section, we consider clustering and community detection methods as the way to search for such patterns.

2.1. Community detection, an overview

The modern approach to data science frequently employs graphs to enhance understanding of complex systems. A variety of problems can be represented and studied using graphs as described in Section 1. The key feature of a graph is a community structure, which relates to the way the nodes are organized in communities. Specifically, many edges connect nodes within the same community, while comparably few edges connect nodes between different communities [5, 6]. These communities can be considered to represent independent structures within the graph, and the detection of those independent communities is one of the key goals in the analysis of large graphs.

In graphs that represent real-world systems or data gathered in a study, the distribution of edges over subgroups of vertices is usually non-uniform. This reflects the possible presence of a hidden structure and patterns in the graph and, hence, in the data based on which the graph was created. Specifically, some groups of vertices may have high concentrations of intra-edges, while the concentrations of inter-edges between these groups of vertices may be low. The groups of densely connected vertices are referred to as communities. Figure 3 illustrates an example of a community structure within the graph that contains three groups of nodes (vertices) with dense internal connections within each group and comparably fewer connections between groups.

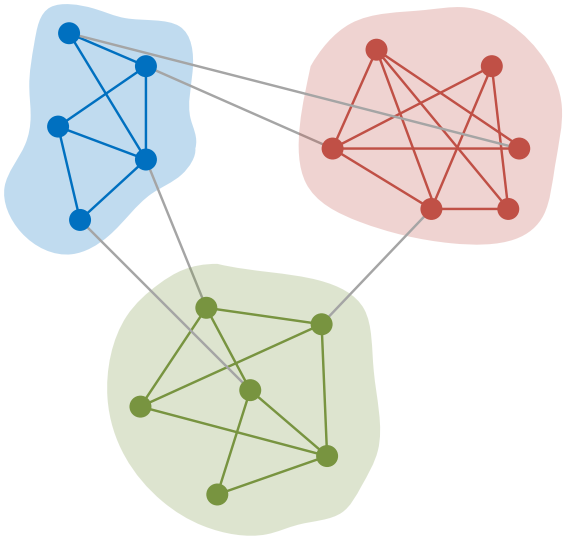


Figure 3. The schematic representation of the simple graph that has a community structure. The graph contains three communities of densely connected nodes with a much lower density of connections (gray edges) between the communities.

A number of algorithms have been developed for community detection (see [5, 6] for a survey), in particular:

- hierarchical methods, which build some dendrogram of communities by merging or splitting them, e.g., the **Girvan-Newman algorithm** described in Section 2.1.1. Such dendrograms can be cut at some level to communities; the number of communities depends on the cut;
- methods based on the specific nature of communities, e.g., methods for maximizing so-called *modularity*, assuming that communities should have structure different in some sense from that of a random graph;
- propagation methods assuming that some structure (like a clique in the **clique percolation method**, described in Section 2.1.2, or the most frequent neighbors label as in the label propagation method) "propagates" through the graph;
- random walk methods which are based on the assumption that the random walker spends more time inside a community and change communities with low probability.

In this paper, we give emphasis to the Girvan-Newman algorithm and the clique percolation method (with modifications made to these methods by the authors, see [7]) as the most efficient and most popular methods when applied for analyzing data.

2.1.1. The Girvan-Newman algorithm

The Girvan-Newman algorithm [8] attempts to identify the edges that are located "between" some pairs of nodes in the graph. In the algorithm, the distance between all pairs of nodes, i.e., the shortest edge-based path, is calculated. Such paths define the edge betweenness characteristic of the edges. The edge betweenness characteristic of an edge is the number of shortest paths between pairs of nodes that run along the edge.

The method of community detection using the Girvan-Newman algorithm is based on calculating the edge betweenness characteristic for all edges in the graph. The method includes steps of removing the edge having the highest edge betweenness characteristic and recalculating the edge betweenness characteristic for all edges affected by the removal. The steps are repeated until no edges remain. The edges that have the highest edge betweenness characteristic are the most "loaded" and, hence, are considered to lie the most "between" communities. The removal of the edges having the highest values of the edge betweenness characteristic from the graph results in the nodes falling into communities. The removal of the edges that have the further highest values of the edge betweenness characteristic separates further communities within the graph. See the main steps in Figure 4.

The Girvan-Newman algorithm has been widely applied to a variety of graphs, e.g., graphs of human and animal social networks, metabolic graphs, gene graphs, graphs representing collaborations between scientists and musicians, and so forth. However, this algorithm is computationally intensive and takes $O(m^2n)$ times for a graph with m edges and n nodes. In view of the large amount of time required to perform the calculations, the use of the algorithm is limited to graphs that contain less than a few thousand vertices. Furthermore, the algorithm does not show how many edges need to be removed to provide the most optimal community detection.

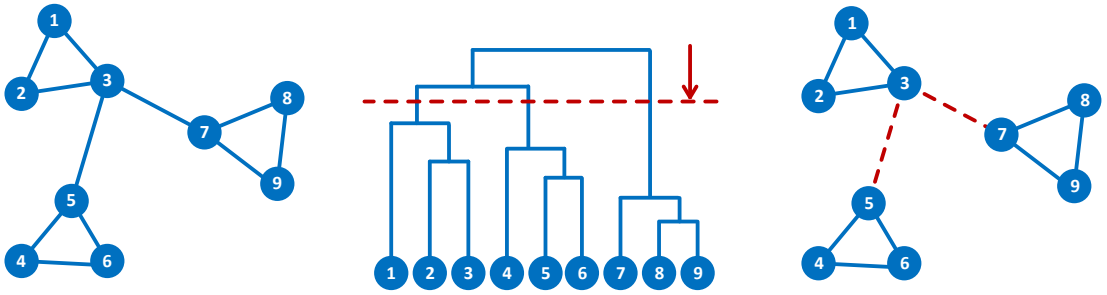


Figure 4. A hierarchical decomposition of the graph.
As one moves down the dendrogram, the detailed partitioning into communities starts to appear.

2.1.2. The clique percolation method

The clique percolation method [9] operates based on the assumption that internal edges within a community form *k*-cliques (i.e., subgraphs with *k* nodes in which every pair of nodes is connected by an edge) and edges that lie between the communities are not likely to form cliques.

The use of this method is based on the assumption that if a clique can "move" in the graph, the clique will get trapped inside the community and will not manage to pass in between two communities due to a lack of connecting paths. In this method, one clique can be "moved" to another if they share all but one node, and a community is defined as a maximal connected subgraph of the original graph so that each node in this graph belongs to some *k*-clique that lies entirely in the subgraph. The classical clique percolation method receives a value of *k* as an input and produces the list of all possible communities (as described above for the given value of *k*) as an output. The peculiarities of the method include the ability of some nodes to belong to several communities as several *k*-cliques may pass through these vertices and the ability of some nodes to occur out of communities as no *k*-clique contains them.

The example of the 3-clique percolation method can be seen at Figure 5: the 3-clique cannot pass through node 3, but the blue community and the green community overlap in node 3.

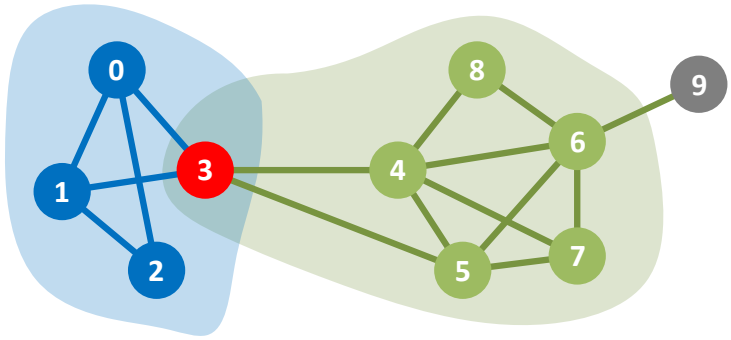


Figure 5. Example of overlapping community detection by the 3-clique percolation on a simple graph

Although theoretically the clique percolation method is computationally intensive because the detection of maximal cliques requires processing time that runs exponentially to the size of the graph, it was shown by the practical applications of this algorithm to real-world systems that this method works reasonably fast due to a limited number of cliques in real-world-based graphs.

2.2. Clustering, an overview

Considering the data as a point cloud in a multidimensional space usually assumes some (dis)similarity measure that can evaluate how close the points are to each other. The subgroups of "relatively close" points are referred to as *clusters*. Although there is no precise and conventional definition of clustering, the common point of different models is to understand clustering as partitioning a set of objects (points) into groups (clusters) in such a way that:

- objects in the same cluster must be similar as much as possible;
- objects in different clusters must be different as much as possible.

Usually, as a result of clustering, each point in the dataset is assigned to a cluster.
It can be noted that the same point cloud can be equipped with different (dis)similarity measures; they are usually chosen due to the dataset specification. Also, the understanding of what "as much as possible" is leads to a variety of clustering algorithms. Traditional clustering algorithms include [10]:

- partition methods (e.g., *k*-means and ***k*-medoids**), which perform the partition of the data into groups usually by grouping data points around a centroid (i.e. a "central point" of the dataset);
- hierarchical methods, which build a dendrogram of nested clusters by merging or splitting them;
- methods based on a model, which assume some specific knowledge on clusters nature, such as being sampled from the same distribution (e.g., a Gaussian mixture model) or high density (e.g., a **density-based spatial clustering of applications with noise (DBSCAN)** method).

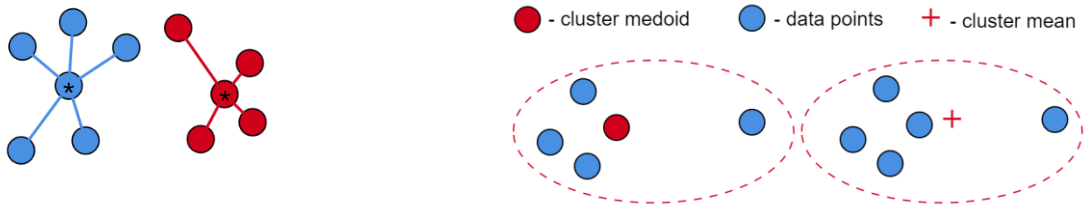
Modern approaches also involve kernel models, ensembles, spectral theories, specific algorithms for large-scale data, and so on.

In this paper, we give emphasis to the *k*-medoids method as the modification of the *k*-means method, which is one of the most popular methods in clustering, and to the DBSCAN method as the one that can find independent patterns in data and that does not require each point in the dataset to be assigned to a cluster. In contrast, the *k*-medoids method results in a complete partition of the dataset into non-overlapping clusters.

2.2.1. *k*-medoids clustering

k-medoids is a variant the classical *k*-means partitioning technique of clustering that splits the dataset of objects into *k* clusters, where the number *k* of clusters is assumed to be known *a priori*. In each potential cluster, the method of *k*-medoids chooses "the most central" representative in the cluster; this central point is called a *medoid*. Afterwards, the method tries to minimize the sum of dissimilarities from the medoid to the other points in the corresponding cluster (see Fig.6a). In contrast to the *k*-means algorithm, the *k*-medoids algorithm chooses actual data points as medoids, not the mean points of clusters as in the *k*-mean algorithm (mean points often do not lie in the dataset, see Fig. 6b).

The other difference is that *k*-medoids can use any dissimilarity measure, whereas in the *k*-means method only the Euclidean distance is used.



a) distances to the medoids (marked with *) b) mean and medoid in the cluster (dotted)

Figure 6. *k*-medoids clustering

To choose the parameter *k*, i.e., the applicable number of clusters, different criteria can be used. These criteria are usually some functions that provide scores of the quality of clustering depending on a given *k*. The value of *k* that optimizes the value of such a function (e.g., the maximal or minimal value) is then chosen as an optimal parameter for the method. Often, especially when the scoring functions are monotone in *k*, one searches for an elbow or a knee in order to determine the optimal *k.* These criteria will be discussed later in Sections 2.4 and 3.3.

2.2.2. DBSCAN clustering

DBSCAN is a clustering algorithm in which data points lying in the region with high density of the datapoint cloud are considered to belong to the same cluster [10]. To construct the cluster, DBSCAN finds the core points of high density (with enough neighbors inside a specified ε-ball) and expands clusters from the core points by adding the points which are *reachable* from the core point (i.e., within distance ε from the core point), as shown in Figure 7.

In contrast to the *k*-means/*k*-medoids methods, DBSCAN does not require specifying the number of clusters in the data *a priori*. It requires two parameters: a minimum number of points (minPts) in a neighborhood for a point to be considered as a core point (the minimum number *k* of neighbors plus the point itself); and the most important parameter ε which bounds the maximal distance needed to reach a neighbor. DBSCAN also has the notion of noise (i.e., not all points are assigned to clusters).

In this paper, to set the parameter minPts, we follow [11] and set it to twice the dimension of the data space (i.e., to 14, see Section 3). To set the parameter ε, it is usually recommended to detect the elbow/knee of the sorted *k*-distance plot or look for elbow/knee, min/max of some function, scoring the quality of clustering. For further discussion see Sections 2.4, 3.3.

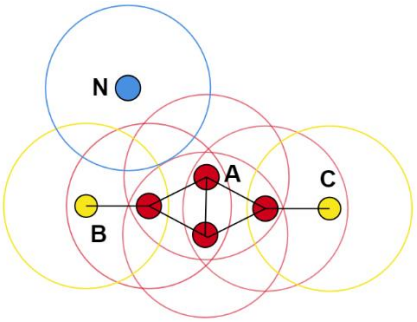


Figure 7. The DBSCAN algorithm: the red points are core points as their ε-balls contain at least 4 points, the yellow points are not core points but are directly reachable from the core points. The point N is an outlier that is neither a core point nor a directly-reachable point.

2.3. Clustering vs. community detection in clinical studies

In this subsection, we compare the chosen above clustering and community detection methods by their goals, prerequisites, and realization details.

Similarly to clustering, community detection is a fairly broad and undetermined problem with no universally accepted definitions of what a cluster or a community is. Community detection can be considered as a special case of clustering, in which there is a grouping of data (points) connected by some relationships. Graphs are used to represent such data, and communities in this case are represented as groups of nodes with close links within the communities and weak links between the communities.

Representing the community detection as a special case of clustering leads to the fact that the ideas of many clustering methods are transferred to the search for communities. In particular, as it was shown above, there are hierarchical methods for building communities, as well as methods based on the specific nature of communities. At the same time, the ideas related to community detection have an impact on the construction of general clustering methods, such as, e.g., graph-based models [12] and spectral graph theory [13].

Both clustering and community detection allow finding groups inside the data that are likely to share common properties and/or play similar roles within the data. Clustering algorithms can do this based on metric properties of the data, but topological properties of the graph are additionally used to identify communities. Hence, clustering cannot show the relationships between clusters and the position of the specific point inside the cluster, while community detection allows classification of nodes according to their structural position on the graph. Thus, nodes with a central position in their communities share the largest number of edges with other nodes within the community, which may indicate the important role they play in the stability of the community. On the other hand, nodes that are located at the boundaries between the communities may play an important role as mediators leading the relationships and exchange between different communities.

The majority of clustering methods form the partition of data points without intersections, i.e., all the points, even the noisy ones, are uniquely categorized to some cluster. This may lead to the "blurring" of clusters and difficulties in understanding what sense and common data properties show the clusters. Communities do not necessarily form a partition of nodes (in this case, "out-of-communities" nodes arise, which can be understood as outliers or noisy points) and, therefore, individual communities can be recognized for further analysis, independently of others. We consider this point to be particularly important for clinical studies, since it allows us to recognize separate homogeneous and well-defined groups of participants, which can be used, for example, to create effective treatment specifically for these groups, while the partition of all participants into groups (clusters) can lead to the appearance of "outlier" participants in clusters and can distort further processing of these outliers.

The last but not the least important advantage of community detection is the possibility to visualize the results as the graphs can be plotted in 2D. The results of clustering should be visualized by some projections of the data or dendrograms that usually misrepresent and distort the picture.

The results of this paper indicate that community detection methods demonstrate a more flexible approach to defining the patterns within the data, while the clustering methods are more general. In our experiment, we focus on detection of some "clear" patterns in data (corresponding to groups of participants/patients) and explore if community detection is more appropriate for this task than clustering.

2.4. Evaluation of the quality of community detection and clustering

Both clustering and community detection need methods for evaluation of accuracy of their results. The survey of evaluation methods can be found in [10] for clustering and in [5, 6] for community detection algorithms. They can be divided into two categories:

I. **external evaluation indicators** which measure if the separation of data given by a clustering (community detection) algorithm is similar to some ground truth separation or if the two separations are in a consensus or not;

II. **internal evaluation indicators** which do not assume any ground truth separation.

For the purposes of our experiment, we use the first group of indicators to compare the results of clustering and community search and to show that these methods provide different results. We use the second group of indicators to measure and compare the (absolute) quality of clustering and community detection methods. Note that in exploratory analysis of clinical data only the internal indicators can be applied because there is no ground truth grouping of the data.

2.4.1. External evaluation indicators

External evaluation indicators compare the results of a clustering algorithm with some external labeling. Ground truth class assignments, other clustering results or splitting dataset points into subsets can be used as such a labeling. We discuss several indicators to compare two clustering results.

Jaccard index: Jaccard index compares two clusters that are viewed as two sets of points. This index is computed as follows:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

where A and B are the sets of points in the clusters, $|A \cap B|$ is a number of common points, and $|A \cup B|$ is the total number of points in A, B. Note that $0 \leq J(A,B) \leq 1$ and $J(A,B) = 1$ in case $A = B$.

Rand index: To compare the partition of a set of points into clusters, Rand index (RI) and mutual information score can be used. It is calculated as follows [14]:

$$RI = \frac{TP + TN}{TP + FP + FN + TN}$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives while considering the first partition into clusters as the ground truth and the second – as prediction.

Rand index measures the similarity of two assignments, while ignoring permutation of cluster labels. Furthermore, it is symmetric: swapping the two clusterings does not change the scores. It can thus be used as a consensus measure. The score of well-matched clusterings is close to 1. It is good if the random labels of the clusters give a score of zero, but that is not always the case. The adjusted Rand index fixes this problem [14].

The mutual information is an information theoretic measure of how much information is shared between two clusterings. The adjusted mutual information score is normalized against chance. Values close to zero indicate two label assignments that are largely independent, while values close to one indicate significant agreement (with or without permutation).

2.4.2. Internal evaluation indicators

The most popular internal evaluation indicators for clustering algorithms are the *silhouette coefficient* and the *Davies-Bouldin indicator* [10]. The most used scores to evaluate the quality of the results of community detection are the *coverage*, *performance*, and *modularity* [5]. Originally, these scores were introduced for partitions of data into clusters/communities, i.e., when each data point is assigned to a cluster/community. We modify these scores to use them in the situation of a non-complete partition, i.e., when not all points in the data cloud/vertices of the graph are assigned to clusters/communities (see the discussion in Section 2.3). We use both clustering and community scores to evaluate the quality of *both* clustering and community detection results. This is done in order to control the bias of scoring in "native" methods (cluster scores for clustering, community scores for community detection). In order to implement this, we project the results of clustering algorithms onto the topological model of the dataset; in this case, we adopt the terminology a 'community' for a projected cluster.

In our experiment, we use the following 6 scores:

1. **Davies-Bouldin maximum and mean indicators.** These indicators compare the distance between clusters with the size of the clusters themselves and are defined as follows:

$$DB_{max} = \frac{1}{k}\sum_{i=1}^{k} \max_{i \neq j} R_{ij}\,, \qquad DB_{mean} = \frac{1}{k}\sum_{i=1}^{k} \operatorname{mean}_{i \neq j} R_{ij}, \qquad R_{ij} = \frac{s_i + s_j}{d_{ij}},$$

where s_i is the average distance between points in the i-th cluster and the medoid of that cluster (i.e., the point with minimum average distance to other points of the cluster), d_{ij} is the distance between the medoids of the i-th and j-th clusters, and k is the number of clusters.

2. **Silhouette coefficient.** This coefficient measures how similar an object is to its own cluster compared to other clusters. It is defined as the mean of the silhouette coefficients for each point. For a given point, the coefficient is defined as follows:

$$s = \frac{b - a}{\max(a, b)},$$

where a is the mean distance between the given point and all other points in the same cluster and b is the mean distance between the given point and all other points in the next nearest cluster.

For both Davies-Bouldin maximum and mean indicators, lower values relate to a model with better separation between the clusters and the silhouette coefficient is higher when clusters are dense and well separated.

3. **Weighted density score.** This score is defined as follows:

$$wd = \sum_{j=1}^{k} \frac{n_j}{n} d_j \,,$$

where n_j is the number of nodes in the j-th community, n is the total number of nodes, and d_j is the ratio of intra-community edges to the total number of node pairs in j-th community.

4. **Performance score.** This score is defined as follows:

$$p = \frac{intra + out_{non} + inter_{non}}{\frac{n(n-1)}{2}},$$

where n is the total number of nodes, $n(n-1)/2$ is the total number of pairs of nodes in the graph, $intra$ is the total number of intra-community edges, $inter_{non}$ is the total number of intra-community non-edges (i.e., pairs of vertices in a community that are not connected by an edge), and out_{non} is the total number of non-edges connecting out-of-community nodes (i.e., pairs of nodes one of which lies in the community while the other lies outside of the community).

5. **Modularity.** This score is defined as follows:

$$Q = \sum_{c} \left(\frac{L_c}{m} - \left(\frac{k_c}{2m} \right)^2 \right),$$

where m is the total number of edges, we sum over all communities c, L_c is the total number of intra-community edges of the community c, and k_c is the sum of degrees of the nodes in the community c.

For communities that form the partition of the set of nodes, the above-defined performance score and modularity coincide with their classical prototypes. The higher values for weighted density, performance, and modularity correspond to better selection of communities, and the maximum possible value for all of them is 1.

6. **Distortion score**. This score is the total sum of all squared distances from the cluster centroids. We use the maximal and the average distortion values over all clusters/communities for a given algorithm and the average of the standard deviation (a score based on distortion that penalizes big clusters) as scores:

$$D(C) = \sum_{u \in C} \big(d(u, m)\big)^2 \qquad std(C) = \frac{1}{|C|-1} \sqrt{D(C)} \,,$$

$$\max D = \max_{i \in \{1,\dots,k\}} D(C_i), \quad av\ D = \frac{1}{k} \sum_{i=1}^{k} D(C_i), \quad av\ std = \frac{1}{k} \sum_{i=1}^{k} std(C_i) \,,$$

where C and C_i are clusters, $|C|$ is the size of a cluster C, $d(u, m)$ is the distance between $u \in C$ and centroid m of the cluster C, and k is the total number of clusters.

The distortion score as a cost function is often used in order to find an optimal number of clusters in *k*-means type algorithms [15].

3. EXPERIMENT

We took the 839-point dataset (767 points after preprocessing) to test the most popular clustering and community detection algorithms (on a corresponding TDA-based data graph) in order to find non-trivial hidden patterns within the data. We define optimal parameters and applicability of algorithms and evaluate efficiency of *k*-medoids, DBSCAN clustering algorithms, the clique percolation method for community detection, and the Girvan-Newman community detection algorithm. Finally, we compare the performance results and select the algorithm with the best scores for further investigation and interpretation.

3.1. Data description and design of the experiment

The experiment considers two techniques used to discover the meaningful patterns in large and complex datasets:

1. Clustering, which became the classics of unsupervised machine learning; and

2. Modern community detection algorithms on graphs built by TDA.

The goal of the experiment is to show that community detection on a TDA-graph can reveal statistically significant and clear patterns different from those revealed by clustering and to understand if the community detection can be used to improve the quality of clustering.

The original dataset to perform the experiment was taken from NIDA research study of nicotine smokers (CENIC P1S1) [16]. The study was a double-blind, parallel, randomized clinical trial between June 2013 and July 2014 at 10 sites. Eligibility criteria included an age of 18 years or older, smoking of five or more cigarettes per day, and no current interest in quitting smoking. Participants were randomly assigned to smoke for 6 weeks either their usual brand of cigarettes (*usual brand*) or one of six types of investigational cigarettes (*study cigarettes*), provided free. The investigational cigarettes had nicotine content ranging from 15.8 mg per gram of tobacco (typical of commercial brands) to 0.4 mg per gram. The primary outcome was the number of cigarettes smoked per day (*CPD*) during week 6 and the goal was to evaluate the relationship between nicotine yield of very low nicotine content cigarettes smoked per day, nicotine exposure, discomfort/dysfunction and other health-related behaviors, nicotine/tobacco dependence, biomarkers of tobacco exposure, intention to quit, compensatory smoking, other tobacco use, cigarette characteristics, cognitive function, cardiovascular function, and perceived risk. The study was provided on 839 participants for 6 weeks, preceding by the baseline data gathering and concluded by a follow-up study.

The results of the original study [17] showed that during week 6, the average number of cigarettes smoked per day was [significantly] lower for participants randomly assigned to cigarettes containing 2.4, 1.3, or 0.4 mg of nicotine per gram of tobacco than for participants randomly assigned to their usual brand or to cigarettes containing 15.8 mg per gram. Participants assigned to cigarettes with 5.2 mg per gram did not reveal significant increasing of the number of cigarettes from the average number among those who smoked control cigarettes. Cigarettes with lower nicotine content, as compared with control cigarettes, reduced exposure to and dependence on nicotine, as well as craving during abstinence from smoking, without significantly increasing the expired carbon monoxide level or total puff volume, suggesting minimal compensation. Adverse events were generally mild and similar among groups.

Opposite to the original study, in our experiment on the CENIC dataset we select the outcomes to be the number of cigarettes smoked per day during a baseline week and all the weeks from 1 to 6 to see the dynamics of cigarettes smoked. The other difference is that we do not primarily divide the participants into treatment groups and investigate the relationships in the dataset as a whole.

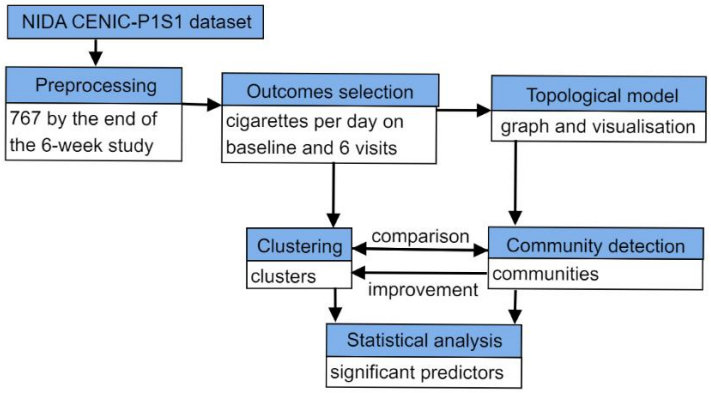


Figure 8. The workflow of the experiment

The overall workflow of the experiment is presented in Figure 8. Table 1 illustrates the order of steps in the experiment.

Table 1. Steps in the experiment

1.	Data **preprocessing**	Select 767 participants that have completed the 6-week study
2.	Defining **outcomes**	Define the average number of cigarettes smoked per day on the baseline week and during every week in the 6-week study. 7 outcomes are defined in total
3.	Building a **graph**	Based on the defined outcomes, apply Topological Data Analysis to build a graph revealing the geometric properties of the dataset and hidden interdependencies
4.	Detecting **communities**	Detect the communities on the graph by the Girvan-Newman and clique percolation algorithms. Pick up the best parameters for each method
5.	Performing **clustering**	Perform clustering with the *k*-medoids and DBSCAN algorithms. Pick up the best parameters for each method
6.	**Comparison** of clusters and communities	Evaluate the results of the clustering and community search and compare the patterns found by the algorithms
7.	**Statistical analysis**	Perform a statistical analysis of discovered patterns, identify significant predictors
8.	Community detection for **improving** the clustering	Exclude non-community nodes to perform new clustering to improve the quality of original clustering

3.2. Building the model using topological data analysis

After preprocessing the data of the original CENIC, 767 subjects are selected into the dataset of our experiment (see Section 3.1). The number of cigarettes smoked per day (**CPD**) are taken as **outcomes** of interest. A 7-dimensional row-vector with CPD rates evaluated at the baseline and through the weeks 1-6 is assigned to each subject (see Figure 9). This vector represents the smoking behavior of the subject.

$$\text{person} = \boxed{CPD_0}\boxed{CPD_1}\boxed{CPD_2}\boxed{\dots}\boxed{CPD_6}$$

Figure 9. CPD_0 is the CPD rate at the baseline, $CPD_1, \dots, CPD_6$ are the CPD rates at weeks 1-6, respectively.

The dissimilarity between participants is measured by a combined metric. This metric was designed to capture the difference in smoking trends (increasing or decreasing) as well as the mean values of CPD smoked during weeks 0-6 (0 stands for the baseline). The metric is composed of the **correlation distance** and the absolute value of **difference between the CPD means.**

I. Correlation distance is computed as follows:

$$d_c(u,v) = 1 - \frac{(u-\bar{u})\cdot(v-\bar{v})}{\|(u-\bar{u})\|_2\|(v-\bar{v})\|_2},$$

where $\bar{v}$ is the mean values of the elements of a vector v, $\|\ldots\|_2$ is the standard Euclidean distance, and $a \cdot b$ is the dot product of vectors a and b.

The correlation distance ranges from 0 to 2 and measures proximity of directions of trends. Thus, it is sensitive to the slopes of the time series. However, it is centered, so it is not sensitive to the shifts and could not distinguish the starting points of the trends (the baseline CPD rate) and thus the mean CPD values during the visits. The latter explains why the following additional metrics are needed.

II. The absolute value of difference between CPD means is defined as follows:

$$d_m(u,v) = 2\left|\frac{\bar{u}-\bar{v}}{\max\limits_{a,i} a_i}\right|,$$

where max is taken over all coordinates a_i of all vectors a of the dataset (i.e., it is maximum CPD rate observed). Means are normalized in such a way that d_m ranges from 0 to 2. It is done in order to balance d_m contribution with the one provided by the correlation distance d_c.

We define the **combined metric** as follows:

$$d_{cm}(\text{🧍}, \text{🧍}) = d_c(\text{🧍}, \text{🧍}) + d_m(\text{🧍}, \text{🧍})$$

Figure 10 shows the result of visualization of our dataset with combined metric using the multidimensional scaling (MDS) projection. Recall that the MDS projection fits a multidimensional dataset into a 2-dimensional plane trying to maintain the distances between points as much as possible. By the construction of the combined metric, the subjects with similar smoking behavior dynamics should correspond to points lying nearby on Figure 10.

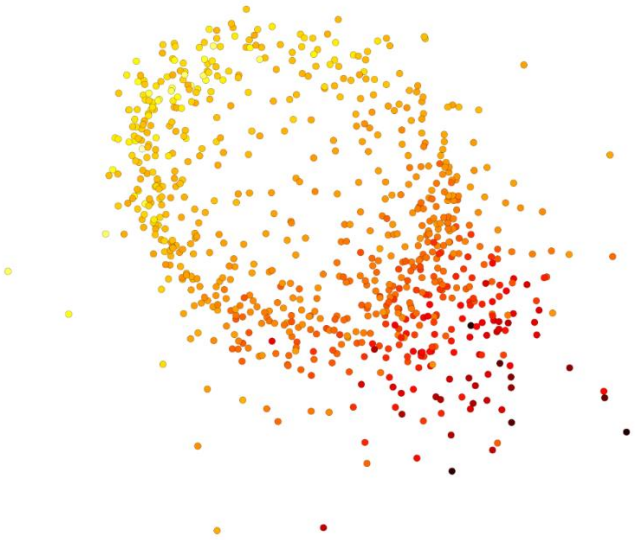


Figure 10. The MDS projection of 767 vector-points fitting the combined metric.
The points are colored by deviation of the mean CPD rate from the baseline CPD rate.

Legend: *smoke less* - - *smoke more*

The methods of Section 1 were applied to the experimental dataset to build a topological model (TDA graph). The combined metric described above and a projection based on density of points in the data cloud with respect to the selected outcomes were used. The resulting graph is shown in Figure 11. In this graph,

- each node represents a single CENIC-P1S1 participant (the total of 767 nodes);
- two nodes are connected with an edge if the participants have similar smoking behavior.

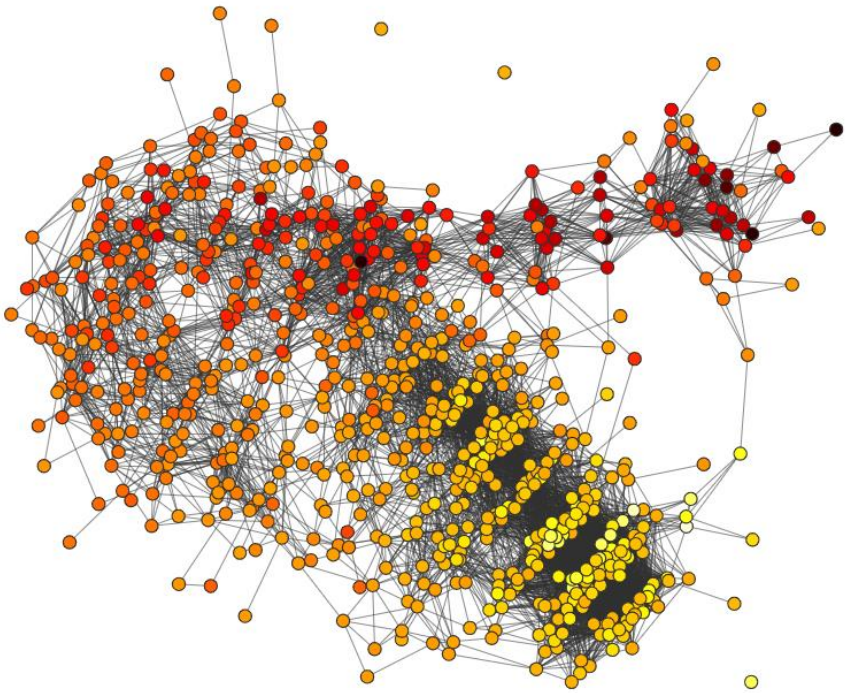


Figure 11. The topological model (TDA graph) of the experimental dataset with 767 preselected participants of the CENIC-P1S1 study based on preselected outcomes.
The coloring reflects the deviation of the mean CPD rate from the baseline CPD rate.

Legend: *smoke less* -  - *smoke more*

3.3. Clustering and community detection on the TDA graph

In Section 2.4, some general scores for evaluation of the quality of clustering and community detection methods were introduced. In this subsection, we use those scores with respect to the combined metric as described in the previous subsection. Furthermore, we add into consideration several more purely data-based scores.

Considering vector representation of smoking behavior of each subject (i.e., the CPD time series), the coordinate-wise descriptive statistical information (e.g. mean, std, sem, outliers) can be considered for each cluster or community C. The outliers are counted via an interquartile range within C and their portion in C is calculated. The **av outliers** is the additional score which measures the average portion of outliers over all clusters/communities for a given algorithm. As described in Section 2.4.2, we define **av distortion cmdm** and **av std cmdm** to be the average distortion and std scores taken over all C's for which they were calculated (with respect to the combined metric). We additionally define the scores **av distortion** and **av std** that are calculated in exactly the same way as **av distortion cmdm**/**av std cmdm** but using the standard Euclidean metric on vectors instead of the combined metric.

Upon testing various clustering algorithms on the dataset in question, it is concluded that most of methods find two expected communities/clusters of those subjects with increasing CPD rate and those participants who smoke less (decreasing CPD rate). However, we were interested in finding nontrivial hidden patterns that could show different behavior or further stratify the two main trends. For this purpose, the following assumptions were made:

A. The total proportion of discovered patterns should exceed 50% of the data (filtered "noise" cannot exceed the maximum of 383 points).

B. There must be more than two communities/clusters of essential size, communities/clusters of size less than 5 subjects are considered too small and are regarded as noise.

In order to choose the methods' optimal parameters (see Section 2), namely,

- the proximity parameter *eps* for DBSCAN
- the number *k* of clusters for the *k*-medoids
- the size k of the clique in the clique percolation
- the *number of iterations* in the Girvan-Newman algorithm,

we consider local maxima/minima or knees/elbows of the evaluating scores under the reasonable restrictions A and B mentioned above. Then, by the majority of the votes cast by all the scores (+1 point for the best score value, -1 point for the worst and ± half point for local maximum/minimum or knee/elbow value) the optimal parameter for the corresponding algorithm is chosen.

Our results on optimal clustering and community detection on the built TDA graph are described in the following four subsections.

3.3.1. Communities by the clique percolation method

To distinguish communities by the clique percolation method, we pick the size of the clique to be 8, see Figure 12.

The corresponding communities and their CPD rate trends are shown in Figure 13 (the trends are computed by averaging point-wise trends in the corresponding community, i.e., by taking the mean trend).

The found communities contain in total 409 nodes, the total number of out-of-community noise nodes is 358. All trends but two show the increase of CPD rate stratifying the smokers based on the baseline CPD value. Trends are clean with small std and sem (the whiskers on Fig. 13c). The only decreasing trend is in the biggest **0-olive** community which smoked about 14 CPD at the baseline and reduced down to 10 CPD. Finally, the **9-orange** community shows non-monotonic behavior – with a CPD rate drop on the first 3 visits and with increase on the last 3 visits up to their baseline rate. As it will be shown, this community is meaningful and not accidentally formed (see discussion in Section 3.5).

3.3.2. Communities by the Girvan-Newman algorithm

To detect the number of iterations and thus the optimal number of Girvan-Newman communities the following procedure is accomplished. For each iteration step only communities of essential size are left (the rest are considered as noise). Then, all the scores are evaluated and contribute by voting for a parameter.

The iteration parameter 10 is chosen by simultaneous improvement of 7 scores (leaps up/down respectfully), see Figure 14. This choice results in 8 communities of essential size shown with the community mean trends in Figure 15.

Although the communities found by the Girvan-Newman algorithm form the partition of dataset and each node belongs to a community, there are still 12 'noise' nodes (communities of size < 5 nodes) and 755 essential communities' nodes. The two major communities (**1-pink** and **2-brown**) contain participants with increasing CPD trend and another big one (**0-green**) with decreasing CPD rate dynamics. One can observe that **3-purple** Girvan-Newman community is similar to **9-orange** community found by the clique percolation method.

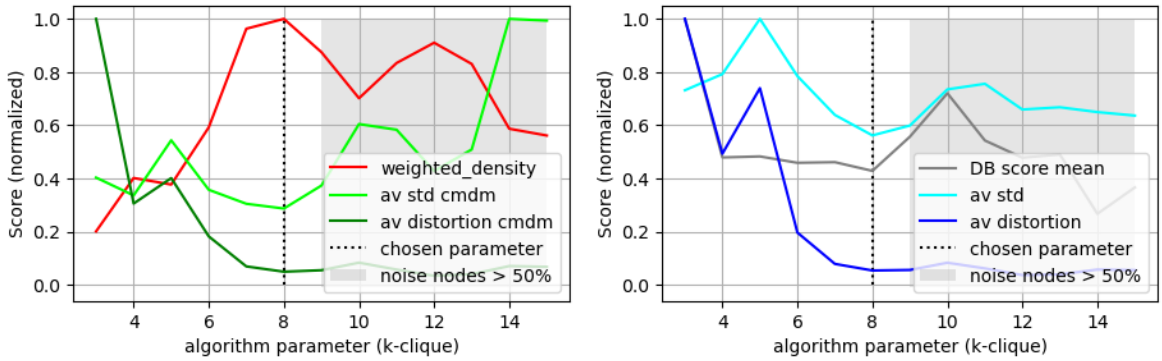


Figure 12. Choosing the optimal size of the clique in the clique percolation method

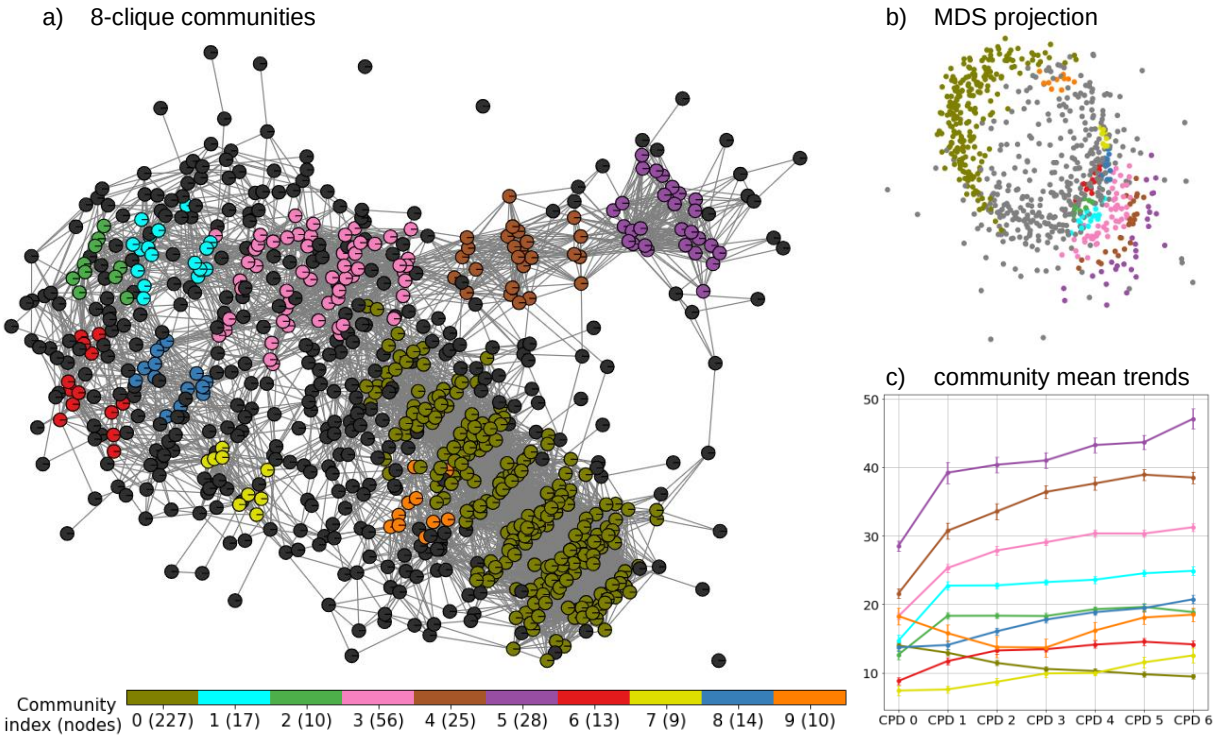


Figure 13. The output of the clique percolation method with the chosen optimal parameters

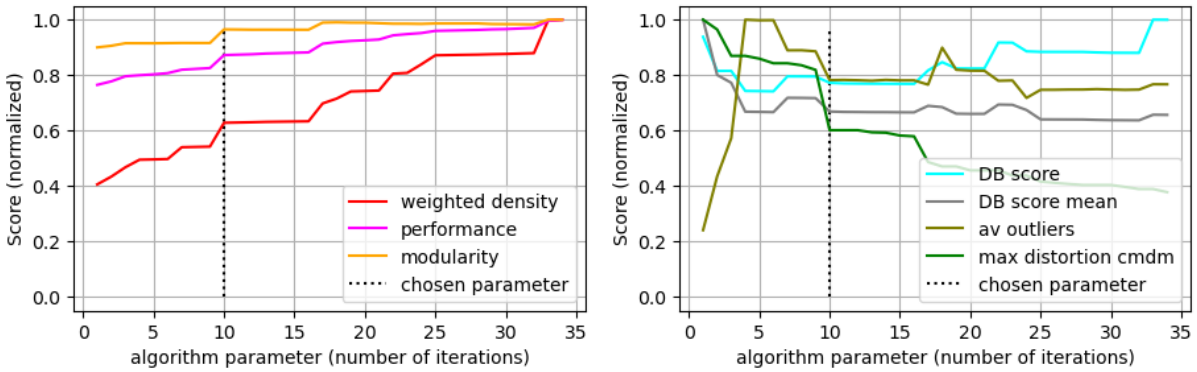


Figure 14. Choosing optimal parameters in the Girvan-Newman algorithm

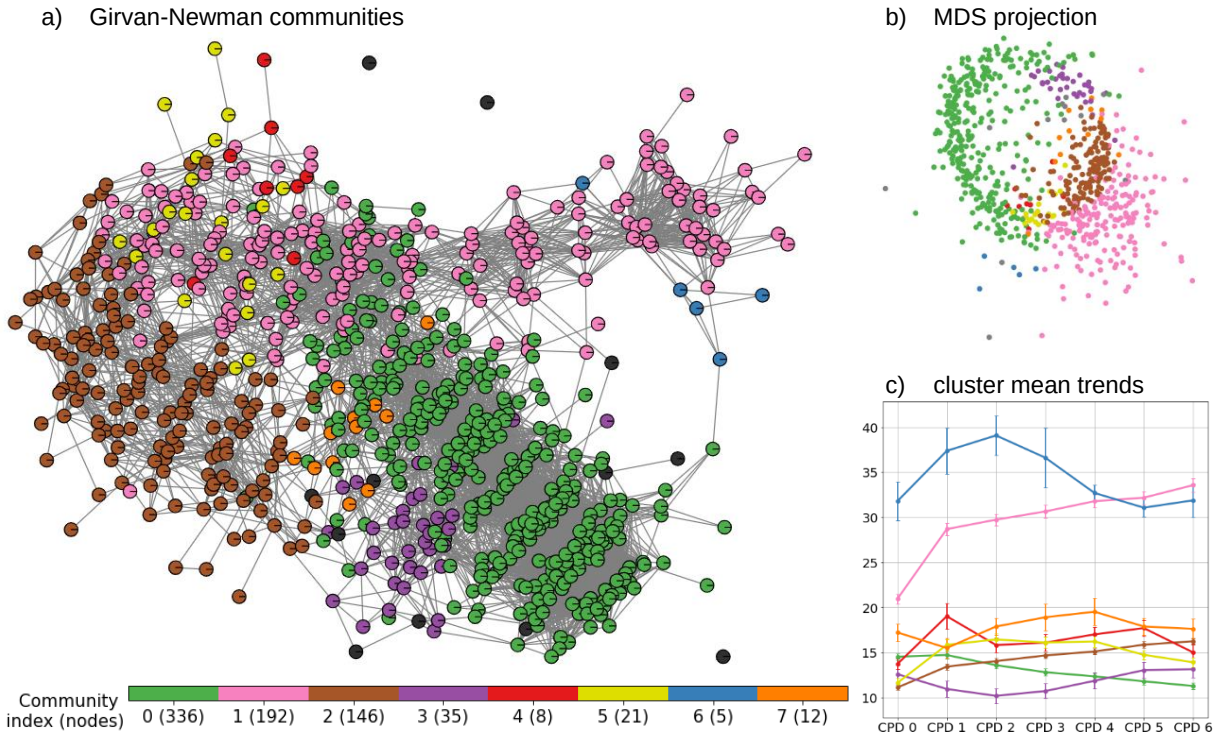


Figure 15. The output of the Girvan-Newman algorithm with the chosen optimal parameters

3.3.3. Clustering by DBSCAN

In our experiment with DBSCAN, it was observed that it has a narrow segment of parameters for reasonable clustering of the dataset. This algorithm finds either several clusters with lots of noise points (>50%) or very few sufficiently large clusters. The only borderline appropriate DBSCAN clustering that could be observed recognizes 3 patterns. As was mentioned before, the *minPts* parameter is taken to be 14 (twice the dimension of the vector space). The DBSCAN clustering parameter *eps* is taken to be 0.1956: this value simultaneously establishes the maximum of

the weighted density score, the local maximum of modularity, and close to the best (min) values of the DB score, DB score mean, av std, av distortion, av std cmdm and av distortion cmdm (see Figure 16).

The results of DBSCAN clustering can be seen in Figure 17.

The DBSCAN clustering with optimal parameters resulted in 3 clusters with 386 nodes and 381 out-of-cluster noise nodes. The two major clusters refer to the ascending and descending trends (**0-orange** and **1-blue**, respectively) and the third small **2-green** cluster which can be thought as a cluster with undecided trends. In most of our experiments, DBSCAN was unable to recognize more than two big clusters in the dataset.

3.3.4. Clustering by *k*-medoids.

The *k*-medoids clustering is performed with 6 clusters since this number simultaneously establishes the best (uppermost) modularity and DB score (the least) with a local minimum of the DB score mean, as indicated in Figure 18. All other scores abstain from voting (as being neither for, nor against this parameter value), while the other number-of-clusters-candidates have received fewer votes.

The results of *k*-medoid clustering are shown in Figure 19.

The clustering partitioned the dataset into two (**5-yellow**, **3-purple**) clusters with strictly increasing CPD mean trends, one (**0-green**) with monotone decreasing CPD mean rate, and the rest of the clusters with fluctuating CPD curves. In contrast to Girvan-Newman partition results, all *k*-medoids' clusters are of essential size and it is the only method in our experiment which gave a partition of the dataset without noise points.

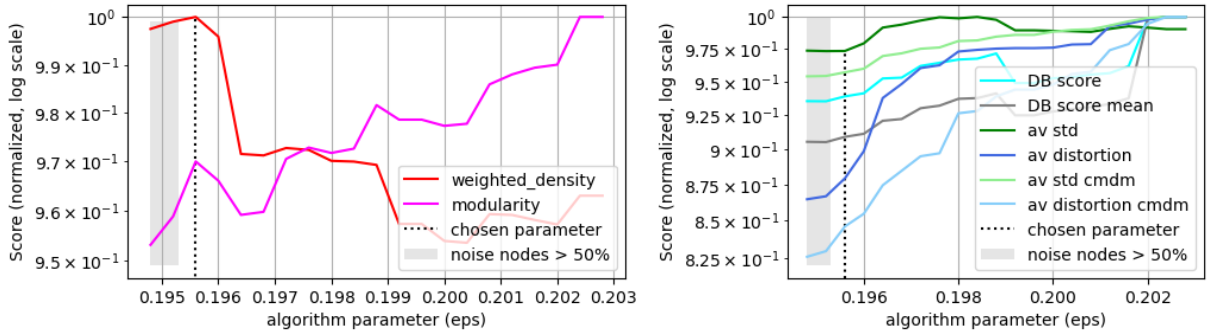


Figure 16. Choosing optimal parameters in the DBSCAN method

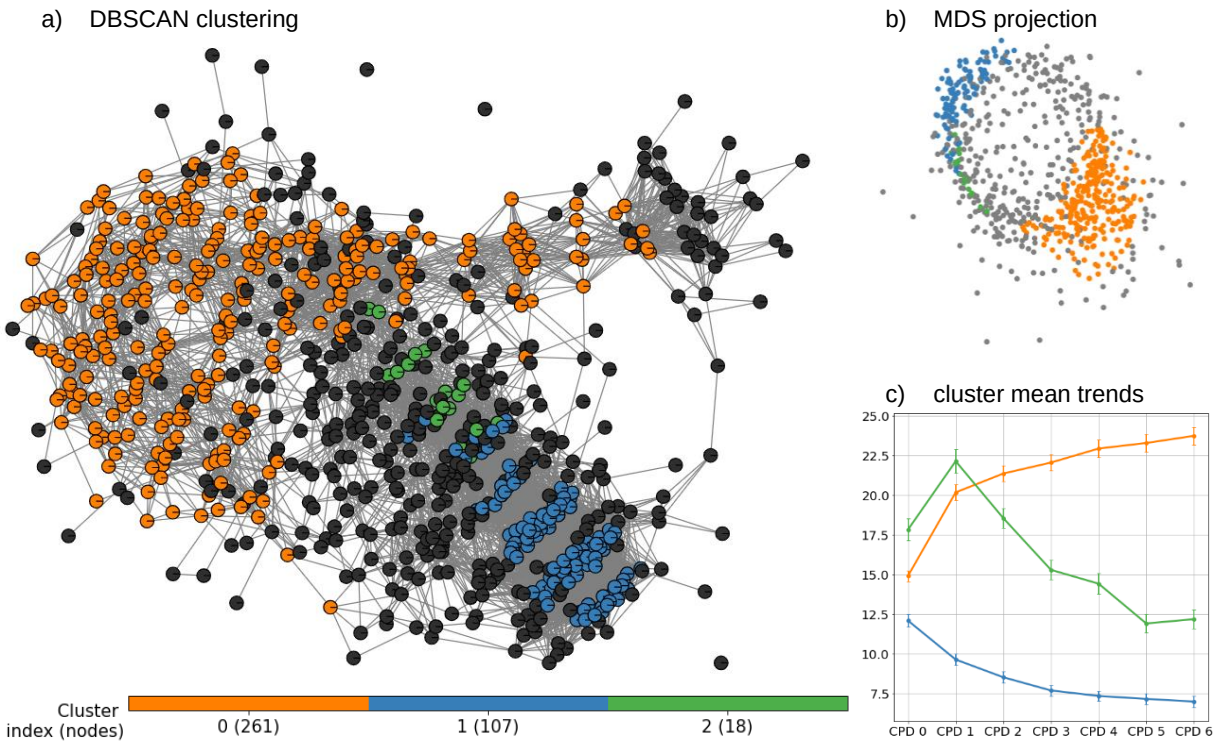


Figure 17. The output of the DBSCAN algorithm with the chosen optimal parameters; projected on the topological model in a)

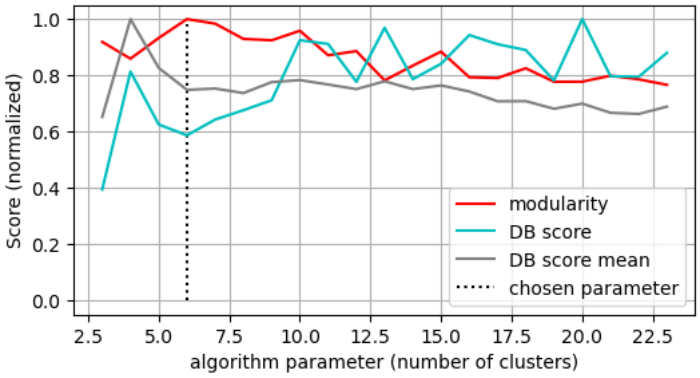


Figure 18. Choosing optimal parameters in the *k*-medoids clustering method

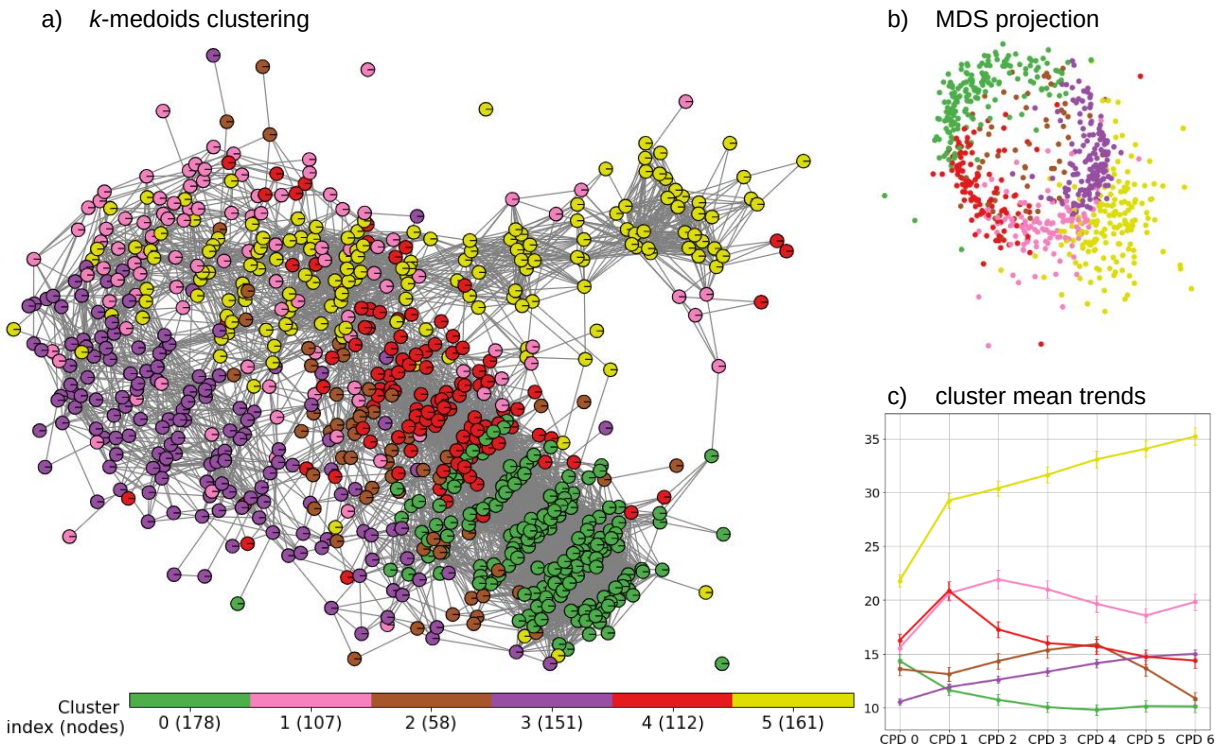


Figure 19. The output of the *k*-medoids clustering with the chosen optimal parameters; projected on the topological model in a)

3.4. Comparison of clusters and communities

3.4.1. Are the clusters and communities different?

The comparison of the pattern recognition results obtained in the previous subsection is shown in Table 2. For this, we use external indices *Adjusted Rand index* (ARI) and *Adjusted mutual information* (AMI) (see Section 2.4.1). The maximal index value of 1 indicates the maximal similarity of the resulted clustering/community search. A random splitting will give a value close to 0. We observe that in our experiment, the similarity of partitions is low, i.e., lower than 0.5.

Table 2: Comparison using external indices

ARI	**Percolation**	**Girvan-Newman**	***k*-medoids**	**DBSCAN**
Percolation	1.000			
Girvan-Newman	0.241	1.000		
***k*-medoids**	0.211	0.396	1.000	
DBSCAN	0.211	0.193	0.150	1.000

AMI	**Percolation**	**Girvan-Newman**	***k*-medoids**	**DBSCAN**
Percolation	1.000			
Girvan-Newman	0.413	1.000		
***k*-medoids**	0.383	0.450	1.000	
DBSCAN	0.355	0.318	0.319	1.000

Table 3 shows the *Jaccard index* for the results of the clique percolation method compared to the results of the clustering methods. The rows of the table correspond to the found communities, and the columns correspond to the found clusters. The Jacquard index for comparing percolation and *k*-medoids shows that the zero cluster and zero community have the greatest similarity (compare Figures 13a and 19a). The fifth *k*-medoids cluster broke up mainly into three communities: the fourth and the fifth. The first DBSCAN cluster is similar to the zeroth community.

Table 3: Jaccard index for percolation vs. clustering

k-medoids	0	1	2	3	4	5
0	0.594	0.000	0.040	0.005	0.228	0.000
1	0.000	0.051	0.000	0.000	0.000	0.066
2	0.000	0.035	0.000	0.032	0.000	0.006
3	0.000	0.019	0.000	0.005	0.000	0.315
4	0.000	0.008	0.000	0.000	0.000	0.148
5	0.000	0.008	0.000	0.000	0.000	0.167
6	0.000	0.000	0.000	0.086	0.000	0.000
7	0.000	0.000	0.000	0.060	0.000	0.000
8	0.000	0.000	0.000	0.093	0.000	0.000
9	0.039	0.000	0.000	0.019	0.000	0.000

DBSCAN	0	1	2
0	0.000	0.471	0.079
1	0.065	0.000	0.000
2	0.038	0.000	0.000
3	0.210	0.000	0.000
4	0.083	0.000	0.000
5	0.021	0.000	0.000
6	0.050	0.000	0.000
7	0.034	0.000	0.000
8	0.054	0.000	0.000
9	0.000	0.000	0.000

A similar comparison of the Girvan-Newman communities and the clustering methods is shown in Table 4. The *k*-medoids and Girvan-Newman algorithms have some similar subsets, e.g., 0 community and 0 cluster, community 2 and cluster 3, 1 and 5 (compare Figures 15a with 17a and 19a).

Table 4: Jaccard index for Girvan-Newman vs. clustering

k-medoids	0	1	2	3	4	5
0	0.460	0.067	0.101	0.012	0.284	0.010
1	0.000	0.141	0.000	0.015	0.017	0.697
2	0.003	0.091	0.010	0.623	0.004	0.023
3	0.070	0.000	0.011	0.114	0.007	0.000
4	0.000	0.000	0.031	0.006	0.043	0.000
5	0.000	0.133	0.068	0.006	0.000	0.000
6	0.000	0.037	0.000	0.000	0.009	0.000
7	0.000	0.008	0.094	0.025	0.000	0.006

DBSCAN	0	1	2
0	0.003	0.318	0.054
1	0.429	0.000	0.000
2	0.366	0.000	0.000
3	0.003	0.000	0.000
4	0.007	0.000	0.000
5	0.041	0.000	0.000
6	0.000	0.000	0.000
7	0.000	0.000	0.000

Finally, Table 5 shows the Jaccard index comparison of the results of the clique percolation method and the Girvan-Newman algorithm (compare Figures 13a and 15a). The rows of the table correspond to the resulting communities of

the percolation method, and the columns correspond to the found Girvan-Newman communities. Zero communities are similar, just like the ninth percolation community is similar to the third Girvan-Newman community. The first Girvan-Newman community splits mainly into three percolation communities: the third, the fourth and the fifth.

Table 5: Jaccard index for percolation vs. Girvan-Newman

	0	1	2	3	4	5	6	7
0	0.676	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1	0.000	0.089	0.000	0.000	0.000	0.000	0.000	0.000
2	0.000	0.015	0.047	0.000	0.000	0.000	0.000	0.000
3	0.000	0.292	0.000	0.000	0.000	0.000	0.000	0.000
4	0.000	0.130	0.000	0.000	0.000	0.000	0.000	0.000
5	0.000	0.146	0.000	0.000	0.000	0.000	0.000	0.000
6	0.000	0.000	0.089	0.000	0.000	0.000	0.000	0.000
7	0.000	0.000	0.062	0.000	0.000	0.000	0.000	0.000
8	0.000	0.000	0.096	0.000	0.000	0.000	0.000	0.000
9	0.000	0.000	0.000	0.286	0.000	0.000	0.000	0.000

3.4.2. Evaluating the models: the final results

Upon optimally detected communities on the graph and optimal clustering of the dataset with each method, it is natural to compare the overall results. The choice of optimal parameters for each of the algorithms guarantees that each method delegates the best of its representatives. Table 6 shows how these representatives compete with each other according to the 13 previously selected scores (the **av distortion, max distortion** and **av distortion cmdm, max distortion cmdm** are presented additionally normalized by the total number of points in the dataset, i.e., divided by 767).

	Weighted density	**Performance**	**Modularity**	**Silhouette score**	**DB score**	**DB score mean**	**av std**	**av distortion**	**max distortion**	**av outliers**	**av std cmdm**	**av distortion cmdm**	**max distortion cmdm**
Percolation	**0.212**	**0.917**	0.363	0.232	1.386	0.586	**9.136**	**9.275**	**65.72**	0.254	0.474	**0.0180**	**0.104**
Girvan-Newman	0.115	0.734	0.472	0.050	1.299	0.729	13.15	41.59	157.01	0.298	0.615	0.0666	0.307
DBSCAN	0.100	0.867	0.330	**0.534**	**0.521**	**0.410**	12.82	57.44	158.51	**0.121**	**0.437**	0.0449	0.114
***k*-medoids**	0.125	0.837	**0.529**	0.243	1.258	0.817	18.04	59.43	123.86	0.191	0.634	0.0672	0.127

Table 6: Community and clustering scores, the final results
Legend: **Hot palette marker** highlights "the larger-the better" scores
Cold palette marker highlights "the smaller-the better" scores
Scores are compared column-wise

From Table 6 it could be seen that the percolation method has the majority of the best scores (7 out of 13). The second place takes DBSCAN (5 of 13). The *k*-medoids clustering wins only in the modularity score. Although Girvan-Newman algorithm does not show any overall best scores, it should be noted that Girvan-Newman has better DB score mean, av std, av distortion, av std cmdm, av distortion cmdm values, if compared with *k*-medoids.

Thus, the clique percolation community detection algorithm used in TDA wins the battle. This algorithm showed itself better in identifying clean patterns, which is verified by **av std, av distortion, max distortion**, **av distortion cmdm,** and **max distortion cmdm** scores. This algorithm was able to produce clean and sufficiently stratified mean trends (see Figure 13c) that reflects the nature of the experiment. In the next section we will perform statistical analysis of the patterns and trends unveiled by the clique percolation method.

3.5. Results, interpretation, and discussion

In this subsection, we incorporate various predictors in order to explain the patterns found by the clique percolation method. Figure 13 indicates that the percolation communities differ by the outcomes. It also suggests that the discovered patterns cannot be further explained by those predictors that either correlate with the selected outcomes or are intrinsically dependent on the CPD rates. For instance, an increase of expired carbon monoxide (ppm) is observed for communities with increase of CPD rate (e.g. 3, 4, 5 communities) and a decrease for the 0-olive community with decreasing CPD rate. We will concentrate on other aspects that can be explained by available predictors.

The predictors in CENIC-P1S1 can be categorized into the following groups:

A. Demographics (age, gender, race, ethnicity, marital status, education, household income)
B. Questionnaires (detailed description of each questionnaire can be found in [16]):
    a. Nicotine dependence:
        i. Fagerström Test for Nicotine Dependence (*FTND*)
        ii. Wisconsin Inventory of Smoking Dependence Motives (*WISDM*)
    b. Craving to smoke:
        i. Questionnaire of Smoking Urges (*QSU*)
        ii. Cigarette Evaluation Scale (*CES*)
        iii. Cigarette Purchase Task (*CPT*)
    c. Intention to quit and nicotine withdrawal:
        i. Stages of Change (*SOC*)
        ii. Intention to Quit (*ITQ*)
    d. Depression and stress:
        i. Perceived Stress Scale (*PSS*), participant's perception of their own stress
        ii. Center for Epidemiologic Studies Depression Scale (*CESD*)
    e. Health biomarkers:
        i. Respiratory Health Questionnaire (*RESP*)
        ii. Health Changes Questionnaire (*HC*)
    f. Perception of risk:
        i. Perceived Health Risk (*PHRR*)
    g. Other dependency:
        i. Short Michigan Alcoholism Screening Test (*SMAST*)
        ii. Drug Abuse Screening Test (*DAST*)
C. Physical measurements:
    a. Vitals/CO (blood pressure, heart rate, height, weight, expired carbon monoxide);
    b. Urine Tox Screen (amphetamines, barbiturates, benzodiazepines, etc.)
    c. Biomarker Variables (total nicotine, total cotinine, nicotine equivalents, etc.)
D. Environmental and Social Influences on Tobacco Use
E. Tobacco Use History
F. 30 Day Follow-up Questionnaire

For our analysis, as predictors, we consider:

- treatment arm
- cigarette purchase task (CPT)
- questionnaire of smoking urges (QSU)
- Wisconsin inventory of smoking dependence motives (WISDM)
- Minnesota nicotine withdrawal scale (MNWS)
- urine tox screen (drug screening test)

- center for epidemiologic studies depression scale (CESD)
- stages of change (SOC) in intention to quit smoking
- demographic predictors (age, gender, marital status, household income, ethnicity, living situation, education and etc).

Focusing our attention on two main communities (the biggest) found by percolation, the **zeroth (olive)** and **third (pink)** (see Figure 13), we proceed as follows.

Running statistical tests, a table of statistically significant predictors with the p-values ≤ 0.05 is calculated to establish that the distribution of the predictors for a selected community of nodes was different from that of another community or all the rest of our dataset (i.e. complementary points). After significance level of any predictor was found to be statistically significant, we are able to construct a histogram representing normalized frequency distributions of the predictor for both the nodes in the selected pair of communities or for the community and its complementary.

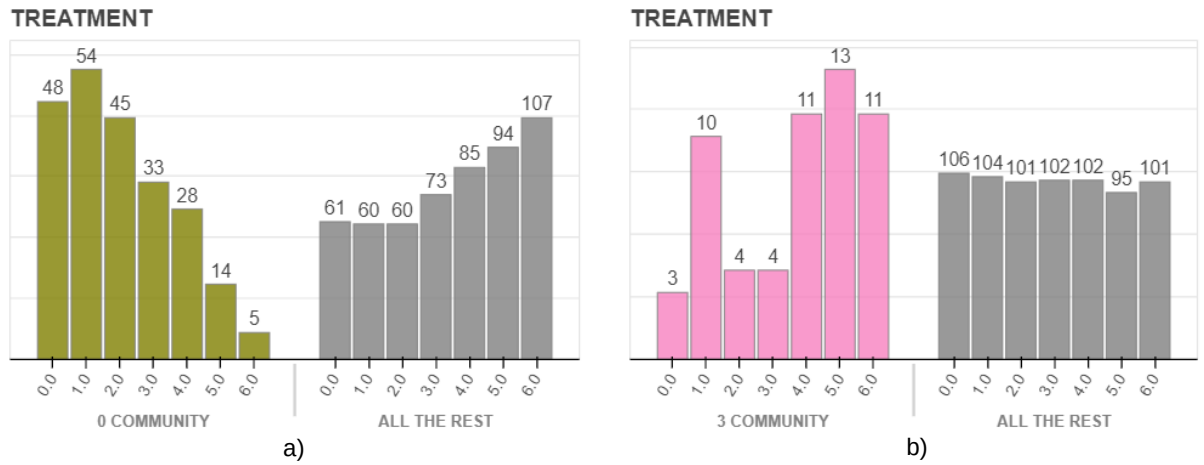
For the purpose of the statistical analyses, continuous, mixed, binary, and categorical (non-binary) univariate predictors were differentiated according to variable type. Continuous predictors were examined using the standard two-sample Mann–Whitney–Wilcoxon test. To examine the statistical association between two samples within the categorical data, Fisher's exact test and the chi-square test were used for the binary and non-binary categorical variables, respectively.

The authors of the original study [17] aimed to assess the behavioral, subjective, and physiological effects of smoking cigarettes at different nicotine (and tar) yields. They established that participants who were assigned cigarettes with low nicotine yield showed decrease of CPD rates. Our experiment verifies this result. Moreover, new findings are discovered.

In order to illustrate our results, in current paper the following labeling of the treatment arms is used (reordered by increase of nicotine yield):

(0) G: 0.4 mg/g
(1) B: 0.4 mg/g, high tar
(2) D: 1.3 mg/g
(3) A: 2.4 mg/g
(4) F: 5.2 mg/g
(5) C: 15.8 mg/g
(6) E: usual brand

Figure 20 shows that the **0-olive** and **3-pink** communities discovered by the *k*-clique percolation method have significant difference (chi-square test, p-value ≤ 0.05) in treatment arm distribution compared with all the rest of the dataset.



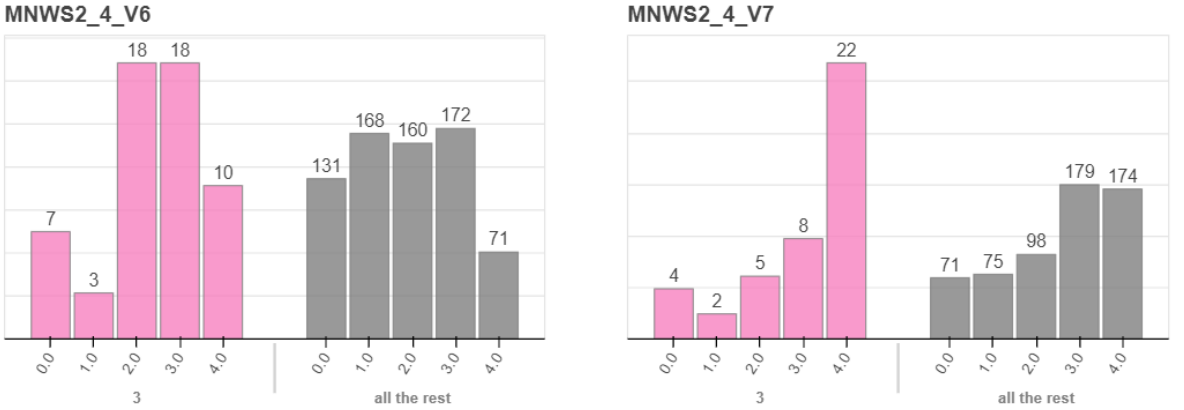
a) b)

Figure 20

The **0-olive** community comprises of participants that reduced CPD consumption according to Figure 13c) and as shown in Figure 20a) they were assigned cigarettes with low nicotine yield, which justifies the results of the original study. The **3-pink** community shows significant shift to higher nicotine yield (Figure 20b) together with increase of CPD consumption.

In the original study it was found that the cigarettes with lower nicotine content, as compared with control cigarettes, reduced exposure to and dependence on nicotine, as well as craving during abstinence from smoking. We can verify the reduce of craving by considering the difference of *QSU* (Questionnaire of Smoking Urges) total score at abstinence visit minus the average of *QSU* total score values taken at weeks 1 through 6. We can also validate the

results of [17] by looking at differences between *WISDM* predictors at visit 6 and at the baseline: *WISDM_PDM* and *WISDM_SDM* (primary and secondary dependence motive scales, respectively) scores and *WISDM_CRAV* (craving subscale) score. All of them have significantly lower values (Mann–Whitney–Wilcoxon test) for the **0-olive** community compared to the rest of the dataset. Apart from the confirmation of the results of the original study, the following information about **0-olive** and **3-pink** communities with regard to the *WISDM* were further discovered: **0-olive** community showed improvement of both the overall *WISDM* total score as well as *WISDM_LOC* (loss of control subscale), *WISDM_AUT* (automaticity subscale), *WISDM_AFFAT* (affiliative attachment subscale) and *WISDM_TASTE_V6* (taste subscale at visit 6), while for the **3-pink** community all but *WISDM_AFFAT* scores got worse. The latter indicates the increase of smoking dependence for the **3-pink** community, which is also confirmed by the dynamics of craving to smoke (the *MNWS2_4* scores, growing from 1st visit to 7th abstinence visit). It is significantly different (chi-square, p-value ≤ 0.05) for each of the 3-7 visits (see Figure 21) compared to the rest of the dataset.



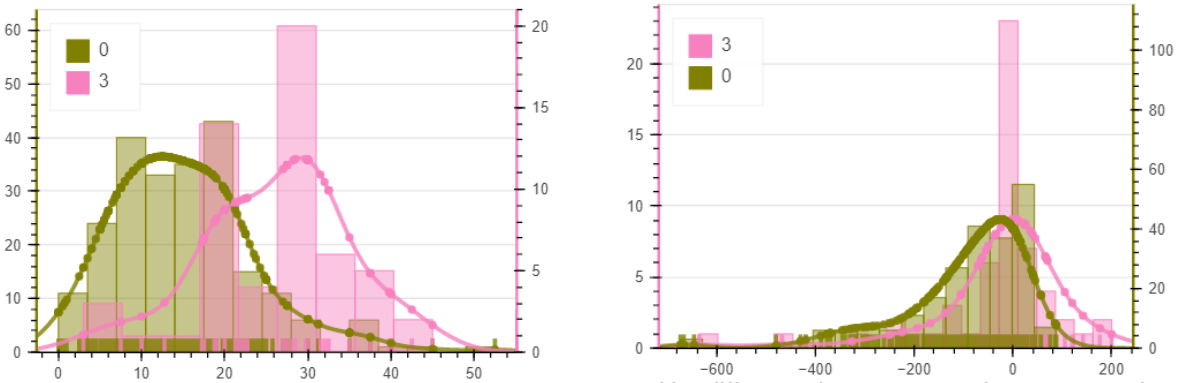
a) "desire or craving to smoke" (visit 6)
scale: 0 = none — 4 = severe

b) "desire or craving to smoke" (abstinence visit 7)
scale: 0 = none — 4 = severe

Figure 21

It should be noted that 27 out of 56 participants in the **3-pink** community had a positive drug test (done on visit 6). And 10 out of these 27 did test positive for cocaine. Wherein, among all 227 members of the **0-olive** community 87 had a positive drug test, 9 of which did a positive test for cocaine.

CPT (cigarettes purchase task) questionnaire reveals that participants of the **3-pink** community not only smoke more than of **0-olive** but are ready to spend more money on their consumption level (Mann–Whitney–Wilcoxon test, p-value ≤ 0.05), see Figure 22a. Moreover, the absolute difference between sum of CPT score values for study and usual brand cigarettes is significantly smaller for **3-pink** compared to **0-olive** (Figure 22b). All these findings remain true for both pairwise and complementary comparisons.



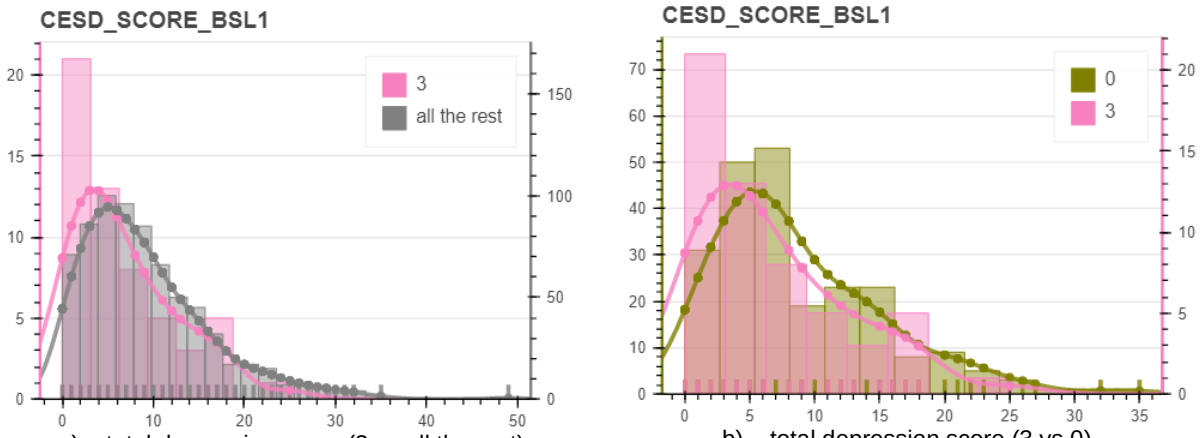
a) CPT range

b) difference between sum of CPT score values for study and usual brand cigarettes

Figure 22

As for explanation, it could be that participants grouped in the **3-pink** community liked their study cigarettes (with high nicotine yield) almost as much as their usual brand cigarettes, which is not true for the participants from the **0-olive** community (with low nicotine yield).

From the *CESD* questionnaire one can discover several additional predictors which further describe the **3-pink** and **0-olive** communities. Firstly, the **3-pink** community shows lower total depression score both on the baseline and after the 6th visit, both compared to all the rest (Figure 23a) and with the **0-olive** community (Figure 23b).

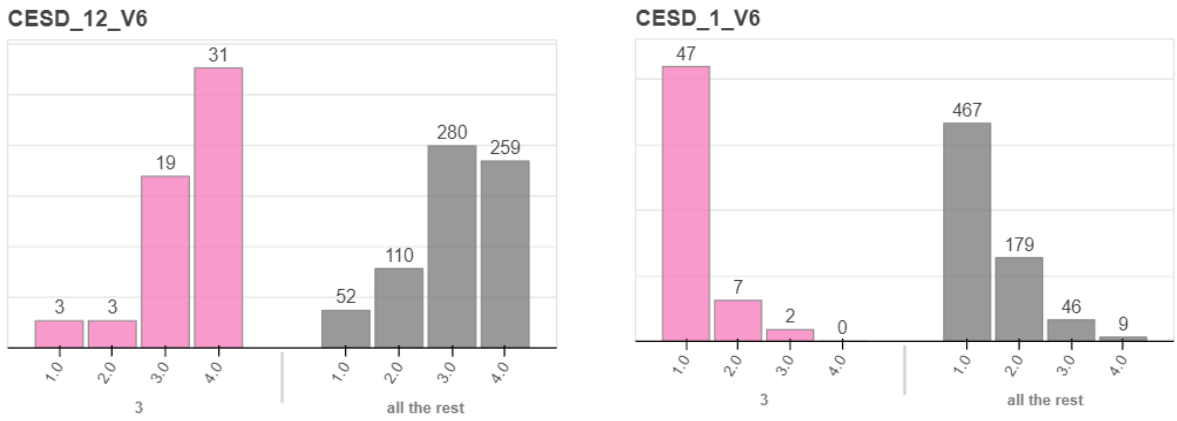


a) total depression score (3 vs all the rest)

b) total depression score (3 vs 0)

Figure 23

Secondly, if **3-pink** community is compared with all the rest of the dataset, then they are significantly different (chi-square, p-value ≤ 0.05) by the predictor *CESD_12_V6*, which says that participants of the **3-pink** community felt happy during week 6 more often than all the rest of participants (Figure 24a). The participants grouped into the **3-pink** community felt also less bothered during week 6 (*CESD_1_V6*), see Figure 24b. The latter characteristic of the **3-pink** community remains true for both pairwise comparison (3 with 0) and complementary comparisons (3 to all the rest).

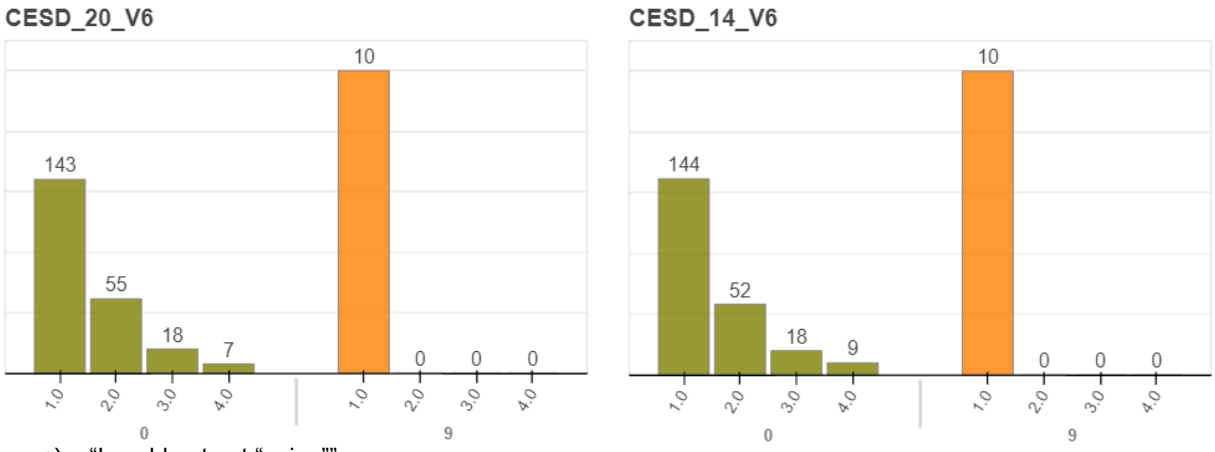


a) "I was happy"
scale: 1 = rarely — 4 = most of the time

b) "I was bothered by things that usually don't bother me"
scale: 1 = rarely — 4 = most of the time

Figure 24

If compared to its neighbors, the **0-olive** community members answered after visit 6 that some of the time they could not get "going" (*CESD_20_V6,* Figure 25a), while **all** the members of its neighboring **9-orange** community answered "none of the time". The same happened with feeling lonely (*CESD_14_V6,* Figure 25b), appetite (*CESD_2_V6*) and bothering by things that usually don't bother (*CESD_1_V6*), none was a problem for all ten members of the **9-orange** community.



a) "I could not get "going""
scale: 1 = rarely — 4 = most of the time

b) "I felt lonely"
scale: 1 = rarely — 4 = most of the time

Figure 25

As to the pairwise comparison with the neighbors of the **3-pink** community (**1-cyan**, **4-brown**, **8-blue**), it could also be seen that the **3-pink** community shows significantly less signs of depression. Contrast to **3-pink**, **1-cyan** talked less than usual at the baseline and were more bothered after week 6 (*CESD_1_V6*). The **8-blue** community is different in feeling that people dislike them comparing with that of **3-pink** (*CESD_19_BSL1*), some time they did not feel like eating (*CESD_2_BSL1*), and **8-blue** comprises of those who some of the time were bothered by things that usually don't bother them (*CESD_1_V6*). The **4-brown** community had significantly higher than **3-pink** total depression score by visit 6 (*CESD_SCORE_V6*), their participants had more often a poor appetite (both at the baseline, *CESD_2_BSL1*, and during week 6, *CESD_2_V6*) and were bothered by things that usually don't bother them (significantly more frequently).

It should be noted that if one compares **3-pink** community with any other discovered community there will be differences in *CESD* scores distributions. Thus, it is quite fair to call the **3-pink** community as a "happy community".

Analyzing the demographic predictors, the 0 and 3 communities can be characterized as follows. The **3-pink** community comprises of participants that roughly could be described as being near 50 years old (*DEMO_2_1_TEXT_SCR*), divorced (*DEMO_6_SCR*) and alone (*DEMO_7_SCR*). The **0-olive** community significantly includes students (*DEMO_10_SCR*) and have friends or relatives (*DEMO_7_SCR*).

A similar in-depth analysis of a greater variety of predictors can be performed on further patterns discovered by clique percolation and other methods, as described in the previous sections.

3.6. Clustering and community detection: better together

The *k*-medoids algorithm assigns each node to some cluster. If the dataset contains the outliers or noisy points it may cause the "unclearness" of clusters. In this section, we investigate if the quality of clustering would increase when the data is filtered by excluding some nodes. These nodes are taken as out-of-community nodes ("the noise") found by the percolation algorithm and the filtration is done in two ways:

- the noise is excluded directly from *k*-medoids clusters ("postfiltartion")
- the noise is excluded from experiment's dataset and *k*-medoids clustering is reapplied ("prefiltation").

The experiment on excluding nodes is implemented on 6 clusters. The results are shown in Table 7**.** In the "Percolation" and "*k*-medoids" rows the results of scoring from Table 6 are given. And the last two rows show the improvement of *k*-medoids scores with the "postfiltration" and the "prefiltration" by the help of percolation. If the number of clusters is not fixed (as 6) the improvement could be even better.

	Weighted density	Performance	Modularity	Silhouette score	DB score	DB score mean	av std	av distortion	max distortion	av outliers	av std cmdm	av distortion cmdm	max distortion cmdm
Percolation	**0.212**	0.917	0.363	0.232	1.386	**0.586**	**9.136**	**9.275**	65.72	0.254	**0.474**	**0.018**	0.104
***k*-medoids**	0.125	0.837	**0.529**	0.243	1.258	0.817	18.037	59.43	123.9	0.191	0.634	0.067	0.127
***k*-medoids with "postfiltration"**	0.139	0.934	0.401	0.220	**1.257**	0.675	14.952	22.08	60.77	0.270	0.545	0.026	**0.069**
***k*-medoids with "prefiltration"**	0.157	**0.945**	0.383	**0.279**	1.630	0.688	13.450	19.05	**55.93**	**0.183**	0.488	0.024	0.071

Table 7: *k*-medoids scores after excluding the noise found by percolation

Legend: **Hot palette marker** highlights "the larger-the better" scores
Cold palette marker highlights "the smaller-the better" scores
Scores are compared column-wise

The results of this section suggest that the percolation algorithm can be used to prefilter or postfilter the data to make better defined clusters.

CONCLUSIONS

In this paper, the authors compare the ability of clustering and community detection methods to reveal hidden patterns in data. The authors consider two clustering algorithms (DBSCAN, *k*-medoids) on the dataset and two community detection algorithms (clique percolation, Girvan-Newman) on its topological model and pick the parameters for their best realizations. The analysis of the results with numerous clustering and community detection quality scores shows that clique percolation algorithm gives the clearest and the most defined patterns which differ from those found by clustering methods and are significant from statistical point of view. Finally, the authors show how the pre- or postfiltration of the data with the help of the clique percolation method can improve the results of clustering.

REFERENCES

[1] G. Carlsson, "Topology and data," *Bulletin of the American Mathematical Society,* vol. 46, no. 2, pp. 255-308, 2009.

[2] A. Zomorodian, Topology for Computing, Cambridge: Cambridge University Press, 2005.

[3] K. Drach and I. Kotenko, "Using the geometric properties of datasets to analyze the accuracy of COVID-19 forecasting models," in *Pharmaceutical Users Software Exchange 2022 (Phuse US Connect, May 1-4, 2022, Atlanta, USA)*, 2022.

[4] S. Glushakov, I. Kotenko and K. Drach, "Understanding the spread of COVID-19 in the United States using topology and machine learning for time series," in *Pharmaceutical Users Software Exchange 2020 (Phuse US Connect, March 8-11, 2020, virtual)*, 2020.

[5] S. Fortunato, "Community detection in graphs," *Physics Reports,* vol. 486, no. 3-5, pp. 75-174, 2010.

[6] S. Fortunato and D. Hric, "Community detection in networks: a user guide," *Physics Reports,* vol. 659, pp. 1-44, 2016.

[7] S. Glushakov and K. Drach, "Sub-population Detection Using Graph-based Machine Learning," in *Pharmaceutical Users Software Exchange 2019 (Phuse EU Connect, November 10-13, 2019, Amsterdam, The Netherlands)*, 2019.

[8] M. Girvan and M. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences,* vol. 99, no. 12, pp. 7821-7826, 2002.

[9] G. Palla, I. Derenyi, I. Farkas and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature,* vol. 435, pp. 814-818, 2005.

[10] D. Xu and Y. Tian, "A comprehensive survey of clustering algorithms," *Annals of Data Science,* vol. 2, pp. 165-193, 2015.

[11] J. Sander, M. Ester, H.-P. Kriegel and X. Xu, "Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications," *Data Mining and Knowledge Discovery,* vol. 2, pp. 169-194, 1998.

[12] E. Hartuv and R. Shamir, "A clustering algorithm based on graph connectivity," *Information Processing Letters,* vol. 76, no. 4-6, pp. 175-181, 2000.

[13] A. Ng, M. Jordan and Y. Weiss, "On Spectral Clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems*, vol. 14, MIT Press, 2001.

[14] D. Steinley, "Properties of the Hubert-Arable Adjusted Rand Index," *Psychological Methods,* vol. 9, no. 3, pp. 386-396, 2004.

[15] C. C. Aggarwal and C. K. Reddy, Eds., Data Clustering: Algorithms and Applications, New York: Chapman and Hall/CRC, 2014.

[16] "National Institute on Drug Abuse," [Online]. Available: https://datashare.nida.nih.gov/study/nidacenicp1s1. [Accessed 19 February 2023].

[17] E. C. Donny, R. L. Denlinger and et al., "Randomized Trial of Reduced-Nicotine Standards for Cigarettes," *The New England Journal of Medicine,* vol. 373, no. 14, pp. 1340-1349, 2015.

ACKNOWLEDGMENTS

We would like to acknowledge **Oleksandr Leonov** (Kharkiv National University, Ukraine), **Lyudmyla Polyakova** (Kharkiv National University, Ukraine) and **Victoriia Shevtsova** (Intego Group, Ukraine) for handling the computational experiment and being core members of the research team. Without you, this research and the experiment would not have been possible.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the group of authors at:

Contact: Iryna Kotenko

Company: Intego Group

Address: 2300 Maitland Center Pkwy, Suite 240, Maitland, FL 32751, USA

Work Phone: +1 407 512-1006

Email: irina.kotenko@intego-group.com

Web: www.intego-group.com