Clustering High-Dimensional Social Media Datasets utilizing Graph Mining

Andreas Kanavos*, Gerasimos Vonitsanos* and Phivos Mylonas*
*Department of Informatics
Ionian University, Corfu, Greece
{akanavos, fmylonas}@ionio.gr
†Computer Engineering and Informatics Department
University of Patras, Patras, Greece
mvonitsanos@ceid.upatras.gr

***Abstract*—Social networks are an essential component of people' daily lives, and as a result, much academic attention has been focused on them. The rapid adoption of machine learning as a problem-solving tool, which simplifies and accelerates numerous tasks while enabling the processing of large volumes of data, has played a significant role in this field of research. This is in contrast to the more traditional approaches that lacked this momentum. Characterization of linkages and cluster identification in social networks are two of the research community's most well-known issues. The goal of this study is to gather data for a set of users who are then divided into groups based on the hashtags they used in their Twitter postings. The procedure performed generates the numerical data, in following reduces the dimensions, and finally performs the clustering.**

***Index Terms*—Social Network Analysis, Graph Mining, Twitter, Knowledge Extraction, Clustering, Dimensionality Reduction**

I. Introduction

Businesses have recently become interested in networks as a platform for grouping individuals as social media has taken off. Several approaches have been successfully applied in this direction, and effective retrieval strategies and techniques have mostly been developed from the class of probabilistic models [5]. The link between the processing of queries and their results is thought to be a key factor in clustering. The level of similarity chosen will make use of the redundant and overlapping information included in the data points. Based on the idea that related points might fulfill the same query or queries, this type of retrieval is performed, as shown in [34]. Along the same lines, more recent study has shown that if strong clusters can be constructed, then the retrieval performance will subsequently increase [27].

Scientists are faced with a difficulty as a result of the diversity of social networks during the past ten years, and consequently, of their users. In order to find commonalities between people in actual contacts, new patterns of interaction are developing as a result of user clustering. Additionally, people that are part of social networks generally interact with one another and form linked clusters. Their examination is a key step in a process known as ”social network analysis” [35]. This leads to the conclusion that this frequently challenging problem may be classified as an optimization constraint since such networks have a complicated and dynamic structure, making it difficult to simply identify these clusters. Therefore, discovering node groups with more interfaces is a crucial research objective that can also work in identifying node groups with less linkages.

A network is based on the idea that the allocation of a clustering factor obeys the rule of strength within social networking. In this context, grouping processes are also recognized; the most fundamental of these is the clustering coefficient, which plays a crucial role in determining how likely it is for nodes to form clusters. This characteristic assumes their discovery for a deeper knowledge of the network architecture and argues that interconnected clusters and communities constitute social networks [11], [17].

This paper looks into the problem of identifying clusters of users by examining various preprocessing techniques as a common ground for cluster formation. The focus of this work is on the aspect of clusters and is thus differentiated from existing and more general work on identifying clusters mainly based on users' static profiles. Our view of creating more dense clusters is to use the “follow” links of Twitter and extract the symbolic links that emerge through Twitter interactions, identifying thus clusters of high information flow. In this work, the steps performed contain the numerical data generation, dimensionality reduction, and clustering.

The rest of the paper is organized as follows. First, section II describes the relevant to the subject works. Besides, Section III analyzes the methodology followed, the algorithms, and the modules of our paper. Then, section IV details the implementation and different methods for practically evaluating our techniques. Moreover, in Section V, the acquired research results are captured. Finally, discussion and conclusions are outlined in Sections VI and VII, respectively.

II. Related Work

Social networks are a platform that is crucial in fostering friendships amongst individuals who have common hobbies or interests. This platform offers a variety of connections between people, along with information on how strong these ties are.

978-1-6654-8045-1/22/$31.00 © 2022 IEEE

In this context, several efforts and research are being carried out for the so-called ”analysis” of Twitter to conclude both the content and the general use of this social network and its services to customers. For example, many research types have been carried out on identifying social groups, verifying fake news, identifying or creating bots, and analyzing sentiment of listeners concerning events or public figures, but also research to conclude the politician and the commercial sector.

Understanding the relationships between members is crucial to comprehend why people tend to build social networks and how these networks work. [15], [16]. Using a grouping identification approach based on modularity, which takes into account the unique personality qualities of individuals, this leads to the establishment of such communities inside the Twitter graphs. Additionally, by including pre-processing sequences, graph vertices originating from the aforementioned personality-based methods are eliminated.

For example, the research in [20] gathers general characteristics of this social network, such as typological and geographical properties, development patterns, as well as behaviors of its users. Another research [13] stated that the activities of Twitter users could be considered information-seeking, information-sharing, or socializing activities. To identify different types of user intent within Twitter, a 2-level framework for detecting user intent was proposed. Several works [12], [18], [19] examined whether a sender, who has a significant influence, can modify the opinions of his followers by publishing content with emotional enlightenment. It also proposes a methodology so that debtors who have a strong power of influence from other debtors can be identified. In [10], an attempt is made to characterize a set of tweets, which had been collected as data, regarding the disposition of the opinion of the person who made the publication.

In a work related to the clustering of hashtags [25], it is argued that two hashtags are similar if they co-occur simultaneously in a tweet. Authors created a summary table where they represent the classes resulting from categorizing hashtags. The concept of ”co-occurrence frequency” was used to measure the similarity between classes. The authors in [4] presented an algorithm to perceive the relationships between the content of a tweet and the set of hashtags that include this tweet. Specifically, each tweet was represented as a frequency list of the words (non-stopword and non-hashtag) included in that tweet. They proved, therefore, that comparing clustering techniques in data from Twitter, the categorization of hashtags and the reduction of the dimension of the data can lead to a precise classification of tweets into categories without affecting the effectiveness of the classification algorithms.

In [33], the modeling of social networks as undirected graphs is presented, and models of the presence of private space, attack models for the case of anonymity shared in class i-hope degree, i.e., the prior knowledge of the opponent includes the degree of the target and the degrees of the neighbors within i-hopes from the target. For this reason, two new and efficient clustering techniques for undirected graphs are presented: the t-Means and union-split clustering algorithms are delimited, which group similar graph nodes into clusters with a minimum size constraint. A new technique is presented for quantifying a local method's effectiveness to predict how different algorithms perform relative to each other in [6]. Furthermore, due to the unique implementation of a local method during cluster initialization, a simple set of ad hoc fallback networks was also developed.

The research in [8] was based on retweets and, after analyzing data received from the Twitter API, addresses the different variations in retweeting messages on Twitter and how the different patterns lead to ambiguity about the state origin, performance, and fidelity of speech, significantly as the content is modified during dissemination. Finally, in [26], an approach based on the scale of the data is proposed for estimating the location of tweets using a new but straightforward approach to Gaussian Mixture Models. At the same time, because real-world applications rely on quantified lifetimes for such estimations, they propose an efficiency and quantification metric and perform this approach.

III. Methodology

A. Word2Vec

Word2vec is a set of related models used to produce word representations in vector space [29]. These models consist of two-layer neural networks trained to reconstruct words in a linguistic context. Word2vec algorithm includes an iterative word vector generation and model generation method proposed by Google. Unlike other methods, which treat the text as a whole when generating the model, Word2vec works iteratively by updating the model for each text it needs to process.

When creating a model, it takes a large set of texts as input and creates a vector space of variable dimensions, where a different vector is created for each word. The magnitude of the dimensions is usually in the order of a few hundred. The model generation process can be parameterized in terms of the architecture used, the control window size, and the number of dimensions. To achieve the sampling of negative examples, the Gensim implementation automates binary search on a matrix the size of the dictionary, making its time complexity technically $O(N * log(V))$, where N is the total text size and V is the vocabulary size of unique words.

B. Dimensionality Reduction

Three different algorithms have been used in terms of dimensionality reduction, namely Principal Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding (t-SNE) and Truncated Singular Value Decomposition (SVD), which will be presented in the following subsection.

1) Principal Component Analysis (PCA): Principal Component Analysis (PCA) is an orthogonal transformation in which the correlation between the variables is removed and their dimensions are reduced. It is a technique that utilizes an orthogonal transformation projecting each data point onto only the first few principal components with the aim to obtain lower-dimensional data while preserving as much of the data variation as possible [2].

2) t-Distributed Stochastic Neighbor Embedding (t-SNE): t-Distributed Stochastic Neighbor Embedding (t-SNE) is an algorithm used to visualize large-dimensional data in small-dimensional space by giving each data point a location in a two- or three-dimensional map. Precisely, it models each high-dimensional object by a two- or three-dimensional point so that nearby points model similar objects, and dissimilar objects are modeled by distant points with high probability [7].

3) Truncated Singular Value Decomposition (SVD): Singular Value Decomposition (SVD) is one of the most essential linear algebra matrix decomposition methods but is used to visualize concepts in vector space. In the new vector space being created, words with similar illustrations tend to appear in documents with similar content. In addition, it is also used as a data dimensionality reduction method in vector space with a special form of SVD, the Truncated SVD [1].

C. Clustering Algorithms

Seven different clustering algorithms have been employed in order to measure the effectiveness of each one, namely k-Means, Bisecting k-Means, DBSCAN, OPTICS, Gaussian Mixture Model (GMM), Hierarchical and Spectral Clustering.

1) k-Means: k-Means algorithm starts with k random points called cluster centroids, which denote the center of each cluster, while k denotes the desired number of clusters that the algorithm will generate. Then, k-Means iteratively performs two steps; in the first step, the assignment to a cluster is made, while in the second step, the centroid of each cluster is redefined and shifted [21]. Running a fixed number of iterations of the standard algorithm takes $O(I * k * N * d)$ in time complexity for N (d-dimensional) points, where k is the number of centers (or the number of clusters), and I is the number of repetitions.

2) Bisecting k-Means: Another implementation of the k-Means algorithm is the Bisecting k-Means, where in this version, the set of points is divided into two clusters. A random binary tree is therefore created where each interval, a node with two children, corresponds to the division of the points of the set into 2 clusters [31].

3) Density-Based Spatial Clustering of Applications with Noise (DBSCAN): DBSCAN is a non-parametric algorithm that uses two parameters, ϵ and $MinPts$, i.e., the minimum number of points required to create a dense region. It starts from a random point where the points at the most distance ϵ from the selected point are retrieved. If there is an excess number of points, i.e. greater than $MinPts$, the algorithm creates a cluster; otherwise, the point is disguised as a hole. If a point is a dense part of a cluster, then indeed its ϵ-neighborhood is a subset of this cluster [32].

4) OPTICS: OPTICS algorithm, like DBSCAN, is considered a density-based algorithm whose purpose is to find density-based clusters in spatial data. The concept of OPTICS is similar to DBSCAN, but it faces a similar disadvantage to DBSCAN, which is the problem of cluster detection in data of varying density. To achieve this, it uses data clustering techniques so that those spatially closer points will become neighbors in the ordering. OPTICS, like DBSCAN, requires the input of two parameters, the minimum radius (Eps) and the minimum number of points ($MinPts$), and produces a dendrogram [3]. Also, OPTICS, unlike DBSCAN, considers points that are members of a denser class so that each point is assigned a kernel distance that describes the distance of the nearest $MinPts$. The algorithm's complexity is $O(n{*}log(n))$.

5) Gaussian Mixture Model (GMM): Gaussian Mixture Model (GMM) is considered a probabilistic concept where the model general distributions estimating the density assuming all the data points are generated from a mix of Gaussian distributions with unknown parameters. The model parameters are selected by maximizing the logarithm of the bound probability of the training data concerning the model itself. GMM consists of mean vectors μ and covariance matrices Σ. A Gaussian distribution is a continuous probability distribution that takes on a bell-shaped curve [28].

In Gaussian Mixture Models, an Expectation-Maximization (EM) algorithm is a powerful tool for estimating the parameters of a GMM. The expectation is defined as E while the maximization is defined as M. Expectation is used to find the Gaussian parameters, which represent each component of Gaussian mixture models and EM chooses some random values for the missing data and calculates a new dataset. These new values are then used to calculate an adequate prior dataset, filling in the missing data until the values are valid [22].

6) Hierarchical Clustering: Hierarchical algorithms are based on the combination or division of existing groups and produce a tree representing a hierarchical grouping of the objects of the set based on specific criteria. The tree's base contains all the objects, its leaves consist of a single object, while the intermediate nodes represent the clusters created by the union of several leaves. Algorithms of this class do not need the number of clusters as input in advance. Instead, each level represents a distance threshold; if two clusters have a distance smaller than this threshold, they are merged into a cluster which is the linkage criterion. In hierarchical algorithms, there is the accumulative approach (bottom-up) and the divisive approach (top-down). In the latter case, each object is initially placed in its cluster and then the individual clusters are iteratively merged into increasingly larger clusters until all objects are joined in a cluster. On the contrary, in the divisive method, the reverse process is followed, with the objects starting from a common cluster, which is in following split into smaller and smaller clusters [24].

7) Spectral Clustering: Complex multidimensional datasets are broken down using the spectral clustering approach into clusters of related data in rarer dimensions. The basic goal is to arrange various disorganized data items into different categories according to how distinctive they are. The connectivity technique identifies communities of nodes that are related to or located adjacent to one another in a network. After that, the nodes are mapped to a low-dimensional space that can be readily partitioned into clusters. More specifically, spectral clustering makes use of data from the eigenvalues of unique matrices created from the graph or data collection [23].

IV. Implementation

A. Twitter Discussion Synopsis

The Twitter graph was collected in a time interval of two weeks, that is $(19/09/2022 - 02/10/2022)$. A topic-based sampling approach was utilized where tweets are collected via a keyword search query. More specifically, data for one particular discussion on Twitter, namely #ChampionsLeague, was downloaded.

Therefore, a social network was created for a number of users, which was depicted in a directed graph whose every edge starts from a server and ends at another if the first one follows the second one. Specifically, the graph contains 8122 directed edges and 3261 nodes. The original nodes were more, e.g. 7785; however 4524 were isolated, i.e., they had no relationship with any of the other nodes and were deleted from the graph as will be presented in following Figure 1.

The properties of this dataset are presented in Table I. The first column has fundamental graph structure properties such as the number of edges and triangles, whereas the second column has Twitter specific properties such as the average number of followers and the maximum number of friends. Note that the vertices are accounts and the directed edges represent "following" relationships.

TABLE I
Graph and Twitter Properties for #ChampionsLeague Dataset

Graph Properties		**Twitter Properties**	
Property	**Value**	**Property**	**Value**
Vertices	3261	Average Followers	10226.35
Edges	8122	Average Friends	1453.34
Triangles	6754	Average Statuses	22503.28
Squares	5232	Maximum Friends	239151
Components	107	Maximum Followers	7145929
Maximum Friends	210	Eggs	96
Maximum Followers	723		
Average Friends	2.84		
Average Followers	4.14		
Average Pagerank	1.00		
Density	0.000764		

We notice that many triangles and squares are created between three or four nodes of the graph, respectively, since these are active users who publish material on the same topics, so it is possible to have relationships between them. At the same time, within this graph, some users have a large number of followers or users who follow either within the graph or on Twitter with the result that, again, it becomes more likely that there will be a triple or quadruple relationship between them. These structural characteristics indicate an active social network. Our aim is to create a social media graph with high degree of dense connections and not just a random graph.

In the graphic representation of the network in Figure 1, we notice that some points of dissemination of information are created, so some popular users around which a large part of the total relations of the network is concentrated. This is shown by the very firmly densely placed red dots (users) and the solid black lines that accumulate around them and thus show that these are the users with the most followers or friends in this network. On the contrary, of course, there are also users in the sparse points of the graph who join the rest with minimal edges, and we could argue that they are the least popular or active members of this social network.

B. Silhouette Method

Silhouette is a method for validating the consistency within clusters. This technique provides a short graphical representation of how well each object has been classified. As an example one can consider the objects that are clustered through any technique, such as the k-Means clustering algorithm, into k groups [30].

The estimated coefficient is calculated for each sample of the collection and consists of two ratings, namely:

- The average distance of a sample from the remaining samples of the same cluster and
- The average distance of a sample from the samples of the nearest cluster.

This coefficient can be defined as in following Equation 1 and in order to calculate the overall coefficient for a collection of samples, we need to calculate the average of the coefficients of each of its samples:

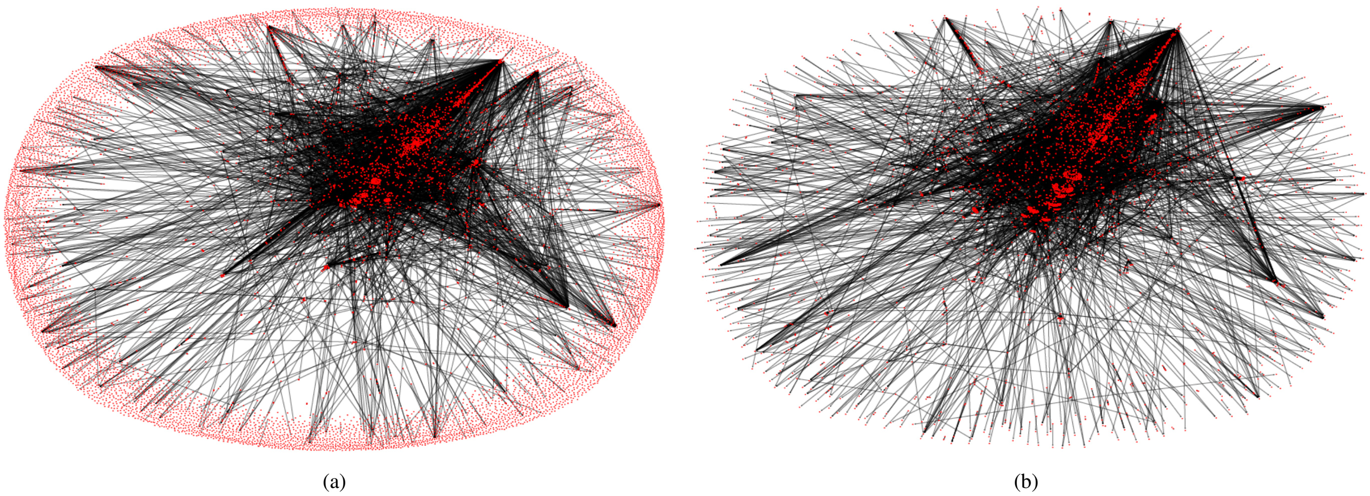
$$\frac{\beta - \alpha}{max(\alpha, \beta)} \tag{1}$$

Concretely, for each item i, $\alpha(i)$ is the average of the dissimilarity of i to all other items within the same cluster. The value of $\alpha(i)$ is interpreted as a metric of how well the item i is clustered within its cluster; that is the smaller its value, the better the clustering. In the following, $\beta(i)$ is defined as the lowest average dissimilarity of item i concerning any other cluster of which this item is not a member. The cluster with the next lower average dissimilarity is considered the "neighboring cluster" of item i because it would be the next best cluster to group this particular item.

Silhouette score values range from -1 to 1, where a value equals to 1 indicates that the object correctly belongs to the current cluster. Conversely, -1 indicates that it should have been placed in the neighboring cluster. Finally, a value close to the neutral 0 indicates that the object lies on the border of the clusters. Table II presents the Silhouette scores for each different value of k where the scores achieve similar values ranging from 0.55 to 0.62.

TABLE II
Silhouette Analysis

k	**Silhouette Score**
2	0.607290490923769
3	0.567850774607643
4	0.553020593769727
5	0.564194559159169
6	0.608938787637735
7	0.612308399507005
8	0.624352659805041
9	0.615360217999776



(a)

(b)

Fig. 1. (**a**) Initial Graph, (**b**) Graph with no Isolates

C. *Elbow Method*

In the elbow method, k-Means clustering algorithm is executed for many different values of k, from very small to very large. For each run, it is calculated an evaluation index of the clusters and this index is the cost syntax of the algorithm defined as the sum of the squared distance between each point and the centroid of the cluster assigned to it [9]. This is defined with the following Equation 2:

$$Cost = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||x_n - \mu_k||^2 \tag{2}$$

where r_{nk} is equal to 1 if point belongs to the specific cluster; otherwise is equal to 0.

Figure 2 plots the cost versus the number of clusters (k). We observe that the cost index decreases as the number of clusters k increases, which is expected. However, we also notice that for a specific value of k, the plot presents a sharp corner, hence the method's name. This particular value of k is chosen as optimal for the implementation of the algorithm.

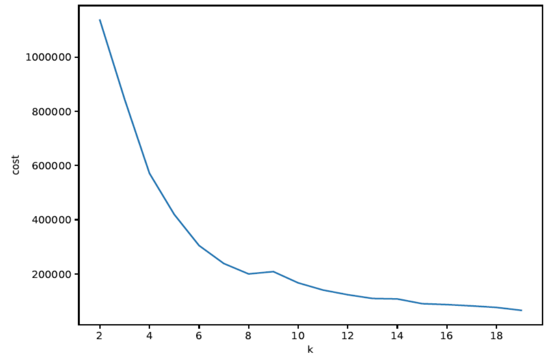


Fig. 2. Elbow Method Result

The conclusion, derived from the application of the two aforementioned methods, was to use the number of clusters, equal to 8, to run the algorithms afterwards, as the two methods converge to this value. This is the best value of k for the specific dataset, as at this specific value, the highest value in terms of the Silhouette score is observed, while from the shape of the Elbow method it appears that the ”elbow” is made for $k = 8$.

D. *Epsilon (ϵ) Parameter*

DBSCAN algorithm is one of the clustering algorithms for which, it is not necessary to predetermine the number of clusters before its execution. However, it is necessary to determine the ϵ value (the minimum radius or otherwise, the maximum distance between two points belonging to the same cluster).

The result is represented graphically in Figure 3. We conclude that the most suitable value for the DBSCAN algorithm to have the best possible result is equal to 5 when the elements that have this distance between them are minimal. As a result, in this case, the algorithm can group the data in a more efficient way.

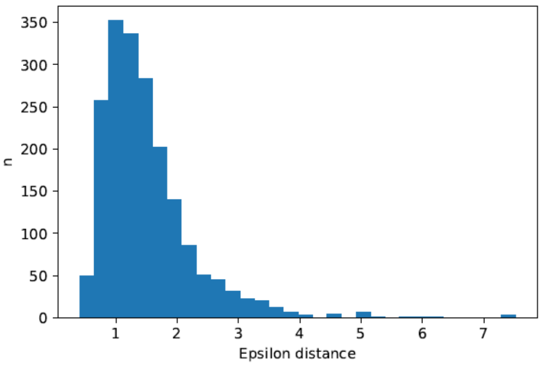


Fig. 3. Epsilon (ϵ) Parameter

E. Explained Variance

PCA is a dimensionality reduction method in which the correlation between the variables is removed, and their dimensions are reduced. It is an algorithm that uses an orthogonal transformation from a set of correlated variables to a set of uncorrelated variables [14].

Figure 4 portrays the sum of variances per each component, as the number of components is 100. We can conclude that important information is lost for the specific dataset by reducing the data dimensions as per our current implementation. However, it needs to be made clear the value of k, for which PCA algorithm could produce a new dataset that would not lose to a significant extent the characteristics of the original dataset but would offer, at the same time, a significant reduction in dimensions. For example, the first 40 out of 100 dimensions contain 85% of the information variance. This percentage can be considered entirely satisfactory but only helps in reducing the dimensions.

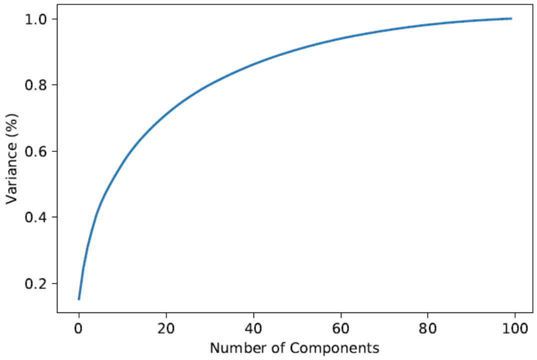


Fig. 4. Explained Variance

V. Experimental Evaluation

A. Dimensionality Reduction

The first experiment concerns the dimensionality reduction from 100 to 2 dimensions, which was implemented with use of the PCA, t-SNE, and Truncated SVD algorithms. Figure 5 presents the output of the execution of these three algorithms applied to the aforementioned graph.

The presentation of the data is optimal in the t-SNE algorithm as it appears to be sparsely distributed, which helps the clustering process. Additionally, the search for specific terms depicts that usernames related to those terms are nearby. The other two algorithms gathered all the data around two dense poles, and the search for terms proves that some related usernames are not nearby found.

B. Dimensionality Reduction Algorithms Execution Times

Figure 6 illustrates the graphical representation of the times for the execution of the above three algorithms, i.e. PCA, t-SNE and Truncated SVD for different data dimensions and different sizes of the set of cross-sectional words.

We observe that t-SNE algorithm is very time-consuming and compared to the other two algorithms, which range in microseconds (μ-seconds) levels most of the time, the range of this algorithm is in seconds. More to the point, PCA and Truncated SVD have similar performance, but with SVD algorithm having a slightly better performance in terms of execution time. However, their difference is significant for larger word sizes.

As a result, t-SNE algorithm was employed for the clustering process, as even though it was much worse in time comparison than the other two, the data it produced was of better quality for the intended purpose.

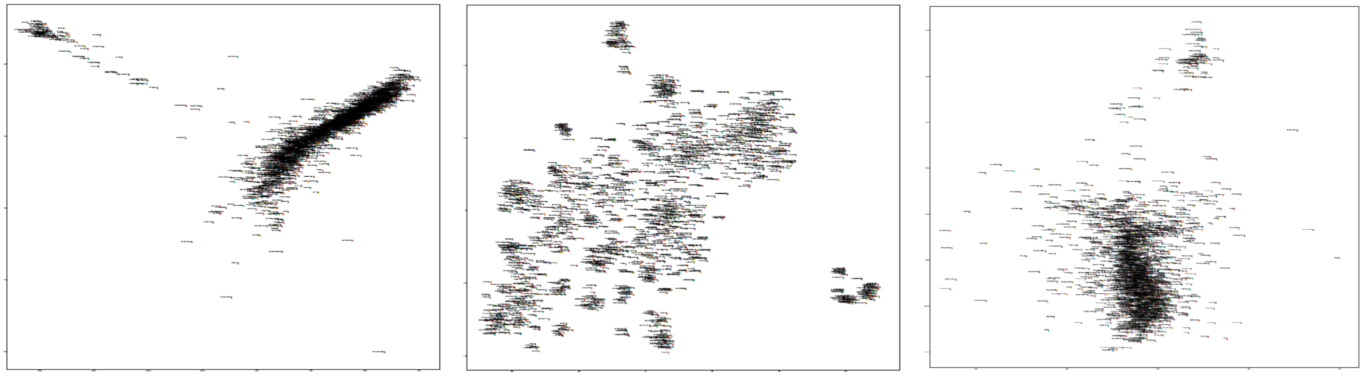
C. Clustering Algorithms Comparison

In this subsection, the graphical representation of the results regarding the seven clustering algorithms are presented. Specifically, Figure 7 illustrates the results of k-Means, Bisecting k-Means, DBSCAN and OPTICS, Gausian Mixture Model, Hierarchical Clustering and Spectral Clustering.

Regarding the number of clusters, DBSCAN and OPTICS algorithms, which do not require a prior determination of this particular number, resulted in 6 and 7 clusters, respectively. All other algorithms were executed for clusters number equal to 8. In following, the analysis of each algorithm will be presented and thoroughly analyzed. Specifically, we can observe that OPTICS algorithm failed to classify a large amount of data into a cluster; as can be seen in Figure 7(d), data with blue color belongs to the category of unspecified data, which was logical due to its inability to determine a cluster when the data is spatial. However, this algorithm needed better clustering for the data produced.

The Expectation-Maximization as well as Spectral clustering algorithms have different techniques for determining the number of components and clusters as presented in Figure 7(e). However, in the present work, the number obtained from the Silhouette and Elbow method for k-Means was taken into account. Regarding DBSCAN algorithm, as illustrated in Figure 7(c), a small amount of data was left in the undefined category. On the other hand, a large amount of data was gathered into a large cluster and created seven other small clusters, some of which contained only one data point. This may occur due to the values given to the ϵ parameter, which was set to value equal to 5, and $minPts$ parameter, which was set to value equal to 6. Similarly, the Spectral clustering algorithm gathered a large amount of data into one large cluster and created seven other small clusters, each containing only one data point, as can be portrayed in Figure 7(g). This can be proved, in a similar way with the DBSCAN algorithm, because the number of clusters was reduced to 8 as in all other algorithms, which was different from the appropriate number of clusters for this current algorithm.

k-Means and Bisecting k-Means algorithms gather the clusters around some particular points, e.g. the centroids of the clusters, which makes them ideal for spatial data geometries as in Figure 7(a) and (b). On the contrary, Expectation-Maximization algorithm converges the clusters around the Gaussian surfaces calculated by the Gaussian Mixture Model; for this reason, it is more efficient in flat data geometries.



(a)

(b)

(c)

Fig. 5. Dimensionality Reduction implemented with use of Algorithms: (**a**) PCA, (**b**) t-SNE, (**c**) Truncated SVD

Based on the patterns of the grouped data, we could claim that the best results are presented by the k-Means, Bisecting k-Means, Expectation-Maximization, and Hierarchical clustering algorithms. On the other hand, the remaining three algorithms failed to present satisfactory results. It is also acknowledged that no algorithm achieved the same result with another.

Finally, we could state that all algorithms could improve the quality of their results if data had higher quality and were more efficient in terms of the particular problem at hand.

D. Clustering Algorithms Execution Times

Table III introduces the average execution times of five iterations of the algorithms for the specific graph we utilized. Spectral clustering is the most time-consuming technique, as it can be proved due to the algorithm's complexity, which is equal to $O(n^3)$. We can observe that the fastest clustering algorithm is DBSCAN for the specific data used, with complexity equal to $O(n^2)$ and can be further reduced in cases of efficient low-dimensional data to $O(n * log(n))$.

Another output is that the experimental procedure cannot verify that the OPTICS is faster than DBSCAN algorithm, as we would expect it to be faster, due to its $O(n * log(n))$ complexity.

TABLE III
AVERAGE EXECUTION TIMES OF CLUSTERING ALGORITHMS

Algorithm	**Average Execution Time (sec)**
k-Means	8.08255
Bisecting k-Means	137.53432
DBSCAN	0.04225
OPTICS	0.93712
Gausian Mixture Model	9.84338
Hierarchical Clustering	0.14176
Spectral Clustering	748.55882

VI. DISCUSSION

Regarding dimensionality reduction, an essential difference in the execution time of t-SNE algorithm compared to PCA and Truncated SVD was observed, which was expected and confirmed. Furthermore, t-SNE algorithm could not be used if results of a dimension greater than four were sought. However, on the contrary, these weaknesses prove that for the specific data used, t-SNE algorithm was the best solution for subsequent clustering of data as it produced more extensive global data that maintained an excellent morphology and characteristics, with users who used similar hashtags being close to each other.

Moreover, data dimensionality reduction should be implemented with additional caution as algorithms that calculate distances between data points are inefficient for high dimensional data. In addition, it is difficult for the user to visualize the multi-dimensional data with aim to performing the required audit. Of course, some techniques make this possible, such as reducing the dimensions after performing the clustering. In this work, the procedure performed was firstly to generate the numerical data, then to reduce the dimensions, and finally to perform the clustering.

Regarding user clustering, it was shown that the morphology of the data did not enhance the results of all algorithms, as some failed to produce satisfactory clusters. If we had to choose one algorithm for clustering the specific data, we would probably select the Hierarchical algorithm, which could quickly group the data into satisfactory clusters. Referring to the time, it is noteworthy to mention that DBSCAN algorithm was the fastest but did not satisfactorily cluster the data.

Another critical aspect is that all clustering algorithms require numerical data, which means that the use of the Word2Vec neural network, due to our natural language data, is required. This will generate a new dataset that depends on the number of features of each sample, as well as its dimensions, which are determined by the total texts size assigned to it.

Finally, we also found that in a social network, just like in real life, some people have few social relationships, while others have many social relationships. In the constructed graph, some users, around whom the most edges (relations) were presented, were identified and captured, which means that it is straightforward for these users to spread information when they create it; or it is easier for them than remote users to access the information if it was created by someone else.

VII. Conclusions and Future Work

The collection of the right data required for dimensionality reduction and in following clustering process plays a huge role in correctly and effectively extracting results. The success of the algorithms is directly dependent on the correct configuration and transformation of the data, as inefficient data lead to unsatisfactory results. Also, the performance of each algorithm is implicitly linked to the morphology and characteristics of the graph and specifically the data points that will be given to it as training data. More in detail, huge variations were observed in the results of both the dimensionality reduction and the clustering algorithms.

By proposing some extensions, one could delve into the necessary evaluation of the performance of the models and algorithms through evaluation methods as well as test the speed of these models on different datasets to understand their behavior on different kinds of data. As a result, one idea could be to group the users based on the texts of the tweets they have published or to cluster users either with use of hashtags (hashtag clustering) or to calculate both text and hashtags simultaneously. Furthermore, two-level clustering could be implemented by automating both the data and the graph created for the users. On the contrary, it would be possible for the division of the users to focus only on the graph and on the community resilience techniques that exclusively concern graphs.

Acknowledgement

This research was funded by the European Union and Greece (Partnership Agreement for the Development Framework 2014-2020) under the Regional Operational Programme Ionian Islands 2014-2020, project title: ”Indirect costs for project ”TRaditional corfU Music PresERvation through digiTal innovation””, project number: 5030952.

References

[1] H. Abdi. Singular value decomposition (svd) and generalized singular value decomposition (gsvd). *Encyclopedia of Measurement and Statistics*, pages 907–912, 2007.

[2] H. Abdi and L. J. Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459, 2010.

[3] M. Ankerst, M. M. Breunig, H. Kriegel, and J. Sander. OPTICS: ordering points to identify the clustering structure. In *ACM SIGMOD International Conference on Management of Data*, pages 49–60, 1999.

[4] D. Antenucci, G. Handy, A. Modi, and M. Tinkerhess. Classification of tweets via clustering of hashtags. *EECS*, 545:1–11, 2011.

[5] R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval: The Concepts and Technology Behind Search, Second edition*. Pearson Education Ltd., Harlow, England, 2011.

[6] J. P. Bagrow. Evaluating local community methods in networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(05):P05001, 2008.

[7] A. C. Belkina, C. O. Ciccolella, R. Anno, R. Halpert, J. Spidlen, and J. E. Snyder-Cappione. Automated optimized parameters for t-distributed stochastic neighbor embedding improve visualization and analysis of large datasets. *Nature Communications*, 10(1):1–12, 2019.

[8] D. Boyd, S. Golder, and G. Lotan. Tweet, tweet, retweet: Conversational aspects of retweeting on twitter. In *43rd Hawaii International International Conference on Systems Science (HICSS)*, pages 1–10. IEEE Computer Society, 2010.

[9] M. Cui. Introduction to the k-means clustering algorithm based on the elbow method. *Accounting, Auditing and Finance*, 1(1):5–8, 2020.

[10] D. Davidov, O. Tsur, and A. Rappoport. Enhanced sentiment learning using twitter hashtags and smileys. In *23rd International Conference on Computational Linguistics (COLING)*, pages 241–249, 2010.

[11] G. Drakopoulos, P. Gourgaris, and A. Kanavos. Graph communities in Neo4j. *Evolving Systems*, 11(3):397–407, 2020.

[12] A. Georgiou, A. Kanavos, and C. Makris. Finding influential users in twitter using cluster-based fusion methods of result lists. In *14th International Conference on Artificial Intelligence Applications and Innovations (AIAI)*, volume 519, pages 14–27. Springer, 2018.

[13] A. Java, X. Song, T. Finin, and B. Tseng. Why we twitter: Understanding microblogging usage and communities. In *9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis*, pages 56–65, 2007.

[14] I. T. Jolliffe and J. Cadima. Principal component analysis: A review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016.

[15] E. Kafeza, A. Kanavos, C. Makris, G. Pispirigos, and P. Vikatos. T-PCCE: twitter personality based communicative communities extraction system for big data. *IEEE Transactions on Knowledge and Data Engineering*, 32(8):1625–1638, 2020.

[16] E. Kafeza, A. Kanavos, C. Makris, and P. Vikatos. T-pice: Twitter personality based influential communities extraction system. In *IEEE International Congress on Big Data*, pages 212–219, 2014.

[17] A. Kanavos, G. Drakopoulos, and A. K. Tsakalidis. Graph community discovery algorithms in neo4j with a regularization-based evaluation metric. In *13th International Conference on Web Information Systems and Technologies (WEBIST)*, pages 403–410, 2017.

[18] A. Kanavos, A. Georgiou, and C. Makris. Estimating twitter influential users by using cluster-based fusion methods. *International Journal on Artificial Intelligence Tools*, 28(8):1960010:1–1960010:26, 2019.

[19] A. Kanavos and I. E. Livieris. Fuzzy information diffusion in twitter by considering user's influence. *International Journal on Artificial Intelligence Tools*, 29(2):2040003:1–2040003:22, 2020.

[20] B. Krishnamurthy, P. Gill, and M. F. Arlitt. A few chirps about twitter. In *1st Workshop on Online Social Networks (WOSN)*, pages 19–24, 2008.

[21] A. Likas, N. Vlassis, and J. J. Verbeek. The global k-means clustering algorithm. *Pattern Recognition*, 36(2):451–461, 2003.

[22] T. K. Moon. The expectation-maximization algorithm. *IEEE Signal Processing Magazine*, 13(6):47–60, 1996.

[23] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14 (NIPS)*, pages 849–856, 2001.

[24] F. Nielsen. Hierarchical clustering. In *Introduction to HPC with MPI for Data Science*, pages 195–211. 2016.

[25] J. Pöschko. Exploring twitter hashtags. *CoRR*, abs/1111.6553, 2011.

[26] R. Priedhorsky, A. Culotta, and S. Y. D. Valle. Inferring the origin locations of tweets with quantitative confidence. In *17th ACM Conference on Computer Supported Cooperative Work (CSCW)*, pages 1523–1536, 2014.

[27] F. Raiber and O. Kurland. The correlation between cluster hypothesis tests and the effectiveness of cluster-based retrieval. In *37th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 1155–1158, 2014.

[28] D. A. Reynolds. Gaussian mixture models. In *Encyclopedia of Biometrics*, pages 659–663. 2009.

[29] X. Rong. word2vec parameter learning explained. *CoRR*, abs/1411.2738, 2014.

[30] P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

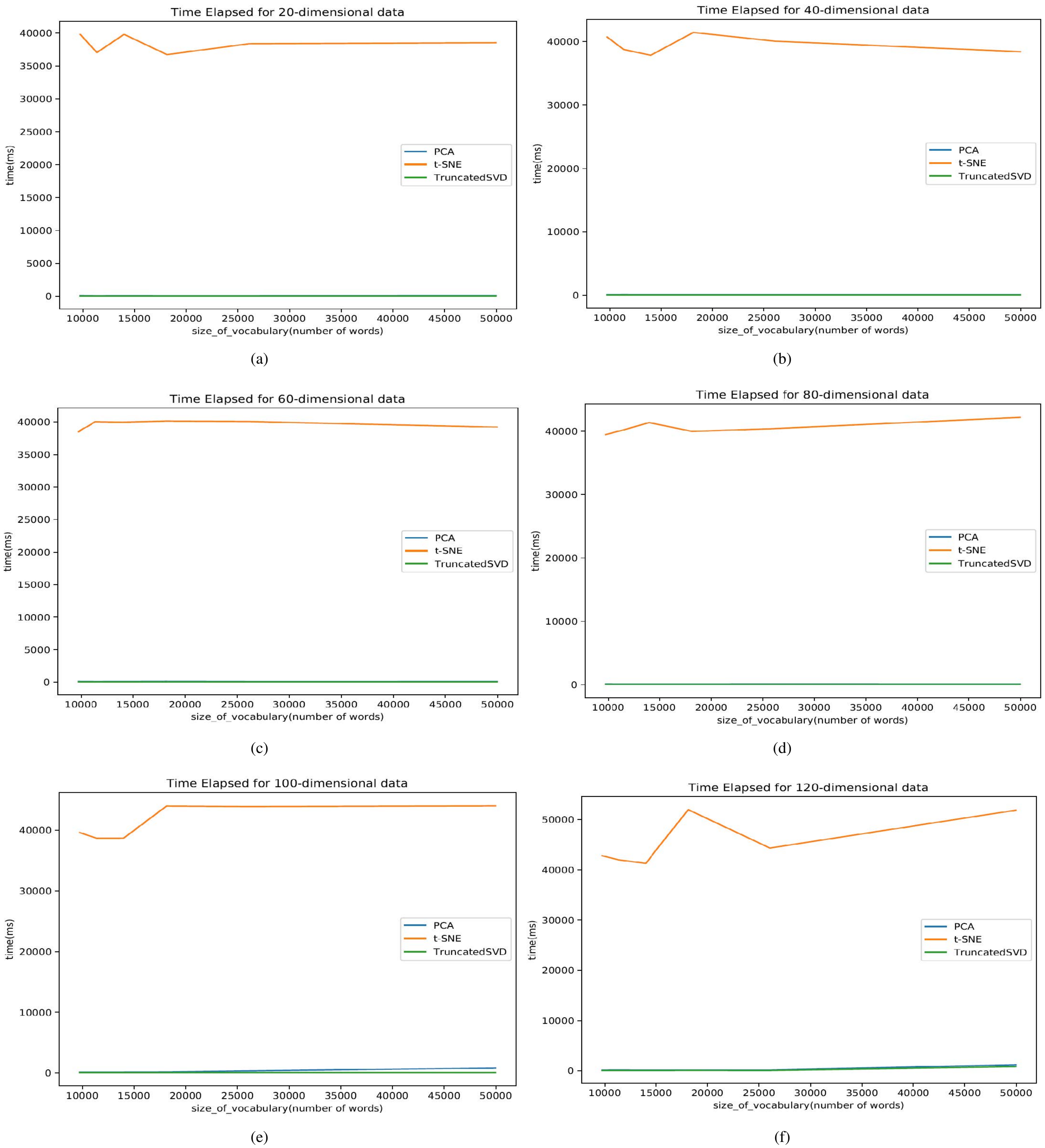
[31] S. M. Savaresi and D. Boley. On the performance of bisecting k-means and PDDP. In *1st SIAM International Conference on Data Mining (SDM)*, pages 1–14, 2001.

[32] E. Schubert, J. Sander, M. Ester, H. Kriegel, and X. Xu. DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems*, 42(3):19:1–19:21, 2017.

[33] B. Thompson and D. Yao. The union-split algorithm and cluster-based anonymization of social networks. In *ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, pages 218–227, 2009.

[34] C. J. van Rijsbergen. *Information Retrieval*. Butterworth, 1979.

[35] S. Wasserman and K. Faust. Social network analysis: Methods and applications. 1994.



(a)

(b)

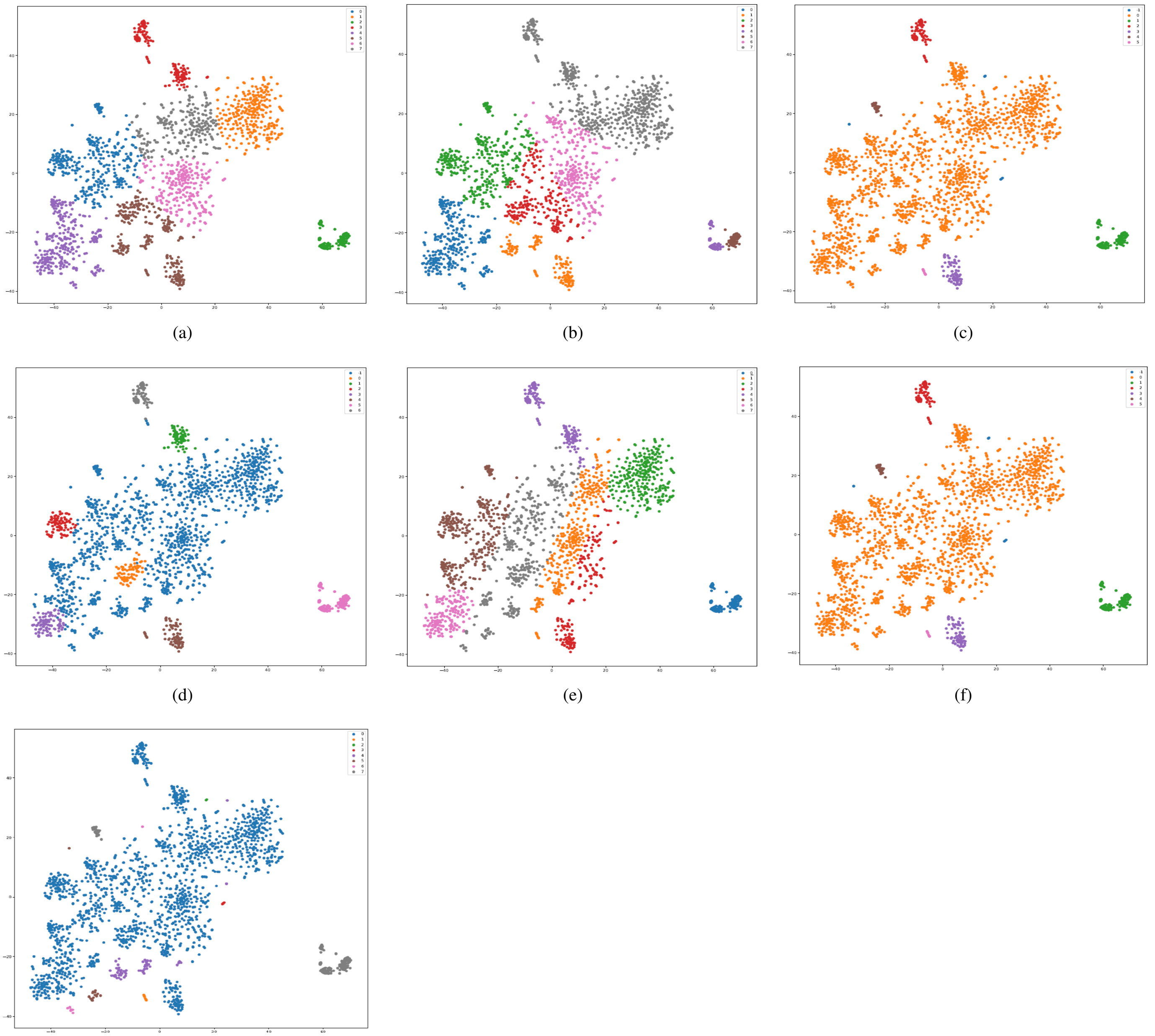
(c)

(d)

(e)

(f)

Fig. 6. Time elapsed for Data having Dimensions equal to (**a**) 20, (**b**) 40, (**c**) 60, (**d**) 80, (**e**) 100, (**f**) 120



(a)

(b)

(c)

(d)

(e)

(f)

(g)

Fig. 7. Clustering Algorithms: (**a**) k-Means, (**b**) Bisecting k-Means, (**c**) DBSCAN, (**d**) OPTICS, (**e**) Gausian Mixture Model, (**f**) Hierarchical Clustering, (**g**) Spectral Clustering