Journal of Soft Computing and Decision Analytics, Volume 1, Issue 1 (2023) 139-160

Journal of Soft Computing and Decision Analytics

Journal homepage: www.jscda-journal.org
ISSN: 3009-3481

Spectral Clustering Approximation For Large Scale Crew Disruption Data Of An Airline Company For Intelligent Crew Recovery

Ahmet Herekoğlu[1,*], Özgür Kabak [2]

[1] Department of Industrial Engineering, Faculty of Management, Istanbul Technical University, 34367 Istanbul, Turkiye

[2] Department of Industrial Engineering, Faculty of Management, Istanbul Technical University, 34367 Istanbul, Turkiye

ARTICLE INFO

Article history:
Received 19 August 2023
Received in revised form 2 September 2023
Accepted 5 September 2023
Available online 6 September 2023

Keywords:
Airline Crew Disruptions; Data Science; Machine Learning; Graph Clustering; Spectral Clustering

ABSTRACT

In the airline industry, after fuel costs, the crew costs constitute airlines' second-highest cost items. For this reason, an airline needs to manage the valuable crew resource efficiently. Deviations from plans are fact in airline business and fixing deviations from crew schedules that occurred during operations by minimizing the crew-related delays and associated costs is one of the most important operational burdens of airlines. In this context, the analysis of crew disruption data is vital in order to find disruption characteristics. Clustering analysis is one of the key methods for analyzing the disruption characteristics. In this context, although there have been satisfactory studies in the literature and applications in the industry for small and medium-sized airlines, there is no good solution or industry practice for airlines with extensive networks and fleets. This study aims to analyze and categorize large-scale crew disruption data of a European airline. The relationship between categories of crew disruption and variables such as flight and crew types etc., are determined, and the disruption characteristics are revealed. For this purpose, clusters hidden in the large data set are extracted by spectral clustering. Due to the large size of the input data, a new approximation approach for spectral clustering is introduced. With the help of this new approximation approach, spectral clustering techniques are applied within a limited computational power and time frame as most real world scenario require. Even if the data set is gathered from one airline, the characteristics that are derived from the data is representing most of the cases an airline may face today. and will serve as a basis for further estimation and analysis of crew disruption.

[1*] *Corresponding author.*
E-mail address: herekoglu@itu.edu.tr
https://doi.org/10.31181/jscda11202315,

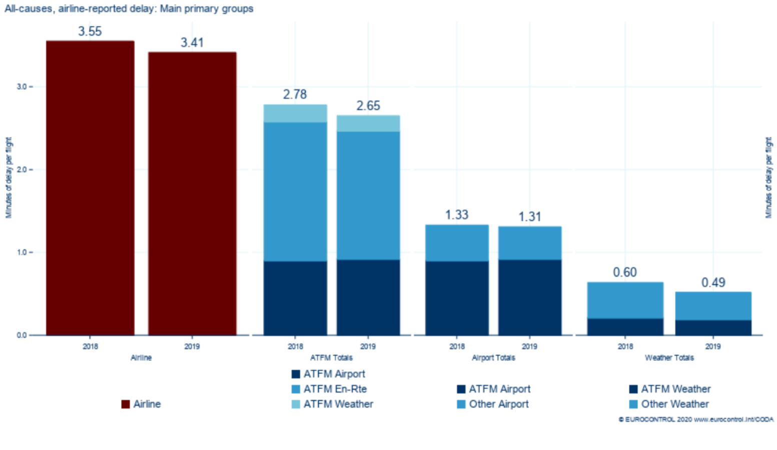
1. Introduction

Thanks to globalization, new travel opportunities and economic development have increased the interest in aviation industry and air transportation. Increasing interaction between developing Asian countries, especially China and India, where there are dense populations, and the United States, which is still the world's financial center, increases the need for transportation between the two points day by day. Even if the COVID-19 pandemic introduces significant changes in the aviation sector [1], according to the pre-pandemic trends in air transport, the International Air Transport Association (IATA) reveals that the number of passengers could increase to 8.2 billion in 2037 [2].

The increase in passenger numbers and compliance to regulations on passenger rights make transformation inevitable for airlines. For that purpose, airlines should rearrange and manage their all resources to be compliant with the change. The most critical resources of commercial airlines are listed as crew and aircraft, which are the essential components of operational efficiency in combination with passengers [4]. The expected rise in the aviation industry after the COVID-19 pandemic also forces airlines to extend their fleet size and increase their operational capacities. Growth in the volume of operations also increases the total size of airline crew, especially flight crew which can easily be identified by increasing total crew costs [5].

Most airlines face crew costs as the second biggest cost item after fuel costs, and small gains in crew cost margins lead to profound positive advances in operational expenses [35]. That situation is the primary motivation of airlines for searching for robust and cost-effective schedules which is a complex and challenging to solve by nature [6]. On the day of operation, unpredictable events such as bad weather conditions, aircraft failures and crew absence can cause deviations from the schedules [8]. These events are called as "Disruption". Disruptions such as flight delays/cancellations and the costs caused by these disruptions are one of the main challenges faced by the industry [8]. According to the report published by Eurocontrol, known as the European Air Navigation Safety Organization, delay statistics are getting worse in the long-term performance analysis even if there are minor and local improvements [9].

As disruptions such as delays are the primary and fundamental factor in passenger satisfaction and the financial situation of the airline, aviation companies are allocating valuable resources for analyzing disruptions and taking necessary actions [10]. As it can be seen from Figure 1, the main primary groups of disruptions such as delays are due to airlines' operations [9]. Therefore, this fact drives aviation companies to look into their way of handling disruptions and understanding them.


All-causes, airline-reported delay: Main primary groups

© EUROCONTROL 2020 www.eurocontrol.int/CODA

Fig. 1. Primary Delay Groups

Traditional solution approaches can provide solutions to problems faced by airlines; they are rarely able to provide enough insights on realities hiding in the daily operation. Instead of these traditional solutions, it is highly advised to build customized models based on the feedback provided by people responsible for daily operation [11]. In order to build that customized models, it is very vital to understand the problem itself in great detail.

This paper aims to examine the characteristics and understand the inner dynamics of the flight crew disruptions in a large airline operation since crew-related problems can be widely seen in disruptions of airlines [7]. Within the scope of this study, data gathered from a large-scale airline company covering thousands of flights between 2018 and 2019 is analyzed, and a common characteristics for flight crew disruption is proposed. By utilizing the data model as a graph, clustering techniques are applied in order to understand features and relationships such as categories hidden in the disruption data. After all, with the help of the model proposed, findings of graph clustering are interpreted for general-purpose utilization in other studies, especially studies focusing on deep learning field.

Data gathering is the most essential part of the study. Still, airline companies are not very eager to share their operational disruption data in large amounts and without limitations which are the main obstacles in this study's analysis. Because of General Data Protection Regulation (GDPR) and similar regulations, airline companies share only a masked version of their operational disruption data which makes domain knowledge a necessity to interpret the data successfully. This paper also contributes by introducing findings of large-scale aviation operation data set since data ownership and sharing is one of the biggest obstacles in the data-driven approaches and machine learning practices in the aviation sector [1].

Even there are many studies on crew recovery in the literature; there are not any published studies solely aiming at data analysis of flight crew disruptions. As already mentioned, due to the lack of the required large volume of data, no published models on flight crew disruptions are found in the field. As an early work in the data analysis field of airline crew disruptions, this article will provide new opportunities for the upcoming studies. It is thought that this study will provide satisfactory answers because flight crew disruption data of a large airline is used. The data is a large-scale set covering a total flight crew size of about 20,000, 1500+ flights daily and 300+ aircraft for the pre COVID-19 times. It is also believed that this study will constitute a basis for the following industry practices.

This study is organized as follows: Section **2. Methodology** briefly describes airline disruptions and recovery and summarizes text processing, graph structure, and graph clustering. Lastly, in this section, we propose a new approximation method for spectral clustering algorithm to process large amounts of textual data. Results covered in this study are shown in Section **3. Result**. This section also interprets the results reported by the clustering phase. Lastly, Section **4. Conclusion** concludes the research and discusses further research opportunities.

2. Methodology

In this section, disruption and recovery concepts are briefly explained. The definition of disruption, types of disruption are discussed, and actions needed to eliminate these problems, simply recovery processes, are defined. Lastly, all information is combined under a high-level context.

2.1. Disruption

A situation that forces executers to deviate from the planned state during the operation is generally called a disrupted situation or simply disruption [12]. The deviation's root cause is called disruption source [13]. Accidents, incidents, capacity constraints, diseases, financial problems, geological events, IT systems, security issues, strikes, and weather conditions are the primary and most known possible disruption sources [3]. Disruptions are also dependent on the airline resources such as aircraft, crew, and passenger. Maintenance, flight delay, a problem with airport infrastructure are some of the sources of disruption, but it can be easily seen that disruptions become connected with events far beyond the originated source [4]. This distributed nature of disruption leads to a chaining effect [14].

The situations which lead to disruptions can be summarized as follows [8]:

- Problematic Aircraft: An aircraft that could not complete its original plan
- Problematic Flight: A flight that cannot take-off and/or arrive at the scheduled time due to an unexpected event never takes place
- Problematic Airport: The airport where the problematic flight takes place
- Problematic Passenger: Passenger who lost one or more flight connections due to a problematic flight
- Problematic Crew: The crew that could not complete the original plan

In this study, we are mainly focused on the intersection subset of the problematic crew and problematic flights or simply flight disruptions which can be defined as disruption originated by a change in flight timetables such as flight cancellation or a 2+ hours delay in the operational window [3]. Possible disruptive events such as aircraft type change, cancellation, or retiming of a flight is a multi-objective process where passengers' inconvenience should be minimized, and airline resources such as crew and aircraft should be utilized effectively [4]. Flight disruptions that lead to problematic crew are investigated and analyzed as the focus of this study.

2.2. Recovery

Based on figures provided by Amadeus, a global technology provider for the travel industry, deviations from plans cost a total amount of $60 billion per year, which is almost 8% of overall total commercial airline revenue [3]. Negative effects of deviations or disruptions should be minimized by making decisions to get back on track as planned by monitoring the uncertainty [3]. Making necessary decisions for replanning and rebuilding plans close to operations day is called as Disruption Management [4], [15]. In disruption management, the overall process consists of identifying and classifying the problem based on criteria, finding out possible solution options and applying the solution which is the most applicable one. The last phase is also known as the recovery phase [3]. Disruptions lead to plans that can no longer be implemented during the operation. The process of finding new plan that is revised and minimizes the harmful effects of disruptions while at the same time taking the constraints and goals of the developing environment into account is called recovery management. The process of recovering the crew operation from the irregularities by minimizing the costs and following the necessary rules is called Crew Recovery[8]. The problem solved in this context is called the Crew Recovery Problem.

Because of the complex nature, analyzing characteristics of disruptions is one of the fundamental building blocks to understand sources of disruptions and find out structural problems which is vital for reducing the probability of occurrence and the effect of disruption [14]. It is also important to formulate more realistic models of problems and make required simplifications or assumptions to find a solution [16]. Analysis of operational realization is a very beneficial method for the punctuality of an airline. In this context, data gathered during operations contains valuable information for all kinds of variations and bottlenecks [17]. Finally, in order to have robust and cost-effective operation, having a detailed view on disruption mechanics can be easily classified as a must [14].

2.3. Text Processing and Graph Clustering

During the flight operations especially recovery operations, extensive amount of text data has been produced and saved into responsible systems. These data sometimes come from sensors and other types of machines and sometimes produced by human operators. Among available information, one of the most important ones is text data provided by crew trackers, which include details of the disruption and clues for recovery actions. This valuable piece of information is essential in order to be used for future disruptions. Utilizing machine learning techniques by converting text data into vector space is critical for extracting data patterns [57]. In order to understand patterns and relationships among chunks of texts, data is processed based on the text processing techniques.

We model the text data as graph and conduct graph analysis on text data about flight crew disruption provided by crew trackers. Graph analysis is one of the key and efficient tools for investigating generic features, characteristics and connectivity of complex networks such as air traffic, which are very dynamic by nature [18]. Since disturbances in transportation networks have huge effects due to the connected and dependent nature of these networks, finding relationships between different network attributes is crucial for understanding the data [20]. Therefore, graph is an adequate representation of complex data structures and applying clustering techniques on graphs reveals information hidden in them [19].

2.3.1. Text Processing

Text processing techniques such as summarization, text minig, text classification or text clustering are active and wide research areas especially with the development in Natural Language Processing applications. This area aims to convert the large set of documents into manageable chunks without sacrificing essential features and information in the original documents by modeling the documents set as graphs [36]. It also includes methods for extracting information from texts that are not easily found [58].

Text clustering which is also called document clustering [59] is multi disciplineary clustering technique based on information retrieval, natural language processing and machine learning [38]. In document clustering, document collections are grouped together where the same documents in the group have similar topics [39].

2.3.1. Graph Clustering

A graph is defined as pair of sets consisting of vertices and edges [21]. Graphs are denoted as

$$G = (V, E)$$

where V is the set of vertices and E is the set of edges.

$$n = |V|$$

is called the order of graph and it is simply the number of vertices. As v and t are the endpoints of edges, if v,t pair is unordered then it is **undirected graph** (digraph) otherwise **directed graph** [21].

The path is sequences of vertices with an edge connecting vertices consecutively. If every pair of vertices has at least one path connecting each other, this graph is known as **complete graph** [22]. On the other hand, if there exists at least one path connecting every vertice combination, this graph is called **connected graph** [23].

$$|E| = m$$

is known as the size of graph [21].

In a weighted graph, there exists a function that assigns a weight on each edge. The weights of edges are used to identify similarities inside a graph. Although different types of functions are used for deciding on the similarity of vertices in a graph such as sigmoid function, the definition of similarity and selection of the similarity measure is highly dependent on the domain of research [21], [22]. In this study, we investigate a weighted undirected complete graph.

Clustering is a set of exploratory data analysis techniques that have been utilized in many research fields such as social sciences, psychology or biology. The primary motivation behind the clustering techniques is to find similar behavior in empirical data by labeling data points as similar if they are in the same group and dissimilar otherwise [24]. Formally, identifying the underlying structures in the heterogeneous data by dividing data based on similarity measures into formerly unknown groups is called clustering [21]. The groups found by running clustering techniques are known as clusters. The key idea behind the clustering algorithms is having maximum similarity inside clusters and minimum similarity between clusters [25]. Since the exact description of what transforms a collection of items into cluster is not clear [39], it is not a common expectation to have clusters with strong similar concepts [39].

Graph clustering is one of the famous research topics of today as the usage of graphs is increasing day by day as can be seen in social networks, supply chains and electronic commerce. It can be defined as finding similar nodes in a graph [40]. Finding vertex partitions in a graph is graph clustering which is a pattern recognition problem [26]. Graph clustering is based on the idea of having many within-cluster edges and fewer between clusters [21]. Although graph clustering is a highly sophisticated research area, there is no universally accepted definition of a good cluster in the literature [21]. Optimal solutions to graph clustering problems are NP-hard and require high computational power in terms of large amount of CPU and memory resources. As the graphs' size increases, manuel analysis of graphs becomes impossible [40].

Spectral clustering methods are being utilized due to the efficient relaxation capacity in order to cluster the graphs without manuel intervention [25]. These methods have shown high performance while partitioning graphs and are easy to implement by utilizing linear algebra software packages [24], [25]. On the other hand, their application to large-scale problems has not been common yet [22].

The documents to be clustered are compared with each other according to the similarity measures. The most common distances used for similarity measures for text processing are cosine and euclidean distances [37].

Cosine Distance: One of the most common distance measures when text documents are modeled as vectors. Cosine distance between vectors calculated based on the angle between document vectors. The cosine distance of vectors v_1 and v_2 is calculated below [37], [39].

$$cosine(v_1, v_2) = v_1 * v_2 / |v_1||v_2|$$

where * is the vector dot product, and | | is the vector length [54].

Euclidean Distance: General distance measure based on euclidean space for all kinds of data analytics problems with multi-dimensional data. The Euclidean distance is calculated as the formula given below. The Euclidean distance of vectors d_1 and d_2 is:

$$euclidean(d_1, d_2) = |d_1 - d_2|^2$$

Some studies suggest fractional distances for calculating similarities [55]. To sum up, in high dimensional space, the selection of distance calculation method is not clear and is mostly based on heuristical approach [55].

2.4. Spectral Clustering

Spectral clustering is becoming very popular and applied to different problems because of its easy and effective mechanism, making it a common clustering technique [41], [43]. It is a clustering algorithm that utilizes spectral features of graph laplacian by finding connected components, leading to clusters of data points where similar data points are grouped based on the eigenvectors of the similarity matrix [42], [36].

2.4.1. Theory of Spectral Clustering

Spectral clustering takes eigenvectors related to eigenvalues of normalized Laplacian or other modifications of the adjacency matrix representing the graph structure into consideration while partitioning the graph [21]. Even if there are some variants of well-known clustering algorithm K-means to overcome specific issues, k-means has some certain drawbacks such as dependency on initialization method and the risk of stuck at local optimum [36]. Also, high dimensionality is not very friendly with K-Means. It leads to poor results even if dimensionality reduction techniques such as Principal Component Analysis improves results, which is also not effective against complex data structures [49].

The ease of implementation and outperforming performance by using standard linear algebra methods are the main advantages of spectral clustering [49]. It is also able to work with different geometries [51]. Because of that spectral clustering is used where K-means can not able to work [52]. On the other hand, heavy need on sorting and decomposition makes the spectral algorithm less performant for large data sets with high dimensions, which are also known as the "curse of dimensionality" and makes it impossible to distinguish neighborhood in a meaningful way [49], [38], [43], [53], [54], [55], [56]. As of our best knowledge, there is no single definition for at what limit high dimensionality starts [54].

Spectral clustering generally consists of three steps which are [43]:

- Preparing graph
- Spectral embedding
- Clustering

In order to utilize spectral clustering methods on graphs, similarity graphs that are based on the pairwise similarities are utilized. There are three common similarity graphs in the spectral clustering literature which are [37] :

- e - Neighborhood Graph: nodes of which pairwise distances are smaller than e are connected.
- k - Nearest Neighbor Graph: nodes are connected with the nearest k neighbors.
- Fully connected Graph: all nodes are connected.

The connection between nodes of a graph is represented by an adjacency matrix known as affinity matrix [38]. In order to eliminate correlations while preparing the adjacency matrix with the complex data structures in high dimensional space, kernel functions are applied [48]. However, the choice of the kernel is still a debating item in the literature [38]. In other words, the benefit of applying a kernel is projecting features into high dimensions [54]. The RBF kernel is a way to define a normalized Laplacian matrix from a feature set distinguished based on Euclidean similarity [48].

Eigengap heuristic is a way to determine the number of clusters while applying spectral clustering [48]. It tries to find the largest eigengap value when eigenvalues are ordered [48]. Since eigengap heuristic is mostly able to process well-defined features but not complex features with different scales, different versions of it have been developed to cope with real-world problems [41]. Even if local scaling is an efficient way to build models by taking every data points' measurement scales, it requires more computational power than global scaling since it introduces extra steps while generating affinity matrix [41]. Besides, it also introduces new parameters such as the number of neighbors, which is a vital element for reaching good results. Even if it is usually set as 7, this parameter is also domain dependent [42].

Although there are different spectral clustering algorithms in the literature by various researchers [24], the spectral clustering algorithms are categorized into three groups [60]:

- Recursive Spectral: Recursively divides data into two groups based on a single eigenvector.
- Multiway Spectral: Multiple eigenvectors are taken into account in order to generate multiple partitions.
- Non-spectral: Simpler group techniques other than the former two groups.

As it is stated in the literature, the application of spectral clustering on large data sets and large graphs leads to complexity and high resource consumption [51-53]. In order to overcome the problem occured when spectral clustering is applied to big data approximation methods are introduced.

2.4.2. Approximation

It can be easily found out that a large data set with millions of rows with double precision requires more than 80 GB of memory even if a small subset is used [47]. This need is basically because of the heavy calculations made while sorting the affinity matrix and decomposing the Laplacian matrix. The complexity of building an affinity matrix is $O(n^2p)$ whereas decomposing the eigenvalues and eigenvectors requires $O(n^3)$ time [44], where n is the number of data items. Spectral clustering methods still are not scalable enough to handle big data [47].

In order to overcome the problems between spectral clustering and big data, numerous approximation methods have been introduced with acceptable accuracy and lighter computational cost [53]. In order to reduce the size of the graph and computational resource need, nodes are randomly selected from the graph, clustered according to distance with the selected nodes and eigen decomposition is applied on the centroids of these clusters [44]. Another way is to approximate by selecting good representatives of data points via sampling [51-43].

Unfortunately, approximations methods come with a drawback of limited performance improvements or taking assumptions into account while running [43]. On the other hand, the quality metrics of our approximation algorithms can be seen below as it can be found in the literature [43]:

- Less computational power compared to the regular version of the algorithm
- Acceptable solution quality

The literature on approximations reveals that current approximation methods are run on small or medium datasets and do not take the computational speed into consideration while introducing additional complexity [66]. In the next section, a new approximation technique which both requires less computational power and provides acceptable solution quality is proposed.

2.5. Proposed Approximation Method for Spectral Clustering The proposed method is a self-tuning approximation method for spectral clustering algorithm which firstly finds out hyperparameters and then runs for providing acceptable solutions after processing large amounts of text data. Benchmark results based on a common text data set are also provided in this section.

2.5.1. Proposed Approximation Method

In this section, we presented a new approximation approach for spectral clustering which requires less computational power and is fast enough to provide acceptable solution quality, as we already mentioned.

The proposed algorithm consists of different stages that are modeled so that it can be independent of the application and learn hyperparameters from the data it is working on. For this purpose, the best needed hyper parameter values, especially the number of clusters, are determined by running simulations with small-sized sets taken from the data in order to learn hyperparameters in the initialization stage. The Eigenvector Selection step is used to determine the number of clusters, and the Clustering step is used for clustering small-sized data. With Processing, the big data stack is processed and the final clusters are determined with the help of the Clustering step. However, at this stage, with the Eigenvector Selection procedure, the number of clusters is also updated throughout the process, if necessary.

Algorithm 1 Mini Batch Spectral Clustering Approximation

```
Input: Set of Documents
Output:  Clustered Documents
Require: documents, batch_size, number_of_iterations, sigma_values
data = read_data(file_name)
data = clean_data(data)
data = lower_text(data)
data = remove_non_letter_characters(data)
data = remove_stopwords(data)
vectors = vectorize_data(data)
vectors = normalize_vectors(vectors)
scores = Initialization(vectors, batch_size, experiment_iterations, sigma_values)
ideal_sigma = get_ideal_sigma(scores)
upper_limit, lower_limit = get_limits(scores)
data = Run(vectors, batch_size, ideal_sigma, upper_limit, lower_limit)
data = load_data(documents)
data = clean_data(data)
data = lower_text(data)
data = remove_non_letter_characters(data)
data = remove_stopwords(data)
vectors = vectorize_data(data)
vectors = normalize_vectors(vectors)
scores = Initialization(vectors, batch_size, number_of_iterations, sigma_values)
ideal_sigma = get_ideal_sigma(scores)
upper_limit, lower_limit = get_limits(scores)
clustered_data = Run(vectors, batch_size, ideal_sigma, upper_limit, lower_limit)
return clustered_data
```

As it can be seen from Algorithm 1, the approximation algorithm takes a set of text data or simply documents as input and outputs clustered documents. Documents are read first, then preprocessed with cleaning such as eliminating empty rows etc. Afterwards, text data is converted to lower case, all non-letter characters and stopwords are removed. The processed data set is converted to word embeddings, also known as word or documents vectors and then they are normalized so that all values are between -1 and 1.

Word embedding is a relatively new technique that provides good performance on NLP models by solving common problems such as scalability. It is used to convert documents into vectors by utilizing pre-trained model based on neural network architecture trained on large scale text data [32]improving, [63], [64]. In this study, Facebook's FastText library is used in order to convert documents into vectors by utilizing word embeddings [65].

Initialization is the step where the algorithm applies search for best parameter values in a predefined space. This step produces all kinds of metrics in order to decide ideal values for affinity matrix calculations and the number of clusters. A user-defined number of iterations are run on normalized vectors and aforementioned parameters are found. After defining parameters, the actual clustering run is applied to the data set and results are appended to the document set as a new column.

The flow of the initialization phase is seen in Algorithm 2; the initalization procedure tries different sigma

Algorithm 2 Initialization

Input: Document Vectors
Output: Parameters

```
Require: vectors, sigma_values, batch_size, maximum_iteration_numbers
number_of_clusters = 0
upper_limit = batch_size
lower_limit = 2
scores = [ ]
mini_batch_data = get_random_data(data)
for sigma in sigma_values do
    for iteration in range(maximum_iteration_numbers) do
        number_of_clusters, normalized_n_eigen_vectors = Eigen_Selection(vectors, sigma, stabilized = False, lower_limit,upper_limit)
        clusters = Cluster(number_of_clusters, normalized_n_eigen_vectors)
        run_time = calculate_run_time()
        silhouette_score = get_silhouette_score(mini_batch_data, clusters)
        calinski_harabasz_score = get_calinski_harabasz_score(mini_batch_data, clusters)
        davies_bouldin_score = get_davies_bouldin_score(mini_batch_data, clusters)
        improvement_per_cluster = calculate_improvement_per_cluster(silhouette_score, calinski_harabasz_score, davies_bouldin_score)
        scores.add(sigma, number_of_clusters, run_time, silhouette_score, calinski_harabasz_score, davies_bouldin_score, improvement_per_cluster)
    end for
end for
return scores
```

values, which is a parameter for calculating affinity matrix for predefined times. In all these iterations, samples from the main data set is selected and spectral clustering is applied. The result of the clustering process is evaluated with performance metrics and these metrics are provided to find best values for the parameters needed for the main run of the algorithm.

It is clear that for real world problems, the ground truth is not available at the algorithm's runtime. Because of this fact, internal indices are utilized for correctly align the algorithm. There are many internal indices in the literature; three of them which are widely used are listed below [46]:

- Silhouette index
- Davies-Bouldin index
- Calinski-Harabasz index

The internal indices can be used to tune the algorithm to provide high performance as a post-processing step. For example; Davies-Boulding and Silhouette indexes have certain applications for this purpose [48], [49].

In our initialization step, we use these three internal indices to find out which clusters are better than the others. Since these indices have different scales we normalize them with min-max normalization and have values on the same scale. The normalized values are summed to get a new feature which is called as "Score":

$$Score = Silhouette\_Score + Davies\_Bouldin\_Score + Calinski\_Harabasz\_Score$$

In order to find out what is the computational cost and gain related with every cluster we get from the algorithm, two new metrics are calculated as below.

$$Cost\_Per\_Cluster = Iteration\_Run\_Time/Number\_Of\_Clusters$$
$$Gain\_Per\_Cluster = Score/Number\_Of\_Clusters$$

After these calculations, we have enough information on what every new cluster provides how much improvement to our scores.

$$Improvement\_Per\_Cluster = Gain\_Per\_Cluster - Cost\_Per\_Cluster$$

These calculated metrics are provided to calculate values of ideal sigma, upper limit and lower limit. Those last two, upper and lower limits, are used to limit the number of clusters in a limited space and stabilize eigengap heuristics which becomes destabilized very easily. The ideal sigma is the value where the algorithm provides the best score. The upper limit is the maximum of number of clusters in scores where sigma is equal to the ideal sigma and the lower limit is the minimum one.

After calculating the parameters, the approximation algorithm is applied to the whole data set to find the real clusters which can be seen in Algorithm 3. This procedure takes all vectorized texts and divides them into chunks based on the batch size provided by the user. Every batch is clustered separately with spectral clustering and all clusters are added to data as a label. After all items in the data set are processed, centroids of clusters are calculated. These centroids are clustered with the K-Means algorithm rather than the whole data. The clusters provided by K-Means are broadcasted to the data set in order to have the final clusters.

There are two important subprocedure in the algorithm which are the selection of eigenvectors to be clustered and clustering the eigen vectors. The first subprocedure which is responsible part for calculating

Algorithm 3 Processing

```
Input: Document Vectors
Output: Clustered Documents
Require: data, batch_size, ideal_sigma, upper_limit, lower_limit
total_item_count = number_of_items_in_data
total_processed_item_count = 0
while total_processed_item_count $<$ total_item_count do
    mini_batch_data = get_random_data(batch_size)
    stabilised = False
    if number_of_clusters is same for n iterations then
        stabilised = True
    end if
    number_of_clusters, normalized_n_eigen_vectors = Eigen_Selection(vectors, sigma, stabilized, lower_limit, upper_limit)
    if number_of_clusters is in range(lower_limit, upper_limit) then
        lower_limit, upper_limit) = set_limits(number_of_clusters)
    end if
    clusters = Cluster(number_of_clusters, normalized_n_eigen_vectors)
    total_processed_item_count += batch_size
    update_data_with_clusters(mini_batch_data, clusters)
end while
centroids = calculate_centroids_by_clusters(data)
clusters = Cluster(number_of_clusters, centroids)
update_data_with_clusters(data, clusters)
return data
```

Algorithm 4 Eigenvector Selection

```
Input: Document Vectors
Output: Number of Clusters, Selected Eigenvectors
Require: vectors, sigma, stabilized, lower_limit, upper_limit
distances = get_euclidean_distances(vectors)
affinity_matrix = get_affinity_matrix(distances, sigma)
degree_matrix = get_degree_matrix(affinity_matrix)
laplacian_matrix = get_laplacian_matrix(affinity_matrix, degree_matrix)
eigen_values, eigen_vectors = get_eigens(laplacian_matrix)
if stabilized then
    number_of_clusters = get_eigen_gap(eigen_values)
end if
if number_of_clusters $\leq$ lower_limit then
    number_of_clusters = lower_limit
end if
if number_of_clusters $\geq$ upper_limit then
    number_of_clusters = upper_limit
end if
largest_n_eigen_vectors = eigen_vectors[number_of_clusters]
transposed_n_eigen_vectors = transpose_eigen_vectors(eigen_vectors)
normalized_n_eigen_vectors = normalize_eigen_vectors(transposed_eigen_vectors)
return number_of_clusters, normalized_n_eigen_vectors
```

spectral clustering parameters can be seen in Algorithm 4. This step is a classical spectral clustering implementation with normalized Laplacian matrix but includes a stabilization process by bounding the eigengap heuristic with parametric bounds.

The distances between vectors are calculated as Euclidean distances which are briefly discussed in Section Graph Clustering. In order to calculate the similarities, this distance is selected based on the benchmarks of different distance metrics discussed in the former sections. Simply, Euclidean distance method gives the best result in the experiments run during the design of the algorithm. After calculating distances, they are used to build an adjaceny matrix. We prefer to use a version of the adjacency matrix which is known as the weighted adjacency matrix, used in different studies such as analyzing high order of networks [29]. It is defined as:

$$A = [a_{ij}]$$

where A is $n \times n$ matrix and

$$A = \begin{cases} w_{ij}, & \text{the weight of edge if there is one between vertices } v_i \text{ and } v_j \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

In our study, we used distance as weights in the adjacency matrix. This decision is computationally efficient but has a significant drawback. Distance is not an indicator of similarity but dissimilarity. The distance-based measurement is converted to similarity and the normalized adjacency matrix is converted to an affinity matrix. Even if the adjacency matrix and affinity matrix are used interchangeably in literature [30], [31], [32], we use distance-based adjacency as a dissimilarity indicator and affinity as a similarity indicator.

Even if there are many sophisticated and complex similarity functions in literature, Gaussian Similarity Function [33] converts the adjacency matrix to an affinity matrix. The Gaussian function, defined below as kernel function, is commonly used since it is one of the best kernel applications to deal with fully connected graphs with complex statistical features [49]. The gaussian Kernel formula is [43], [53], [27]:

$$exp(-|p_i - p_j|^2/2\sigma^2) \tag{2}$$

Since Gaussian requires iterations in order to find out the ideal parameters [34] deciding on these parameters is a human-driven process [27], tirals based on different parameters values are already conducted in Initialization phase.

Affinity matrix is used to calculate the degree matrix(d_i), an important component for calculating Laplacian matirx of the graph represented with affinity matrix [24].

$$d_i = \sum_{j}^{n} a_{ij}$$

By using affinity and degree matrices, Laplacian matrix (L) is calculated which is used to find eigenvalues and eigenvectors [27].

$$L = D^{1/2} A D^{1/2}$$

If the algorithm is not stabilized yet, we utilize the eigengap heuristics to find out number of clusters. After getting the number of clusters, we update our upper or lower bounds. After that update step, eigenvalues are calculated in ascending order. Based on the values, the largest eigenvectors as many as the number of clusters are selected. The eigenvectors are firstly transposed and then normalized. The procedure sends back the number of clusters and normalized eigenvectors to main flow.

Algorithm 5 Clustering

```
Input: Document Vectors
Output: Number of Clusters, Selected Eigenvectors
Require: number_of_clusters, normalized_n_eigen_vectors
clusters = apply_kmeans_clustering(normalized_n_eigen_vectors, number_of_clusters)
return clusters
```

The second important subprocedure is the clustering step which is utilized after calculating desired number of clusters and normalized eigenvectors. The flow of the subprocedure can be seen in Algorithm 5. This step divides the eigen vectors into the given number of clusters.

2.5.2. Benchmarks

Clustering is one of the challenging fields in machine learning compared to supervised learning due to the unknown ground truth and lack of a universally accepted definition of a good cluster [50]. In order to find out how good the results of clustering algorithms, cluster validation of cluster evaluation metrics are utilized. Since there is no universal best clustering metric and algorithm [40] which can be applied to all kinds of problems, this process is a mandatory but complex and long process. The quality of clusters and the performance of the clustering algorithm depend on parameters such as choice of similarity measures etc. [46].

Two types of performance metrics are utilized for clustering algorithms known as internal end external indices. When ground truth is known before the clustering, external indices are used and internal ones otherwise [46]. As external index for the benchmarking purposes, we have selected Normalized Mutual Information (NMI) score as a reference score for the tests because it has already been used as a metric for comparing spectral clustering algorithms and we already know the ground truth for benchmark test set [45].

The experiments are run on a workstation with 3.4 GHz Intel CPU and 32 GB memory running Ubuntu 20.04 operating system. The proposed algorithm runs 30 times and the average of the performance scores is reported. The data set for the benchmarks is Reuters-21578 data set which covers Reuter's 21.578 news documents [61]. We have selected documents with a single class and omit the documents with multiple classes. Since NMI scores for K-Means and Spectral Clustering algorithms are not given in the comparison study [45], we have made benchmarks with these algorithms by using reference implementations provided by the SKLearn library [62]. Our approximation method can correctly cluster 39% of documents clustered with K-means and 37% of documents clustered with the reference implementation of spectral clustering.

The performance of our method is not as good as the ones given above in terms of clustering performance metrics. It should be noted that the proposed method, which is an approximation method, is compared to the exact methods. We do not expect to beat the exact methods in terms of clustering performance metrics. The key point is that our approximation method can process large data sets that can not be handled by most of the exact methods. As it can be seen from the literature, approximation and parallel algorithms are compared based on computational efficiency such as CPU time or memory consumption [66].It is already mentioned in Section 2.5.1 that an approximation algorithm should provide not the best but acceptable performance in a reasonable time. According to our applications that is made here for benchmarking and presented in Section 3, the method presented here provides acceptable results in a proper time frame. Our main aim with this proposal is to create a baseline for the studies focusing on clustering algorithms that work with large data sets and be able to learn hyper parameter values from these sets. As its clustering performance is acceptable and it can provide computational efficiency, our method can be considered as a starting point for further studies on clustering with large data sets.

3. Result

Data structures in the transportation industry can be easily modeled as graphs, networks of nodes. Some studies in literature aim to understand structures in networks such as the one covering air transportation network [28]. Generally, data sets being analyzed are not suitable for building graph at the beginning. Data transformation methods are applied to convert data set into a building graph is possible in order to manage computationally demanding graph processes [21]. A weighted undirected complete graph is prepared which is connecting 25.000 nodes in sample data with more than 400,000,000 edges.

In the analysis within the scope of the study, crew operation data of a large-scale European airline is used whose name can not be disclosed due to the airline's legally-binding procedures. This data covers more than 60 millions entry elated with crew disruptions between 2012 and 2019. Since the airline has different types of narrow and wide body aircrafts covering many types flying all around the world and it flies to many different destinations all over the world, results produced from this data set can easily be generalized to other airlines having same aircraft types and flying the same destinations. In short, the structure of this large scale airline covers many characteristics found in the aviation sector. There are different disruption types in the data set such as flight, ground activity etc. As it is already indicated, flight disruptions are selected from this data set for analysis purposes.

A total of 22,592,770 crew disruptions related to flight duties, covering 2018 and 2019 have been selected due to the low data quality before 2018. After all data cleaning process, 3,250,000 document records are selected as mature enough for analysis.

The most frequent disruptions can be seen in Table 1.

Table 1

Crew Disruption Distribution

Disruption	Count	Frequency(%)
Late for Duty Check-in	4.008.976	17.78%
Flight Delay	2.970.943	13.18%
Min Connection Time in Domestic Stations	1.721.046	7.63%
Open Time for Flight Attendant	1.459.628	6.48%
Duty End Time Change	1.302.066	5.78%
Duty Start Time Change	1.013.114	4.49%
Flight Cancellation	587.931	2.61%
Aircraft Change in International Station	587.031	2.60%
Open Time for First Officer	586.872	2.60%
Open Time for Cabin Chef	521.717	2.31%
Others	699.818	34.53%

The disruption data includes flight number, month of departure, day of departure, hour of departure, departure airport, crew type and disrupted rule. For instance if the first officer is late for duty start for the flight AB0001 departs in June, 5 at 20:00 from airport XYZ, this disruption is classified as "late for duty check in". If flight AB0002 departs in June, 5 at 21:00 from airport XYZ is 1 hour late, then this disruption is classfied as "flight delay".

It is impossible to process the volume of data we have mentioned before with the reference spectral clustering implementations such as the one in the SKLearn library [62] due to computational capacity limitations. We have applied the proposed methodology to the data set and successfully cluster the data. The initialization process is run 30 times for every sigma values in the sigma values set which includes values of $[0.001, 0.01, 0.1, 1, 2, 5, 7, 10, 20, 50, 70, 100, 150, 200]$. The initialization scores can be seen in Figure 2.

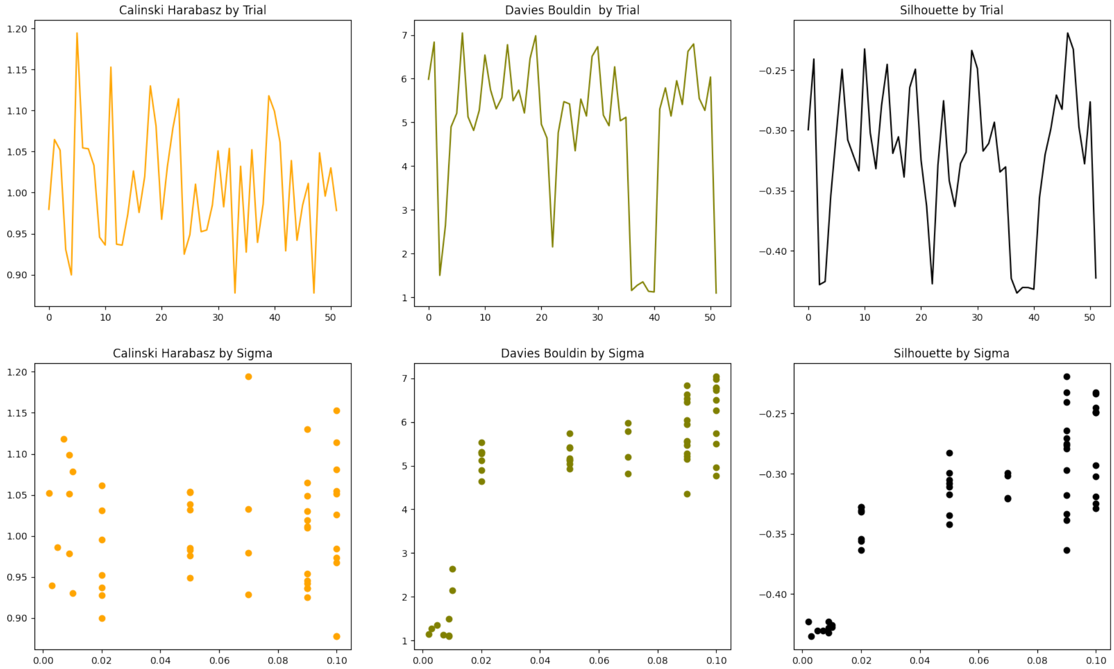


Fig. 2. Initialization Scores

The initialization scores are analyzed as defined in Section 2.5.1 and ideal sigma score is 0.1, upper limit as 92 and lower limit as 47. After the algorithm is run on the data set, it converges to the lower limit and finds 47 clusters. The total processing time is 2 hours 10 minutes for 3.250.000 documents with the configuration mentioned in Section 2.5.2. As of now there is not a universal reference document set and hardware set up for comparison the duration. Therefore, this duration is evaluated as expected because without proposed approximation method, it is not possible to process such amount of documents with current hardware setup.

Our findings summarized in this section will provide some insights to researchers dealing with the problem of crew recovery. Even if the data is gathered from only one airline both the volume of data and the characteristics of the airline making it possible to generalize these findings to most airlines. The utilization of narrow and wide-body aircrafts, large network of destinations and high number of crew make it possible to cover many daily problems most airlines face today.

The 47 clusters in 3.250.000 documents are investigated with airline experts in order to find the recovery actions or some insights for recoveries. As we found out, the main feature to decide on the recover actions is disrupted rule. Furthermore, crew type and departure airport are also important features for airline experts to decide on recovery actions. As an example, recency information which is known as formerly flown airports in a certain time frame, is initially clustered based on rule and airport information.

The close enough clusters are concetaneted if possible. For example, deassignment of cabin chief and flight attendant clusters are merged and named as "DEASSIGN CABIN CREW". All technical qualification clusters are also merged into "SWAP CREW WITH CREW HAS TECHNICAL QUALIFICATION" cluster. This human in loop process leads to 23 recovery actions listed below.

- ASSIGN CABIN CREW: Assign a new cabin crew
- ASSIGN COCKPIT CREW: Assign a new cockpit crew
- CANCEL FLIGHT: Cancel the flight disruption that occurs
- DEADHEAD: Transport crew as passengers between two stations
- DEASSIGN CABIN CREW: Deassign one assigned cabin crew
- DEASSIGN COCKPIT CREW: Deassign one assigned cabin crew
- DELAY FLIGHT: Delay the flight disruption occurs for a certain amount of time
- SWAP AIRCRAFT: Change the aircraft
- SWAP AIRCRAFT WITH AIRCRAFT HAS AVAILABLE SEAT: Change the aircraft with available seats
- SWAP AIRCRAFT WITH AIRCRAFT HAS RESTBUNK OR RESTSEAT: Change the aircraft with rest area for crew
- SWAP CREW WITH CREW AT SPECIFIC AIRPORT: Change assignment of crew with another one who is currently at a specific station
- SWAP CREW WITH CREW HAS CATEGORY QUALIFICATION: Change assignment of crew with another one who has CAT qualification
- SWAP CREW WITH CREW HAS HEALTH QUALIFICATION: Change assignment of crew with another one who has health qualification
- SWAP CREW WITH CREW HAS LANDING QUALIFICATION: Change assignment of crew with another one who has already landed at destination airport before
- SWAP CREW WITH CREW HAS LANGUAGE QUALIFICATION: Change assignment of crew with another one who has language qualification
- SWAP CREW WITH CREW HAS LINE CHECK QUALIFICATION: Change assignment of crew with another one who has passed in flight controls
- SWAP CREW WITH CREW HAS PASSPORT OR VISA: Change assignment of crew with another one who has passport/visa for the destination country
- SWAP CREW WITH CREW HAS SPECIAL AIRPORT QUALIFICATION: Change assignment of crew with another one who has specific qualification for the destination station
- SWAP CREW WITH CREW HAS TECHNICAL QUALIFICATION: Change assignment of crew with another one who has legitimate qualification for flying with a specific aircraft
- SWAP CREW WITH CREW HAS VALID FLIGHT CREW DOCUMENTS: Change assignment of crew with another one who has necessary crew documents for the destination station

- SWAP CREW WITH CREW NOT FLOWN TO AIRPORT IN THE MONTH: Change assignment of crew with another one who has not flown to the destination station in current month
- SWAP CREW WITH EXPERIENCED CREW:Change assignment of crew with another experienced one
- SWAP CREW WITH STANDBY CREW: Change assignment of crew with another one who is on airport / home standby duty

By using these actions, recovery strategies can be determined or solutions to be used during recovery can be modeled more accurately. In particular, during the preparation of recovery optimization models, more effective models will be produced by using the actions determined by this study as input to the model. For example, providing the listed actions to the optimization model as a constraint will result in the production of optimization models that produce faster solutions.

Lastly, more details about business rules, flight numbers or departure airports could not be revealed because of data sharing regulations that do not allow to state or mean name of the airline. If these results were shared, there would be a risk of finding out name of the airline with proper reengineering techniques.

4. Conclusion

In this study, crew disruption data of a large scale European airline is analyzed in order to find disruption characteristics and clusters in the data set produced by the aforementioned airline's operation. Clustering analysis is conducted and the disruption characteristics are revealed. To have the proper clusters, a modified Spectral Clustering algorithm is applied and based on summarization techniques, a summarized version of data is consolidated in order to use for future studies.

We believe that more computational power will lead to new insights about the data set but the methodology proposed in this study will be a basis for such researches.

Even if the data set is gathered from one airline, the characteristics that are derived from the data is representing most of the cases an airline may face today. Due to the wide range of aircraft types, large flight network and high number of flight crew, this study will serve as a basis for further prediction and recovery solutions.

As a continuing study, by utilizing deep learning techniques on the clustered data we have produced in this study, recovery actions and constraints that are hidden in the large data set are being planned to be revealed. AutoML techniques will be utilised in order to generate a neural network model for multi-class classification problem. Disruptions will be used as inputs for the model to predict appropriate recovery actions. We are planning to use these recovery actions and constraints as an input to crew recovery optimization model in order to have an intelligently reduced solution space leading to effective solutions for crew recovery problem.

Acknowledgements

This study was funded by Research Fund of the Istanbul Technical University (ITU-BAP) (Grant Number 42736). This article does not contain any studies with human participants or animals performed by any of the authors. The data that support the findings of this study are available from a large scale European airline but restrictions apply to the availability of these data, which were used under license for the current study, and so are not publicly available. Data are however available from the authors upon reasonable request and with permission of the aforementioned airline.

References

[1] Mitsokapas, E., Schafer, B., Harris, R., & Beck, C. (2021). Statistical characterization of airplane delays. *Scientific Reports*, *11*, 1-11. https://doi.org/10.1038/s41598-021-87279-8.

[2] IATA IATA Forecast Predicts 8.2 billion Air Travelers in 2037. (2018), https://www.iata.org/pressroom /pr/Pages/2018-10-24-02.aspx, [Online; accessed 30-July-2021].

[3] Jimenez Serrano, F., & Kazda, A. (2017). Airline disruption management: yesterday, today and tomorrow. *T*ransportation Research Procedia, *28* 3-10. https://doi.org/10.1016/j.trpro.2017.12.162.

[4] Kohl, N., Larsen, A., Larsen, J., Ross, A., & Tiourine, S. (2007). Airline disruption management—Perspectives, experiences and outlook. *J*ournal Of Air Transport Management , *13* 149-162. https://doi.org/10.1016/j.jairtraman.2007.01.001.

[5] Deveci, M., & Demirel, N. (2018). A Survey of the literature on airline crew scheduling. *E*ngineering Applications Of Artificial Intelligence , *74* 54-69. https://doi.org/10.1016/j.engappai.2018.05.008.

[6] Schaefer, A., & Johnson. (2005). Airline Crew Scheduling Under Uncertainty. *T*ransportation Science , *39* 340-348. https://doi.org/10.1287/trsc.1040.0091.

[7] Novianingsih, K., Hadianti, R., Uttunggadewa, S., & Soewono, E. (2015). A Solution Method for Airline Crew Recovery Problems. *I*nternational Journal Of Applied Mathematics And Statistics , *53 (4)* 137-149.

[8] Castro, A., Rocha, A., & Oliveira, E. (2014). A New Approach for Disruption Management in Airline Operations Control. *S*tudies In Computational Intelligence , *562*. http://dx.doi.org/10.1007/978-3-662-43373-7.

[9] Eurocontrol All-Causes Delay and Cancellations to Air Transport in Europe-2019. (2019), https://www.eurocontrol.int/publication/all-causes-delay-and-cancellations-air-transport-europe-2019, [Online; accessed 30-July-2021].

[10] Khaksar, H., & Sheikholeslami, A. (2019). Airline delay prediction by machine learning algorithms. *T*ransactions On Civil Engineering (A), *26 (5)* 2689-2702. http://dx.doi.org/10.24200/sci.2017.20020.

[11] Hewitt, M., & Frejinger, E. (2020). Data-driven optimization model customization. *E*uropean Journal Of Operational Research, *287* 438-451. https://doi.org/10.1016/j.ejor.2020.05.010.

[12] Clausen, J., Larsen, A., Larsen, J., & Rezanova, N. (2010). Disruption management in the airline industry—Concepts, models and methods. *C*omputers & Operations Research , *37* 809-821. https: //doi.org/10.1016/j.cor.2009.03.027.

[13] Xu, P., Corman, F., & Peng, Q. (2016). Analyzing Railway Disruptions and Their Impact on Delayed Traffic in Chinese High-Speed Railway. *I*FAC-PapersOnLine, *49 (3)* 84-89. https://doi.org/10.1016/j.ifac ol.2016.07.015.

[14] Ionescu, L., Gwiggner, C., & Kliewer, N. (2016). Data Analysis of Delays in Airline Networks. *B*usiness & Information Systems Engineering. *58* 119-133. https://doi.org/10.1007/s12599-015-0391-3.

[15] Hoeben, N. (2017). Dynamic Crew Pairing Recovery. *D*elft University of Technology

[16] Vos, H., Santos, B., & Omondi, T. (2015). Aircraft Schedule Recovery Problem – A Dynamic Modeling Framework for Daily Operations. *T*ransportation Research Procedia, *10* 931-940. https: //doi.org/10.1016/j.trpro.2015.09.047.

[17] Goverde, R. (2005). Punctuality of railway operations and timetable stability analysis. *N*etherlands TRAIL Research School.

[18] Dunn, S., & Wilkinson, S. (2016). Increasing the resilience of air traffic networks using a network graph theory approach. *T*ransportation Research Part E: Logistics And Transportation Review, *90* 39-50. https://doi.org/10.1016/j.tre.2015.09.011.

[19] Vathy-Fogarassy, A., & Abonyi, J. (2013). Graph-Based Clustering and Data Visualization Algorithms. *S*pringer.

[20] Sohouenou, P., Christidis, P., Christodoulou, A., Neves, L., & Presti, D. (2020). Using a random road graph model to understand road networks robustness to link failures. *I*nternational Journal Of Critical Infrastructure Protection, *29* 100353. https://doi.org/10.1016/j.ijcip.2020.100353.

[21] Schaeffer, S. (2007). Graph clustering. *C*omputer Science Review , *1 (1)* 27-64. https://doi.org/10.1016/j.cosrev.2007.05.001.

[22] Nascimento, M., & Carvalho, A. (2011). Spectral methods for graph clustering – A survey. *E*uropean Journal Of Operational Research , *211* 221-231. https://doi.org/10.1016/j.ejor.2010.08.012.

[23] Luisa, B. (1995). Handbook of Combinatorics Volume 1. North Holland

[24] Luxburg, U. (2007). A tutorial on spectral clustering. *S*tat Comput , *17* 395-416. https://doi.org/10.1007/s11222-007-9033-z.

[25] Beauchemin, M. (2015). On affinity matrix normalization for graph cuts and spectral clustering. *P*attern Recognition Letters , *68* 90-96. https://doi.org/10.1016/j.patrec.2015.08.020.

[26] Tautenhain, C., & Nascimento, M. (2020). An ensemble based on a bi-objective evolutionary spectral algorithm for graph clustering. *E*xpert Systems With Applications , *141* 112911. https://doi.org/10.1016/j.eswa.2019.112911.

[27] Ng, A., Jordan, M., & Weiss, Y. (2002). On spectral clustering: analysis and an algorithm. *A*dvances In Neural Information Processing Systems, *14* 849-856.

[28] Guimera, R., Mossa, S., Turtschi, A., & Amaral, L. (2005). The worldwide air transportation network: Anomalous centrality, community structure, and cities' global roles. *P*roceedings Of The National Academy Of Sciences , *102* 7794-7799. https://doi.org/10.1073/pnas.0407994102.

[29] Benson, A., Gleich, D., & Leskovec, J. (2016). Higher-order organization of complex networks. *S*cience , *353* 163-166. https://doi.org/10.1126/science.aad9029.

[30] Boughanem, M., Berrut, C., Mothe, J., & Soule-Dupuy, C. (2009). Advances in Information Retrieval. *3*1th European Conference On IR Research , *31.*

[31] Dana, K. (2018). Computational Texture and Patterns: From Textons to Deep Learning. Morgan & Claypool.

[32] Zhang, X. (2020). A Matrix Algebra Approach to Artificial Intelligence. *S*pringer. https://doi.org/10.1007/978-981-15-2770-8.

[33] Gopal, M. (2018). Applied Machine Learning. McGraw Hill Education.

[34] Favati, P., Lotti, G., Menchi, O., & Romani, F. (2020). Construction of the similarity matrix for the spectral clustering method: Numerical experiments. *J*ournal Of Computational And Applied Mathematics , *375* 112795. https://doi.org/10.1016/j.cam.2020.112795

[35] Barnhart, C., Belobaba, P., & Odoni, A. (2003). Applications of Operations Research in the Air Transport Industry. *T*ransportation Science . *37(4)* 368-391. https://doi.org/10.1287/trsc.37.4.368.23276.

[36] Alami, N., Meknassi, M., En-nahnahi, N., El Adlouni, Y., & Ammor, O. (2021). Unsupervised neural networks for automatic Arabic text summarization using document clustering and topic modeling. *E*xpert Systems With Applications, *172* 114652. https://doi.org/10.1016/j.eswa.2021.114652.

[37] Janani, R., & Vijayarani, S. (2019). Text document clustering using spectral clustering algorithm with particle swarm optimization. *E*xpert Systems With Applications, *134* 192-200. https://doi.org/10.1016/j.eswa.2019.05.030.

[38] Andrews, N., & Fox, E. (2007). Recent developments in document clustering. Department of Computer Science, Virginia Polytechnic Institute & State.

[39] Steinbach, M., Ertöz, L., & Kumar, V. (2004). The challenges of clustering high dimensional data. *N*ew Directions In Statistical Physics, 273-309.

[40] Almeida, H., Guedes, D., Meira, W., & Zaki, M. (2011). Is there a best quality metric for graph clusters?. *J*oint European Conference On Machine Learning And Knowledge Discovery In Databases, 44-59.

[41] Afzalan, M. & Jazizadeh, F., (2019). An automated spectral clustering for multi-scale data. *N*eurocomputing, *347* 94-108. https://doi.org/10.1007/s12594-019-1275-9.

[42] Correa, C., & Lindstrom, P. (2012). Locally-scaled spectral clustering using empty region graphs. *P*roceedings Of The 18th ACM SIGKDD International Conference On Knowledge Discovery And Data Mining. 1330-1338. https://doi.org/10.1016/j.jspi.2011.12.010.

[43] Tremblay, N., & Loukas, A. (2020). Approximating spectral clustering via sampling: a review. *S*ampling Techniques For Supervised Or Unsupervised Tasks. 129-183. https://doi.org/10.48550/arXiv.1901.10204.

[44] Liu, J., Wang, C., Danilevsky, M., & Han, J. (2013). Large-scale spectral clustering on graphs. *T*wenty-Third International Joint Conference On Artificial Intelligence.

[45] Cadot, M., Lelu, A., & Zitt, M. (2018). Benchmarking seventeen clustering methods on a text dataset. LORIA.

[46] Wang, K., Wang, B., & Peng, L. (2009). CVAP: validation for cluster analyses. *D*ata Science Journal. 0904220071-0904220071. http://dx.doi.org/10.2481/dsj.007-020.

[47] Li, M., Lian, X., Kwok, J., & Lu, B. (2011). Time and space efficient spectral clustering via column sampling. *C*VPR 2011, 2297-2304.

[48] Talebi, H., Peeters, L., Mueller, U., Tolosana-Delgado, R., & Boogaart, K. (2020). Towards geostatistical learning for the geosciences: A case study in improving the spatial awareness of spectral clustering. *M*athematical Geosciences, *52* 1035-1048. https://doi.org/10.1007/s11004-020-09867-0.

[49] Duan, L., Ma, S., Aggarwal, C., & Sathe, S. (2021). Improving spectral clustering with deep embedding, cluster estimation and metric learning. *K*nowledge And Information Systems, *63* 675-694. https://doi.org/10.1007/s10115-020-01530-8.

[50] Thalamuthu, A., Mukhopadhyay, I., Zheng, X., & Tseng, G. (2006). Evaluation and comparison of gene clustering methods in microarray analysis. *B*ioinformatics, *22* 2405-2412. https://doi.org/10.1093/bioinformatics/btl406.

[51] Yan, D., Huang, L., & Jordan, M. (2009). Fast approximate spectral clustering. *P*roceedings Of The 15th ACM SIGKDD International Conference On Knowledge Discovery And Data Mining. 907-916.

[52] El-Bhissy, K., El-Faleet, F., & Ashour, W. (2014). Clustering Using Optimized Gaussian Kernel Function. *I*nternational Journal Of Artificial Intelligence And Application For Smart Devices IJAIASD, *2*. https://doi.org/10.14257/ijaiasd.2014.2.1.04.

[53] Wang, L., Leckie, C., Ramamohanarao, K., & Bezdek, J. (2009). Approximate spectral clustering. *P*acific-Asia Conference On Knowledge Discovery And Data Mining, 134-146.

[54] Assent, I. (2012). Clustering high dimensional data. *W*iley Interdisciplinary Reviews: Data Mining And Knowledge Discovery, *2* 340-350. https://doi.org/10.1002/widm.1062.

[55] Aggarwal, C., Hinneburg, A., & Keim, D. (2001). On the surprising behavior of distance metrics in high dimensional space. *I*nternational Conference On Database Theory 420-434.

[56] Wu, S., Feng, X., & Zhou, W. (2014). Spectral clustering of high-dimensional data exploiting sparse representation vectors. *N*eurocomputing, *135* 229-239. https://doi.org/10.1016/j.neucom.2013.12.027.

[57] Cheng, M., Kusoemo, D., & Gosno, R. (2020). Text mining-based construction site accident classification using hybrid supervised machine learning. *A*utomation In Construction, *118* 103265. https://doi.org/10.1016/j.autcon.2020.103265.

[58] Zhang, F., Fleyeh, H., Wang, X., & Lu, M. (2019) Construction site accident analysis using text mining and natural language processing techniques. *A*utomation In Construction, *99* 238-248. https://doi.org/10.1016/j.autcon.2018.12.016.

[59] Dörpinghaus, J., Schaaf, S., & Jacobs, M. (2018). Soft document clustering using a novel graph covering approach. *B*ioData Mining, *11* 1-20. https://doi.org/10.1186/s13040-018-0172-x.

[60] Verma, D., & Meila, M. (2003). A comparison of spectral clustering algorithms. *U*niversity Of Washington Tech Rep UWCSE030501, *1* 1-18.

[61] Lewis, D. (1999). Reuters-21578. http://www.daviddlewis.com/resources/testcollections/reuters21578/, [Online; accessed 30-July-2021].

[62] Learn scikit-learn Machine Learning in Python. (2021). https://scikit-learn.org/stable/modules/classes.html#module-sklearn.cluster, [.Online; accessed 30-July-2021]

[63] Kumhar, S., Kirmani, M., Sheetlani, J., & Hassan, M. (2021). Word Embedding Generation for Urdu Language using Word2vec model. *M*aterials Today: Proceedings. https://doi.org/10.1016/j.matpr.2020.11.766.

[64] Ruas, T., Ferreira, C., Grosky, W., França, F., & Medeiros, D. (2020). Enhanced word embeddings using multi-semantic representation through lexical chains. *I*nformation Sciences, *532* 16-32. https://doi.org/10.1016/j.ins.2020.04.048.

[65] FastText Library for efficient text classification and representation learning. (2021). https://fasttext.cc, [Online; accessed 30-July-2021].

[66] Alguliyev, R., Aliguliyev, R., & Sukhostat, L. (2021). Parallel batch k-means for Big data clustering. *C*omputers & Industrial Engineering, *152* 107023. https://doi.org/10.1016/j.cie.2020.107023.