CZECH TECHNICAL UNIVERSITY IN PRAGUE

FACULTY OF MECHANICAL ENGINEERING

Department of Instrumentation and Control Engineering

Bachelor Thesis

2022 MHD Kussay Nadar

CZECH TECHNICAL UNIVERSITY IN PRAGUE

FACULTY OF MECHANICAL ENGINEERING

Department of Instrumentation and Control Engineering

Clustering Algorithms for Power Load Profiles

Bachelor Thesis

Study program:	Theoretical Fundamentals of Mechanical Engineering
Study branch:	Mechanical Engineering
Supervisor:	Ing. Adam Peichl

MHD Kussay Nadar

Prague, August 2022

BACHELOR'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Nadar Mhd Kussay** Personal ID number: **481259**

Faculty / Institute: **Faculty of Mechanical Engineering**

Department / Institute: **Department of Instrumentation and Control Engineering**

Study program: **Theoretical Fundamentals of Mechanical Engineering**

Branch of study: **No Special Fields of Study**

II. Bachelor's thesis details

Bachelor's thesis title in English:

Clustering algorithms for power load profiles

Bachelor's thesis title in Czech:

Shluková analýza pro výkonové profily

Guidelines:

- Perform research on topic of clustering
- Implement K-means and two additional algorithms
- Compare algorithms and validate on data

Bibliography / sources:

[1] XU, Rui; WUNSCH, Donald. Survey of clustering algorithms. IEEE Transactions on neural networks, 2005, 16.3: 645-678.
[2] RAJABI, Amin, et al. A comparative study of clustering techniques for electrical load pattern segmentation. Renewable and Sustainable Energy Reviews, 2020, 120: 109628.

Name and workplace of bachelor's thesis supervisor:

Ing. Adam Pelchl U12110.3

Name and workplace of second bachelor's thesis supervisor or consultant:

Date of bachelor's thesis assignment: **25.05.2022** Deadline for bachelor thesis submission: **18.08.2022**

Assignment valid until: _____

Ing. Adam Pelchl
Supervisor's signature

Head of department's signature

doc. Ing. Miroslav Španiel, CSc.
Dean's signature

III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

02.06.2022
Date of assignment receipt

Kussay
Student's signature

Declaration

I hereby declare that I have completed this thesis having the topic Clustering Algorithms for Power Load Profiles and I have included a full list of used references.

I do not have a compelling reason against the use of this thesis within the meaning of Section 60 of the Act No 121/2000 Sb., on copyright and rights related to copyright and on amendment to some other acts (The Copyright Act), as amended.

In Prague……. ……..…….

Student's signature

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my thesis advisor Ing. Adam Peichl, for the continuous support in my BSc. Study and research, for his motivation, immense knowledge and patience.

Finally, I would like to thank my family and friends for the support they have showed throughout my studies and my life.

Bachelor's Thesis title:

Clustering Algorithms for Power Load Profiles

Abstract:

This Thesis is aimed at understanding what is clustering and the methods of clustering. General information about some examples of clustering methods and their important characteristics, such as how to decide the number of clusters and clustering large scale datasets. This thesis also discusses visualizing high dimensional clusters in different methods, such as dimension reduction. This work also contains four cases where clustering algorithms were ran and graphically displayed the results of clustering and compared the results, in order to get a grasp on which of the used clustering algorithms works best with the applications of electrical power demand. Essentially the clustering algorithms DBSCAN and K-means are used for clustering electrical power demand datasets and can be plotted nicely to find the patterns between the graphically represented results of clustering and the dataset graphically represented with respect to time, in order to predict future loads and create the optimum grid with no losses.

Keywords: Euclidian space, Clustering, Centroids, Sequential data, Indices, Outliers.

Table of Contents:

1 Introduction 9

2 Power loads 9

3 Clustering 11

4 Clustering Methods 12

4.1 *Centroid-Based Clustering* 13

4.1.1 K-means 15

4.2 *Density-Based Clustering* 16

4.2.1 DBSCAN 16

4.3 *Hierarchical Clustering* 18

4.3.1 Self-Organizing Maps 20

5 Important Characteristics for Clustering 20

5.1 *Number of Clusters* 20

5.2 *Clustering Sequential Data* 22

5.3 *Clustering Large-Scale Data Sets* 26

5.4 *Visualizing High Dimensional Clusters* 30

5.4.1 Dimension Reduction 30

6 Preparation for the Experiments 31

6.1 *Preparations for the first experiment* 32

6.2 *Preparations for the second experiment* 33

6.3 *Preparations for the third experiment* 35

6.4 *Preparations for the fourth experiment* 35

7 Results and Validation..... 36
7.1 *Results for the first experiment*..... 36
7.2 *Results for the second experiment* 39
7.3 *Results for the fourth experiment*..... 41
7.4 *Results for the fourth experiment*..... 43

8 Conclusion..... 44

9 References: 46

1 Introduction

We live in a world full of data. People encounter a large amount of information and store or represent it as data, for further analysis and management on a daily basis. One of the most vital means in dealing with these data is to classify or group them into a set of categories or clusters. Classification has played a very important role in the long history of human development as one of the most primitive activities of human beings.

In order to understand a new phenomenon or learn a new object, people always seek the feature that can describe it, and compare it with other know objects or phenomena, based on the similarity or dissimilarity, generalized as proximity, according to some certain standards or rules. “Basically, classification systems are either supervised or unsupervised, depending on whether they assign new inputs to one of a finite number of discrete supervised classes or unsupervised categories, respectively [1], [2], [3].

2 Power loads

The electrical industry plays a very vital role in human life from various angles. The electricity demand is increasing day-by-day with the rapid increase in population [4]. The traditional power grid became an old version and isn’t efficient enough now; therefore, a new intelligent and smart version of the power grid was introduced known as The Smart Grid (SG). The most important task of the grid is to effectively control the consumption, generation and distribution of the electricity. It becomes much easier to manage the distribution of electric load of utility companies, remain in touch with the consumers help reduce the variation between the power demand supply through the SG system. The utility supplies electricity to consumers based on their demand. The rate of electricity

consumption sometimes rises, and the utility does not have enough energy to supply the rise in demand. To overcome the issue of balancing between consumption and utility support, the utility uses the electricity load forecasting model, which is one aspect of the SG system.

The approximate energy consumption pattern of the consumers is predicted through load forecasting by their historical data. Generally, there are four types of forecasting: Very Small Term Forecasting (VSTF), Short-Term Forecasting (STF), Medium-Term Forecasting (MTF), and Long-Term Forecasting (LTF). In VSTF electric load of some hours to one day is predicted, STF is for predictions of one day-ahead to some weeks-ahead, MTF predicts data of one week to one year and through LTF, one year to several years ahead load can be forecasted, as they were used in [5], [6], [7].

Data analysis is a process of getting useful information from hidden patterns of data. Data analysts measures the price and load consumption by taking historical data in the form of datasets, in order to perform some tasks which then allows us to obtain useful information, such as in [8] a detailed review of data in available. The volume of real-world data is intensively increasing day-by-day, and the large volume of data is referred to as big data. Effective information is being collected from massive quantities of historical power data to implement analysis over it, through data analytics, which helps to make more enhancements in the market operations planning and management. Big data is multifaceted and very excessive in volume, that leads to redundant features arising as a main issue in the big data sets; so traditional methods are not very supportive for handling such a large amount of data. Many techniques are tested and applied to handle big data and extract useful information. Although, big data is still an issue of the current era, and in the is paper we will discuss some algorithms that can be used to analysis a dataset of a power load to find the best approach for analysing datasets of electric loads.

3 Clustering

In its basic form clustering is the problem of finding homogeneous group of data points in a given data set. Each one of these groups is called a cluster and can be defined as a region in which the density of objects is higher than in other regions.

Manual clustering is possible only in some special cases by visualizing the vectors in space and finding the clusters by eye, but it has many downsides. A very clear downside would be that the visualization of the data points in a multi-dimensional ($d > 3$) space is not possible. Another would be that the result of manual clustering is objective because different people might be able to see different clusters. It's usage in high frequency applications is very inconvenient because it's repetitive and time consuming.

For the reasons mentioned previously and many others, clustering is done by algorithms which led to the raise of automatics classification procedures, as a matter-of-fact manual clustering is only used in a very limited scope of applications. Many algorithms have been developed for cluster analysis for over the last 40 years and more will be developed in the future.

The reasons for the variety of methods of clustering are probably twofold. One of main reason for the diversity of algorithms is because there exists no general definition of a cluster, which means there's different kinds of clusters, such as linear clusters, spherical clusters and so on. There also exists different types of data, such as continuous variables, discrete variables, dissimilarities, and similarities which are used in different applications. Therefore, different clustering methods are needed in order to adapt to the kind of application and the type of cluster sought. There are three main motivations to group objects into clusters, a more in-depth analysis can be found in [9].

First, a good clustering has a predictive power; in this case, we perform clustering because we believe the underlying cluster labels are meaningful, will lead us to a more efficient description of our data and help us choose better actions (as mentioned previously with The Smart Grid). This clustering type is called "mixture density modelling", and the objective function that measures how well the predictive model is working is the information content of the data. In a load consumption for example, this property would suggest that clusters can be used to predict future energy consumption.

Secondly, clusters allow people to compress the information into a single information, corresponding to the centre of the cluster, i.e., centroid. For instance, by classifying load consumption into two categories (working days and holidays), it is possible to identify two clusters and their two corresponding centroids. The centroids can be used to identify the "usual" consumption of that day of the week. Thus, it summarizes in the only one 24-hour profile the information content of the load profiles during, let's say, whole year. This type of clustering is sometimes called "vector quantization".

A third reason would be to identify the "outliers", i.e., the cases in which clusters fail to accurately represent particular data. An example of this, in the load consumption case, it is represented by working day that for some reasons present load consumptions that are very close to holidays (e.g., working days in the middle of two holidays also known as Bridge Days). Clearly, such anomalous days should be identified and not considered when building the profile of typical working days.

4 Clustering Methods

As we mentioned previously there exists no general definition of a cluster, which means there's different kinds of clusters. This is where different methods

come to role, to help identify different kinds of cluster. Here are some examples of different clustering methods:

4.1 *Centroid-Based Clustering*

Centroid based clustering represents all of its objects on par of central vectors which need not be a part of the dataset taken. The main underlaying advantage for any centroid based clustering is the aspect of calculating the distance measure [10] between the objects of the data set considered. The basic aspect of distance measure in general is derived using one of Euclidian, Minkowski or Manhattan distance measuring mechanism [11].

In which mean is used in Euclidian distance measure, median in Manhattan and steepest descend method for calculating the distance measures [12].

To get a bit more in depth in Distance measure methodologies:

1) ***Euclidean Distance measure:***

   Euclidian distance becomes a metric space, being processed using Pythagorean formula. The position in a Euclidean n-space is termed as Euclidean vector.

   Suppose that X and Z are two samples of pattern vectors,

$X = (x_1, x_2, \dots, x_n)^T$ $\quad Z = (z_1, z_2, \dots, z_n)^T$

   And we define the distance between *X* and *Z* as:

$$D = \|X - Z\| = [\sum_{i=1}^{n}(x_i - z_i)^2]^{\frac{1}{2}} \qquad (1)$$

   Easy to know that the smaller D is, the more similar are X and Z (D is the distance of X and Z in n-dimensional space)

   Euclidean distance in general is used in nonlinear dimensionality measuring. Correlation analysis is calculated using the metric. The

main drawback of this measure is that it is sensitive to high noise and sensitive in determining correlation between similar trends [13].

2) ***Manhattan Distance measure:***

Manhattan distance measure is a typical method that could be adapted even if a grid-like path is being traced in the data sets [14]. Typically, it is the distance measure between corresponding correlated objects. The measure in general is given by

$$d = \sum_{i=1}^{n} |x_i - y_i| \qquad (2)$$

The Manhattan distance could determine distance metric for unevenly distributed objects which makes it a heuristic-based algorithm. The major degrading aspect is that it is still sensitive in measuring correlation dissimilarity between similar trends.

3) ***Minkowski distance measure:***

In general Minkowski measure on Euclidian space is regarded as a generalization of both Euclidian and Manhattan distance measures. It can be treated as a power mean multiple of distance between objects. The Minkowski measure between two objects with order p is given by

$$d = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p} \qquad (3)$$

In typical cases, Minkowski distance measure suffers with limiting factor value being reached to infinity [15]. In such cases a new measure termed as Chebyshev measure is being calculated.

An example for such clustering method would be:

4.1.1 K-means

The *k*-means algorithm finds locally optimal solutions with respect to the clustering error. It is a fast iterative algorithm that has been used in many clustering applications, and has advantages of briefness, efficiency and celerity. To seek the optimizing outcome this algorithm tries to find *K* divisions to satisfy a certain criterion. Firstly, choose some dots to represent the initial cluster focal points (Usually, the first *K* sample dots is chosen to represent the initial cluster focal point); secondly, it gathers the remaining sample dots to their focal points in accordance with criterion of minimum distances, we then get the initial classification , and if the classification if unreasonable, we will modify it (Calculate each cluster focal points again), iterate repetitively till we get a reasonable classification to minimize the clustering error.

However, this algorithm depends quite much on initial dots and the difference in choosing initial samples which always leads to different outcomes.

Input X = $\{x_1, x_2, \ldots x_n\}$ where n being a set of objects

K: Number of desired clusters

Output: A set of *k* clusters.

Steps:

1) Randomly choose k objects from X as primary centroids.
2) Repeat
    a. Assign each data item di to the cluster which has the closest centroid.
    b. Calculate the new mean of each cluster; Until convergence criterion is met.

Advantages:

1) It is considered the speediest centroid based algorithm.
2) It is very lucid and can sustain large amount of data sets.
3) It reduces intra-cluster variance measure.

Disadvantages:

1) It suffers when there is more noise in the data.
2) Outliers can never be studied
3) Even though it reduces intra-cluster variance, it could not deal with global minimum variance of measure.
4) Very sensitive at clustering data sets of nonconvox shapes.

4.2 *Density-Based Clustering*

Density-Based clustering was introduced to discover clusters of arbitrary shape. It is based on the fact that within each cluster there is a typical density of points, and this density is higher than outside the cluster. The outside points with the lower density are recognized as noise points. DBSCAN is one of the most commonly known Density-Based clustering algorithms, an example of its usage would be [16] in Applications with Noise.

4.2.1 DBSCAN

This algorithm finds all clusters properly, independent of the size, shape and location of clusters to each other, and it is based on two main concepts: density reachability and density connectability. These two concepts depend on two input parameters of the clustering algorithm: the size of epsilon neighbourhood (e) and the minimum points in a cluster (m) in its neighbourhood, and the algorithm

usually supports the user in determining the appropriate values for them. The size parameter controls the size of the clusters and the size of the neighbourhood. An open set in the Euclidean space can be divided into a set of its connected components. The implementation of this idea for partitioning of a finite set of points requires concepts of density, connectivity and boundary. The key idea of this clustering method is that the numbers of data objects in the "neighbourhood" are considered to determine density. For each object of a cluster the neighbourhood of a given radius (r) has to contain at least a minimum number of objects, i.e., the cardinality of the neighbourhood has to exceed a threshold. Regions of high density are defined in a separate cluster from the regions of no or low density.

To find the neighbourhood of a data point a spatial index is employed, which has improved the complexity of finding clusters by other methods $O(n^2)$ to O(nlogn). It does not preform well if the data is high dimensional, and the Euclidean distance is used to find proximity of objects.

Pseudocode for DBSCAN

1) DBSCAN(D, eps, MinPts)
    a) C = 0
    b) for each unvisited point P in dataset D
        I. mark P as visited
        II. NeighborPts = regionQuery(P, eps)
        III. if sizeof(NeighborPts) < MinPts
            i. mark P as NOISE
        IV. else
            i. C = next cluster
            ii. expandCluster(P, NeighborPts, C, eps, MinPts)

2) expandCluster(P, NeighborPts, C, eps, MinPts)
    a) add P to cluster C
    b) for each point P' in NeighborPts
        I. if P' is not visited
            i. mark P' as visited
            ii. NeighborPts' = regionQuery(P', eps)
            iii. if sizeof(NeighborPts') >= MinPts
                a. NeighborPts = NeighborPts joined with NeighborPts'
    c) if P' is not yet member of any cluster
        I. add P' to cluster C
3) regionQuery(P, eps)
    a) return all points within P's eps-neighborhood (including P)

4.3 *Hierarchical Clustering*

Hierarchical clustering is defined as a method in which clusters take the form of a tree or hierarchy. Every node in the tree represents a cluster and the clusters in the hierarchy are known as dendrograms. Hierarchical clustering can be performed in two ways based on splitting and merging of clusters: divisive method and agglomerative method.

Divisive method of hierarchical clustering is the top-down approach, in which a large data set is given initially, and this data set is further divided into a number of smaller clusters (subsets) until a threshold is reached.

General steps of hierarchical clustering:

1) We start by assigning each item to a cluster, so that if we have N items, we now have N clusters each containing just one item. Let the

distances between the clusters the same as the distances between the items they contain.

2) Find the closest pair of clusters and merge them into a single cluster, so that we have on cluster less now.

3) Compute distances between the new cluster and each of the old clusters.

4) Repeat the previous two steps until all items are clustered into K number of clusters (The previous steps were taken from [67]).

In order to decide if the splitting of a cluster will take place or if two clusters should be combined or merged, a measing criteria known as dissimilarity among the sets of data is required.

Pseudocode for Hierarchical Clustering taken from [17]:

1) Start with a number of n sub clusters at level L(0) = 0 and a counter C = 0.
2) Locate the nearest neighbours that is the neighbours with minimum distance say pair (A), (B), as indicated by D[(A),(B)] = min d[(i),(j)]
3) Increase the counter number: C = C + 1. Merge the clusters (A) and (B) to a single group. Set the level of this hierarchy to L(C) = D[(A),(B)]
4) Upgrade the similarity lattice, by erasing the lines and segments comparing to clusters (A) and (B) and including a column and segment relating to the recent hierarchy. The similarity between the new cluster, most recent cluster(A,B) and old cluster(k) can be calculated as:

$$D[(k), (A, B)] \; min[D[(k), (A)], D[(k), (B)]]$$

5) Stop, if only one clusters is remaining. Else, go to step 2.

An example for this clustering method would be Self-Organizing Maps (SOM).

4.3.1 Self-Organizing Maps

The self-organizing map (SOM) method was invented by Kohonen in [18]. It's an unsupervised artificial neural network that is mapped from the high-dimensional data space to low-dimensional neuron grid and are able to learn both the distribution (as competitive layers do) and the topology of the input vectors on which they are trained, while preserving statistical and topological information. Consequently, excellent clustering results are obtained, and easy evaluation of the result is possible the graphical representation on maps whose different labels (vector identifiers) can be grouped by visual inspection. Applying some index functions, it is possible to obtain an optimum clustering, but some supervision is necessary to filter the results of the maps (i.e., the operator selects the maximum number of clusters). For the purpose of load identification, the basic SOM is extended to be supervised and thus function as a classifier. An advantage of SOM is that it classifies all data in several groups by their inherent relationships, known as "Clustering by Nature".

5 Important Characteristics for Clustering

5.1 *Number of Clusters*

The clustering process splits data into an appropriate number of clusters. For some applications users can determine the number of clusters (K) based on terms of their expertise, and in other applications, the value of K is unknow and needs to be estimated exclusively from the data themselves. Many clustering algorithms ask to be provided with K as an input parameter, and it is obvious that the quantity of resulting clusters depends largely on the estimation of K.

A division with too many clusters complicates the result, which in returns makes the result harder to interpret and analyse, and a division with too few clusters causes the loss of information and misleads the final decision. Dubes called the problem of determining the number of clusters "The Fundamental Problem of Cluster Validity" [19].

A large number of attempts have been made to estimate the appropriate *K* and some of the representative examples are illustrated in the following:

1) **Visualization of the data set.** The value of *K* can be determined by a direct observation of a histogram or scatterplot, which depicts a projection of the data points onto a two-dimensional Euclidean space. This strategy is restricted to only a small scope of applications, due to the complexity of most data sets (data sets with a low number of dimensions).

2) **Construction of certain indices** (stopping rules). The compactness of intra-cluster and isolation of inter-cluster are emphasized usually by the indices and consider the comprehensive effects of several factors, including the defined squared error, the statistical or geometrical properties of the data, the number of patterns, the number of clusters, and the similarity (dissimilarity). Milligan and Cooper compared and ranked 30 indices based on their performance over a series of artificial data sets in [20]. Among these indices, the Calinski and Haarabasz index in [19] achived the best performance and is represented as

$$\mathrm{CH}(K) = \frac{T_r(S_B)}{K-1} \Big/ \frac{T_r(S_W)}{N-K} \qquad (4)$$

where N is the total number of patterns and $T_r(S_B)$ and $T_r(S_w)$ are the trace of the between and within class scatter matrix, respectively. The K that maximises the value of $CH(K)$ is selected as the optimal. Everitt, Landau, and Leese pointed out in [21] that "it is advisable not to depend on a single rule for selecting the number of groups, but to synthesize the results of several techniques". Therefore, a good performance of an index for a certain data doesn't guarantee the same behaviour with different data.

3) **Other heuristic approaches based on a variety of techniques and theories.** In Kothari and Pitts [22], described a scale-based method, in which the distance from a cluster centroid to other clusters in its neighbourhood is considered (added as a regularization term in the original squared error criterion). The neighbourhood of clusters works as a scale parameter and the K that is persistent in the largest interval of the neighbourhood parameter is considered as the optimal. Girolami performed eigenvalue decomposition on the kernel matrix in the high-dimensional feature space and used the dominant K components in the decomposition summation as an indication of the possible existence of K clusters [23].

5.2 *Clustering Sequential Data*

Sequential data are sequences with variable length and many other distinct characteristics, e.g., dynamic behaviours, time constraints, and large volume [24], [25]. Sequential data is usually generated from: DNA sequencing, speech processing, text mining, stock market, web data mining and many more applications, a few examples [26], [25], [27]. Cluster analysis explores potential patterns hidden in the large number of sequential data in the context of

unsupervised learning and therefore provides a crucial way to meet the current challenges.

Generally, strategies for sequential clustering falls into three categories:

1) **Sequence Similarity:**

The first category is based on the measure of the distance (similarity) between each pair of sequences. Then a proximity clustering algorithm, such as hierarchical clustering groups the sequences. Conventional measure methods are inappropriate because many sequential data are expressed in an alphabetic form, such as DNA. The distance between two sequences can be defined by virtue of the minimum number of required operations such as substitution, insertion and deletion operations, if a sequence comparison is regarded as a process of transforming a given sequence to another. A common analysis processes is alignment, as illustrated in Figure 1. The defined distance is known as edit distance or Levenshtein distance, a more in-depth explanation can be found in [24], [28]

C	L	U	S	-	-	-	-	-	-	T	E	R	I	-	N	G
M	M	S	M	I	I	I	I	I	I	M	D	D	M	I	M	D
C	L	A	S	S	I	F	I	C	A	T	-	-	I	O	N	-

M: Match; D: Deletion; I: Insertion; S: Substitution

Figure 1 - Illustration of a sequence alignment. Series of edit operations is performed to change the sequence CLUSTERING into the sequence CLASSIFICATION

2) **Indirect Sequence Clustering:**

The Second category employs an indirect strategy, which begins with the extraction of a set of features from the sequences. All the sequences are then mapped into the transformed feature space, where classical vector space-based clustering algorithms can be used to form clusters, and it becomes obvious that feature extraction becomes the essential factor that decides the effectiveness of these algorithms. Guralnik and Karypis discussed the potential dependency between two sequential patterns and suggested the global and local approaches to prune the initial feature sets in order to represent sequences in the new feature space in a better way, as explained in [29]. Morzy *et al.* utilized the sequential patterns as the basic element in the agglomerative hierarchical clustering and defined a co-occurrence measure, as the standard of fusion of smaller clusters [30]. These methods greatly reduce the computational complexities and can be applied to large-scale sequence databases. However, the process of feature selection has disadvantages, such as the inevitable loss of some information in the original sequences and needs extra attention.

3) **Statistical Sequence Clustering:**

The two categories that were mentioned previously are used to deal with sequential data composed of alphabets, while the third method aims to construct statistical models to describe the dynamics of each group of sequences, can be applied to numerical or categorical sequences. The most method in this category is hidden Markov models (HMMs) [31], [32], [33], which first gained its popularity in the application of speech recognition [34]. A discrete

HMM describes an unobservable stochastic process consisting of a set of states, each of which is related to another stochastic process that emits observable symbols. Accordingly, the HMM is specified by the following:

1) A finite set $V = \{V_1, V_2, \dots, V_Q\}$ with Q states.

2) A discrete set $O = \{O_1, O_2, \dots, O_M\}$ with M observation symbols.

3) A state transition distribution $A = \{\alpha_{ij}\}$, where α_i $j = P$ (jth state at time $t + 1$, ith state at time t).

4) A symbol emission distribution $B = \{\beta_{il}\}$, where $\beta_{il} = P$ [V_l at t , ith state at time t].

5) An initial state distribution $\pi = \{\pi_i\}$, where $\pi_i = P$ [ith state at time t].

After an initial state is selected according to the initial distribution π, a symbol is emitted with emission distribution E. The next state is decided by the state transition distribution T and it also generates a symbol based on E. The process repeats until reaching the last state. Note that the procedure generates a sequence of symbol observations instead of states, which is where the name “hidden” comes from. Dynamic programming was developed to solve the basic three problems of HMMs, which are:

1) **Likelihood** (forward or backward algorithm). Computing the probability of an observation sequence given a model.

2) **State interpretation** (also known as Vertbi algorithm). Optimizing some criterion function given the observation sequence and the model to find an optimal state sequence.

3) **Parameter estimation** (Baum¬Welch algorithm). Maximize the probability ability of observation sequence under the model, by designing a suitable model parameter.

HMMs are well founded theoretically [34].

5.3 *Clustering Large-Scale Data Sets*

With the increasing complexity of data, mainly through two aspects: enormous data volumes and high dimensionality, scalability becomes more and more important for clustering. The examples illustrated in the sequential clustering section, are some of many applications that requires this capability. With the advancement of databases and Internet technologies, clustering algorithms will face many more severe challenges in handling the rapid growth of data. In *Table I* we summarize the computational complexity of some classical clustering algorithms with several newly proposed approaches, which are designed to deal with large-scale data sets.

1) Classical hierarchical clustering algorithms, including single-linkage, complete linkage, average linkage, centroid linkage and median linkage, are not appropriate for large-scale data sets due to the quadratic computational complexities in both execution time and store space.

2) *K*-means algorithm has a time complexity of $O(N\ K\ d)$ and space complexity of $O(N + K)$. The complexity becomes near linear to the number of samples in the data sets, since *N* is usually much larger than both *K* and *d*. *K*-means algorithm is effective in clustering large scale data sets, and there has been an effort put into overcoming its disadvantages in [35] and [36].

3) Many novel algorithms have been developed to cluster large-scale data sets, especially in the context of data mining some examples would be [37], [38], [39], [40], [41]. Most of these algorithms can scale the computational complexity linearly to the input size and demonstrate the possibility of handling enormous datasets.

    a) Density-based approach, e.g., density based spatial clustering of applications with noise (DBSCAN) such as in [39] and density-based clustering (DENCLUE) as in [40]. DBSCAN requires that the density in a neighbourhood for an object should be high enough if it belongs to a cluster, and the neighbourhood needs to satisfy the user-specified density threshold. For more efficient queries DBSCAN uses a R*-tree structure. DENCLUE seeks clusters with local maxima of the overall density function, which then reflects the comprehensive influence of data objects to their neighbourhoods in the corresponding data space.

b) Random sampling approach, e.g., Clustering large applications (CLARA) [42] and CURE [43]. The important geometrical properties of clusters can be effectively maintained by the appropriate sample, which is the key point of this approach. Furthermore, as shown in [43] Chernoff bounds can provide estimation for the lower bound of the minimum sample size, given the low probability that points in each cluster are missed in the sample set. Each cluster is represented with a medoid while CURE chooses a set of well-scattered and centre-shrunk points in CLARA.

4) Most algorithms listed previously lack the capability of dealing with data with high dimensionality. The increase of dimensionality degenerates their performance. Some algorithms such as DENCLUE, have shown some successful applications in such cases, but still aren't completely effective yet.

   In addition to approaches mentioned previously, several other techniques and approaches play significant roles in clustering large-scale datasets. Bradley, Fayyad, and Reina proposed in [37] a scalable clustering framework, considering seven important relevant characteristics in dealing with large databases. Parallel algorithms can more effectively use computational resources, and greatly improve overall performance in the context of both time and space complexity [44], [45], [46]. Incremental clustering techniques do not require the storage of the entire data set and can handle it in a one-pattern-at-a-time way. The pattern gets assigned to a cluster, if the patterns display enough closeness to a cluster according to some predefined criteria, an example would be the ART family in [47] and [48]. Most incremental clustering algorithms are dependent on the order of the input patterns as explained in [47] and [49].

Table 1
Computational Complexity of Clustering Algorithms

Cluster Algorithm	Complexity	Capability of tackling high dimensional data
K-means	$O(NKd)$ (time) $O(N + K)$ (space)	No
Fuzzy *c*-means	Near $O(N)$	No
Hierarchical Clustering*	$O(N^2)$ (time) $O(N^2)$ (space)	No
CLARA	$O(K(40 + K)^2 + K(N - K))$* (time)	No
CLARANS	Quadratic in total performance	No
BIRCH	$O(N)$ (time)	No
DBSCAN	$O(N \log N)$ (time)	No
CURE	$O(N^2_{sample} \log N_{sample})$ (time) $O(N_{sample})$ (space)	Yes
WaveCluster	$O(N)$ (time)	No
DENCLUE	$O(N \log N)$ (time)	Yes
FC	$O(N)$ (time)	Yes
CLIQUE	Linear with the number of objects, Quadratic with the number of dimensions	Yes
OptiGrid	Between $O(Nd)$ and $O(Nd \log N)$	Yes
ORCLUS	$O(K_0^3+K_0Nd+K_0^2d^3)$ (time) $O(K_0d^2)$ (space)	Yes

Table 1: Computational Complexity of Clustering Algorithms

*include single-linkage, complete-linkage, average-linkage, etc + based on the heuristic for drawing a sample from the entire data set in [42]

5.4 *Visualizing High Dimensional Clusters*

High-dimensional datasets show up in numerous fields of study, such as physics, biology, chemistry, political science, and economics, to name a few. Their wide availability, increasing size, and complexity have led to new challenges and opportunities for their effective visualization. For example, genomic microarrays in biology in [50] and [51], spectrometry data in air quality research [52], simulation parameters in nuclear safety engineering [53], and chemical compositions in combustion simulations [54] can all be mapped to high-dimensional spaces for exploration.

The physical limitations of our visual systems prevent the direct display and rapid recognition of structures with dimensions higher than two or three. In the past decade, a variety of approaches have been introduced to visually convey high-dimensional structural information by utilizing low-dimensional projections or abstractions: from dimension reduction to visual encoding, and from quantitative analysis to interactive exploration. Several surveys have focused on different aspects of high dimensional data visualization, such as parallel coordinates [55], [56], quality measures [57], clutter reduction [58], visual data mining [59], [60], [61], and interactive techniques [62]. A high-dimensional dataset can be described through the perspective of the range and domain of a function, which provides a unified view of several related but different types of datasets.

5.4.1 Dimension Reduction

Dimension reduction is one of the fundamental techniques for analysing and visualizing high-dimensional and makes the high-dimensional data addressable, reduces the computational cost, and also provides users with a clearer

picture and visual examination of the data of interest. However, dimensionality reduction methods inevitably cause some loss of information, and may damage the interpretability of the results, even distort the real clusters. Dimension reduction techniques can be roughly divided into two major categories: linear dimension reduction and nonlinear dimension reduction (manifold learning). Linear projection uses linear transformation to project the data from high-dimensional to lowdimensional space. It includes many classical methods, such as Principal Component Analysis (PCA) [63], Multidimensional Scaling (MDS), Linear Discriminant Analysis (LDA), and various factor analysis methods.

Nonlinear dimension reduction can occur in either a metric or nonmetric setting. The graph-based techniques are designed to handle metric inputs, such as isomap [64], Locally Linear Embedding (LLE) [65], and Laplacian Eigenmap (LE) [66], where a neighbourhood graph is used to capture local distance proximities and build a data-driven model of the space.

6 Preparation for the Experiments

In the works cited so far, and in the literature in general, clustering algorithms were used very little to identify aggregate electrical daily patterns. It is clear, intuitively, that during different climate temperatures, days of the week and times during the day different electrical behaviour can be observed and classified and clustering algorithms can help with predicting these behaviours. This can help in optimizing the Smart Grid. The objective of this experiment is to run three different clustering algorithms on a data set in three different cases to describe and compare the results of each clustering algorithm.

The dataset we will use during this experiment will be the electrical demand and temperatures recorded in Italy, from the beginning of March 2003 until the end of December 2014.

Clustering algorithms to be used and the reason;

We will be using three different algorithms, each of which belongs to a clustering method:

1. **K-means:** As mentioned earlier is a centroid based clustering algorithm, which decides to which cluster a certain datapoint belongs based on the measure of the distance between points and another datapoint. we have chosen this algorithm because it's one of the most commonly used algorithms in clustering for many different applications.

2. **DBSCAN:** This algorithm is density-based clustering algorithm, which means it decides where the clusters are located based where the datapoints gather and have a high density. we chose this algorithm because it's commonly used in applications of electrical demand and has a good predictive treat.

3. **Hierarchical clustering:** This algorithm splits the dataset into different levels based on the datapoints properties. It takes a tree like shape. We will use this algorithm in order to compare the results of three different clustering methods.

6.1 *Preparations for the first experiment*

For the first experiment, we grouped the dataset based on the date and then created two table each of which has the date as a key.

Steps:

1) To start we import the dataset.

2) We then group the dataset based on date.

3) We then create the first table with the electrical demand average of the whole day, and the date as an index.

4) Then we create the second table with the temperature average of the whole day, and the date as an index.

5) We then merge both tables with the date as the key for the newly created table.

6) Now we can plot the dataset based on the clusters that were generated by the algorithm.

7) Then we can plot another graph, but instead of it being grouped based on the clusters, we can use the day of week (Monday, Tuesday, Wednesday, ..., etc.).

After we have the plot based on the clusters, we then compare the results from the plots of each algorithm, with the plot which is based on day of the week. We can then conclude if any of the algorithm best suits this case.

6.2 *Preparations for the second experiment*

For this experiment, we will split each part of every-day into 4 parts, each of which represents six hours; the first part consists of hours from 00:00 (Start of Day) until 06:00, the second part starts from 06:00 until 12:00, the third part from 12:00 until 18:00, and the fourth from 18:00 until 00:00 (midnight / end of day).

Steps:

1) To start we import the dataset.

2) We then group the dataset based on date and the hour of the day.

3) Then we calculate the mean for the demand and unstack the newly created table.

4) We now calculate the mean and we re-represent the demand as a ratio between the demand during that hour and the demand during the whole day.

5) Now, we can add two new columns.

    a. The first one represents the ratio of the ratio of demand during the part of the day over the ratio of demand of the whole day.

    b. The second column represents the difference between the ratio of demand during the part of the day and the ratio of demand of the whole day.

6) We now run the clustering algorithms and plot based on the clusters which are given by the algorithms.

7) We can also plot another graph based on the four sections of the day, in order to compare the graphs from the algorithms with the graph we just plotted, and decide which of the algorithms gave the best result in this case

6.3 *Preparations for the third experiment*

For this experiment, we will split the day into twenty-four sections, each representing an hour of the day, then rescale the electrical demand for a better visualization.

We can follow steps 1 to 4 from the previous experiment and then run K-means in order to obtain clusters. We can run this experiment with two clusters (Working days and Weekend days) and seven clusters (each day of the week). We then can compare the obtained graphs with the graph we plot where we group the data based on the day of the week. We only run this algorithm with K-means because it's difficult to represent the results we would get from clustering using DBSCAN and Hierarchical in a line graph.

6.4 *Preparations for the fourth experiment*

For the fourth case, we tried replotted the first case again as a three-dimensional graph. where x, y and, z axis each represent the average demand of the day, the average temperature of the day and, the day of the week respectively, using the Hierarchical algorithm.

We can follow steps 1-5 from case 1, and then follow these two steps;

1) We add a new column to our newly created table as the day of week.

2) We now run the Hierarchical clustering algorithm and plot with respect to dimensions mentioned earlier, and we should be able to see our results.

7 Results and Validation

7.1 *Results for the first experiment*

We can instantly notice that the KMEANS algorithm gave a very similar result when we ran it with 2 clusters as the Hierarchical clustering algorithm. DBSCAN gave a very different result due to the graph having high density in a big area. We can also see that none of the algorithms showed a pattern similar to the graph based on days of the week, not even when we tried to run KMEANS with 7 clusters.

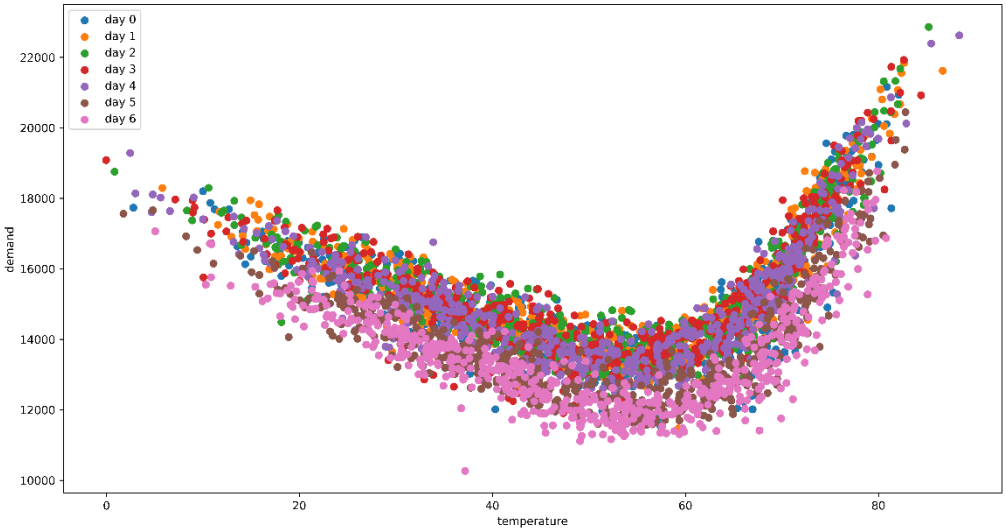

Figure 2 – Scatter Plot of the datapoints grouped by days of the week for Experiment 1

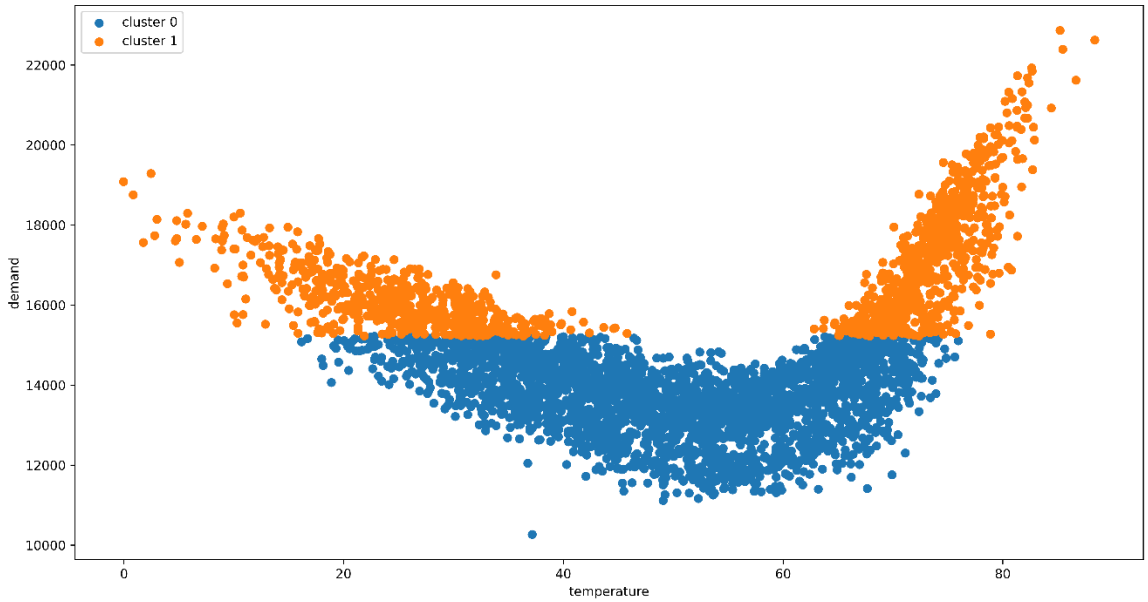

Figure 3 – Scatter Plot of the datapoints grouped by clusters found by K-means for Experiment 1 (2 clusters)

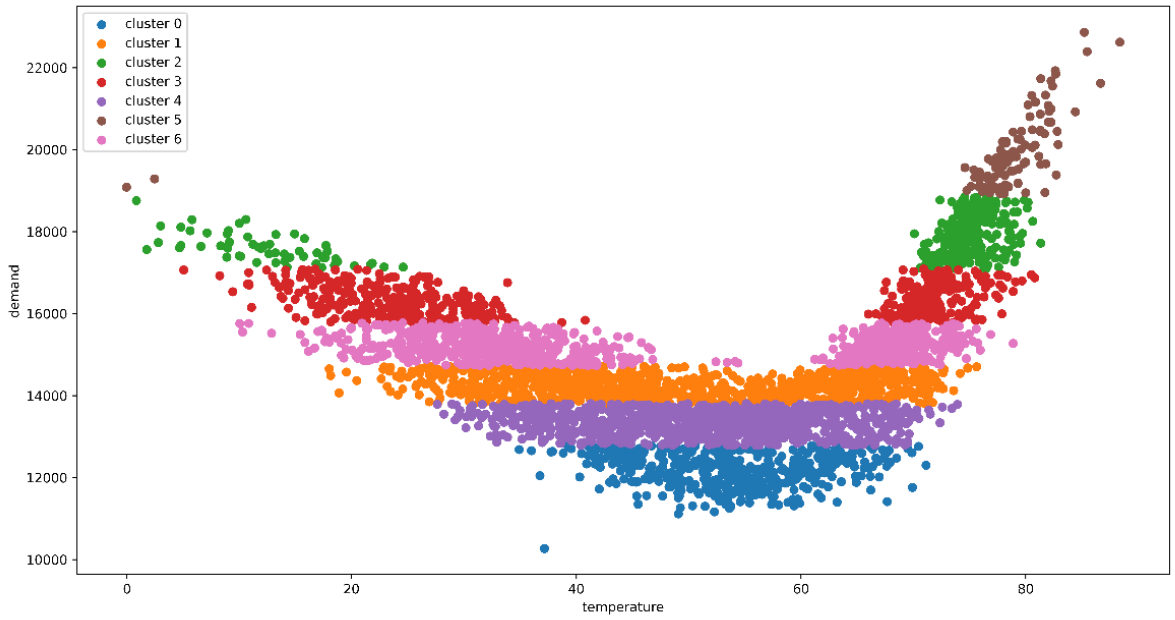

Figure 4 – Scatter Plot of the datapoints grouped by clusters found by K-means for Experiment 1 (7 clusters)

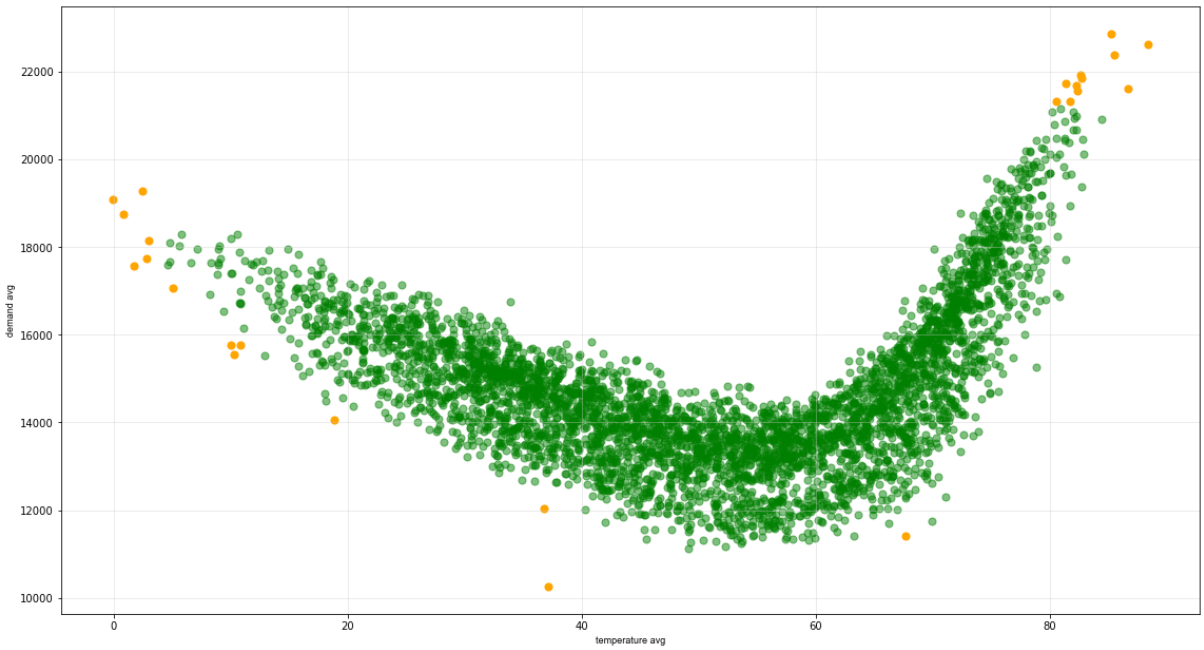

Figure 5 – Scatter Plot of the datapoints grouped by clusters and outliers found using DBSCAN for Experiment 1

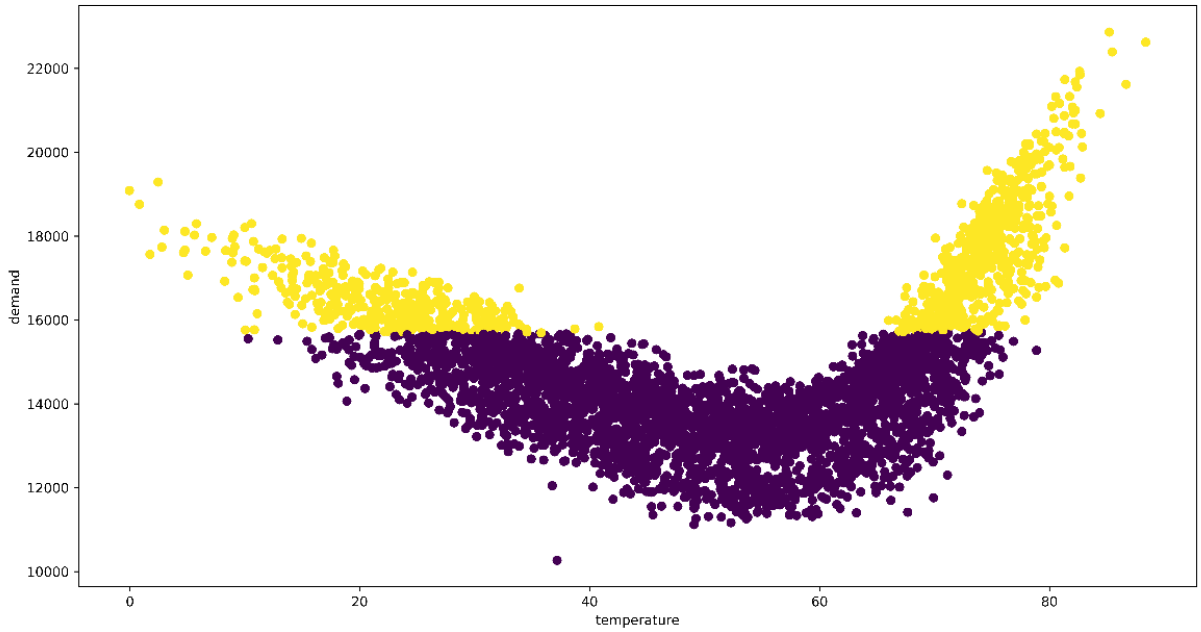

Figure 6 – Scatter Plot of the datapoints grouped by clusters found by the Hierarchical algorithm for Experiment 1

7.2 *Results for the second experiment*

KMEANS showed good results, it was able to identify clusters 1 (orange) and 2 (green), matching the pattern in the graph we plotted based on section of day. As for Hierarchical clustering, it was only able to identify cluster 2 (light blue). Finally, we can clearly see that DBSCAN showed a great result in this experiment, we can see that each cluster matches certain part of the day.

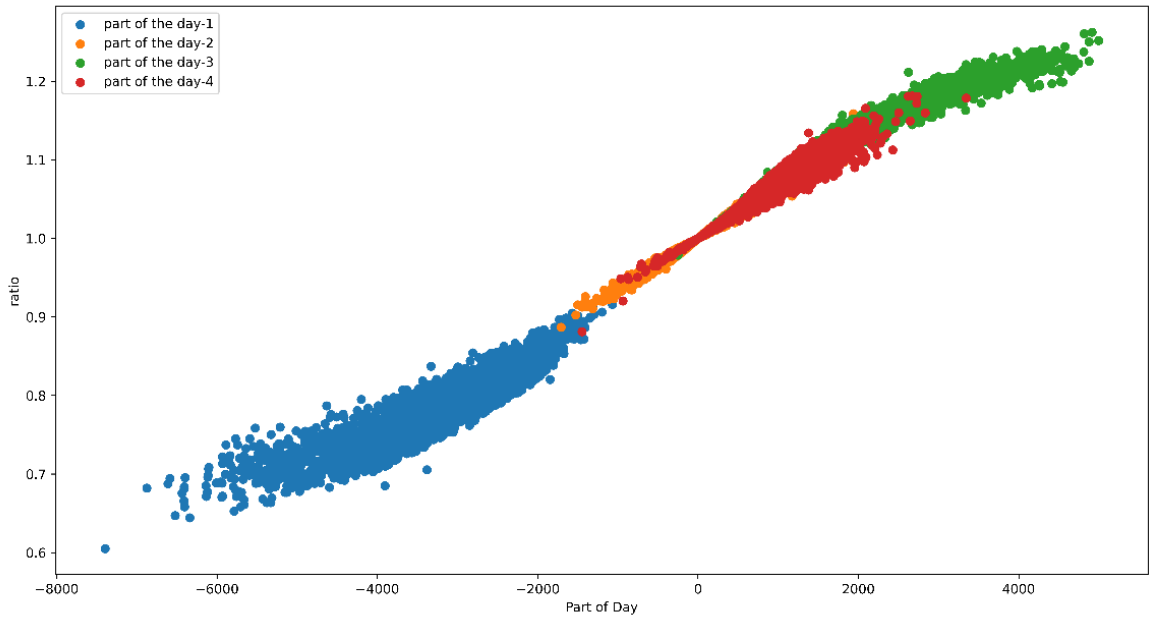

Figure 7 – Scatter Plot of the datapoints grouped by the four parts of the day (6 hours) for Experiment 2

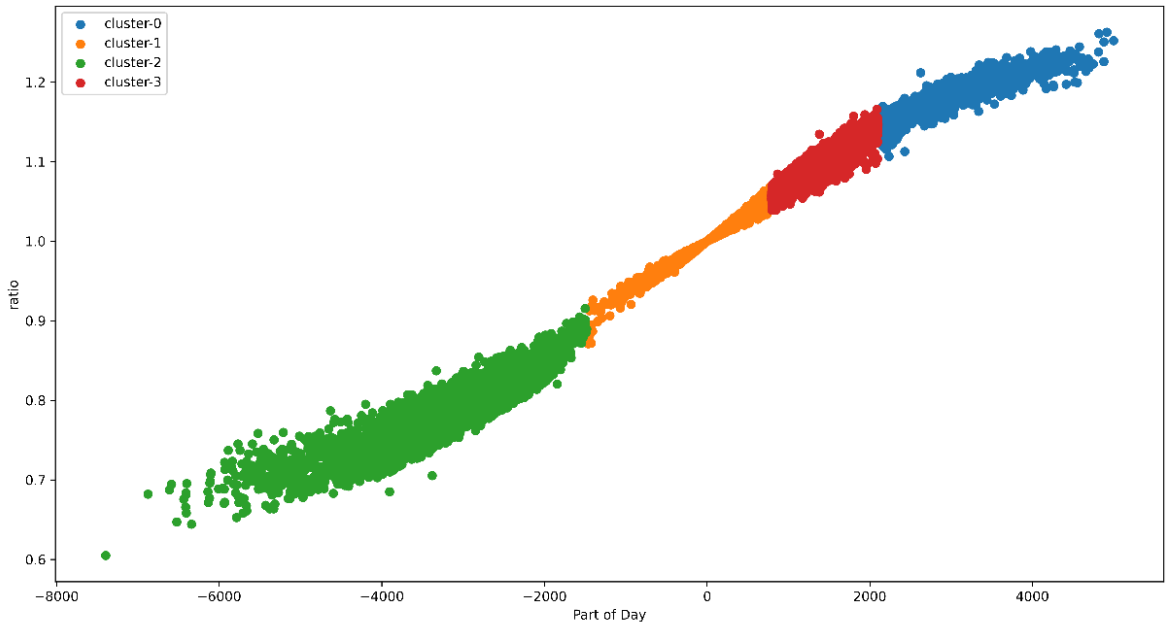

Figure 8 – Scatter Plot of the datapoints grouped by clusters found by K-means for Experiment 1 (4 clusters)

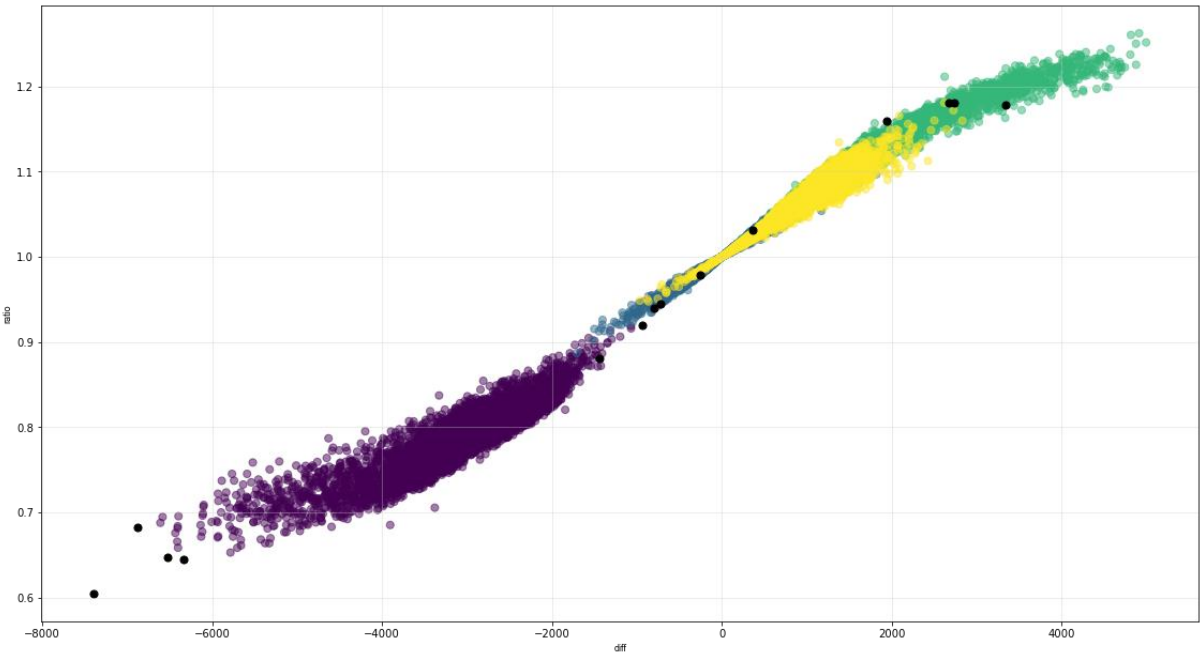

Figure 9 – Scatter Plot of the datapoints grouped by clusters and outliers found using DBSCAN for Experiment 2

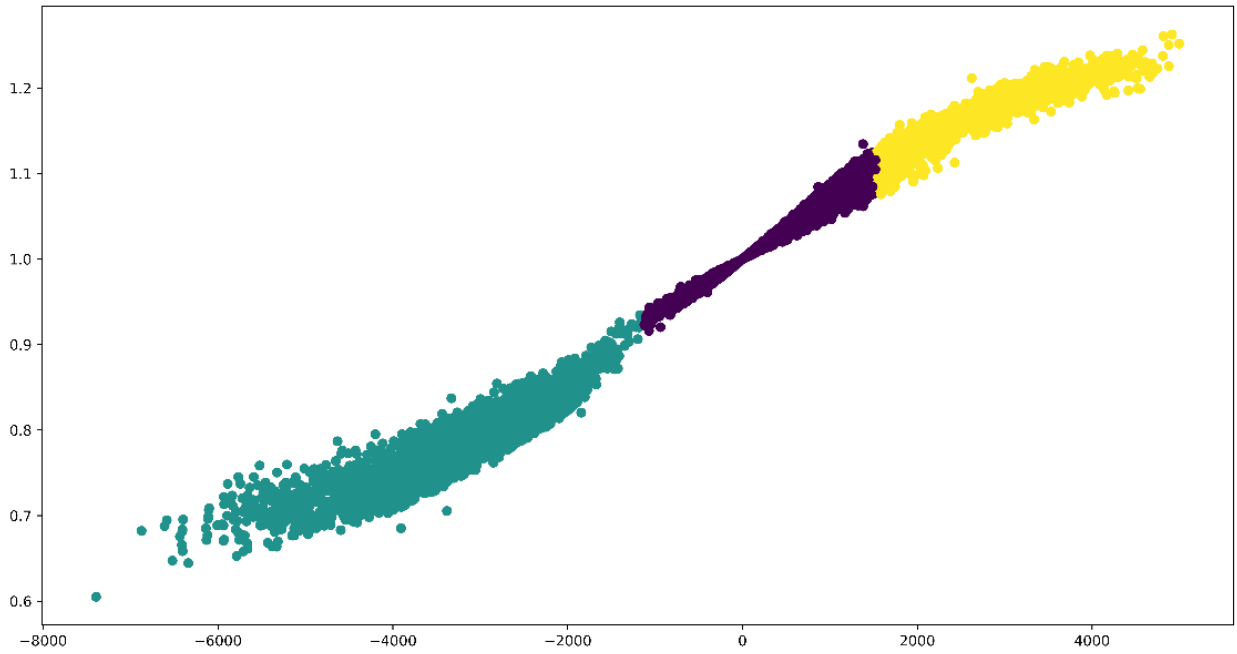

Figure 10 – Scatter Plot of the datapoints grouped by clusters found by the Hierarchical algorithm for Experiment 2

7.3 *Results for the fourth experiment*

KMEANS showed great results when ran with 2 clusters. We can see a pattern where one cluster (orange) represents the working days of the week, and the other cluster (blue) represents the weekend days. The results from when we ran the algorithm with 7 clusters aren't that great. We can see a pattern forming between some of the clusters and days, but not all of them.

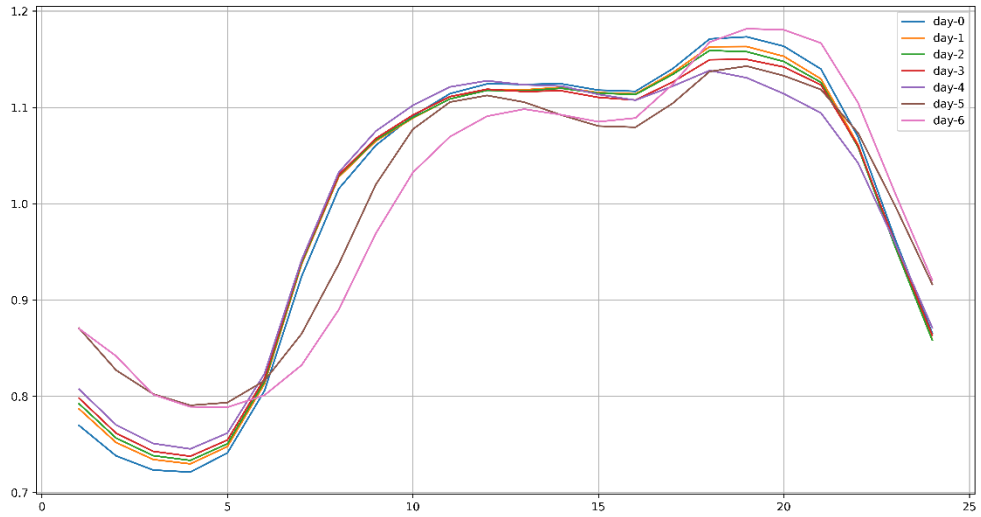

Figure 11 – Line Plot of the datapoints grouped by days of the week for Experiment 3

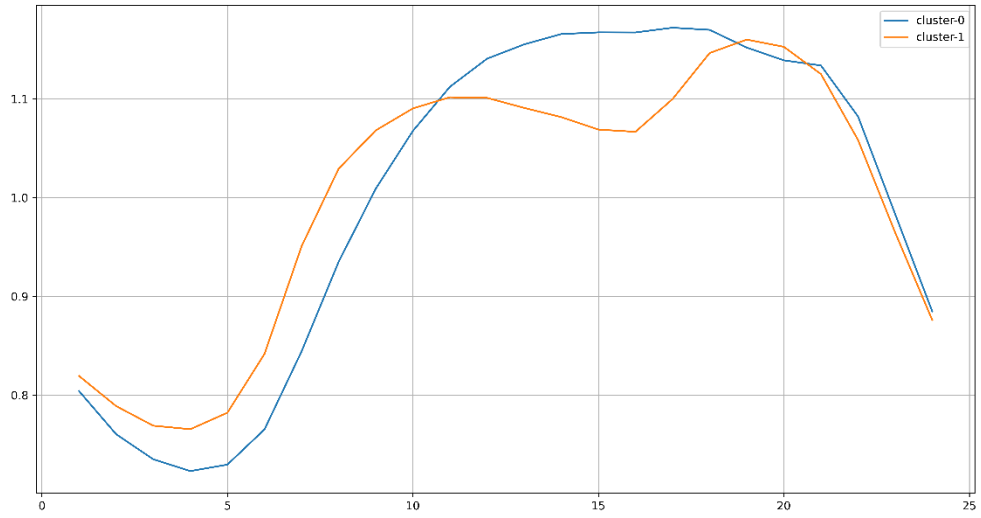

Figure 12 – Line Plot of the datapoints grouped by clusters found by K-means for Experiment 3 (2 clusters)

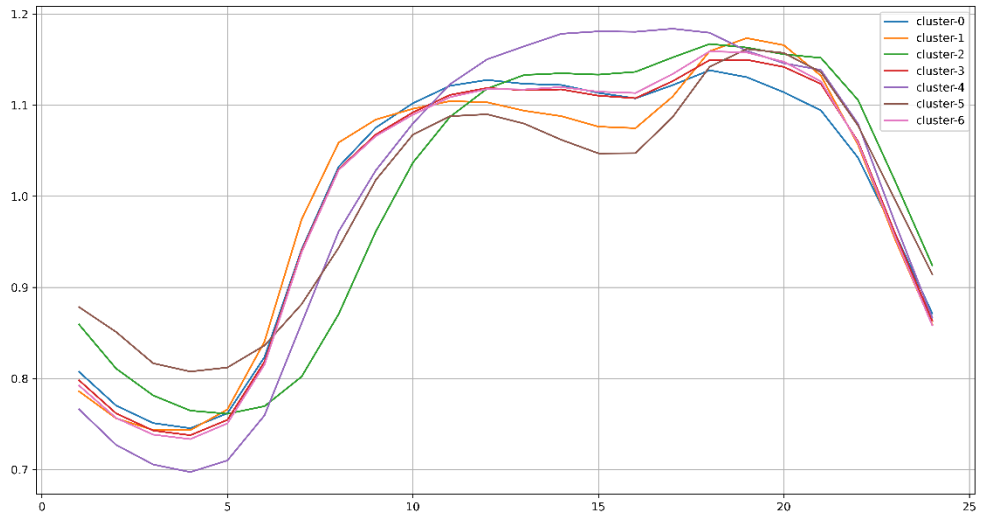

Figure 13 – Line Plot of the datapoints grouped by clusters found by K-means for Experiment 3 (7 clusters)

7.4 *Results for the fourth experiment*

In this experiment we can see the graph we obtained from the first experiment, but we added one dimension representing the days of the week, where 0 is Monday, 1 is Tuesday, 2 is Wednesday, 3 is Thursday, 4 is Friday, 5 is Saturday and 6 is Sunday. We can see the pattern that emerges throughout the days, and the similarity between the working days of the week. We can also notice in Figure 14 the tree shape that Hierarchical clustering takes the form of, where it starts with green, yellow, red, and blue respectfully.

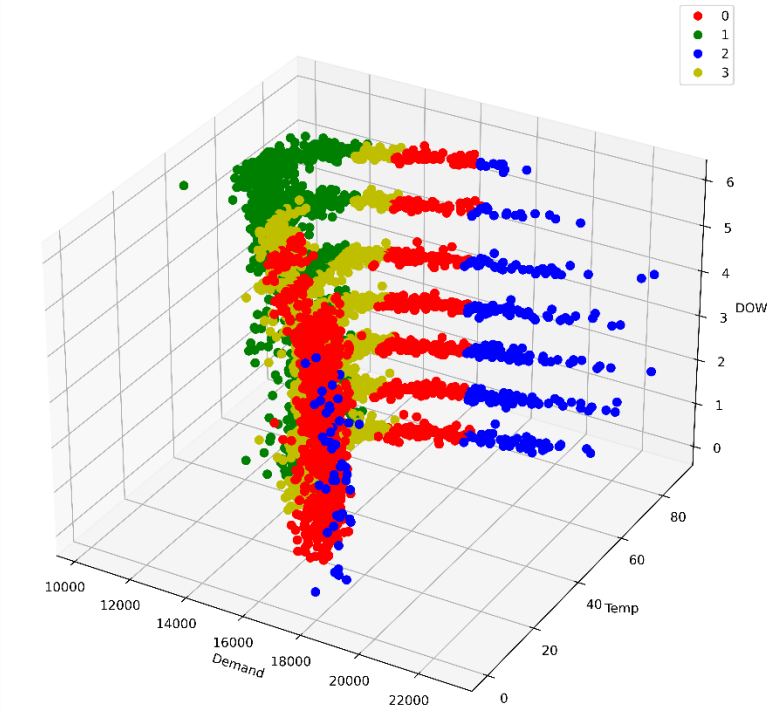

Figure 14 – Scatter Plot of the datapoints grouped by clusters found by the Hierarchical algorithm for Experiment 4

8 Conclusion

Different clustering methods and algorithms were mentioned in the literary review, in addition to some important characteristics that have huge effect on them. In the applications of electrical demand, as the data shows, clustering algorithms can be used to find the patterns hidden in the datasets. This analysis can help with predicting the electrical demand, which can help with creating the optimized grid.

DBSCAN is one of the most commonly used algorithms, and in our second experiment we can see that it gave the best results, in comparison to K-means and Hierarchical clustering. Nonetheless, K-means also showed some promising results and can provide us with some good results as well. Hierarchical clustering

also can be used in applications of electrical power demand but wouldn’t return results with the same quality as DBSCAN or K-means.

9 References:

[1] C. Bishop, Neural Networks for Pattern Recognition. New York: Oxford Univ. Press, 1995.

[2] V. Cherkassky and F. Mulier, Learning From Data: Concepts, Theory, and Methods. New York: Wiley, 1998.

[3] R. Duda, P. Hart, and D. Stork, Pattern Classification, 2nd ed. New York: Wiley, 2001.

[4] Zhu, Z.; Tang, J.; Lambotharan, S.; Chin, W.H.; Fan, Z. An integer linear programming-based optimization for home demand-side management in smart grid. In Proceedings of the Innovative Smart Grid Technologies (ISGT), Washington, DC, USA, 16–20 January 2012; pp. 1–5.

[5] Samadi, P.; Wong, V.W.S.; Schober, R. Load Scheduling and Power Trading in Systems with High Penetration of Renewable Energy Resources. IEEE Trans. Smart Grid 2016, 7, 1802–1812.

[6] Chen, X.; Zhou, Y.; Duan, W.; Tang, J.; Guo, Y. Design of intelligent Demand Side Management system respond to varieties of factors. In Proceedings of the China International Conference on Electricity Distribution (CICED), Nanjing, China, 13–16 September 2010; pp. 1–5.

[7] Hahn, H.; Meyer-Nieberg, S.; Pickl, S. Electric load forecasting methods: Tools for decision making. Eur. J. Oper. Res. 2009, 199, 902–907.

[8] Wang, K.; Yu, J.; Yu, Y.; Qian, Y.; Zeng, D.; Guo, S.; Xiang, Y.; Wu, J. A survey on energy internet: Architecture, approach and emerging technologies. IEEE Syst. J. 2017, 12, 2403–2416.]. In [Jiang, H.; Wang, K.; Wang, Y.; Gao, M.; Zhang, Y. Energy big data: A survey. IEEE Access 2016, 4, 3844–3861.

[9] D.J.C. MacKay, Information Theory, Inference, and Learning Algorithms, VIth ed., Cambridge University Press, 2007.

[10] Bailey, Ken (1994). "Numerical Taxonomy and Cluster Analysis". Typologies and Taxonomies. p. 34.ISBN 9780803952591.

[11] Practical approach towards data mining and its analysis, AnuradhaSrinivas (2013).].

[12] The choice of metrics for clustering algorithms, Peter Grabusts, Environment. Technology. Resources Proceedings of the 8th International Scientific and Practical Conference. Volume I1 © RēzeknesAugstskola, Rēzekne, RA Izdevniecība, 2011.

[13] Aloise, D.; Deshpande, A.; Hansen, P.; Popat, P. (2009). "NPhardness of Euclidean sum-of-squares clustering". Machine Learning 75: 245–249. doi:10.1007/s10994-009-5103-0

[14] D.L., D., 2006. For Most Large Underdetermined Systems of Linear Equations the Minimal L(1)-norm Solution Is also the Sparsest Solution. Communications on Pure and Applied Mathematics.

[15] AR Mohazab, SS Plotkin,"Minimal Folding Pathways for CoarseGrained Biopolymer Fragments" Biophysical Journal, Volume 95, Issue 12, Pages 5496–5507.

[16] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of KDD'96, 1996.]

[17] Jai Kaur, P., 2022. Cluster quality-based performance evaluation of hierarchical clustering method.

[18] T. Kohonen, Self-Organising Maps, 3rd ed.: Springer, 2001. ISBN: 978-3540679219.

[19] Handbook of Pattern Recognition and Computer Vision, C. Chen, L. Pau, and P.Wang, Eds.,World Scientific, Singapore, 1993, pp. 3–32. R. Dubes, "Cluster analysis and related issue".

[20] G. Milligan and M. Cooper, "An examination of procedures for determining the number of clusters in a data set," Psychometrika, vol. 50, pp. 159–179, 1985.

[21] B. Everitt, S. Landau, and M. Leese, Cluster Analysis. London: Arnold, 2001.

[22] R. Kathari and D. Pitts, “On finding the number of clusters,” Pattern Recognit. Lett., vol. 20, pp. 405–416, 1999.

[23] M. Girolami, “Mercer kernel based clustering in feature space,” IEEE Trans. Neural Netw., vol. 13, no. 3, pp. 780–784, May 2002.

[24] D. Gusfield, Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology. Cambridge, U.K.: Cambridge Univ. Press, 1997.

[25] R. Sun and C. Giles, “Sequence learning: Paradigms, algorithms, and applications,” in LNAI 1828, . Berlin, Germany, 2000.

[26] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge, U.K.: Cambridge Univ. Press, 1998.

[27] SWISS-PROT Protein Knowledgebase Release 45.0 Statistics.

[28] D. Sankoff and J. Kruskal, Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison. Stanford, CA: CSLI Publications, 1999.

[29] V. Guralnik and G. Karypis, “Ascalable algorithm for clustering sequential data,” in Proc. 1st IEEE Int. Conf. Data Mining (ICDM’01), 2001, pp. 179–186.

[30] T. Morzy, M. Wojciechowski, and M. Zakrzewicz, “Pattern-oriented hierarchical clustering,” in Proc. 3rd East Eur. Conf. Advances in Databases and Information Systems, 1999, pp. 179–190.

[31] T. Oates, L. Firoiu, and P. Cohen, “Using dynamic time warping to bootstrap HMM-based clustering of time series,” in Sequence Learning. ser. LNAI 1828, R. Sun and C. Giles, Eds. Berlin, Germany: Springer-Verlag, 2000, pp. 35–52.

[32] L. Owsley, L. Atlas, and G. Bernard, “Self-organizing feature maps and hidden Markov models for machine-tool monitoring,” IEEE Trans. Signal Process., vol. 45, no. 11, pp. 2787–2798, Nov. 1997.

[33] “Clustering sequences with hidden Markov models,” in Advances in Neural Information Processing, M. Mozer, M. Jordan, and T. Petsche, Eds. Cambridge, MA: MIT Press, 1997, vol. 9, pp. 648–654.

[34] L. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” Proc. IEEE, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[35] Z. Huang, “Extensions to the K-means algorithm for clustering large data sets with categorical values,” Data Mining Knowl. Discov., vol. 2, pp. 283–304, 1998.

[36] C. Ordonez and E. Omiecinski, “Efficient disk-based K-means clustering for relational databases,” IEEE Trans. Knowl. Data Eng., vol. 16, no. 8, pp. 909–921, Aug. 2004.

[37] P. Bradley, U. Fayyad, and C. Reina, “Scaling clustering algorithms to large databases,” in Proc. 4th Int. Conf. Knowledge Discovery and Data Mining (KDD’98), 1998, pp. 9–15.

[38] “Clustering very large databases using EM mixture models,” in Proc. 15th Int. Conf. Pattern Recognition, vol. 2, 2000, pp. 76–80.

[39] M. Ester, H. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in Proc. 2nd Int. Conf. Knowledge Discovery and Data Mining (KDD’96), 1996, pp. 226–231.

[40] A. Hinneburg and D. Keim, “An efficient approach to clustering in large multimedia databases with noise,” in Proc. 4th Int. Conf. Knowledge Discovery and Data Mining (KDD’98), 1998, pp. 58–65.], [R. Ng and J. Han, “CLARANS: A method for clustering objects for spatial data mining,” IEEE Trans. Knowl. Data Eng., vol. 14, no. 5, pp. 1003–1016, Sep.-Oct. 2002.

[41] G. Sheikholeslami, S. Chatterjee, and A. Zhang, “WaveCluster: A multiresolution clustering approach for very large spatial databases,” in Proc. 24th VLDB Conf., 1998, pp. 428–439.

[42] L. Kaufman and P. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis: Wiley, 1990.

[43] S. Guha, R. Rastogi, and K. Shim, “CURE: An efficient clustering algorithm for large databases,” in Proc. ACM SIGMOD Int. Conf. Management of Data, 1998, pp. 73–84.

[44] E. Dahlhaus, “Parallel algorithms for hierarchical clustering and applications to split decomposition and parity graph recognition,” J. Algorithms, vol. 36, no. 2, pp. 205–240, 2000.

[45] C. Olson, “Parallel algorithms for hierarchical clustering,” Parallel Comput., vol. 21, pp. 1313–1325, 1995.

[46] K. Stoffel and A. Belkoniene, “Parallel K-means clustering for large data sets,” in Proc. EuroPar’99 Parallel Processing, 1999, pp. 1451–1454.

[47] G. Carpenter and S. Grossberg, “A massively parallel architecture for a self-organizing neural pattern recognition machine,” Comput. Vis. Graph. Image Process., vol. 37, pp. 54–115, 1987.

[48] “The ART of adaptive pattern recognition by a self-organizing neural network,” IEEE Computer, vol. 21, no. 3, pp. 77–88, Mar. 1988.

[49] B. Moore, “ART1 and pattern clustering,” in Proc. 1988 Connectionist Models Summer School, 1989, pp. 174–185.

[50] R. Clarke, et al., “The properties of high-dimensional data spaces: Implications for exploring gene and protein expression data,” Nature Rev. Cancer, vol. 8, no. 1, pp. 37–49, 2008.

[51] C. Turkay, P. Filzmoser, and H. Hauser, “Brushing dimensions—A dual visual analysis model for high-dimensional data,” IEEE Trans. Vis. Comput. Graph., vol. 17, no. 12, pp. 2591–2599, Dec. 2011.

[52] D. Engel, et al., “Visual steering and verification of mass spectrometry data factorization in air quality research,” IEEE Trans. Vis. Comput. Graph., vol. 18, no. 12, pp. 2275–2284, Dec. 2012.

[53] D. Maljovec, B. Wang, V. Pascucci, P.-T. Bremer, and D. Mandelli, “Analyzing dynamic probabilistic risk assessment data through topology-based clustering,” in Proc. Int. Topical Meeting Probabilistic Safety Assessment Anal., 2013, pp. 1839–1854.

[54] S. Gerber, P. Bremer, V. Pascucci, and R. Whitaker, “Visual exploration of high dimensional scalar functions,” IEEE Trans. Vis. Comput. Graph., vol. 16, no. 6, pp. 1271–1280, Nov./Dec. 2010.

[55] A. Inselberg, Parallel Coordinates: Visual Multidimensional Geometry and Its Applications. Berlin, Germany: Springer, 2009.

[56] J. Heinrich and D. Weiskopf, “State of the art of parallel coordinates,” STAR Proc. Eurographics, vol. 2013, pp. 95–116, 2013.

[57] E. Bertini, A. Tatu, and D. Keim, “Quality metrics in highdimensional data visualization: An overview and systematization,” IEEE Trans. Vis. Comput. Graph., vol. 17, no. 12, pp. 2203–2212, Dec. 2011.

[58] G. Ellis and A. Dix, “A taxonomy of clutter reduction for information visualisation,” IEEE Trans. Vis. Comput. Graph., vol. 13, no. 6, pp. 1216–1223, Nov./Dec. 2007.

[59] P. E. Hoffman and G. G. Grinstein, “A survey of visualizations for high-dimensional data mining,” in Information Visualization in Data Mining and Knowledge Discovery. San Francisco, CA, USA: Morgan Kaufmann, 2002, pp. 47–82.

[60] D. A. Keim, “Information visualization and visual data mining,” IEEE Trans. Vis. Comput. Graph., vol. 8, no. 1, pp. 1–8, Jan.–Mar. 2002.

[61] M. C. F. De Oliveira and H. Levkowitz, “From visual data exploration to visual data mining: A survey,” IEEE Trans. Vis. Comput. Graph., vol. 9, no. 3, pp. 378–394, Jul.–Sep. 2003.

[62] A. Buja, D. Cook, and D. F. Swayne, “Interactive high-dimensional data visualization,” J. Comput. Graphical Statist., vol. 5, no. 1, pp. 78–99, 1996.

[63] I. Jolliffe, Principal Component Analysis. Hoboken, NJ, USA: Wiley, 2005.

[64] J. B. Tenenbaum, V. De Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” Science, vol. 290, no. 5500, pp. 2319–2323, 2000.

[65] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” Science, vol. 290, no. 5500, pp. 2323–2326, 2000.

[66] M. Belkin and P. Niyogi, “Laplacian eigenmaps for dimensionality reduction and data representation,” Neural Comput., vol. 15, no. 6, pp. 1373–1396, 2003.

[67] Manish Verma, Mauly Srivastava, Neha Chack, Atul Kumar Diswar, Nidhi Gupta / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 2, Issue 3, May-Jun 2012, pp.1379-1384
.