Anomaly Clustering based on Correspondence Analysis

June 2016

By

Humayra Islam

(Student ID: **11261008**)

In partial fulfillment of the requirements for the Degree Of

Master of Science in Electrical and Electronic Engineering

Department of Electrical and Electronic Engineering

APPROVAL FORM

We hereby certify that:

Name	: Humayra Islam
Student ID	: 11261008
Thesis Title	: Anomaly Clustering based on Correspondence Analysis
Submission Date	: June 2016

Has passed the thesis exam and confirmed that this thesis had been thoroughly examined, improved, and approved by advisors.

Approved by

Dr. Tarem Ahmed
Supervisor
Dept. of EEE

Supriyo Shafkat Ahmed
Internal Examiner
Dept. of EEE

Amitabha Chakrabarty, Ph.D
External Examiner
Dept. of CSE

Dr. Mohammed Belal Hossain Bhuian
Exam Committee Chair
Dept. of EEE

Acknowledged

Dr. Md. Sayeed Salam
Chairperson
Dept. of EEE

DECLARATION

This thesis report is based upon research conducted by myself. Results used for reference, that are obtained by others is appropriately cited. This thesis has not been previously submitted for any degree.

Signature of Student

DEDICATION

This thesis is dedicated to my beloved parents who have supported me all the way since the beginning of my studies.

ACKNOWLEDGEMENTS

By the Grace of Almighty Allah, I have reached the goal of completing my thesis in such an ongoing and promising topic in the present world. I would like to thank my supervisor Dr. Tarem Ahmed for giving me the opportunity to work under his supervision. He has always helped me by giving suggestions, ideas, advice and support to solve all my problems and guided me to develop this thesis for this semester.

I also would like to thank my parents and family for making it possible for me to study and for their constant help and support. I have worked hard and gave the best. Hopefully, this work will be appreciated by my supervisor and respected faculties.

ABSTRACT

The huge amount of traffic in backbone IP networks produces various kinds of anomalies in data packets. Distinct classifiers have been developed to deal with this anomalous data. These classifiers typically have predefined number of classes and use supervised learning methods. Some classifiers apply windowing method to make the huge data scalable into small groups. In this work, a new method for the classification of anomalous data have been applied with unsupervised learning using Correspondence Analysis (CA). Correspondence Analysis does not need a predefined number of clusters to begin with, and can handle comparatively large amounts of data. Results have been compared with other clustering techniques, which are applied on real data from the US Abilene backbone network. The results indicate that the proposed method is promising in classifying anomalies on the basis of frequencies of anomalous facade.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

This chapter of the work contains the background of the field related to the research, problem that is faced in networking now a days, proposed solution to the problem and finally the sequential organization of the paper is mentioned at the end.

1.1 Background

Network traffic often deviates from norm and as a result, it has gained much concern in studies of network behavior. These deviations are termed as anomalies and there have been many works that concentrate on identifying them which is known as anomaly detection. However, few works have been done to place them in proper categories, i.e. anomaly classification for further usage. The main idea of this work was focused on placing the detected anomalies in proper category.

1.2 Problem Statement

Most of the works related to classification of anomaly involve using predefined number of classes or categories with predefined parameters. Most data clustering algorithms require the setting of many input parameters [1]. Two main disadvantages of working with parameter-dependent algorithms are the following. First, incorrect settings may cause an algorithm to fail in finding the true patterns. Second, a perhaps more subtle problem is that the algorithm may report spurious patterns that do not really exist, or greatly overestimate the significance of the reported patterns. This is especially likely

when the user fails to understand the role of parameters in the data clustering process [2]. Methods for extracting patterns from continuous streams of data are known as incremental (online) learning algorithms. The basic idea of incremental initiation is that upon receiving a new instance, it is much less expensive to update an existing model than to build a new one. On the other hand, as indicated in [3], the incremental algorithms suffer from several shortcomings, like high sensitivity to the order of training examples and longer preparation times than the non-incremental (batch) methods. Pure incremental methods consider every new instance, which may be impractical in environments, where connections arrive at the rate of thousands per second. Some of the very recent works related to anomaly clustering include operator dependency defining of sample entropy as an estimator [4]. The drawback of this approach is that the entropy increases as the sample size increases. Another work suggested sliding window approach with predefined window size and target layer node [5].The disadvantage with this model was maintaining the window size as well as limited number of data processing. In [6], principal component analysis (PCA) was used where the components are to be adjusted beforehand. STAGGER and FLORA (built on forgetting mechanism)- sliding window approaches are two other models with limitations of data processing order of packets, classifier based approach and with time dependency [3]. In this work, the endeavor was to design an algorithm that takes the alternative approach of learning the number of clusters consisting anomalies based on their behavior.

1.3 Contributions of the thesis

The proposed algorithm classifies using correspondence analysis based on a hierarchical clustering using QR decomposition. The advantages of this algorithm include having no

predefined number of clusters, parameter independence and data processing order independence. This algorithm allows visual representation of the data clusters. For completing the task, hierarchical algorithm with agglomerative clustering have been used. Most of the prior works can be found based on divisive algorithm. Agglomerative algorithm is used because in this case the output clusters found are more outstanding and also the rate of false positive occurrence is less than the divisive algorithm.

The approach followed in current study have following features, which will be empirically demonstrate with extensive experiments:

1) It allows true exploratory data clustering, rather than forcing to impose assumptions on the data.

2) The accuracy of the proposed algorithm can be greatly superior to those of parameter-based algorithms, even if this approach allows these algorithms to search exhaustively over their parameter spaces.

3) This approach is based on compression as its foundation, and compression algorithms are typically space and time efficient. As a result, the proposed method is generally much more efficient than other algorithms, in some cases by three or four orders scale.

4) The algorithm proposed works for time series of different lengths, sampling rates, dimensionalities etc.

1.4 Organization of the remainder of the thesis

The rest of the paper is organized as follows. Section II describes the basic idea of anomaly, clustering algorithm and state of the art. Section III discussed the idea of CA

algorithm, then with alteration with QR decomposition, distance measured with advanced weighted Euclidean distance measures. Section IV shows algorithm step by step. In section V, the results from my proposed algorithm has been shown and also relative comparison with existing works. In the next two sections, in VI and VII, summary and problem faced during research have been placed. Section VIII concludes with future works in following section.

CHAPTER 2

LITERATURE REVIEW

This chapter of the work contains the chronological theoretical viewpoints related to the topic. Starting with anomaly classification, it moves towards anomaly clustering methods and existing works related with clustering. The state of the art on this topic has been stated thoroughly at the end of this chapter.

Theoretical background

2.1 Defining anomaly

Any deviation from norm of data packets in networking is known as anomaly. At first, the researchers applied their interest in detecting anomaly. Many works related to anomaly detection have taken place in last few years. But very few research works regarding clustering of detected anomalies has been done. So basically, the previous work of the proposed approach was on anomaly detection. Now it is time to focus on to know how a potential anomaly will exhibit network traffic, so that they can be put into different classes and requires less time to solve the anomaly related problems. Thus comes the part anomaly classification, which is also known as clustering.

Fig 2a. and Fig 2b. gives the graphical overviews of clustering with the help of K-medoid and K- mean algorithm respectively.

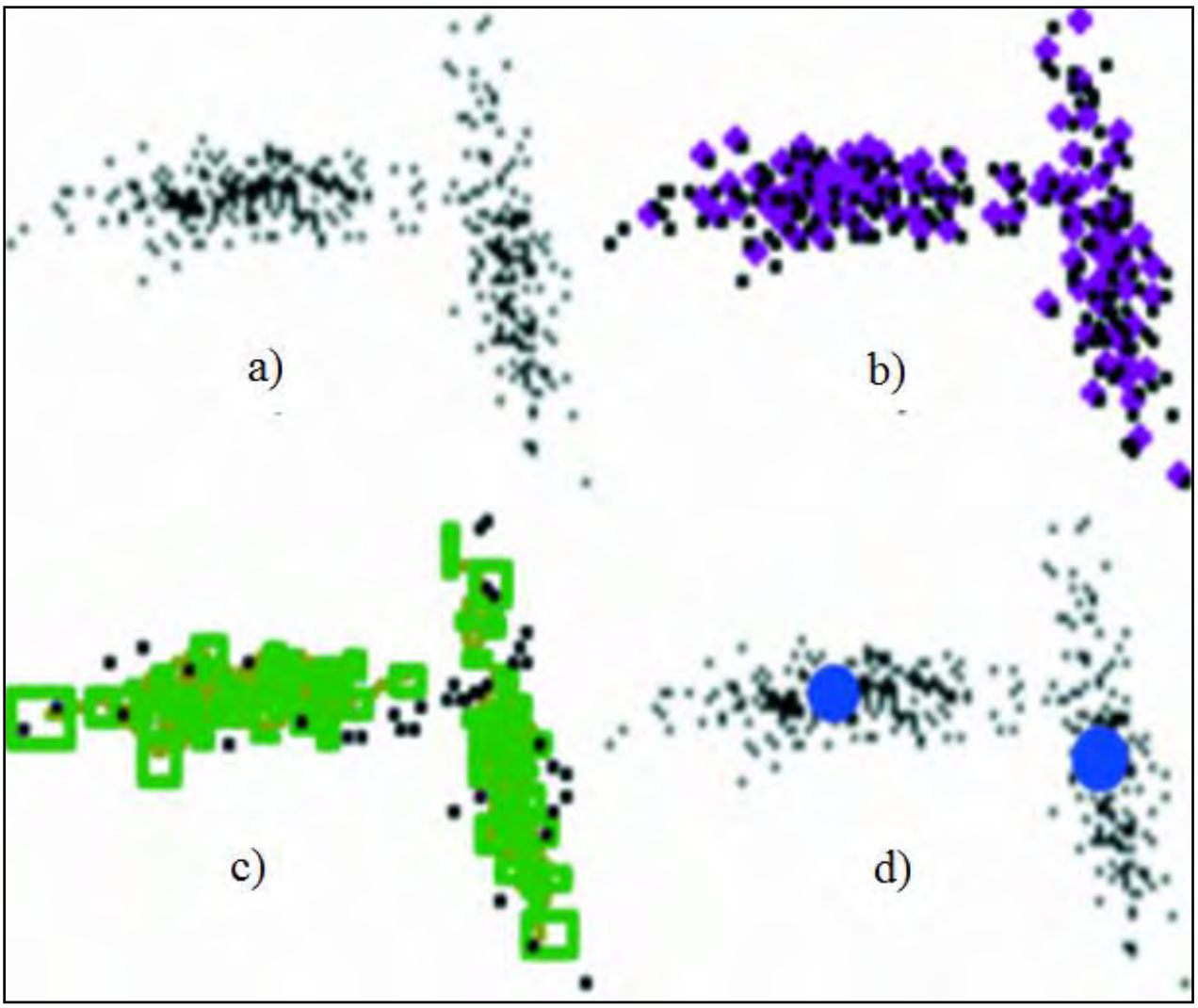


Fig 2a. An overview of data clustering (with K- medoid)

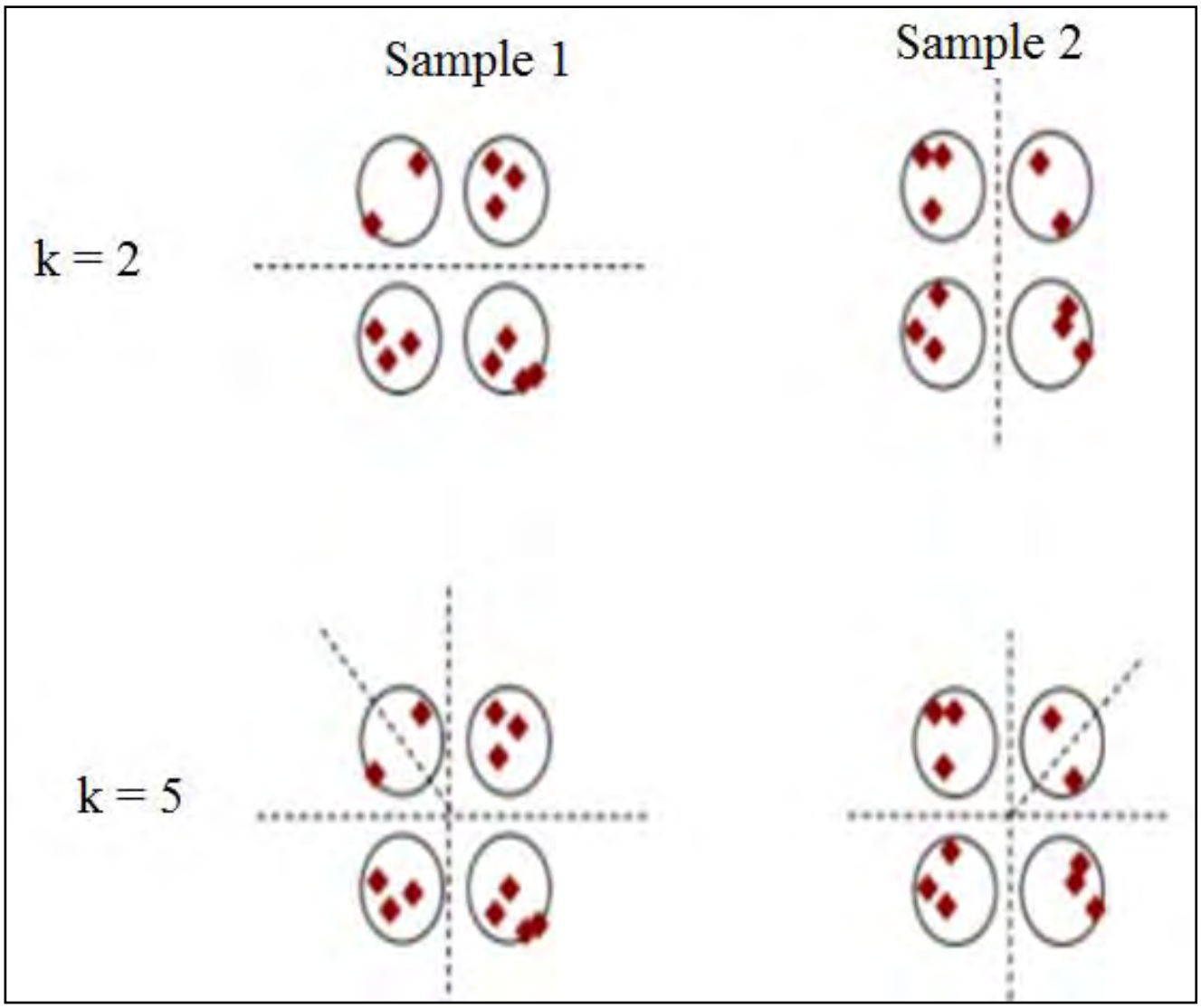


Fig 2b. An overview of data clustering (with k - mean)

2.2 Concept of Clustering

Clustering can be of different types. Such as-

- ***Demographics***: Factors based on ethnicity, economics or religion
- ***Graph theory***: Clusters of linked nodes in a network, by clustering coefficient
- ***In computing***:

  Computer cluster: Technique of linking many computers together.

  Data Cluster: Allocation of contiguous storage in databases and file systems.

  Hashtable: Mapping of keys to nearby slots**.**
- ***Statistics*** and ***data mining***: An algorithm for cluster analysis or a result thereof.

Demographic clustering is basically distribution-based clustering. Typically, demographic data contains lots of categorical variables. The mining function works comparatively well with data sets consisting of this type of variables.

It provides fast and usual clustering of very large databases. Clusters are characterized by the value distributions of their members. It automatically determines the number of clusters to be generated.

Numerical variables can also be used in demographics. The Demographic Clustering algorithm treats numerical variables by assigning similarities according to the numeric difference of the values.

This types of clustering is an iterative process over the input data. Each input record is read in succession. The similarity of each record with each of the currently existing clusters is calculated.

In graph theory, algorithms treat the patterns as points in a pattern space, so distances are available between all pairs of patterns. A complete graph is formed by connecting each pattern with all its neighbors. The edge weights are distances between pairs of patterns.

Fig 2c. shows the basic classification of hierarchical clustering, i.e.; agglomerative and divisive.

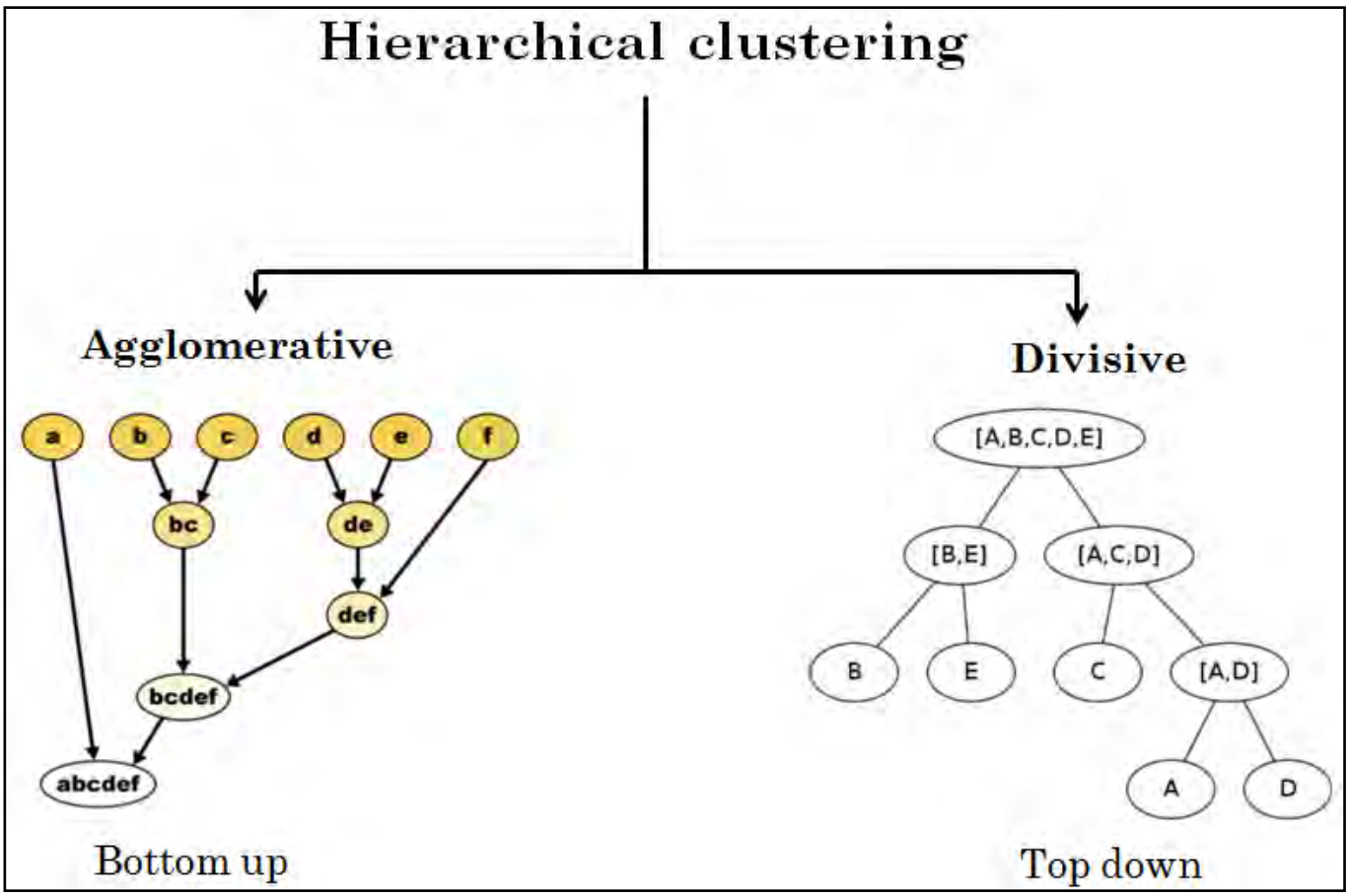


Fig 2c. Classification of Hierarchical clustering

Clustering is required in networking mainly because of its ability to group similar objects that can be used to find flaws in backbone data networks. The necessities of clustering also consists following points:

- **Scalability** − Highly scalable clustering algorithms are needed to deal with large databases.

- **Ability to deal with different kinds of attributes** − Algorithms should be capable to be applied on any kind of data such as interval-based (numerical) data, categorical, and binary data.

- **Discovery of clusters with attribute shape** − The clustering algorithm should be capable of detecting clusters of arbitrary shape. They should not be bounded to only distance measures that tend to find spherical cluster of small sizes.

- **High dimensionality** − The clustering algorithm should not only be able to handle low-dimensional data but also the high dimensional space.

- **Ability to deal with noisy data** − Databases contain noisy, missing or erroneous data. Some algorithms are sensitive to such data and may lead to poor quality clusters.

- **Interpretability** − The clustering results should be interpretable, comprehensible, and usable.

The current research is focused on Hierarchical Agglomerative clustering. As the Fig 2c. referred, there are two types of hierarchical clustering, naming- Agglomerative clustering

and Divisive clustering. In most of the previous works, divisive clustering has got much attention.

One of the main reasons for working with divisive clustering in most of the previous works is the order of computational complexity in case of agglomerative clustering is much less than the divisive clustering. Working with predefined number of clusters is much easier than working with unknown number of clusters, this is another reason behind working with divisive clustering. Because unknown number of clusters comes with a lot of iterations and also with computational complexity, hence, memory needs extra amount of space to store those extra calculations.

The proposed algorithm in this work has found a solution to address this problem, and as a result while using agglomerative clustering, needs less computations with less amount of memory usage.

Fig 2d. indicates the method for density based clustering. This clustering method detects the most densest area of data, and finds the total number of clusters. Density based clustering is used for detecting clusters from unknown shapes.

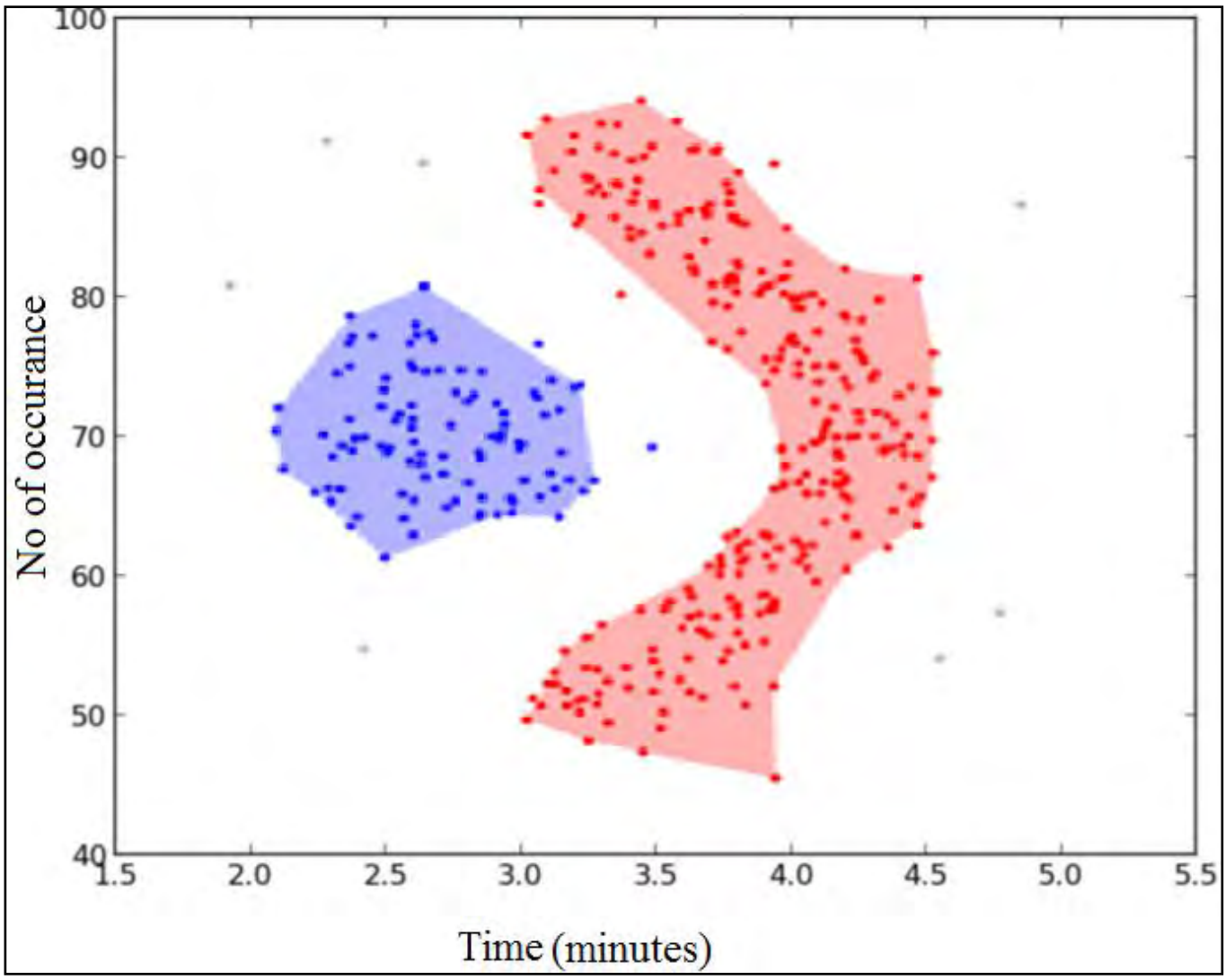


Fig 2d. Density based clustering [3]

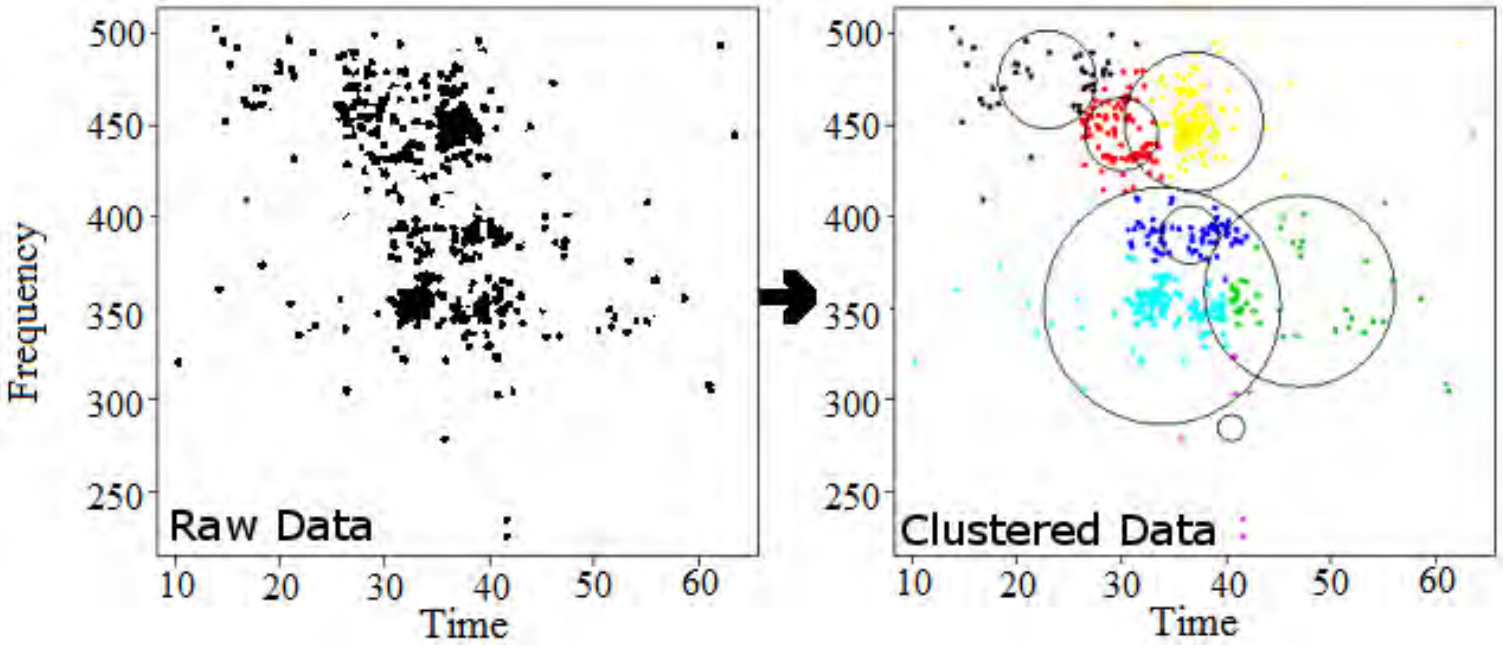


Fig 2e. K- mean clustering [3]

Fig 2e. refers one of the examples of popular K-mean clustering. In case of K-mean clustering, this algorithm needs predefined number of clusters to start with and this way it groups the clusters. Initial centroids are used to detect the centre of each cluster. After iteration, new centroids are created merging the old ones.

Finally in case of distribution clustering method shown in Fig 2f., the algorithm find out the center of the clusters and then increases the boundary to set the cluster numbers. In distribution based clustering, distance vectors are used to find the distances among different clusters.

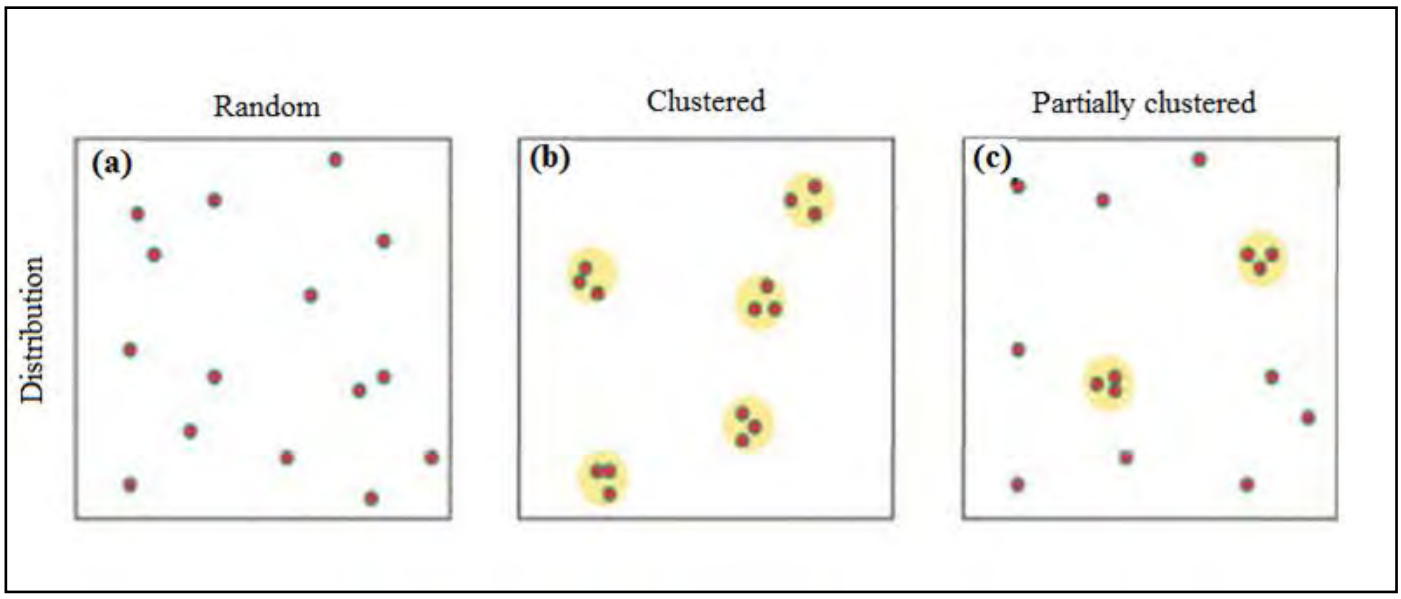


Fig 2f. Distribution clustering [7]

2.3 State of the Art

In this section, the algorithms related to clustering that is mostly related to the proposed work will be discussed. In [7], OLIN, an online classification system, dynamically adjusts the size of the training window and the number of new examples between model reconstructions to the current rate of concept drift. OLIN proposes an online classification system, which uses an info-fuzzy network or IFN shown in Fig 2g., as a base classifier.

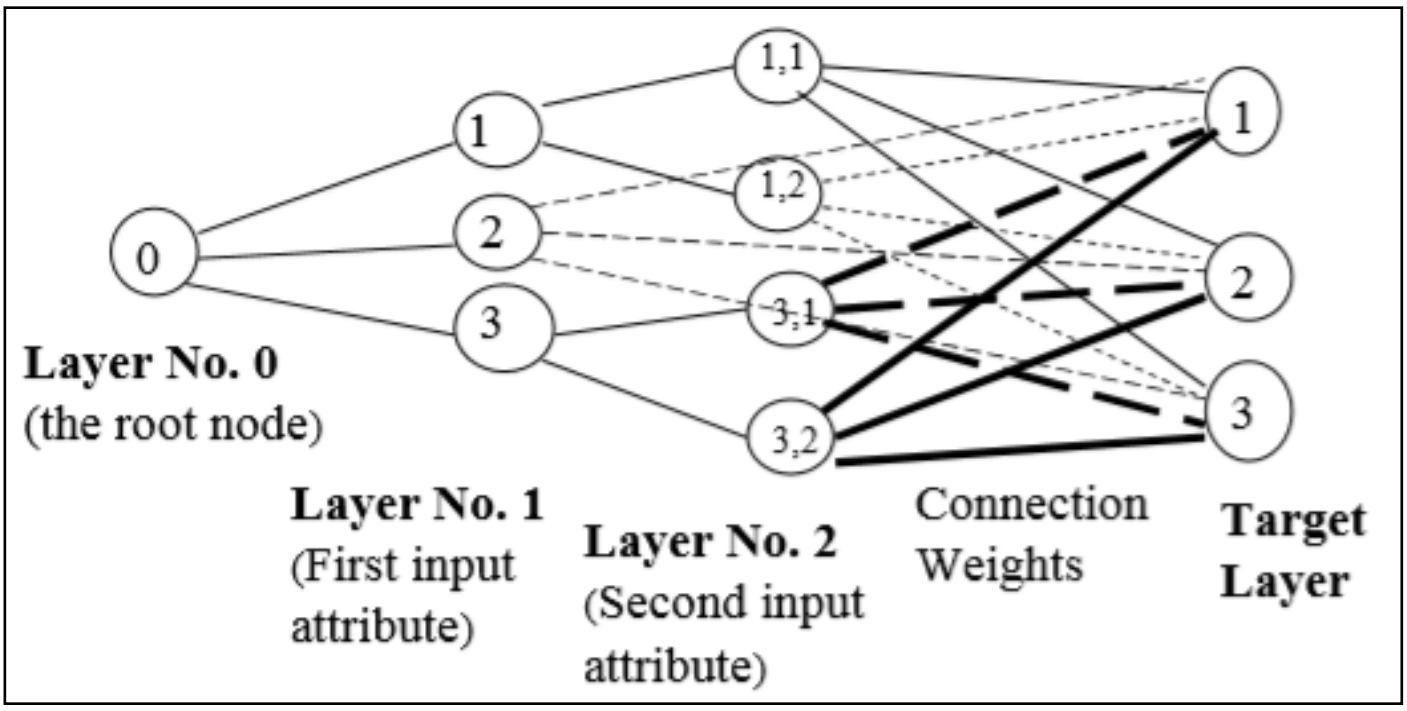


Fig 2g. An info-fuzzy network (IFN) [7]

The proposed system, called OLIN for On-Line Information Network shown in Fig 2g., adapts itself automatically to the rate of concept drift in a non-stationary data stream by dynamically adjusting the size of the training window and the rate of model update. The system does not impose any limitations on the rate, the extent, or the type of change in the underlying concept.

Like the batch version of the IFN method, it can handle both discrete and continuous attributes, the whole comparison is shown in Fig 2h. and Fig 2i. OLIN saves computer resources by increasing the update cycle when the concept appears to be stable and it shrinks the size of the training window, whenever a concept drift is detected. Thus, OLIN can be applied to a time-changing data stream of arbitrary duration. The cumulative accuracy of the models produced by OLIN tends to be higher than the accuracy obtained with a fixed-size sliding window though it may be slightly lower than the accuracy of an incremental system that does not “forget” any past examples.

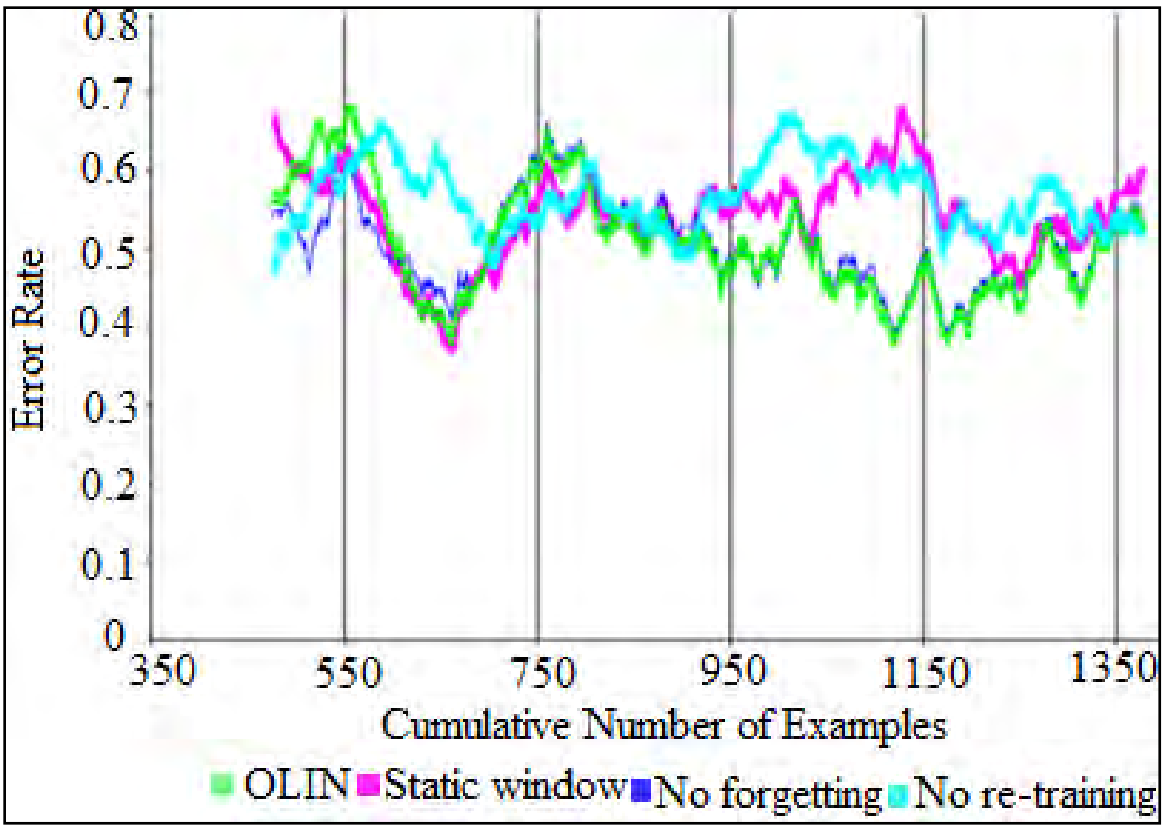


Fig 2h. Compensation rate in case of OLIN with other windowing methods [7]

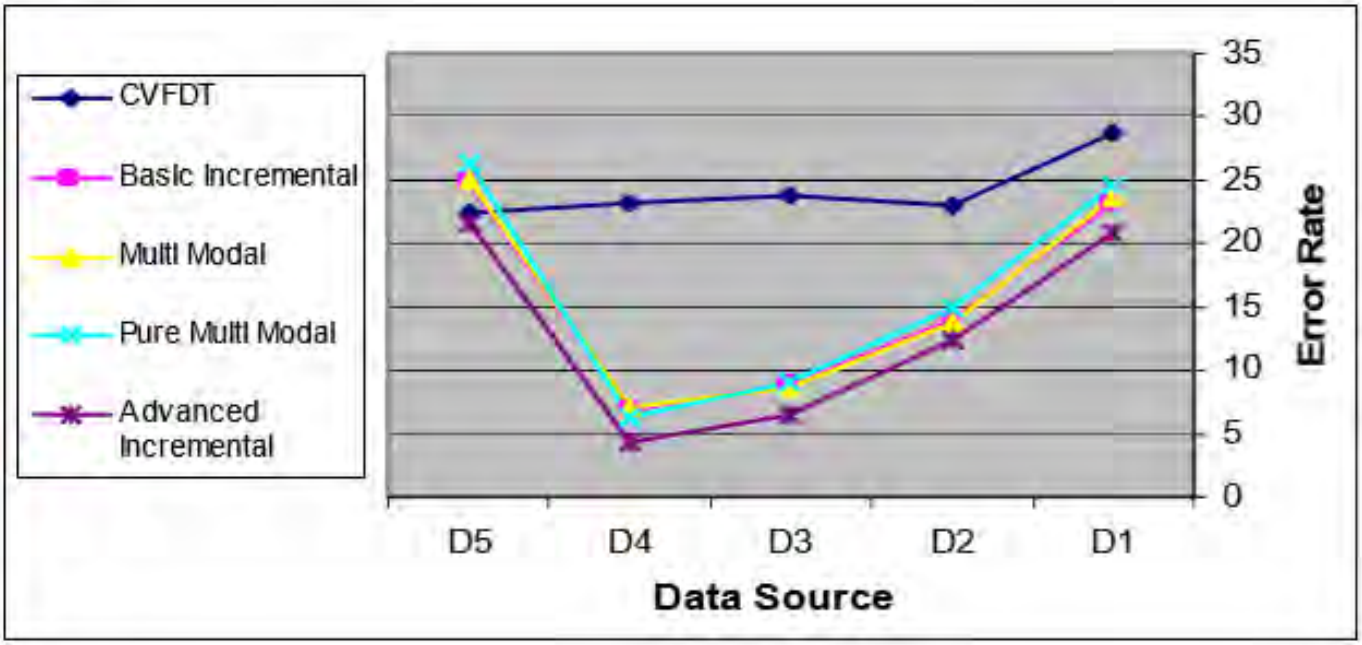


Fig 2i. Training error rate [8]

The recent research literature has proposed more tractable techniques for anomaly detection and classification [8,9,10,11]. These proposals rely on a common approach to data analysis: they apply dimensionality reduction techniques such as sketches [12, 3] or principal components [13,14] to the aggregate network traffic.

Dimensionality reduction enables computationally efficient methods for identifying outliers (or anomalies) in the data set. Dimensionality reduction or dimension reduction is the process of reducing the number of random variables under consideration via obtaining a set "uncorrelated" principle variables. In some cases, data analysis such as regression or classification can be done in the reduced space more accurately than in the original space.

Most relevant to this proposed work is that of [14], which addresses this problem by aggregating netflow into origin-destination (OD) flows, making a much smaller set of dimensions which can then be mined (using, for example, the subspace method) to find out anomalies. However, this approach can only identify which OD flow is anomalous; the particular IP flow(s) conscientious for the anomaly cannot be identified without manual examination.

Another incremental algorithms shown in Fig 2j., FLORA2, maintains a dynamically regulating window of the latest training examples. Whenever a concept drift is assumed, due to a drop in predictive accuracy or an explosion in the number of descriptions, the window size is decreased, by discarding the oldest examples. If the concept appears to be stable, the window size is left unchanged. As long as the presence of drift is indecisive, no examples are forgotten, thus incrementally increasing the window size.

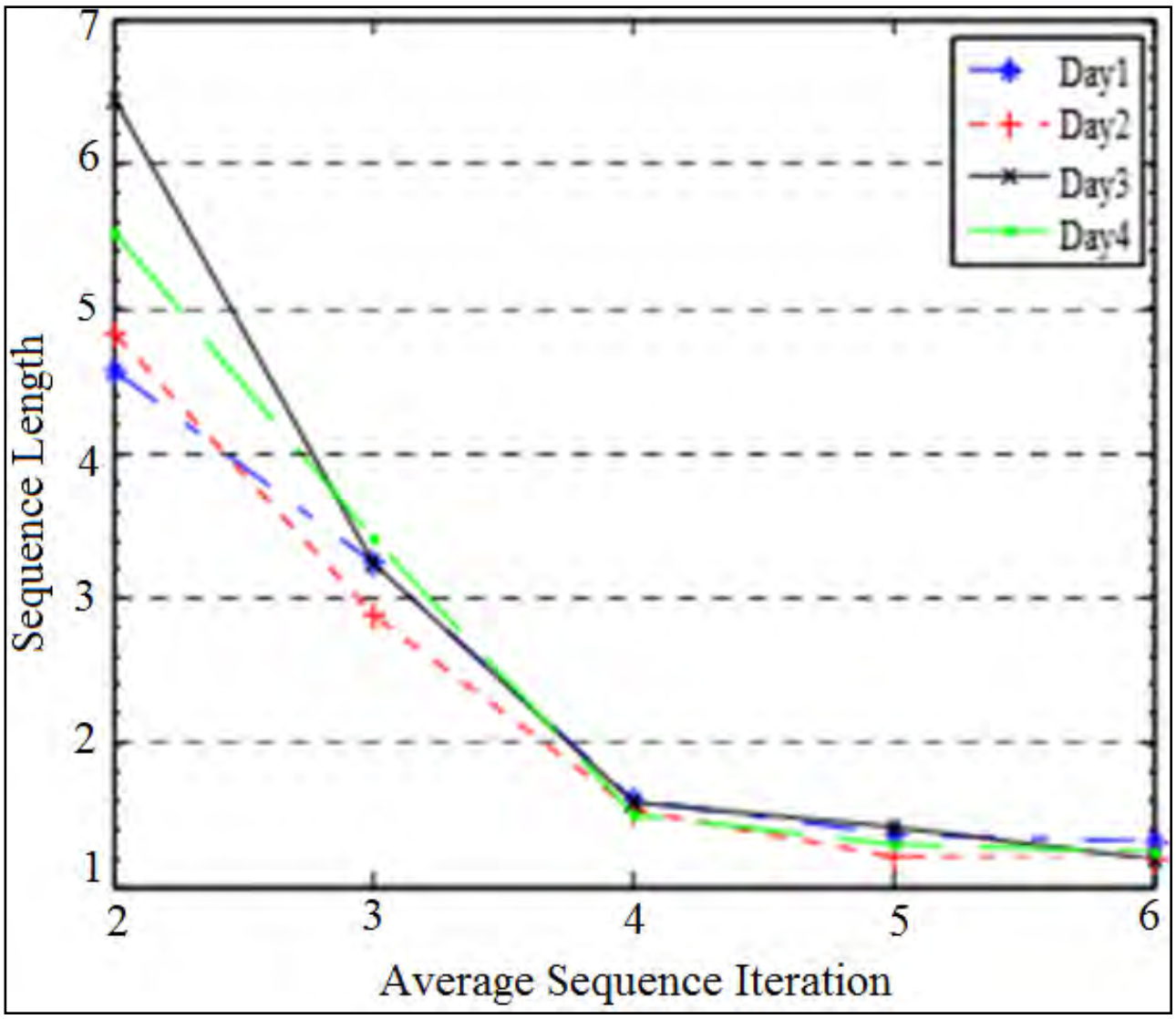


Fig 2j. Average sequence iteration with respect to sequence length [15]

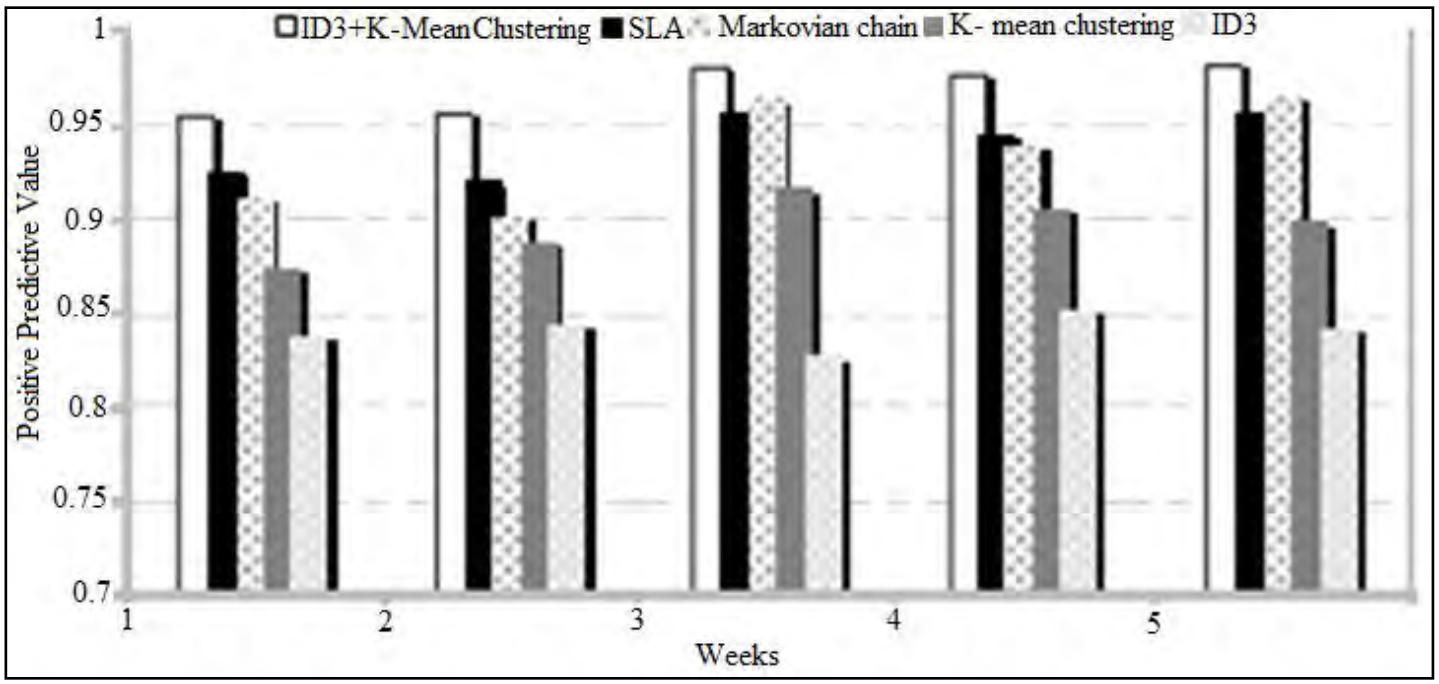


Fig 2k. Comparison of positive predictive value characteristics among approaches[19]

According to [15], this window adjustment strategy may efficiently detect radical changes in the underlying concept, subject to a relatively low rate of change. The FLORA algorithms also assume a limited rate of data arrival, since they process one example at a time.

A recent paper by Domingos and Hulten [31] deals directly with the problem of mining high-speed streams of data. Their data mining system, called VFDT(Very Fast Decision Trees learner), shown in Fig 2k., builds decision trees from symbolic attributes by using sub-sampling of the entire data stream generated by a stationary process. A similar assumption of stationary concepts is used by the incremental method of Fan et al. [19]. The sample size is determined in VFDT from distribution-free Hoeffding. This paper suggests to intend an online classification system, which also uses an info-fuzzy network or IFN, as a base classifier bounds.

A new version of the VFDT system, called CVFDT, learns decision trees from continuously changing data streams by repeatedly applying the VFDT algorithm to a sliding window of fixed size. CVFDT is aimed at detecting only one type of concept drift at the node level of the tree: namely, the importance of the current input attribute vs. other attributes. The algorithm grows an alternative subtree for each attribute having a relatively high information gain and replaces the old subtree when a new one becomes more accurate.

This paper recommends that the IFN method is able to produce much more compact models than other decision-tree methods, like CART and C4.5, while preserving nearly

the same level of predictive accuracy. Moreover, it can also be used as an efficient feature selection method.

Many learning methods use the information theory to induce classification models. One of the methods, developed by Last & Maimon [32] is the Info-Fuzzy Network algorithm (also known as Information Network-IN). IN, is an oblivious tree-like classification model, which is designed to minimize the total number of predicting attributes. Info Fuzzy Networks (IFN) is a Greedy machine learning algorithm for supervised learning. The data structure produced by the learning algorithm is also called Info Fuzzy Network. IFN construction is quite similar to decision tree construction.

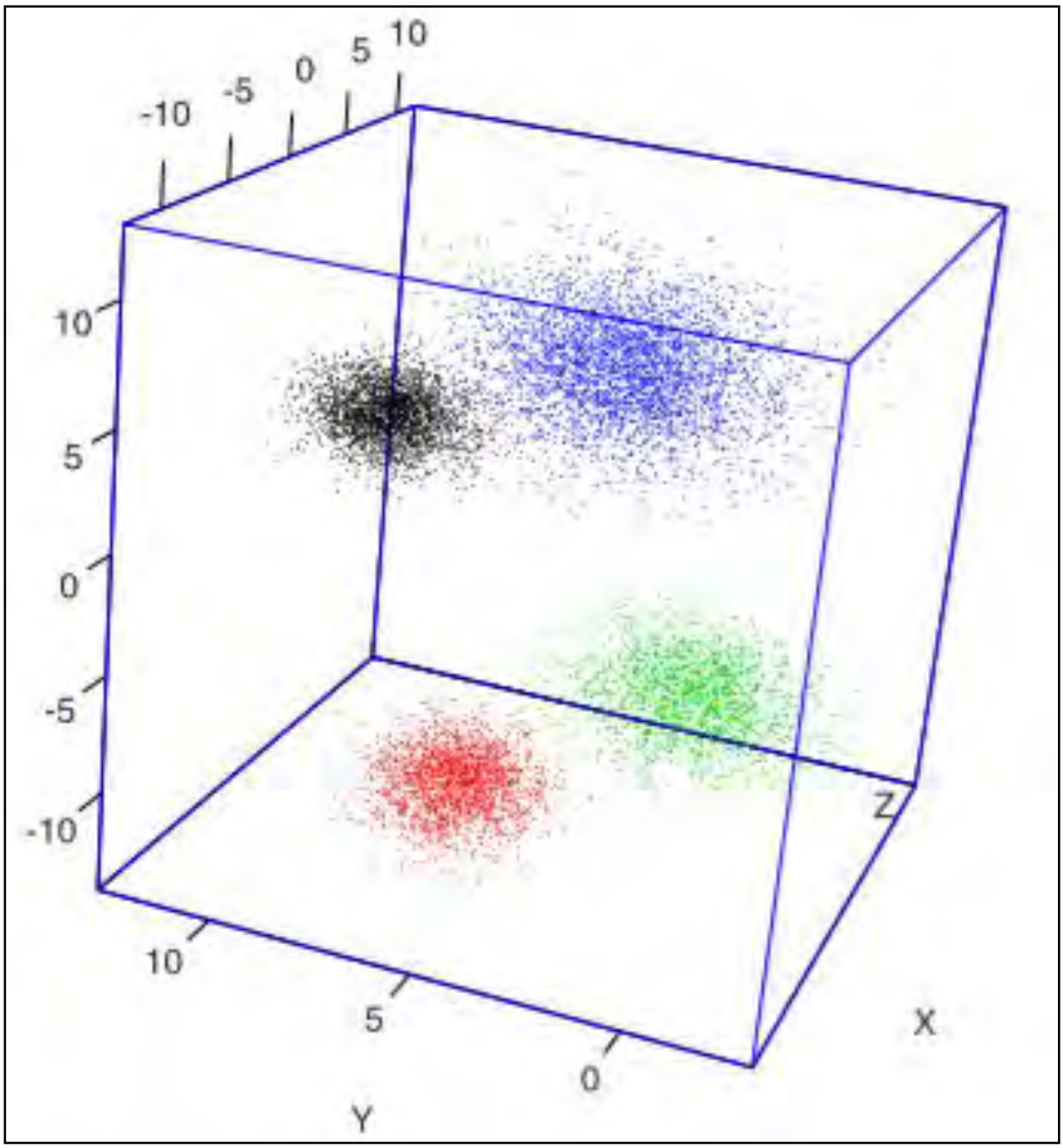


Fig 2l. The final clustering using conditional metric [23]

However, IFN constructs a directed graph and not a tree. IFN also uses the conditional mutual information metric, shown in Fig 2l., in order to choose features during the construction stage while decision trees usually use other metrics like entropy or gini.

The underlying principle of the IN-based methodology is to construct a multi-layered network in order to test the Mutual Information (MI) between the input and target attributes. Each hidden layer is related to a specific input attribute and represents the interaction between this input attribute and those associated with previous layers. The IN algorithm is using a pre-pruning strategy: a node is split if this procedure brings about a statistically significant decrease in the conditional entropy of the target attribute (equal to an increase in the mutual information). If none of the remaining input attributes provides a statistically significant increase in the mutual information, the network construction stops. The output of this algorithm is a network, which can be used as a decision tree to predict the values of the target attribute. Fig 2m. illustrates a sample structure of an information network [21].

This process of decision tree learning is a bit complicated process. Although one epoch at a time has to be done in their process, that makes the whole algorithm. Suppose a data partition is recently collected and we are supposed to build a new decision tree classifier for the collected partition that includes D_1 to D_4. The classifier C_3 has been the current classifier until D_4 becomes available. When C_3 was constructed, the set U_3 was collected and preserved as well, with the combination of S_3 and U_2. Now D_4 is used to construct T_4 from which a new set of samples S_4 are extracted. S_4 is in turn is combined with U_3 to make the new decision tree classifier T_4.

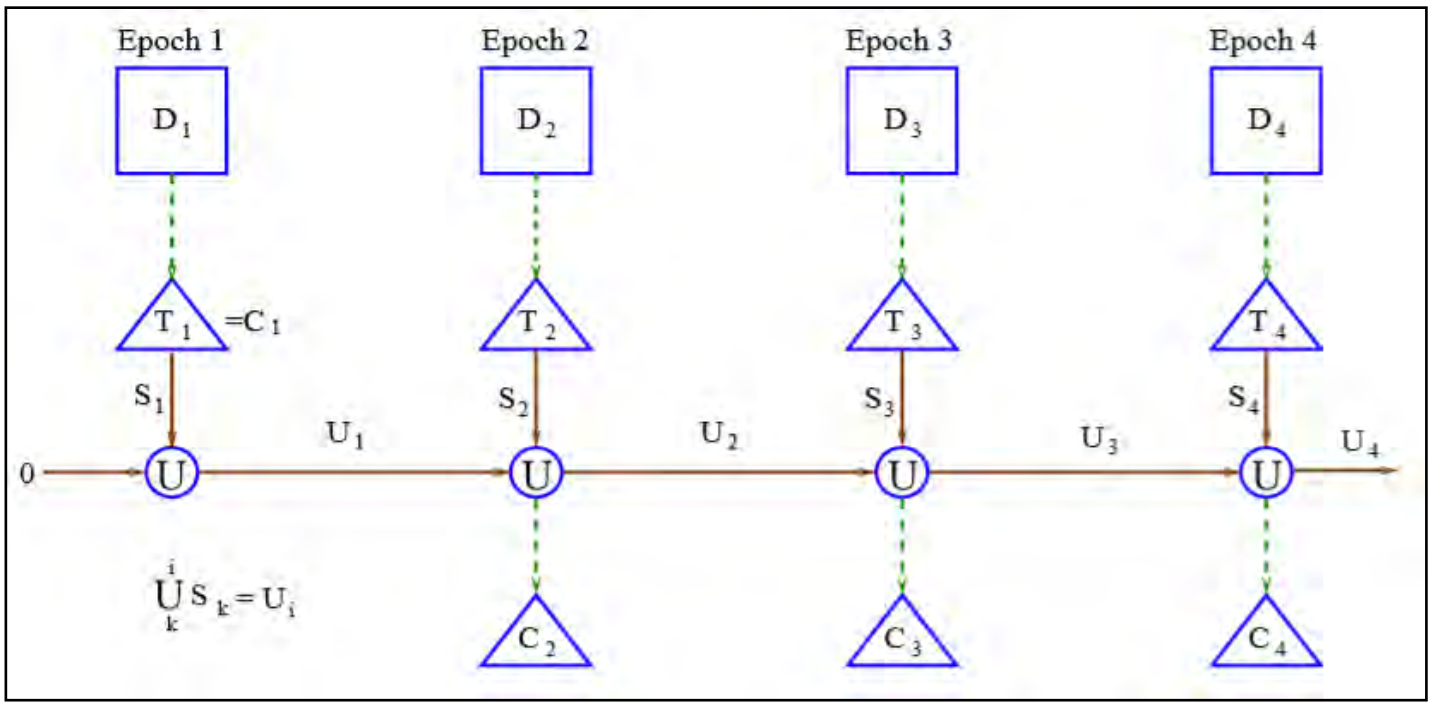


Fig 2m. Basic framework for incremental classification [21]

Comparing all of the previous works on clustering, conclusions can be made that each of the works matches with some of the following criterions. Those includes:

- Classifier-based approach, depends on some defined criterion
- STAGGER and FLORA(built in forgetting mechanism) is a sliding window approach which needs to adjust after each iterations
- Weights have to be dynamically updated before algorithm runs
- Time dependency seems a large factor in some of the works
- Principal Components are predefined in PC analysis, which are to be adjusted beforehand
- For some of the algorithms, handful of data packets are processed at a time
- Determining size of window
- Target layer node number are sometimes predefined
- High computational cost and complexity
- Operator dependency seems a large factor
- Sample entropy is used as an estimator and entropy increases as sample size increases

It is not possible to address all of the problems in hand and to find a new clustering algorithm. The proposed algorithm does not need any estimator, it is time, number of data processing and processing order independent. Any predefined number of clusters to find the actual clusters has not been used. As a result, the result of the current study contains much less false positive clustering.. The algorithm in details and step by step process to achieve the final result is explained in the following section.

CHAPTER 3

RESEARCH METHODOLOGY

The work presented in this paper proposes an algorithm based on Correspondence Analysis which uses Hierarchical Agglomerative clustering with QR decomposition method and Euclidean distance vector. In this proposed algorithm, no predefined cluster number is required. It is also parameter independent and as mentioned earlier, this algorithm is data processing order independent. At first the whole correspondence analysis method has been discussed in the beginning of this chapter. After that discussion related to distance vector and QR decomposition method have taken place. And finally, all the mathematics related to the proposed algorithm have been stated.

3.1 Correspondence Analysis

CA (correspondence analysis) is a process which allows one to analyze the pattern of relationships of several categorical dependent variables. As such, it can also be seen as a generalization of principal component analysis when the variables to be analyzed are categorical instead of quantitative. Because CA has been discovered many times, equivalent methods are known under several different names such as optimal scaling, optimal or appropriate scoring, dual scaling, homogeneity analysis, scalogram analysis, and quantification method.

8 Vars / 12 Obs	1960	1965	1970	1971	1972	1973	1974	1975
	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric
Engineering	794.00	2073.00	3432.00	3495.00	3475.00	3338.00	3144.00	2959.00
Mathematics	291.00	685.00	1222.00	1236.00	1281.00	1222.00	1196.00	1149.00
Physics	530.00	1046.00	1655.00	1740.00	1635.00	1590.00	1334.00	1293.00
Chemistry	1078.00	1444.00	2234.00	2204.00	2011.00	1849.00	1792.00	1762.00
EarthSciences	253.00	375.00	511.00	550.00	580.00	577.00	570.00	556.00
Biology	1245.00	1963.00	3360.00	3633.00	3580.00	3636.00	3473.00	3498.00
Agriculture	414.00	576.00	803.00	900.00	855.00	853.00	830.00	904.00
Psychology	772.00	954.00	1888.00	2116.00	2262.00	2444.00	2587.00	2749.00
Sociology	162.00	239.00	504.00	583.00	638.00	599.00	645.00	680.00
Economics	341.00	538.00	826.00	791.00	863.00	907.00	833.00	867.00
Anthropology	69.00	82.00	217.00	240.00	260.00	324.00	381.00	385.00
Others	314.00	502.00	1079.00	1392.00	1500.00	1609.00	1531.00	1550.00

Table 3a. Example for CA: Doctorates data sheet [6]

Technically CA is obtained by using a standard correspondence analysis on an indicator matrix. The percentages of explained variance need to be corrected, and the correspondence analysis interpretation of interpoint distances needs to be adapted.

CA is used to analyze a set of observations described by a set of nominal variables, here one of the examples is shown in Table 3a and in Fig 3a. and Fig 3b. is the after calculation plots of the example. Each nominal variable comprises several levels, and each of these levels is coded as a binary variable. For example gender, (F vs. M) is one nominal variable with two levels. The pattern for a male respondent will be 0 1 and 1 0 for a female.

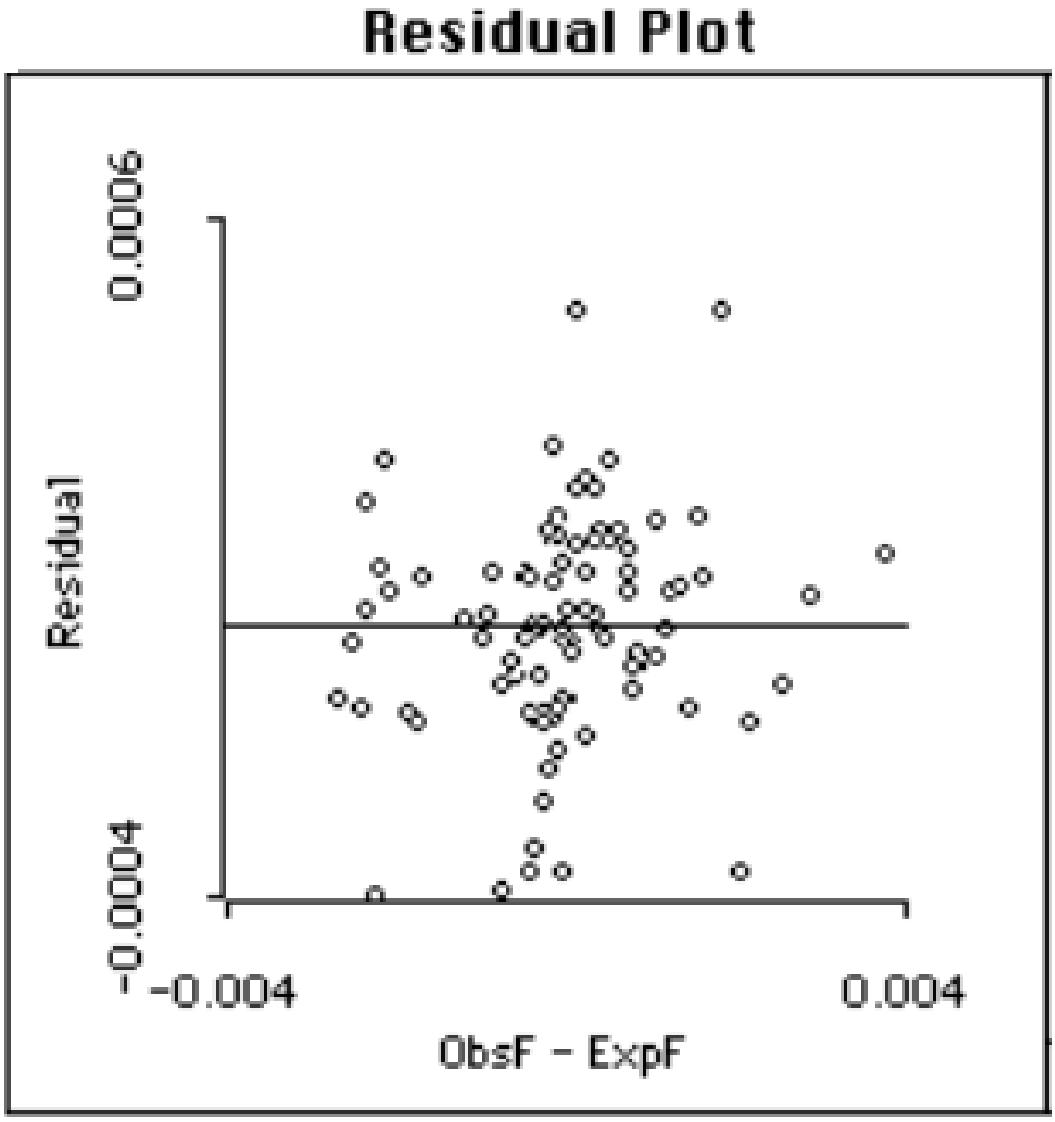


Fig 3a. Examples of Doctorate's Datasheet: Correspondence Analysis visualization [10]

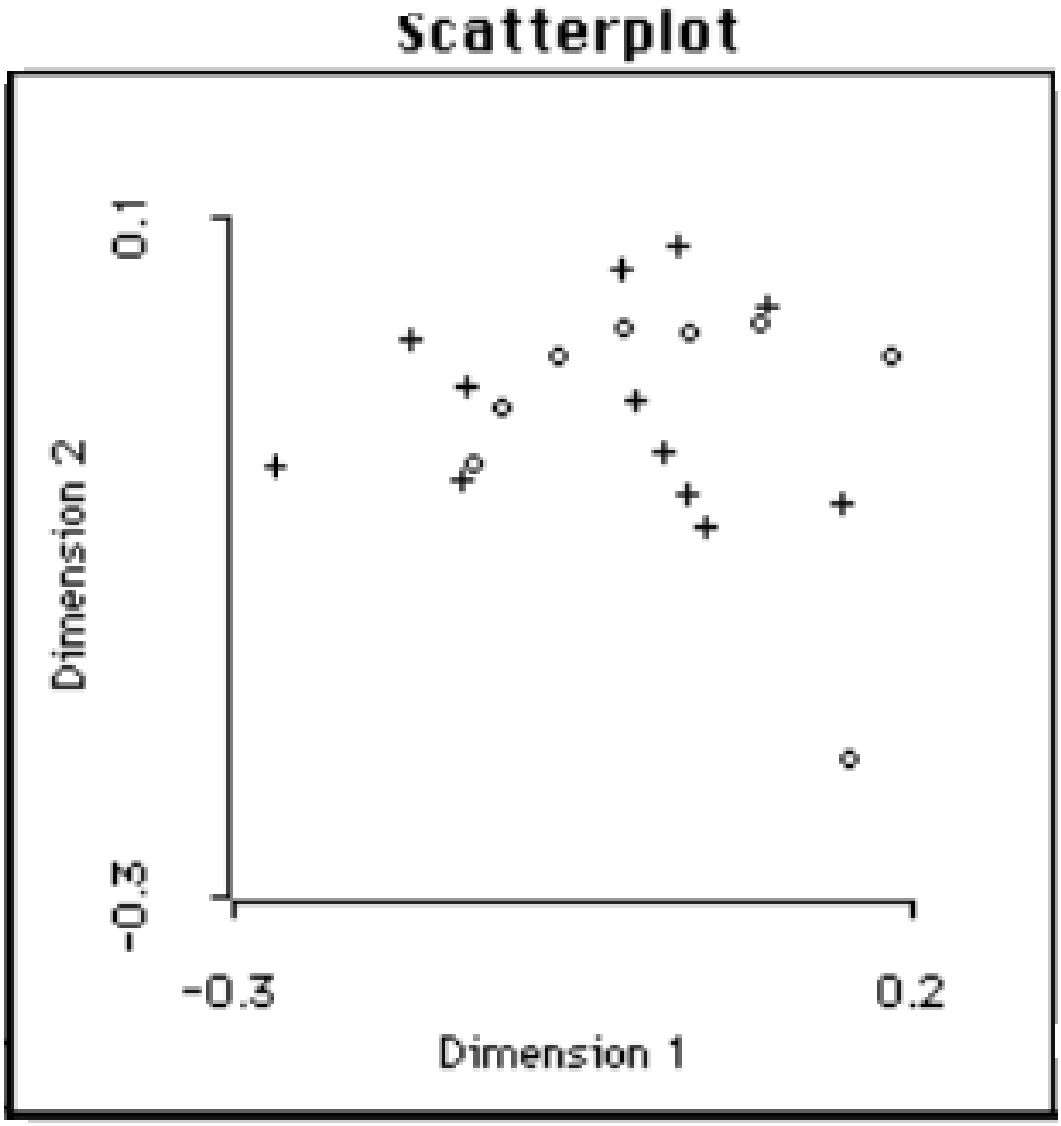


Fig 3b. Example from Doctorate's datasheet: Correspondence Analysis Visualization [10]

The complete data table is composed of binary columns with one and only one column taking the value "1" per nominal variable. Correspondence Analysis can also accommodate quantitative variables by recoding them as "bins." For example, a score with a range of −5 to +5 could be recorded as a nominal variable with three levels: less than 0, equal to 0, or more than 0. With this schema, a value of 3 will be expressed by the pattern 0 0 1. The coding schema of correspondence analysis implies that each row has the same total, which for correspondence analysis implies that each row has the same mass.

The process for CA is, there are K nominal variables, each nominal variable has J_k levels and the sum of the J_k is equal to J. There are I observations. The $I \times J$ indicator matrix is denoted $\mathbf{X}$. Performing correspondence analysis on the indicator matrix will provide two sets of factor scores: one for the rows and one for the columns. These factor scores are, in general scaled such that their variance is equal to their corresponding eigenvalue (some versions of CA compute row factor scores normalized to unity).

A codes data by creating several binary columns for each variable with the constraint that one and only one of the columns gets the value 1. This coding schema creates artificial additional dimensions because one categorical variable is coded with several by the first columns. As a consequence, the inertia (i.e., variance) of the solution space is artificially inflated and therefore the percentage of inertia explained dimension is severely underestimated. In fact, it can be shown that all the factors with an eigenvalue less or equal to 1 K simply code these additional dimensions (K = 10 in the above example). Two corrections formulas are often used, the first one is due to Benzécri (1979), the second one to Greenacre[32]. These formulas take into account that the eigenvalues

smaller than 1 K are coding for the extra dimensions and that CA is equivalent to the analysis of the Burt matrix whose eigenvalues are equal to the squared eigenvalues of the analysis of **X**. Specifically, if we denote by λ` the eigenvalues obtained from the analysis of the indicator matrix, then the corrected eigenvalues, denoted $_{c}\boldsymbol{\lambda}$ are obtained as

$$ {}_{c}\lambda_{\ell} = \begin{cases} \left[\left(\frac{K}{K-1}\right)\left(\lambda_{\ell} - \frac{1}{K}\right)\right]^{2} & \text{if } \lambda_{\ell} > \frac{1}{K} \\ 0 & \text{if } \lambda_{\ell} \leq \frac{1}{K} \end{cases} \qquad (1) $$

Using this formula gives a better estimate of the inertia, extracted by each eigenvalue. Traditionally, the percentages of inertia are computed by dividing each eigenvalue by the sum of the eigenvalues, and this approach could be used here also. However, it will give an optimistic estimation of the percentage of inertia. A better estimation of the inertia has been proposed by Greenacre who suggested instead to evaluate the percentage of inertia relative to the average inertia of the off-diagonal blocks of the Burt matrix. This average inertia, denoted $\overline{\mathscr{I}}$ can be computed as

$$ \overline{\mathscr{I}} = \frac{K}{K-1} \times \left(\sum_{\ell} \lambda_{\ell}^{2} - \frac{J-K}{K^{2}}\right) \qquad (2) $$

According to this approach, the percentage of inertia would be obtained by the ratio

$$ \tau_{c} = \frac{{}_{c}\lambda}{\overline{\mathscr{I}}} \text{ instead of } \frac{{}_{c}\lambda}{\sum {}_{c}\lambda_{\ell}} \qquad (3) $$

The interpretation in CA is often based upon proximities between points in a low-dimensional map (i.e., two or three dimensions). As well as for CA, proximities are meaningful only between points from the same set (i.e., rows with rows, columns with

columns). Specifically, when two row points are close to each other they tend to select the same levels of the nominal variables. For the proximity between variables there are need to distinguish two cases. First, the proximity between levels of different nominal variables means that these levels tend to appear together in the observations. Second, because the levels of the same nominal variable cannot occur together, we need a different type of interpretation for this case. Here the proximity between levels means that the groups of observations associated with these two levels are themselves similar.

Let X be an (n x m) matrix of observed frequencies of rank q such that the row and column sums are nonzero. Let l be a row vector of ones and I be an identity matrix, each of appropriate order. Denote a matrix-valued function that creates a diagonal matrix from a vector by diag (). Define

i. s = l'Xl as the sum of all elements in X;

ii. P = $\frac{l}{s}X$ as the matrix of relative frequencies (the correspondence matrix);

iii. r = Pl as the vector of row marginal proportions (row masses);

iv. c = $P'l$ as the vector of column marginal proportions (column masses);

v. D_r = diag(r) a diagonal matrix of row masses; and

vi. D_c = diag(c) a diagonal matrix of column masses.

The generalized singular value decomposition (abbreviated SVD) of P provides the required solution to the point coordinates of correspondence analysis: ,

$$P = AD_uB' \tag{4}$$

where

i. A is an (n x q) matrix whose columns are the left generalized singular vectors;

ii. D_u is a (q x q) diagonal matrix of generalized singular values;

iii. B is an (m x q) matrix whose columns are the right generalized singular vectors; and where

iv. $A'D^{-1}_{r}A = B'D^{-1}_{c}B = I.$

There is a trivial part of the generalized SVD of consisting of a singular value of 1 and associated left and right singular vectors, which is discarded before any results are displayed.

The remaining left and right singular vectors define the orthogonal principal axes of the column and row points, respectively. In practice, the generalized SVD is computed indirectly by performing an ordinary SVD, where the ordinary SVD of any matrix Q is given by ,

$$Q = UD_{\alpha}V' \tag{5}$$

under the constraint $U'U = V'V = I$. Thus, to compute the generalized SVD of P, we perform the following steps:

i. Let $Q = D_r^{-1/2} PD_c^{-1/2}$.

ii. Obtain the ordinary SVD of Q, giving $Q = UD_{\alpha}V'$.

iii. Let $A = D_r^{1/2}U$, $B = D_c^{1/2}V$, and $D_u = D_{\alpha}$.

iv. Then $P = AD_uB'$ is the required generalized SVD.

v. The row coordinates *F* and column coordinates *G* are then computed according to the appropriate selection of the formulas. [32]

3.2 Distance vector

The squared length of a vector x = [x_1 x_2] is the sum of the squares of its coordinates and the squared distance between two vectors x = [x_1 x_2] and y = [y_1 y_2] is the sum of squared differences in their coordinates[30]. To denote the distance between vectors x and y, $d_{x,y}$ can use the notation so that this last result can be written as:

$$d^2 x, y = (x_1 - y_1)^2 + (x_2 - y_2)^2 \quad (6)$$

that is, the distance itself is the square root

$$d_{x,y} = \sqrt{(x1 - y1)^2 + (x2 - y2)^2} \quad (7)$$

which is called the squared length of x, is the distance between the vector x = [x_1 x_2] and the zero vector 0 = [0 0] with coordinates all zero:

$$d_{x,0} = \sqrt{x_1^2 + x_2^2} \quad (8)$$

which could just denote by d_x . The zero vector is called the origin of the space.

If immediately can be moved to a three-dimensional point x = [x_1 x_2 x_3], the squared length of x is the sum of its three squared coordinates and so

$$d_x = \sqrt{x_1{}^2 + x_2{}^2 + x_3^2} \quad (9)$$

A vector can be described as a directed line segment from the origin of the Euclidean space (vector tail), to a point in that space (vector tip). If it is considered that its length is

actually the distance from its tail to its tip, it becomes clear that the Euclidean norm of a vector is just a special case of Euclidean distance: the Euclidean distance between its tail and its tip.

The standardized Euclidean distance between two J-dimensional vectors can be written as:

$$d_{\mathbf{x},\mathbf{y}} = \sqrt{\sum_{j=1}^{J} \left(\frac{x_j}{s_j} - \frac{y_j}{s_j}\right)^2} \qquad (10)$$

where s_j is the sample standard deviation of the *j-th* variable. Notice that it is not needed to subtract the *j-th* mean from x_j and y_j because they will just cancel out in the differencing. Now (8) can be rewritten in the following equivalent way:

$$d_{\mathbf{x},\mathbf{y}} = \sqrt{\sum_{j=1}^{J} \frac{1}{s_j^2}(x_j - y_j)^2}$$

$$= \sqrt{\sum_{j=1}^{J} w_j (x_j - y_j)^2} \qquad (11)$$

where $w_j = 1/s^2_j$ is the inverse of the *j-th* variance. If it is assumed w_j as a weight attached to the *j-th* variable: in other words, the usual squared differences between the variables on their original scales are compared, as did in the (unstandardized) Euclidean distance, but then multiply these squared differences by their corresponding weights. Notice in this case how the weight of a variable with high variance is low, while the weight of a variable with low variance is high, which is another way of thinking about the compensatory effect produced by standardization.

3.3 QR decomposition

The QR decomposition (also called the QR factorization) of a matrix is a decomposition of the matrix into an orthogonal matrix and a triangular matrix. A QR decomposition of a real square matrix A is a decomposition of A as

$$A = QR \tag{12}$$

where Q is an orthogonal matrix (i.e. $Q^TQ = I$) and R is an upper triangular matrix. If A is nonsingular, then this factorization is unique.

In this decomposition method, it helps to summarize response patterns in both rows and columns of data matrix. It also defines a space in which a graphical representation of data is possible. For clustering the data, residuals are needed to find. QR decomposition contributes to find residuals. In case of residual matrix, the number of axes determined by eigenvalues of square matrix is obtained by multiplying input data matrix with its transpose [16].

If the Gram Schmidt procedure is considered, with the vectors to be measured in the process as columns of the matrix A. That is,

$$A = [a_1 \mid a_2 \mid \dots\dots\dots a_n]$$

Then

$$u_1 = a_1, \ e_1 = \frac{u_1}{\|u_1\|}$$

$$u_2 = a_2 - (a_2 \,.\, e_1)e_1, \quad e_1 = \frac{u_2}{\|u_2\|}$$

$u_{k+1} = a_{k+1} - (a_{k+1}. e_1)e_1 - \ldots - (a_{k+1} . e_k)e_k, \quad e_{k+1} = \frac{u_{k+1}}{\|u_{k+1}\|}$ (13)

Note that once $e_1, \ldots, e_n$ are found, it is not hard to write the QR factorization.

3.4 Mathematical explanations

Correspondence Analysis uses patterns for the representation of data matrix using both row and column profiles. For defining these matrices with other parameters, characterizations of these parameters will be predefine. Input data matrix have used as ***N*** *(I, J)* with elements n_{ij} of *n* independent objects. As these have different correspondences, while using the interpretation for both the rows and columns, either rows or columns must be reduced to the same base. rows are considered as I, (where i= 1, 2, …I) and columns =J, (where *j* = 1, 2 ….*J*). Marginal frequencies are defined as

$$n_{i+} = \sum_i n_{ij} \text{ and } n_{+j} = \sum_j n_{ij}$$

that combines total frequency

$$n = \sum_i . \sum_j n_{ij} \qquad (14)$$

Row profile is stated as n_{ij} / n_{i+} and column profile is n_{ij} / n_{j+}. i' and j' contains rows and column repectivly without *i* and *j*.

If the distance between rows is

$$d^2(i, i') = \sum_{j=1}^{J} 1/n_{+j} \left[\left(n_{ij} / n_{i+}\right) + \left(n_{i'j} / n_{i'+}\right)\right] \qquad (15)$$

and row profile is stated in Table 3b.

ROWS	***COLUMNS***		***TOTAL***
	1	**2 J**	
1. **2.** **3.** **.** **I**	n_{11}/n_{1+} n_{21}/n_{2+} n_{31}/n_{3+} . n_{i1}/n_{i+}	n_{12}/n_{1+} ..... n_{1j}/n_{1+} n_{22}/n_{2+} ..... n_{2j}/n_{2+} n_{32}/n_{3+} ..... n_{3j}/n_{3+} . n_{i2}/n_{i+} ..... n_{ij}/n_{i+}	**1** **1** **1** **1**
Column mass	n_{+i}/n_{++}	n_{+2}/n_{++} n_{+j}/n_{++}	**1**

Table 3b. Matrix of ROW profile

and distance between columns is

$$d^2(j,j') = \sum_{i=1}^{I} 1/n_{i+} \left[\left(n_{ij}/n_{j+}\right) + \left(n_{ij'}/n_{+j'}\right)\right] \qquad (16)$$

and the column profile is stated in Table 3c.

ROWS	***COLUMNS***		***ROW MASS***
	1	**2 J**	
1. **2.** **3.** **.** **I**	n_{11}/n_{+1} n_{21}/n_{+1} n_{31}/n_{+1} . n_{i1}/n_{+1}	n_{12}/n_{+2} ..... n_{1j}/n_{+j} n_{22}/n_{+2} ..... n_{2j}/n_{+j} n_{32}/n_{+2} ..... n_{3j}/n_{+j} . n_{i2}/n_{+2} ..... n_{ij}/n_{+j}	n_{+1}/n_{++} n_{+2}/n_{++} n_{+i}/n_{++}
Column mass		**....1 1**	**1**

Table 3c. Matrix of COLUMN profile

then the inertia or i^{th} row profile will be

$$x_i \sum_i (s_{ij} - \bar{s}j)^2 / \bar{s}j \qquad (17)$$

where

$$s_{ij} = n_w / n_i \quad \text{and} \quad \bar{s} = n_{+j} / n.$$

j^{th} column profile can be calculated simultaneously.

The correspondence matrix **C** is defined as the original table **N** divided by the grand total *n,* **C** = (1/*n*) **N**. Thus, each cell of the correspondence matrix is given by the cell frequency divided by the grand total. The correspondence matrix shows how one unit of *mass* is distributed across the cells. The row and column totals of the correspondence matrix are the row mass and column mass, respectively. So now the matrix of row and column profiles are defined as $D_i^{-1}C$ and $D_j^{-1}C$ respectively. In a low *p*-dimensional subspace, where *p* is less than *I* or *J*, these two *p*-dimensional subspaces (one for the row profiles and one for the column profiles) have a geometric correspondence that enables us to represent both the rows and columns in the same display. For geographically representing the distance between both the profiles, the orientation of the points would have to be the centers of gravity, or the centroids. The centroid of the set of row points in its space is ***c***, the vector of column masses. The centroid of the set of column point in its space is ***s***, the vector of row masses. This is the average column profile. To perform the analysis with respect to the center of gravity, ***C*** is centered "symmetrically" by rows and columns, *i.e.*, $\boldsymbol{Csc}^T$ so that it corresponds to the average profiles of both sets of points. The solution to finding an illustration of both sets of points is the QR decomposition of the matrix of

standardized residuals, i.e., $X' Y$ matrix with elements. With the help of classical Gram-Schmidt procedure, the iterative procedure operates starting with

$$q_0 = h_0/\|h_0\|$$

where $\|h_0\|$ defines Euclidean norm of h_0. Then the standard residuals can be found as

$$R = h_{ij} - \sum_{j=0}^{i-1}(q *_j . h_{ij}) . q_j \quad (18)$$

which is simply the residual vector that results from projection and by orthogonality property of least-square solutions, this residual is orthogonal. The main function of QR decomposition is, it summarizes response patterns of both row and column data matrices, defines a space where graphical representation is possible and contributes to finding residuals. The number of axes is determined by eigenvalues of the square matrix which is obtained by multiplying the input data matrix with its transpose.

Another important step of the algorithm is to find the distances among cluster centroids. This distance measure helps to determine the boundary of the clusters before splitting further. Here new distance measure have been implemented, the standard weighted Euclidean distance measure, which is defined as

$$T_{c_i,c_j} = \sqrt{\sum_{l=1}^{L}(c_{il}/m_l - c_{jl}/m_l)^2} \quad (19)$$

Here m_l is the sample standard deviation from of l-th variable. Primary grouping was started with the very first row and columns. It is important to well-define the boundary of a cluster and also make sure the elements inside the cluster are closely related to each other. At the same time, it should be ensured that the different cluster elements shows different distinctiveness among themselves. To ensure these criterions, first the proposed algorithm was applied for primary sub clustering with fundamental profiles, then to the

large scale data matrix of the real life data from the Abilene network. The algorithm automatically splits into clusters, but for better results, the algorithm needs a threshold to stop clustering, for which quality measure for clusters have been applied. The measure to cease splitting is:

$$Q\ (C) = \sum_{C_f \in C}(|C_f|/|T\ |)^2\ \sum_{g=1}^{m}\sum_{v \in h_g}[(p(u_f = v))^2 - (p^{'}(u_f = v))^2] \quad (20)$$

In the above equation, probability of cluster is $p(u_f = v)$, probability over all the objects for clustering is $p^{'}(u_f = v)$ and weight, which penalizes split with extremely small clusters is $|C_f|/|T\ |$.

After comparing with existing works, the results show more prominent outcome with least false positive alerts.

CHAPTER 4

ALGORITHM STEP BY STEP

As have been explained earlier each of the steps in details, this is the summarized form of the whole algorithm. From the data from Abilene network, first relative frequencies were computed such that the total entries is equals 0. Then comes the splitting part, with row profiles and column profiles with the help of mass calculation. Inertia calculation is then done using multiplications between integral mass and squared distance to the centroids, which is using the Euclidean distance vector. Finally, after calculating the profiles, both row and column, distance statistics and deviation from norm of the distances are measured. Lastly, after finding variance, the splitting stops.

Below in Fig 4a., is given the flow chart for the proposed algorithm.

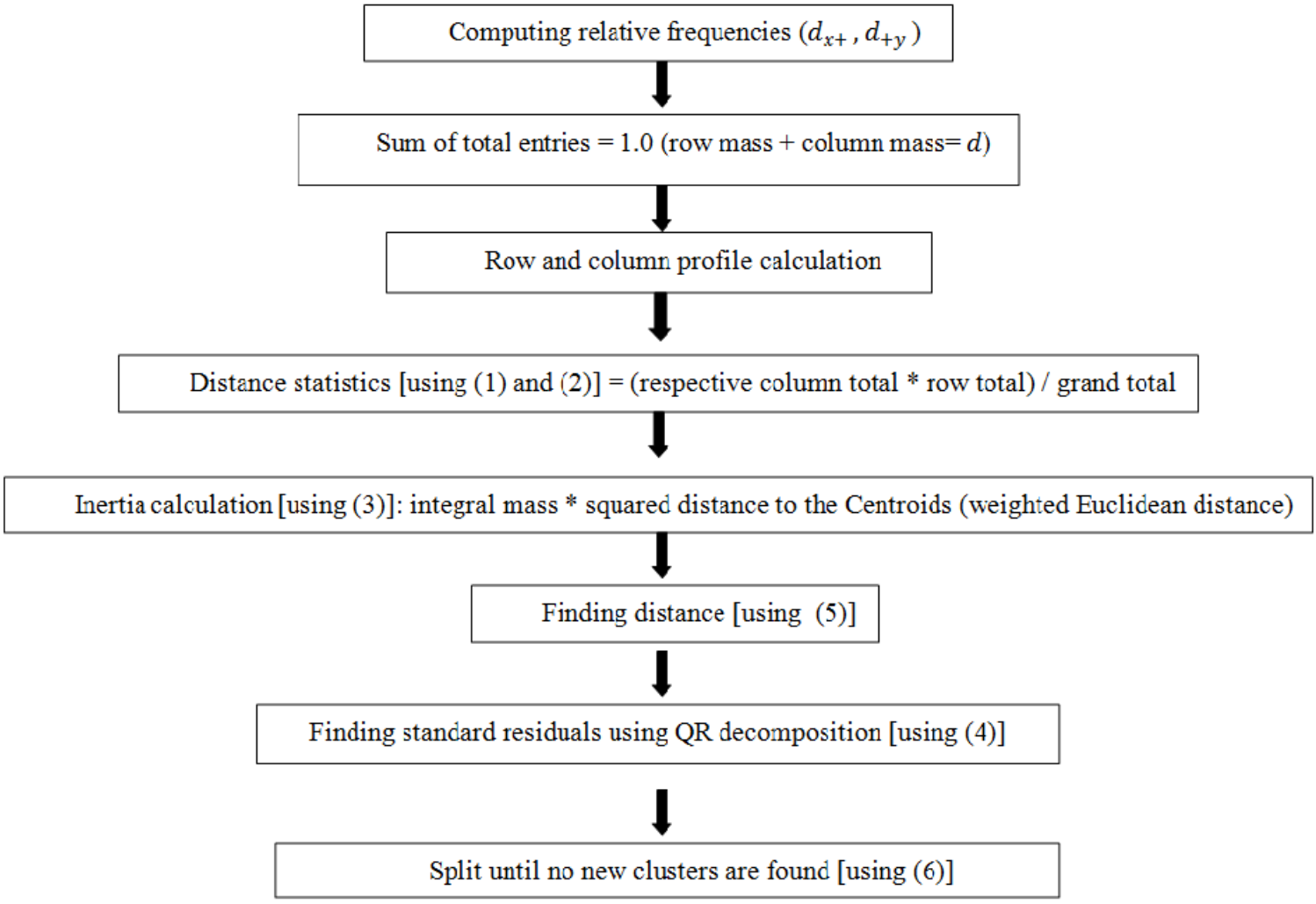


Fig 4a. Flow chart of Correspondence analysis

CHAPTER 5

Experimental Results

5.1 Results using proposed algorithm

To evaluate the proposed algorithm, performance on network-wide traffic datasets analyzed by Lakhina et al. in [7] was examined. This data was collected from 11 core routers in the Abilene backbone network for a week (December 15 to December 21, 2003). It comprises two multivariate timeseries, one being the number of packets and the other the number of individual IP flows in each of the Abilene backbone flows (the traffic entering at one core router and exiting at another), binned at 5 minute intervals. Both datasets, X(1) and X(2), are of dimension F ×T, where T = 2016 is the number of timesteps and F = 121 is the number of backbone flows. For the performance evaluation of the algorithm proposed, real-life marked data from the Abilene network have been used. The results were compared with CHAID (Chi-squared Automatic Interaction Detector), *k*-means and correlation analysis. These are the most recent works on clustering of anomalous data [22], [16], [17].

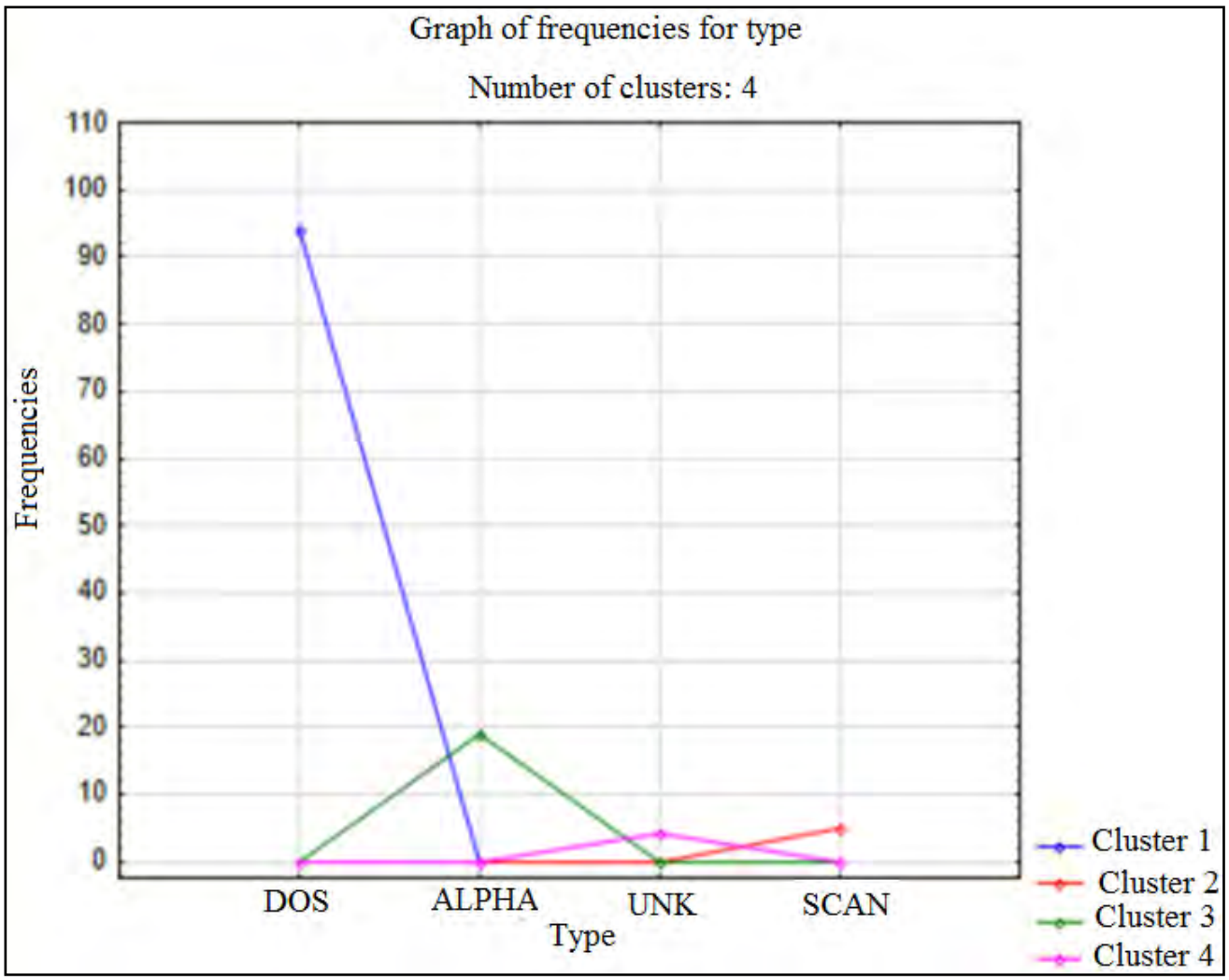


Fig5a. Relative frequencies of Correspondence Analysis

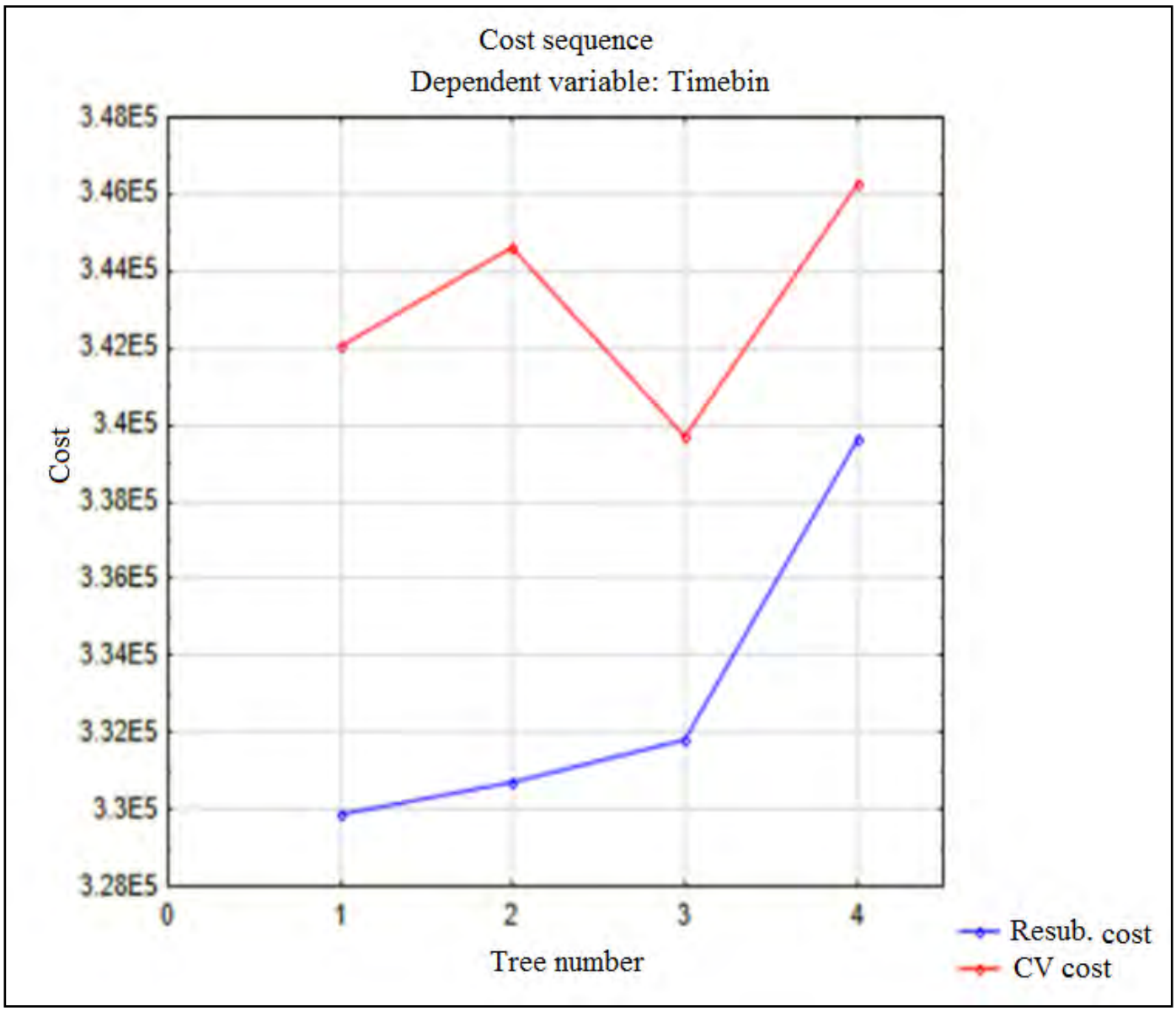


Fig 5b. Cost sequence found from proposed CA

Fig 5a. shows the types of anomaly with respect to their corresponding number of occurrence, simply termed as frequency. The algorithm has detected four categories of anomalous clusters and provided visual representation of them. Which was the main target of the proposed work, to indentify number of clusters among a bunch of anomalies. In case of Fig 5b., the Coefficient of Variation (CV) cost and resubstitution cost has been shown. CV represents the ratio of the standard deviation to the mean, which is a useful statistic for comparing the degree of variation from one data series to another, even if the means are considerably different from each other. Resubstitution cost finds out the cost after minimizing the errors from the CV cost. Both of the results are below 3.5E5 which is the standard threshold value. Here again the results found using the proposed algorithm are satisfactory.

Column Name	Column Number	Coordin. Dim.1	Coordin. Dim.2	Mass	Quality	Relative Inertia	Inertia Dim.1	Cosine² Dim.1	Inertia Dim.2	Cosine² Dim.2
Column Coordinates and Contributions to Inertia (Spreadsheet1) Input Table (Rows x Columns): 122 x 5 Standardization: Row and column profiles										
Type	1	2.792290	-6.54110	0.000000	0.999941	0.000335	0.000052	0.154132	0.994761	0.845809
Bytes	2	-0.003657	0.00000	0.998830	1.000000	0.001170	0.001170	1.000000	0.000000	0.000000
Frac	3	1.374839	-0.78480	0.000000	0.469581	0.000003	0.000001	0.354174	0.001187	0.115407
Pkts	4	3.120799	0.00039	0.001170	1.000000	0.998480	0.998769	1.000000	0.000055	0.000000
Frac	5	3.176316	-1.23409	0.000000	0.742596	0.000012	0.000008	0.645200	0.003997	0.097396

Table 5a. Column coordinates

Row Name	Row Number	Coordin. Dim.1	Coordin. Dim.2	Mass	Quality	Relative Inertia	Inertia Dim.1	Cosine² Dim.1	Inertia Dim.2	Cosine² Dim.2
Row Coordinates and Contributions to Inertia (Spreadsheet1) Input Table (Rows x Columns): 122 x 5 Standardization: Row and column profiles										
10	1	-0.014064	0.000028	0.013193	0.999998	0.000229	0.000229	0.999994	0.000003	0.00000
24	2	-0.014356	0.000026	0.013193	0.999998	0.000238	0.000238	0.999995	0.000003	0.00000
28	3	0.027026	-0.042178	0.000045	0.997892	0.000010	0.000003	0.290448	0.024551	0.70744
46	4	-0.014356	0.000026	0.013193	0.999999	0.000238	0.000238	0.999995	0.000003	0.00000
58	5	-0.014064	0.000029	0.013193	0.999999	0.000229	0.000229	0.999995	0.000003	0.00000
70	6	-0.014648	0.000024	0.013192	0.999999	0.000248	0.000248	0.999997	0.000002	0.00000
202	7	-0.014065	0.000138	0.013193	0.999999	0.000229	0.000229	0.999903	0.000077	0.00009
209	8	0.042119	-0.031532	0.000061	0.978437	0.000015	0.000009	0.627006	0.018574	0.35143

Table 5b. Row profiles

After calculating the row and column coordinates, results presented in Tables 5a and 5b are found.

The graph for the coordinates has been shown in Fig 5c. below. The violation arrives whenever the value deviates from norm. This also helps to identify the clusters in row profile. The Table 5c. shows the tree sequence in tabular form and the parallel coordination plot for the data clusters. In the plot, the arrivals of frequencies for anomalies with respect to types are shown. Each of the timebin has been considered here to plot the graph for all of the data from backbone network.

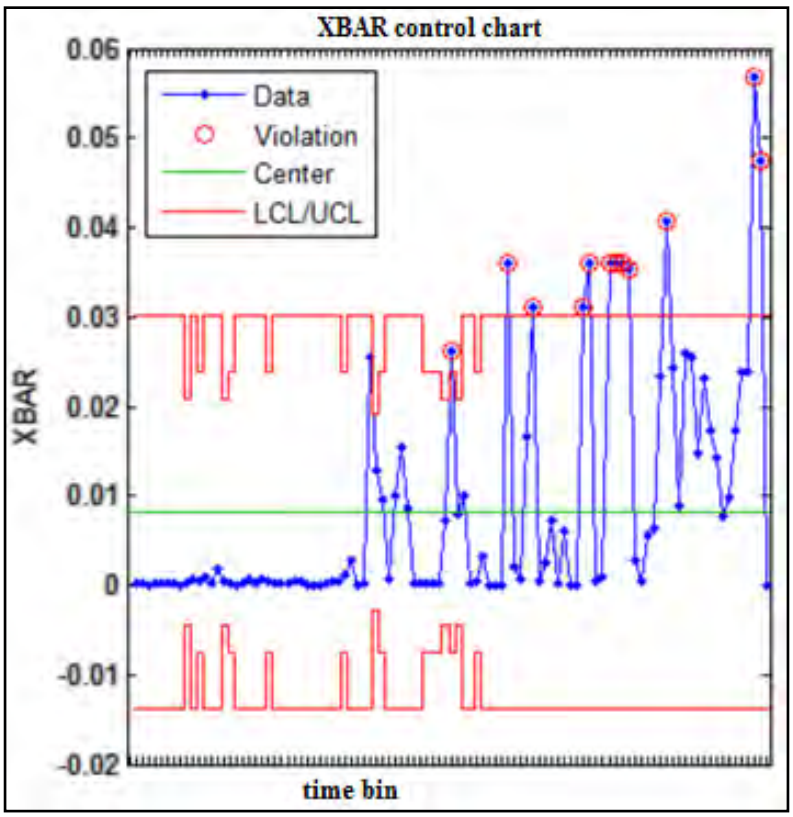


Fig 5c. XBAR chart of row profiles

	Standardized Deviates (Spreadsheet1) Input Table (Rows x Columns): 122 x 5					
	Type	Bytes	Frac	Pkts	Frac	Total
10	-1.2983	4.81	0.02813	-140.60	-0.4379	-137.49
24	-1.2982	4.91	0.04076	-143.52	-0.4271	-140.29
28	24.7505	-0.54	0.94879	15.59	0.2611	41.01
46	-1.2982	4.91	-0.00972	-143.52	-0.4595	-140.37
58	-1.2983	4.81	-0.03497	-140.60	-0.4595	-137.58
70	-1.2982	5.01	-0.04758	-146.44	-0.4703	-143.24
202	-2.3884	4.81	-0.06022	-140.60	-0.4812	-138.72
209	21.2243	-0.98	5.15194	28.42	4.3980	58.21
210	20.7684	-1.07	5.76891	31.04	4.6144	61.12
210	8.7326	-7.61	0.20082	222.23	0.8085	224.36
210	20.1705	0.89	0.09933	-26.16	-0.1205	-5.12
211	9.3607	-8.34	0.22147	243.69	0.9529	245.88
212	20.7684	-1.07	4.85893	31.04	4.7703	60.37
212	47.0809	-9.32	0.20594	271.75	1.0556	310.78
213	9.5934	-9.99	0.23188	291.75	1.1296	292.72
214	-1.2983	4.81	-0.03497	-140.60	-0.4704	-137.59
215	9.5787	-14.09	0.68183	411.57	1.4989	409.24
226	10.1212	-10.67	0.20952	311.54	1.0442	312.25
226	-1.0137	3.04	0.29935	-88.85	-0.1265	-86.65

Table 5c. Standardized deviation from expected value (Parametric model evaluation)

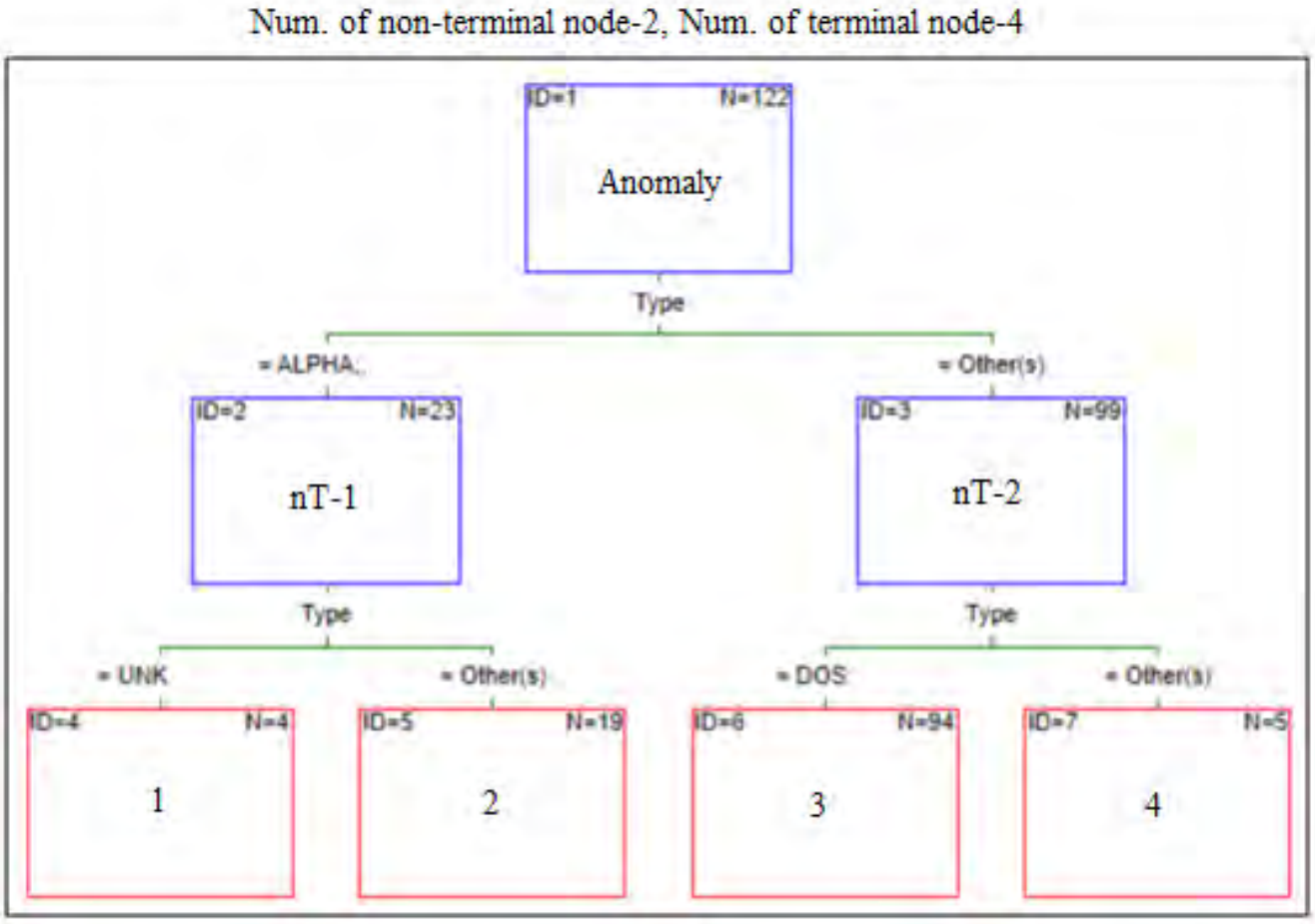


Fig 5d. Tree graph for timebin for anomaly clusters

In the Fig 5d., tree graph has been shown for data clustering. After using the termination equation, it was possible to identify the number of anomaly types in the backbone data network. As mentioned earlier, implementation of the termination equation is comparatively new addition to the existing clustering algorithm. In the above Table 5d., tree sequence for time bin has been shown. It is clear from the table that as the node increases, the complexity also increases. In Fig 5e. and in Fig 5f., anomalous data type with time bin has been shown. From there, four clusters can be detected also.

	Tree Sequence (Week3-BPF-Spreadsheet)		
	Terminal node	Resubstitution cost	Node Complexity
Tree 1	4	329878.2	0.000
Tree 2	3	330702.6	824.464
Tree 3	2	331804.8	1102.147
Tree 4	1	339618.7	7813.945

Table 5d. Tree sequence for timebin with anomaly

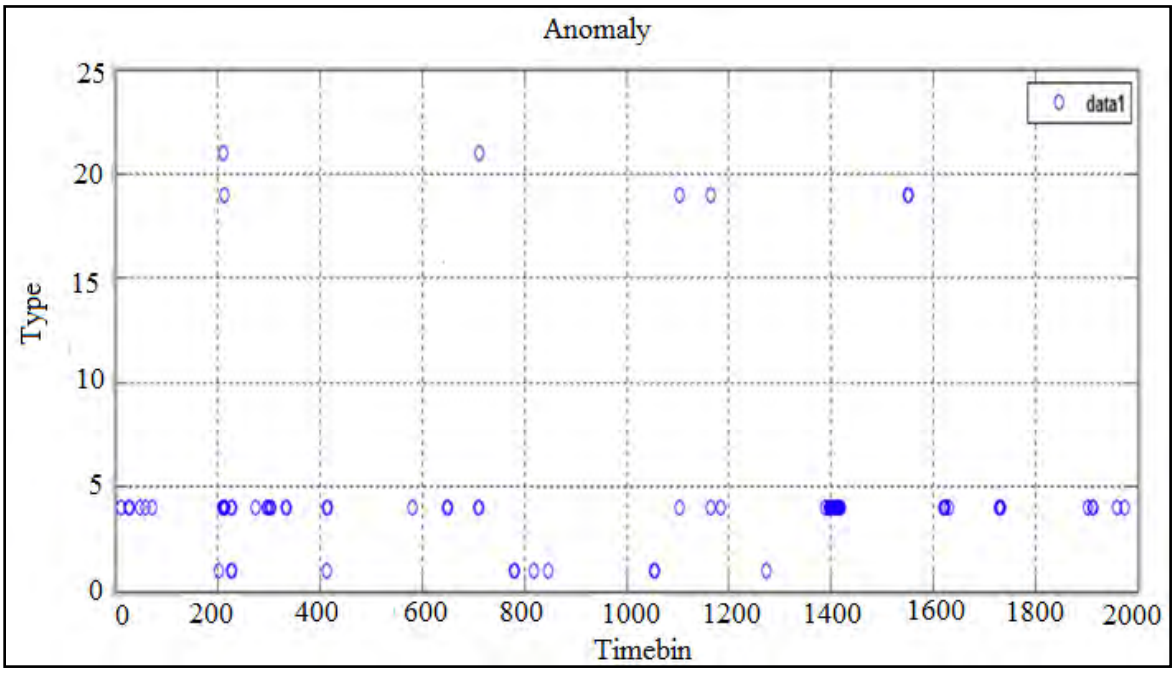


Fig 5e. Anomalous data type

Predictable variable calculation is one of the major calculations for data clustering. It shows how the variables are related with each other. The row mass variable and the column mass variables are found from the mass data set and can be used as estimator for further splitting. Fig 5g. shows the predictable variable importance ranking of CA algorithm. It is a must to calculate the importance of predictable variable ranking for the clarity of the steps of the algorithm towards final result.

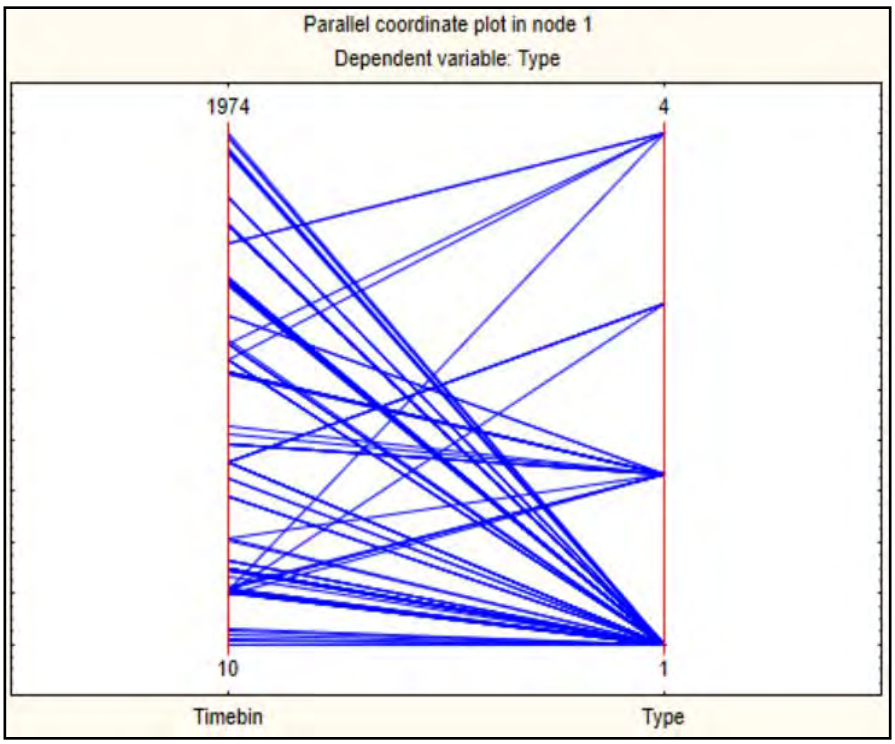


Fig 5f. Parallel coordinate plot

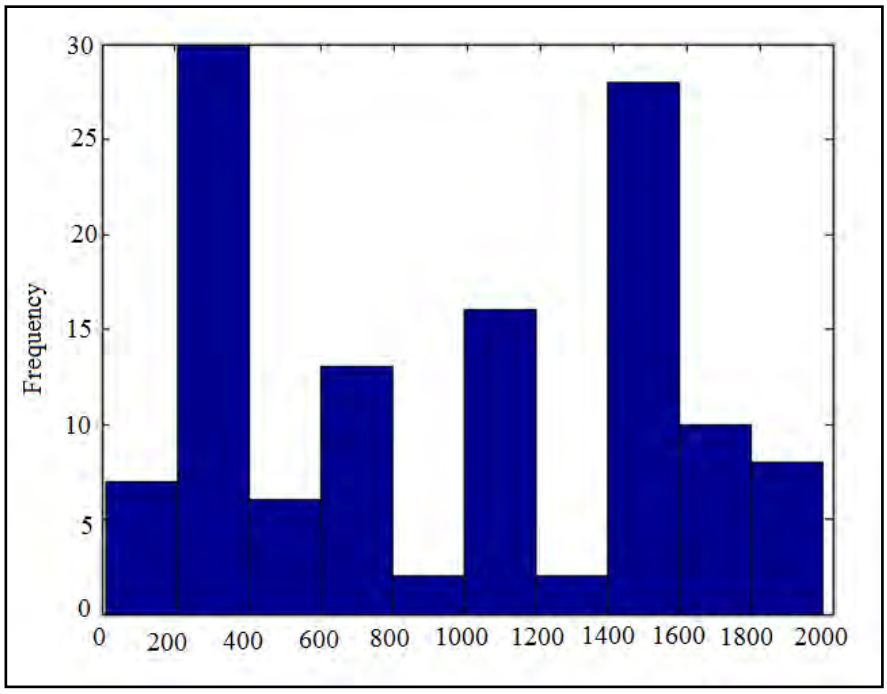


Fig 5g. Predictable variables importance ranking

5.2 Comparison with other clustering algorithms

Algorithms, whether it satisfies the core of the research interest or not, finding out the answer one need to verify through comparison. Here are some of the many examples to show that the proposed algorithm works up to the mark. Results from the latest CHAID (Chi-square Automatic Interaction Detector) algorithm, the Correlation Algorithm and finally the proposed Correspondence Analysis algorithm are being compared in this section of the paper.

In the figures below (Fig 5h.), after measuring the outputs of clustered data it is clear that the correlation analysis algorithm cannot detect clusters below a certain point, i.e.; some certain threshold level. Whereas, using the proposed algorithm, clusters are found, includes anomalies arriving in different frequencies.

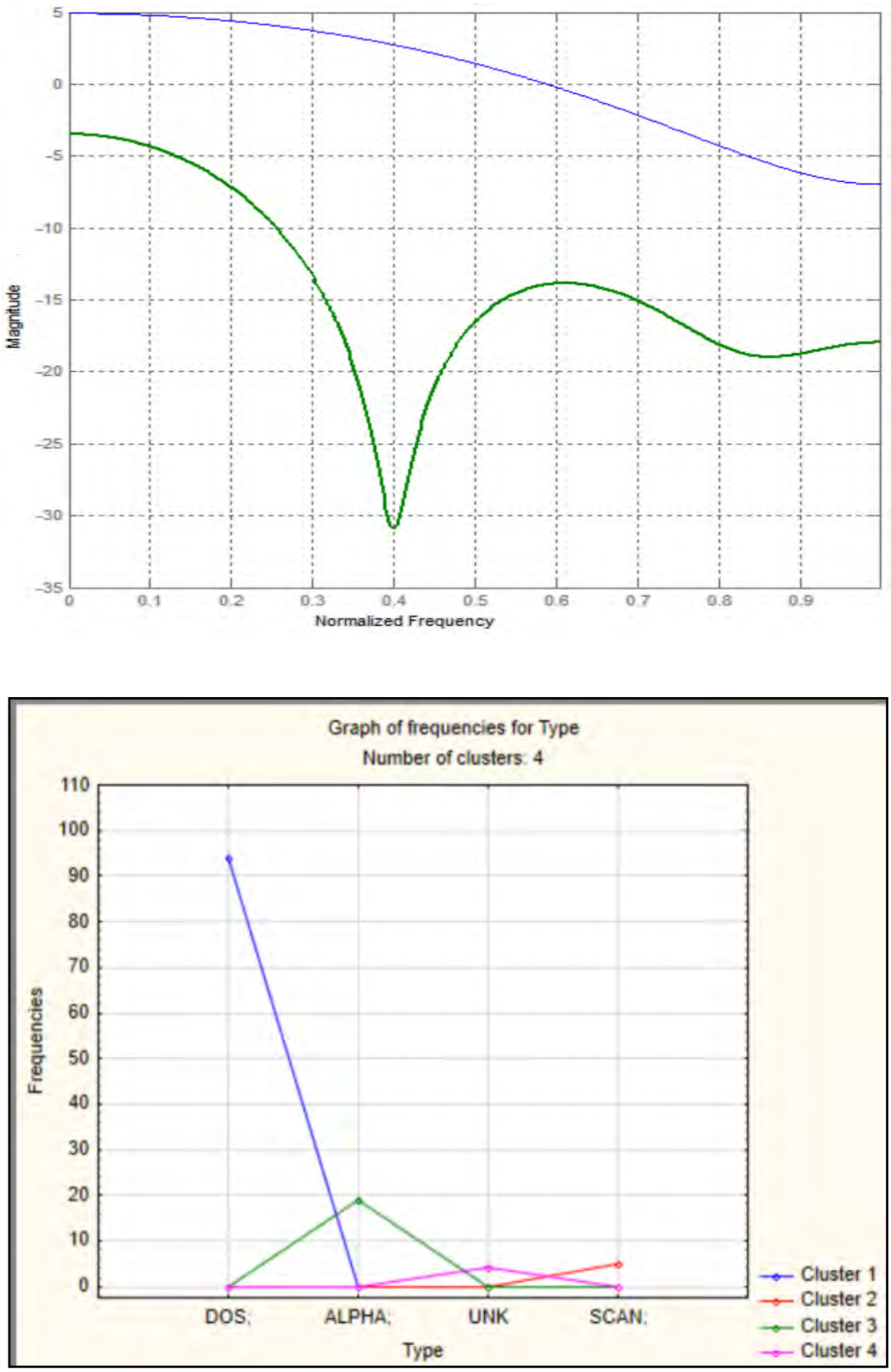
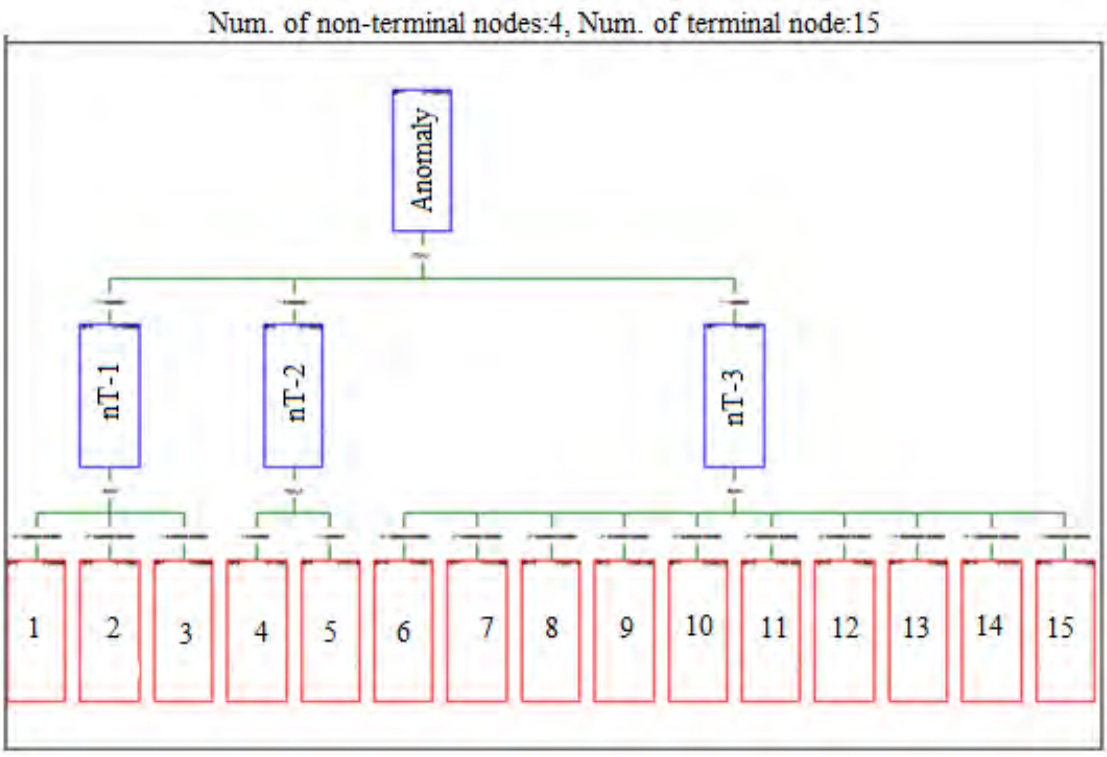


Fig 5h. Comparison for anomaly clustering between
Correlation analysis(upper) and Correspondence analysis(lower)



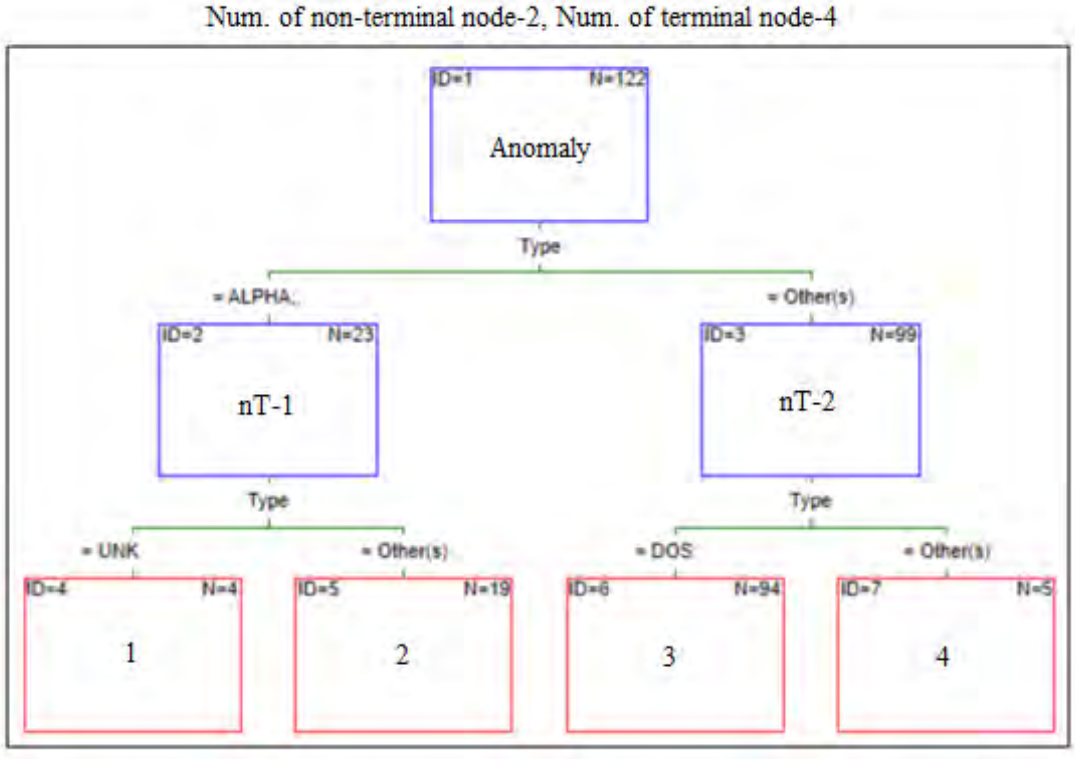
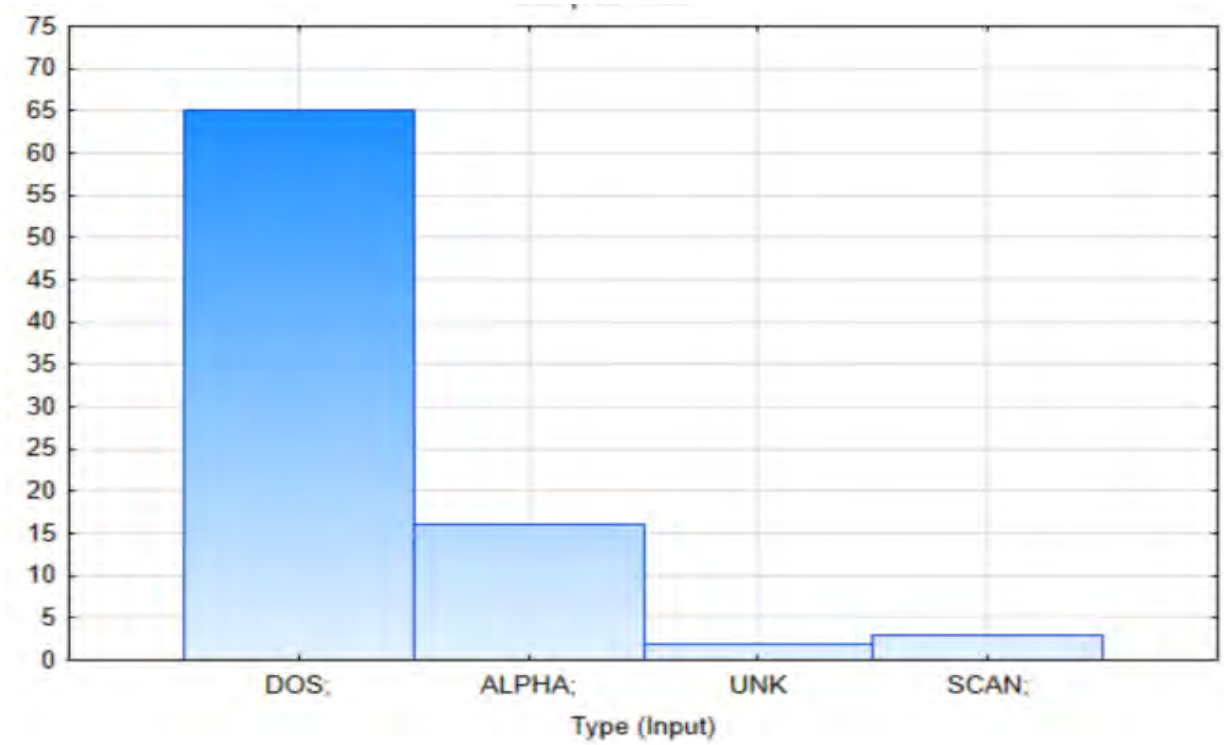


Fig 5i. Tree graph for CHAID (upper) clustering and CA (lower)

In Fig 5i, in case of CHAID, it can be seen that the terminal nodes are 15 and in case of Correspondence Analysis, the number of terminal nodes are 4. It is clear from the output that, in CHAID, as because they have not used any equation to terminate the algorithm, the nodes keep splitting until many false positive nodes are found. But in case of CA, an equation has been added to find when to terminate the splitting, as a result, confusion related nodes with clusters are almost equals to zero. The two tree graphs show the rate of original anomalous data cluster number with terminal and non-terminal node for my proposed algorithm and benchmark CHAID algorithm. In the figure below number of terminal nodes is found to be four. This is because of the splitting equation that has been used as the termination criteria. Which again, satisfies the main purpose of clustering.

In case of histogram also (Fig 5j.), CHAID shows significant difference than CA. It is because CHAID only takes handful amount of data to deal with at a time, when CA is capable of processing quite large number of data and the proposed algorithm is data processing order independent.



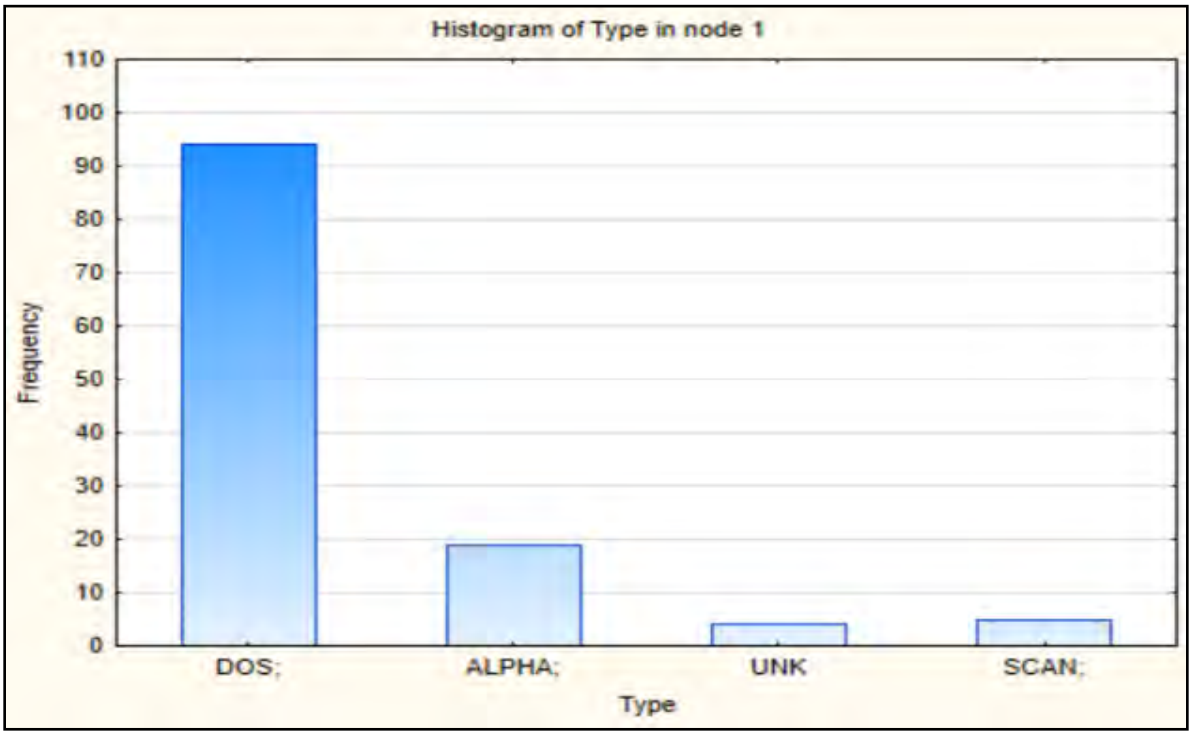


Fig 5j. Histogram of anomaly type for CHAID(upper) and CA(lower)

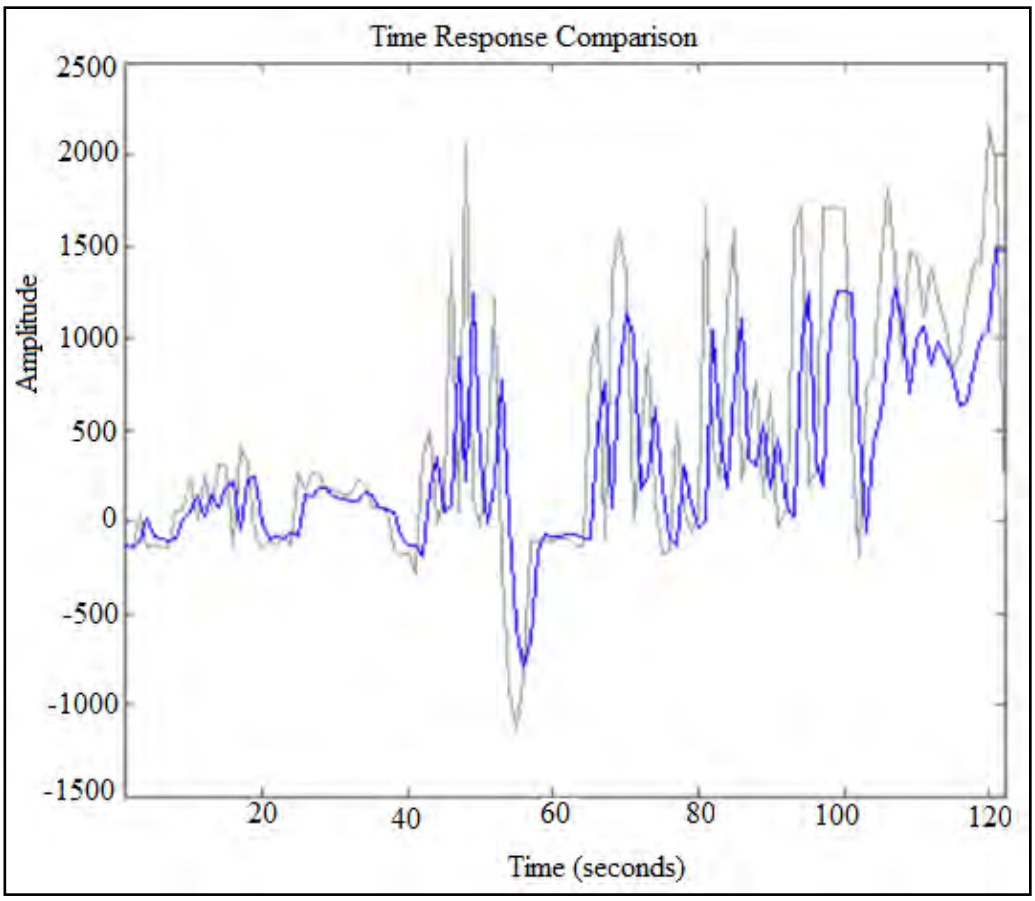


Fig 5k. Violation from norm and related time (CA-Blue & CHAID- Gray)

Data set (Abilene) No. of Obs	Efficiency with respect to time of CA algorithm	Efficiency with respect to time of CHAID algorithm
0-40	33%	38%
41-90	70%	67%
91-120	88%	83%
121-140	93.87%	92%

Table 5e. Comparison of efficiency between CHAID and CA

In Table 5e, the efficiency of the algorithm was found for the data processing of different ranges. Clearly, as the number of data processing increases, efficiency decreases, it is because the computational complexity of the algorithms. In future work, one of the works will be to minimize the complexity of the algorithm.

All of the comparisons in hand, shows that the proposed CA algorithm is much more efficient than the existing algorithms.

Comparing response with respect to time has been shown between proposed algorithm and CHAID algorithm in Fig 5k. The response time is comparatively less for the proposed algorithm, whereas, for CHAID algorithm, the data processing time increases with the increasing amplitude.

5.3 When to stop splitting

For completing the splitting, the quality measure for clustering is:

$$Q(C) = \sum_{C_i \in C} \left(\frac{|C_i|}{|T|} \right)^2 \sum_{j=1}^{m} \sum_{v \in D_j} [(p(x_j = v))^2 - (p'(x_j = v))^2]$$

where,

$p\ (x_j = v)$ *is p*robability of cluster

$p'\ (x_j = v)$ *is p*robability over all the objects for clustering

$(|C_i| / |T|)^2$ is weight which penalizes split with extremely small clusters

The contribution of this work also includes consideration of the splitting equation. It helps the algorithm to identify final clusters of anomalies causing to avoid false splitting. Results show that, this technique also makes this research more promising and outstanding than the other clustering algorithms.

CHAPTER 6

SUMMERY

Finally, to summarize the whole process, it can be said that, after collecting data, with no pre-defined number of clusters for anomalies, can be divided in to two categories, naming- row mass with variance and column mass with variance. The combination of

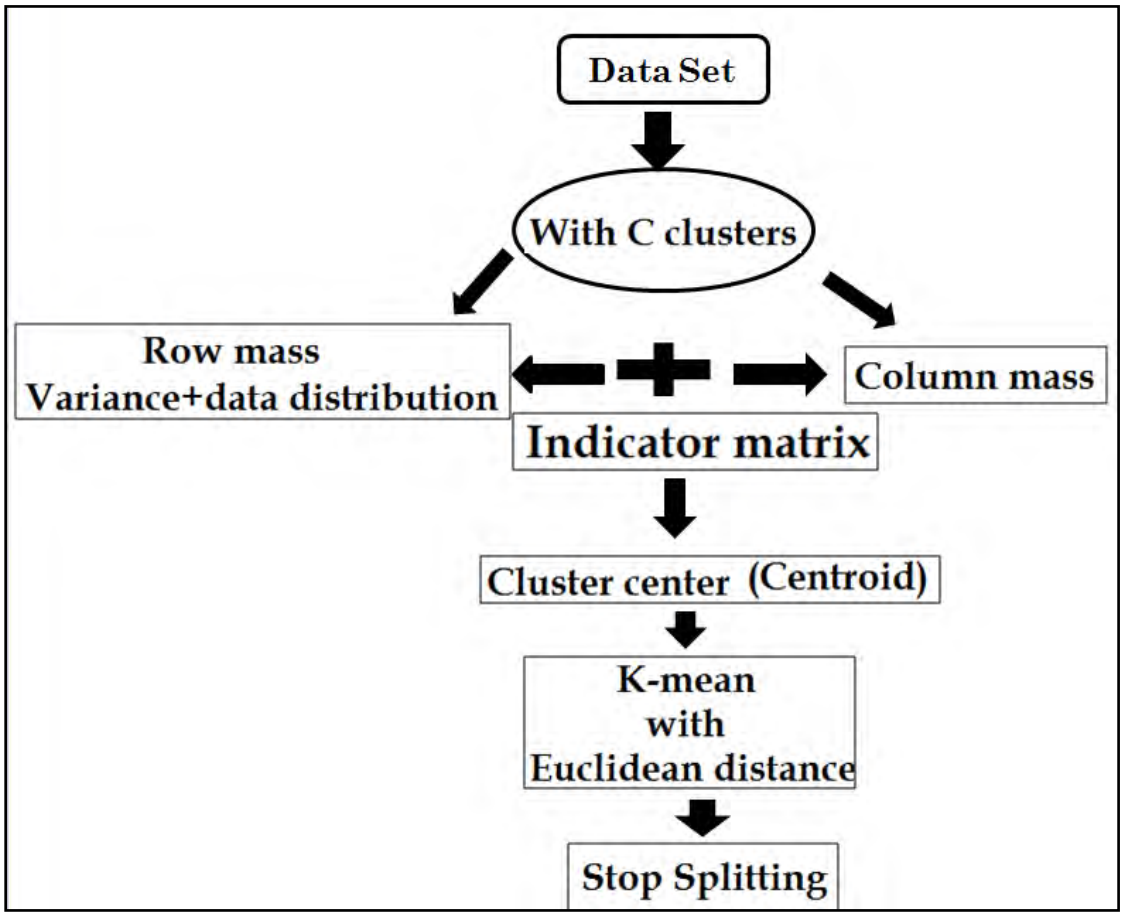


Fig 6a. At a glance the whole CA algorithm

these masses are known as the indicator matrix. Next QR decomposition can be applied and with Euclidean weighted vector, distance can be found. This way, whenever any new data arrives, these processes run until equation for stopping the split start to work. Finally a visual representation of clustered anomalies can be found.

CHAPTER 7

PROBLEMS FACED

While implementing the algorithm, it was necessary to go through tons of previous works related to the work and time to time several ideas occurred before this proposed algorithm finally came up. Some of the works with their drawbacks include:

a. At first, computation of online anomaly classifier has been tried, it is also known as gini indexing shown in Fig 7a. It gives each of the input equal importance. So to do splitting with gini indexing, it was needed to use another classifier just to prioritize anomalies.

Again, unweighted gini index put all the data's in same class. So using weighted function, splitting could be done, but again priority problems come as a drawback. Not to mention, gini indexing needs predefined number of grouping. Almost like k-mean clustering. And it starts with common splitting point, say x.

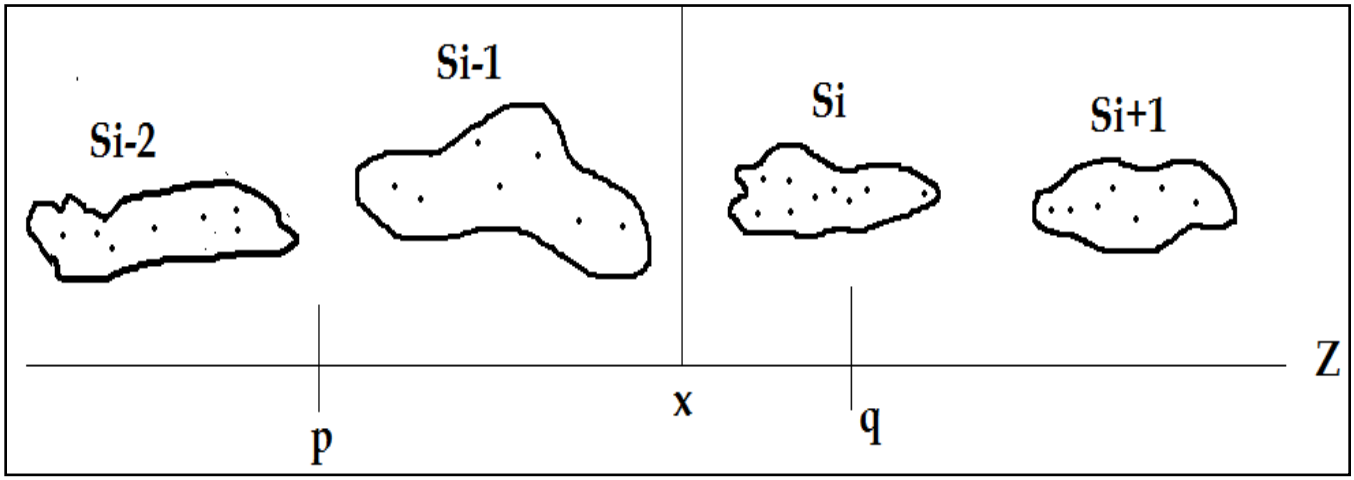


Fig 7a. The gini split index at splitting point x [3]

b. Classification using Sketch: with the help of sketch function, hash estimate was needed to use [30]. Also, it depends on unbiased estimator, known by the equation

$$\hat{R} = \frac{1}{k}\sum_{i=1}^{k}[\min(h_i(A)) = \min(h_i(B))]$$

c. The estimator variance decreases with k-mean clustering. So it is quite difficult to continue with k- mean.

$$\mathrm{Var}(\hat{R}) = \tfrac{1}{k}R(1-R)$$

d. In bottom-k sketches hash function *h* must be evaluated per element of the input set

e. Sketches is being used in conjunction with support vector machines (SVMs)

f. Order of computation increases at the order of $O(|A| \cdot \lg k)$

g. Converting to Hilbert space of input data needed totals mass convergence [29]

h. Addition and scalar manipulations in Hilbert space, need component wise considerations. But anomalies of same class, do not occur one after another. It can come randomly.

CHAPTER 8

CONCLUSION

For understanding the pattern of anomaly in backbone networks and also to know a priori how they will affect the network, clustering of anomalous data is a necessity. The amount of potential anomaly has no bounds, which makes it more difficult for the clustering algorithms to cope with frequently shifting environments. Clustering methods are used to automatically organize anomalous data sets so that the similar anomalies are grouped together with respect to their attribute values. Agglomerative clustering method as well as other techniques such as k-means are widely used for clustering. Each clustering method and each similarity measure affect the quality of the clustering in different ways. In most cases, it is not possible to know the best clustering method and the best similarity measure which may change with accordance to the data set. Moreover, most of the existing algorithms on clustering pre-specifies the number of clusters.

The algorithm that is proposed in this paper, no predefined number of clusters is needed. The reason to apply the correspondence analysis for data clustering is due to its strong capability of learning, memorizing and self-adaptation. Agglomerative clustering with CA and QR decomposition is considerably new era of clustering and the results are showing promising outcomes. Comparisons have been made with existing works based on real data from Abilene backbone network, USA. High competence and better results show that the algorithm proposed, is reasonably better for large scale data processing.

CHAPTER 9

FUTURE WORK

Future endeavor will be to undertake an investigation of online clustering approaches. It would also be interesting to combine the algorithm of the current research with density clustering based on the border-expanding algorithm [12]. BEDBSCAN employs border objects as seeds to expand the cluster. This will alleviate the need for calculating the profiles separately. Experiments on the real dataset and synthetic datasets indicated in [12] that the BEDBSCAN algorithm can find all clusters correctly and enhance the speed greatly. High effectiveness and efficiency of BEDBSCAN algorithm show that the online clustering is feasible for large-scale data processing.

Also, there has been a fairly new invention named Kuramoto Model. It has the specifics stated below:

- It considers only the timing and frequency of when the anomalies are occurring
- It is not important to consider an absolute time-scale of when each anomaly occurs
- It is only important to consider relational time series of when anomalies are 'occurring' (i.e. their phase) in relation to the other parts of the system [31].

Considering all these recent inventions, future approach will be to build a dictionary of anomalies, and also to make sure that the algorithm will not need to store data before analyzing, i.e., will be able to process data online.

BIBLIOGRAPHY

1. R. Elwell, R. Polikar, "Incremental Learning of Concept Drift in No stationary Environment", IEEE *Transactions on Neural Networks*, VOL. 22, NO. 10, October 2011.
2. E. Keogh, S. Kasetty, "On the need for time series data mining benchmarks: A survey and empirical demonstration", In Proc. of SIGKDD, 2002.
3. J. Shao, X. He, "Synchronization-Inspired Partitioning and Hierarchical Clustering", Issue No.04 - April (2013 vol.25), pp: 893-905.
4. A. Lakhina, M. Crovella and C. Diot, "Mining anomalies using traffic feature distributions", SIGCOMM‟05, August 21-26, 2005 USA, ACM 1-59593-009-4/05/0008
5. L. Cohen, G. Avrahami, M. Last, A. Kandel, O. Kipersztok, "Incremental Classification of Nonstationary Data Streams", Boeing Phantom Works Mathematics & Computing Technology, Jan 2011.
6. Y. Law, C. Zaniolo, "An Adaptive NNC algorithm for data stream", KDD 2005, LNAI 3721, pp. 108–120 in Springer-Verlag Berlin Heidelberg 2005.
7. H. Yoon, K. Alsabit, S. Ranka, "Tree based Incremental Classification for Large Dataset", CISE department of Florida, TR-99-013, University of Florida.
8. Y. Kanda, K. Fukuda, T. Sugawara, "Evaluation of Anomaly Detection Based on Sketch and PCA", 978-1-4244-5637-6/10, IEEE Globecom 2010.
9. Michael F. Fox, "A Multi-dimensional Exploration of the Decision Process Using Correspondence Analysis", Marketing Bulletin, 1993, 4, 30-42, Article 4, Page 1of 12.
10. K.Xu, Z.-L.Zhang, "Profiling internet traffic backbone: Behavior model and applications", ACM, SIGCOMM‟2005.
11. T. Xiong, S. Wang, A. Mayer, E. Monga, "A New MCA-based Divisive Hierarchical Algorithm for clustering categorical data", 9th IEEE International Conference on Data Mining, 2009.

12. C. Estan, S. Savage, G. Varghese, "Automatically Inferring Patterns of Resource consumption in network traffic", SIGCOMM‟03, August 25-29, Karlsruhe, Germany, ACM.
13. E. Cesario, G. Manco and R. Ortale, "Top-down parameter-free clustering of high-dimensional categorical data", IEEE *Transactions on knowledge and data engineering*, 19(12), 2007.
14. D. Chen, Y. Yan, D. Wang, "Density clustering based on border-expanding", IEEE, 10th *International Conference on Natural Computation*, 2014.
15. N.X. Vinh, J. Epps, and J. Bailey, "Information Theoretic Measures for Clusterings Comparison: Is a Correction for Chance Necessary?" *Proc. Int'l Conf. Machine Learning (ICML),* pp. 1073-1080, 2009.
16. T. Ahmed, M. Coates ; A. Lakhina, " Multivariate Online Anomaly Detection Using Kernel Recursive Least Squares", INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE
17. E. Keogh, S. Lonardi, C. Ratanamahatana, "Towards Parameter-Free Data Mining", KDD ‟04, August 22–25, 2004, Seattle, WA, U.S.A., 2004 ACM.
18. C.A. Ratanamahatana , E. Keogh, *Making Time-series Classification More Accurate Using Learned Constraints.* In proceedings of SIAM International Conference on Data Mining (SDM '04), Lake Buena Vista, Florida, April 22-24, 2004
19. G. Widmer and M. Kubat, Learning in the Presence of Concept Drift and Hidden Contexts, Machine Learning, Vol. 23, No. 1, pp. 69-101, 1996.
20. P. Domingos and G. Hulten, Mining High-Speed Data Streams, Proc. of KDD 2000, pages 71-80, 2000
21. B. Krishnamurthy, S. Sen, Y. Zhang, Y. Chen. "Sketch-based change detection: Methods, evaluation, and applications", In Proceedings of ACM Internet Measurement Conference, Oct. 2003.
22. A. Lakhina, M. Crovella, C. Diot, *Diagnosing network-wide traffic anomalies*. In Proceedings of ACM SIGCOMM, Aug. 2004.
23. A. Lakhina, M. Crovella, and C. Diot. "Mining anomalies using traffic feature distributions". SIGCOMM, Aug. 2005, ACM

24. A. Soule, K. Salamatian, N. Taft. "Combining filtering and statistical methods for anomaly detection". In ACM Internet Measurement Conference, Oct. 2005.
25. M. Thorup, Y. Zhang. "Tabulation based 4-universal hashing with applications to second moment estimation". In Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA), 2004.
26. M. Thottan, C. Ji. "Anomaly detection in IP networks", IEEE Transactions in Signal Processing, 51(8), Aug. 2003.
27. Y. Zhang et al. "Online identification of hierarchical heavy hitters: Algorithms, evaluation, and application", In Proceedings of ACM Internet Measurement Conference, Oct. 2004.
28. K.Chen, L.Liu, "The „best-K" for entropy-based categorical data clustering",17th international conference on Scientific and statistical database management, 2005.
29. C.-C. Chang, C.-J. Lin, "LIBSVM: A library for support vector machines," ACM Transactions on Intelligent Systems and Technology, vol. 2, pp. 27:1–27:27, 201, 2008
30. K.Xu, Z.-L.Zhang, "Profiling internet traffic backbone: Behavior model and applications", ACM,SIGCOMM,2005
31. Junming Shao, Xiao He, "Synchronization-Inspired Partitioning and Hierarchical Clustering", IEEE Transactions, vol.25, Issue 4. 2013
32. H. Abdi, D. Valentin, N. Salkind, "Multiple Correspondence Analysis", Encyclopedia of Measurement and Statistics. Thousand Oaks(CA):Sage. The University of Texas at Dallas, Richardson,TX75083–0688,USA, 2007
33. S. Vega-Pons, J. Correa-Morris, and J. Ruiz-Shulcloper, "Weighted Partition Consensus via Kernels," Pattern Recognition, vol. 43, no. 8, pp. 2712-2724, Aug. 2010