TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Brau- und Getränketechnologie

Wissenschaftszentrum Weihenstephan

für Ernährung, Landnutzung und Umwelt

Non-Parametric Methods for Data Processing and Knowledge Mining in Bioprocesses

Kalutara Koralalage Lasantha Britto Adikaram

Vollständiger Abdruck der von der Fakultät Wissenschaftszentrum Weihenstephan für Ernährung, Landnutzung und Umwelt der Technischen Universität zur Erlangung des akademischen Grades eines

Doktor der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. Klaus Richter

Prüfer der Dissertation:

1. Univ.-Prof. Dr. Thomas. Becker
2. Univ.-Prof. Dr. Andreas Gronauer

Die Dissertation wurde am 15.01.2018 bei der Technischen Universität München eingereicht und durch die Fakultät Wissenschaftszentrum Weihenstephan für Ernährung, Landnutzung und Umwelt am 09.04.2018 angenommen.

Acknowledgements

Firstly, I would like to extend my gratitude to Prof. Dr. Thomas Becker for giving me the opportunity to conduct my research work at his chair and for his vluaable guidance, support, and encouragement that enabled the successful completion of this research.

Then, I would like to give my thanks to the examining committee, for spending their valuable time and effort in reviewing this thesis. I am also very grateful to the German Academic Exchange Service (DAAD: Deutsche Akademische Austauschdienst) for providing financial assistance for my family and me during our stay in Germany. I must also express my thanks to TUM graduate school for the friendly and very professional assistance they provided to me; this allowed me to successfully complete my research work. I would also like to thank Bayerische Landesanstalt für Landwirtschaft (LfL) for providing the data required for my research work.

My special thanks go to Dr. Mohamed A. Hussein and Dr. Mathias Effenberger for their vital feedbacks, which enhanced the scientific quality of this research work. Furthermore, I must again be thankful to them for their friendship, support, and valuable advice they gave to me, especially during the hard periods of the research work.

I take this opportunity to thank Prof. Andreas Gronauer and Prof. K.D.N. Weerasinghe for opening the doors for me to study in Germany. I cannot forget the assistance given to me by the University Grants Commission, Sri Lanka and the Faculty of Agriculture, University of Ruhuna, Sri Lanka. I would like to thank all the staff members of the University Grants Commission and the University of Ruhuna for their support. It is my pleasure to give my thanks to Prof. S.G.J.N. Senanayake, Prof. R. Senaratne, Prof. S. Subasinghe, Prof. R.T. Serasinghe, Prof. W.M.M.P. Wijeratne, Prof. Champa Nawarathna, Prof. P. A. Jayantha, and Dr. M.K.D.K. Piyarathna for kindly supporting me to complete my research work in Germany.

I sincerely thank all the kind and helpful people of Germany, especially the anonymous lady who helped me on my first day (Sunday, 11th April 2010). On that day, I could not find my apartment in Hans-Sachs-Ring, Manheim. She stopped her car and showed me the way to my apartment, which was about 300m away. This unexpected experience on my very first day in Germany gave me great confidence to stay in Germany and bear any hard situations with a positive attitude. During my stay in Germany with my family, I saw this helpfulness many times from the German people,

especially from the friends and colleagues of Bayerische Landesanstalt für Landwirtschaft (LfL) in Freising, the friends and colleagues of Brau- und Getränketechnologie, TUM Weihenstephan, and the members of Buddhistisches Kloster Bodhi Vihara in Freising.

Finally, I thank my wife Disna, my son Himsara, and my daughter Sandathara for their patience during this period of work. I would also like to thank all my friends and relatives for their assistance and kindness. And last but not least, I thank my parents for everything.

Freising, 01.10.2015

K.K.L.B. Adikaram

Publications

Peer reviewed publications

1. Adikaram, K.K.L.B., Hussein, M.A., Effenberger, M., Becker, T.: Continuous Learning Graphical Knowledge Unit for Cluster Identification in High Density Data Sets. Symmetry. 8 (12) (2016). DOI 10.3390/sym8120152.

2. Adikaram, K.K.L.B., Hussein, M.A., Effenberger, M., Becker, T.: Non-Parametric Local Maxima and Minima Finder with Filtering Techniques for Bioprocess. Journal of Signal and Information Processing. 7 (2016). DOI 10.4236/jsip.2016.74018.

3. Adikaram, K.K.L.B., Hussein, M.A., Effenberger, M., Becker, T.: Multi-Variable, Multi-Layer Graphical Knowledge Unit (MVML-GKU) for Storing and Representing Density Clusters of Multi-Dimensional Big Data. Applied Sciences. 6 (4) (2016). DOI 10.3390/app6040096.

4. Adikaram, K.K.L.B., Hussein, M.A., Effenberger, M., Becker, T.: Universal Linear Fit Identification: A Method Independent of Data, Outliers and Noise Distribution Model and Free of Missing or Removed Data Imputation. PLoS ONE 10 (11) (2015). DOI 10.1371/journal.pone.0141486.

5. Adikaram, K.K.L.B., Hussein, M.A., Effenberger, M., Becker, T.: Data Transformation Technique to Improve the Outlier Detection Power of Grubbs' Test for Data Expected to Follow Linear Relation. Journal of Applied Mathematics (2014). DOI 10.1155/2015/708948.

6. Adikaram, K.K.L.B., Hussein, M.A., Effenberger, M., Becker, T.: Multiple memory structure bit reversal algorithm based on recursive patterns of bit reversal permutation. Mathematical Problems in Engineering (2014). DOI 10.1155/2014/827509.

7. Adikaram, K.K.L.B., Hussein, M.A., Effenberger, M., Becker, T.: Outlier detection method in linear regression based on sum of arithmetic progression. The Scientific World Journal (2014). DOI 10.1155/2014/821623.

8. Adikaram, K.K.L.B., Hussein, M.A., Becker, T.: Impact of Microsoft Visual C++ version on the performance of arrays and vectors. Research Journal in Engineering and Applied Science 3 (2014), 262-266.

.

Technical reports

1. Kissel, R., Adikaram, K.K.L.B., Pohl, A., Gracia, E.R., Effenberger, M.: Betriebs-Monitoring: Vergleichende Untersuchung für die Einwerbung und Vergärung von Grünlandaufwüchsen Abschlussbericht - Schwerpunkt Anlagen-Monitoring. Bayerisches Staatsministerium für Ernährung, Landwirtschaft und Forsten, München, Germany (2015).

Conference contributions

1. Adikaram, K.K.L.B., Hussein, M.A., Effenberger, M., Becker, T: Non-parametric high and low extrema filtering method for filtering extrema in dynamic domains. SAITM 6th Annual International Research Symposium on Engineering Advancements, Malabe, Sri Lanka; 04/06/2016

2. Adikaram, K.K.L.B., Hussein, M.A., Effenberger, M., Becker, T: Novel nonparametric extrema identification method. 12th Academic Sessions, University of Ruhuna, Matara, Sri Lanka; 02/03/2016

3. Adikaram, K.K.L.B., Hussein, M.A., Effenberger, M., Becker, T: A Simple Reliable Unstructured Data Synchronization Technique for Biogas Data. ISAE 2016, Kamburupitiya, Sri Lanka; 13/01/2016

4. Adikaram, K.K.L.B., Hussein, M.A., Effenberger, M., Becker, T: Outlier Detection Method for Identifying Outliers that are not in Gaussian Distribution. 12th Academic Sessions, University of Ruhuna, Matara, Sri Lanka; 04/03/2015

5. Adikaram, K.K.L.B., Hussein, M.A., Effenberger, M., Becker, T: Improving the performance of an algorithm by using multiple single dimensional memory structures for index mapping. 2nd Ruhuna International Science and Technology Conference (RISTCON 2015), Matara, Sri Lanka; 22-23/01/2015

6. Adikaram, K.K.L.B., Hussein, M.A., Effenberger, M., Becker, T: Data Transformation Technique to Improve the Outlier Detection Power of Grubbs Test for Data Expected to Follow Linear Relation. International Symposium on Agriculture & Environment - 2014 (ISAE 2014), Kamburupitiya, Sri Lanka;
   01/11/2014

7. Adikaram, K.K.L.B., Hussein, M.A., Effenberger, M., Becker, T: Data warehouse framework for unstructured biogas data. International Conference of Agricultural Engineering, Zurich; 09/07/2014

Contents

Abstract

In parametric methods, detection criteria of methods are based on one or several domain dependent values such as numerical averages, standard deviations, and numbers of nearest neighbours. Thus, certain detection criteria of parametric methods are valid only for considered data model or considered conditions in the domain. Also, the accuracy of the outcome of parametric methods depends on the values of variables. In contrast, non-parametric methods depend on a less number of underlying assumptions and they are known as distribution-free (data model independent) methods. Because of that, non-parametric methods are considered as robust methods. Due to the dynamic nature of biological processes such as biogas plants, it is a big challenge to analyse, control, and monitor biological processes using parametric methods. When the expected data model or domain conditions (e.g.: value range of data) change, it is necessary to recalibrate parameters or develop new models for monitoring, controlling, and analysing. Therefore, the usage of non-parametric methods is a good solution for overcoming or minimizing afore-mentioned drawbacks.

In this research, new non-parametric techniques were developed for linear fit identification, extrema detection, extrema filtering, and knowledge representation as density clusters. In the linear fit identification method (UniLiFI: Universal linear fit identification) *2/n* is used as the detection criteria, where *n* is the number of data points. Extrema identification is performed by comparing two ratios in relation to maximum, minimum, middle point, and sum (there is no involvement of any external criterion). The threshold criteria for methods developed for filtering non-dominating extrema (MMS-Window based filter or MMS-WBF) and sharp and gradual (flat) extrema (MMS-SG filter) are values based on the number of data points (n). The threshold criterion of the method developed for locating low and high extrema (MMS-LH filter) is a value between 0 and 1. In the method for knowledge representation (GKU: Graphical Knowledge Unit), properties of the marker (graphical symbol of the data point) such as colour, size, and shape are used as cluster formation and identification. Thus, all the developed methods are non-parametric.

All the developed methods were evaluated with the automatically captured biogas data. Results proved that the developed methods are capable of identifying linear fit, identifying extrema, filtering extrema, and density cluster formation, with a high level of robustness. In addition, I have very promising expectations that the new methods will open new windows in the field of data processing, data analysing, process monitoring, and process controlling in bioprocesses.

Zusammenfassung

In der Prozessanalyse ist die Genauigkeit der Ergebnisse parametrischer Methoden nur innerhalb eines definierten Wertebereiches gegeben. Damit sind diese Methoden für dynamische Bioprozesse ungeeignet. Nichtparametrische Methoden sind hingegen Modell unabhängig und erfordern nur wenige Grundannahmen. Sie können daher auch für sehr dynamische Bioprozesse robuste Ergebnisse liefern. Die hier vorliegende Arbeit stellt nicht-parametrische Methoden für die lineare Approximation, Extremwertermittlung, Extremwertfilterung und Identifikation von Clustern für die Bioprozess-Analyse vor.

1. Introduction

1.1 Challenges in data analysis and knowledge mining in biogas plants

Biogas production is a good example of a biological process that converts organic materials into biogas. The biological process in biogas plants is a recreation of natural anaerobic (oxygen-free) digestion process, which occurs in marshes and wetlands. During the digestion process, bacteria (micro-organisms) in biogas plant produce mainly methane (CH_4) and carbon dioxide (CO_2) known as biogas [1-3]. Figure 1.1 shows typical components of an industrial scale biogas plant and data capturing points.

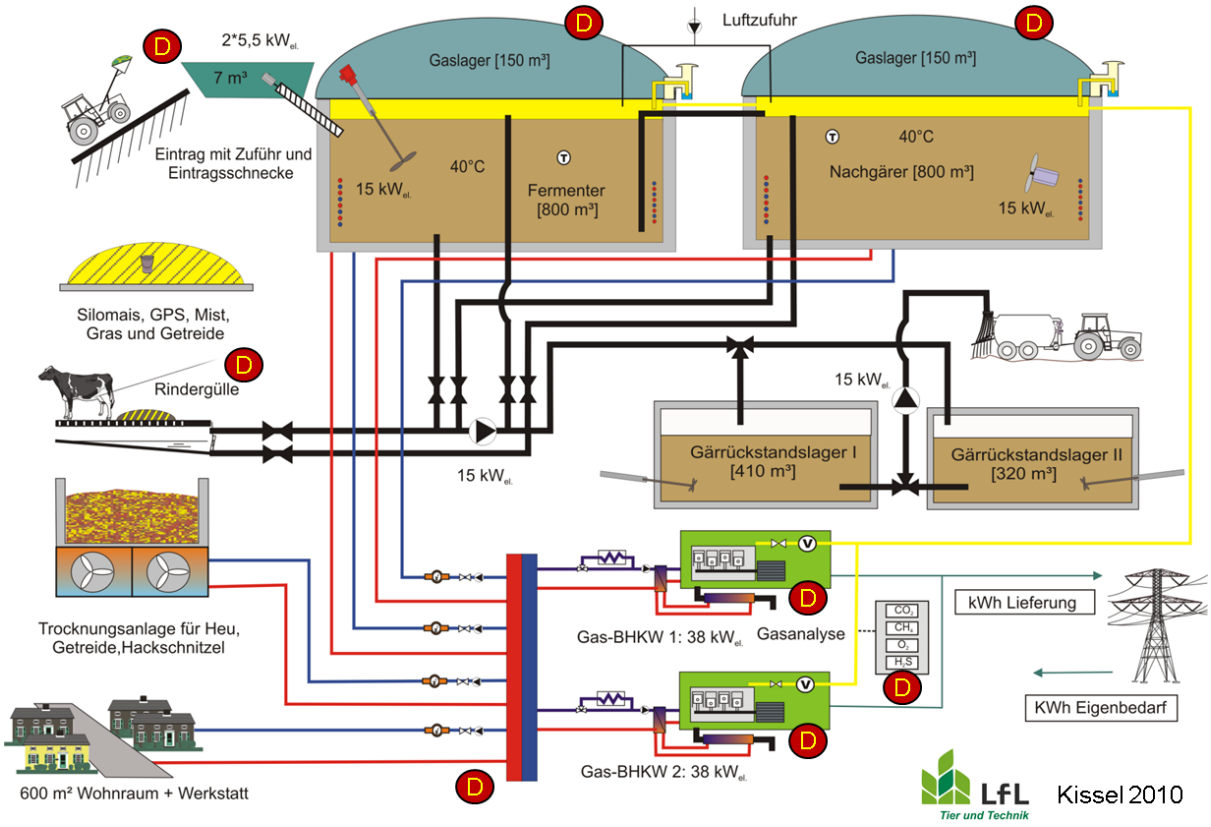


Figure 1.1 : Diagram of an industry scale biogas plant

Because of the complex nature, most of the bioprocesses require close and intensive analysis of all the steps of the entire process cycle. This leads to capturing huge amounts of data. However, with huge amounts of data, knowledge discovery, data mining, feature identification, and data processing are considered as challenges [4-6]. The information extract via knowledge discovery, data mining, or feature identification is used for monitoring, controlling, and understanding the process.

In the field of biogas, there are many different types of plant designs and variety of technologies for biogas production. Thus, biogas industry is considered as an industry that has very complex bioprocess [7]. Compositions of possible input materials are diverse are due to the huge range of input materials. Even for a single biogas plant, the composition of input materials is not constant. For example, type, amount, and mixing ratio of materials prone to be changed according to the availability and sometimes due to policy decisions. In an industry scale biogas plant, in addition to the data directly obtained from the biogas plant there are data from experiments as well as data from analytical laboratories. As a result of the aforementioned factors, biogas plants produce huge amounts of heterogeneous data. This makes the data collection more complicated for knowledge discovery, data mining, feature identification, and data processing. Composition and diverse nature of biogas data are shown in Figure 1.2.

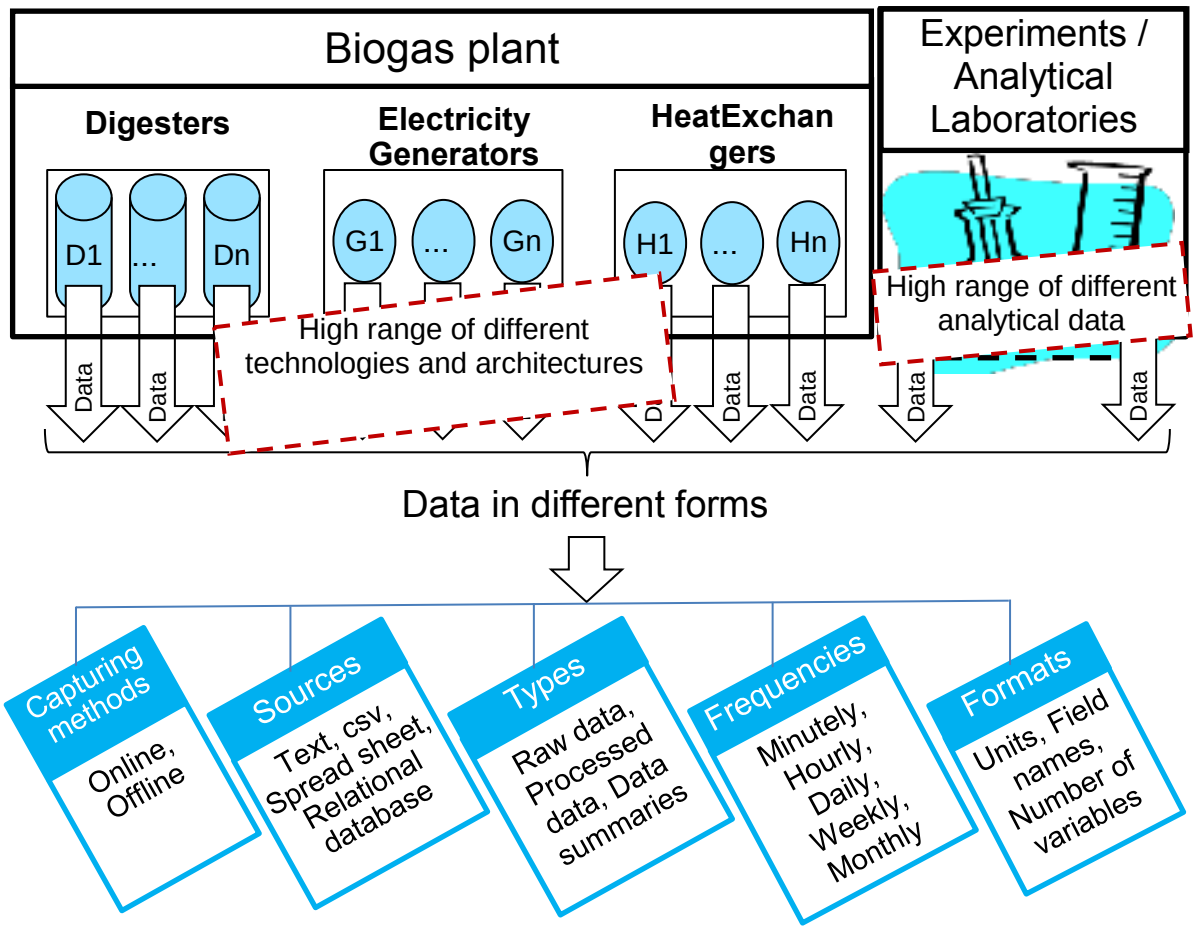


Figure 1.2 : Data diversity in biogas plants.

Since the behaviour of a biogas plant depends on a multitude of different factors, it is hard to find out all possible combinations of factors. For example, the behaviour of a certain plant design can be changed by modifying the physical size of the input material. Still, there are many unidentified factors that directly affect the behaviour of a biogas plant, particularly chemical and biological factors. Therefore, the accuracy of simulated models of a biogas plant that is developed based on already known factors, relations, and dependences totally depends on the underlying assumptions. As a result of this, there exists a large number of methodologies which are applicable only to a certain plant design [8-10].

In general, it is impossible to apply one system developed for controlling and monitoring a certain biogas plant directly to another biogas plant with a different design. Either a new system has to be developed, or the existing system has to be modified accordingly. Also, when developing a system, it is being tested with lab scale digesters with controlled conditions and determining the controlling conditions and constraints of the system. However, when deploying plant design as an industrial scale plant, it may not function as it did in the laboratory.

While knowledge of chemical and biological processes in biogas plants is being extended continuously, existing methodologies are not capable of absorbing new knowledge and implement it in an easy and cost-efficient manner [4]. This is reflected by the structure of existing systems that is used to control and monitor biogas plants. Still, there is no global model for biogas plants, which is capable of absorbing new knowledge and updating/changing its functionality. Such a system would be able to deal with different types of plant designs with different behaviours. It would eliminate the huge effort of creating design-dependent, unique systems. Deploying such a system in research laboratories and institutes will create a platform for storing and analysing data about the behaviour of biogas plants with different designs.

1.2 Non-parametric methods

As aforementioned, knowledge discovery, data mining, feature identification, and data processing are the main challenges in the whole process of process controlling and monitoring. Available techniques in afore-mentioned fields can be classified as parametric and nonparametric methods [11]. Particularly, parametric methods use

one or several domain dependent values as detection criteria. Numerical averages, standard deviations, and numbers of nearest neighbours are some examples of such parametric values that are used as detection criteria in parametric methods. These criteria based on parameter values are valid only for considered data model or considered conditions in the domain. The major drawback of parametric methods is that the accuracy of the outcome depends on the values of variables [12]. In contrast, non-parametric methods are known as distribution-free (data model independent) methods. Thus, non-parametric methods depend on fewer numbers of underlying assumptions [11, 13, 14] and considered as robust methods [12, 15]. Because of that, if there are non-parametric (data model independent) methods, the aforementioned drawbacks can be overcome or minimized.

In this research, linear fit identification, extrema detection, extrema filtering, andknowledge representation as density clusters were considered. At present, except extrema detection, all other areas mainly depend on parametric methods. Thus, in this research, novel non-parametric methods were introduced for identifying linear fit, extrema filtering (non-dominating extrema filtering, sharp and gradual extrema filtering, and low and high extrema filtering), and density cluster formation and identification. Furthermore, another novel non-parametric method for extrema identification was introduced. The non-parametric properties used in proposed methods are as follows.

1. In the linear fit identification method (UniLiFI: Universal linear fit identification) *2/n* is used as the detection criteria, where *n* is the number of data points. Also, the linear fit identification method is totally independent of data, outlier, and noise distribution models and free of missing/removed data imputation.
2. The extrema detection method is capable of detecting all the local extrema, and extrema identification is performed after comparing two ratios in relation to maximum, minimum, middle point, and sum.
3. The threshold criteria for methods developed for filtering non-dominating extrema (MMS-Window based filter or MMS-WBF) and sharp and gradual (flat) extrema (MMS-SG filter) are values based on the number of data points (*n*). The threshold criterion of the method developed for locating low and high extrema (MMS-LH filter) is a value between 0 and 1.

4. The knowledge representation is based on properties of the marker (graphical symbol of the data point) such as colour, size, and shape. The method was named as “Graphical knowledge Unit” (GKU). GKU is a collection of continuous learning knowledge cells formed out of pixels in a bitmap. The GKU is not only a density cluster identification method, but it is also capable of indicating the density of missing data and out of range data, which convey an indicator of the quality of the data especially with big data. Furthermore, GKU can be used as a data cleaning technique, data visualization technique, and portable database. The GKU is extended for multivariate versions for representing density clusters in multivariate data sets.

1.3 Linear fit identification

In any domain, clean data is the data that follows the assumed data distribution model [16], while noise is the data that follows the assumed probability distribution [16, 17]. Then, outliers are data that are not in agreement with the assumed clean data and noise models. In the concept of “linear fit”, the clean data is the data that agree with $y = mx + c$ model (linear regression model). Statistical and model-based approaches are two of the most popular linear fit identification techniques. Majority of the popular statistical and modal based approaches use parametric properties such as variance, average, median, and standard deviation as construction components.

In the process of linear regression identification, the best approach is to first remove outliers / noise with reference to the assumed models and then do the regression analysis on identified linear fit. Most of the outlier/noise detecting methods depend on certain data model such as Gaussian (Normal) distribution model. On the other hand, except methods such as angel based and some cluster based outlier detection methods [18, 19], most of the outlier/noise detection methods such as Sigma filter, Grubbs method, and moving average are considered as parametric methods. Sigma filter usually uses average and multiples of standard deviation as the criteria values. However, in 2013, Leys et al. [20] showed that the combination of median and multiples of median absolute deviation provides more robust and reliable results [20]. Grubbs method uses tabulated significant levels as criteria values. Moving average uses average of an advancing window as the filtering method.

Besides the major disadvantage in relation to parametric methods, if there is no standardization (standardization process requires additional computation time), the values of detection criteria are domain dependent. Therefore, it is not possible to compare two criteria even if they are numerically the same.

“Anscombe's quartet” [21] is a good example for identifying the disadvantages of statistical methods or paramedic methods. In 1973, Francis Anscombe [21] used “Anscombe's quartet” to demonstrate the data sets that have nearly identical statistical properties (Table 1.1) and have considerable variation when graphed (Figure 1.3). Furthermore, Anscombe demonstrated the importance of the effect of outliers on nonparametric methods. Despite the distribution dissimilarities of data sets in “Anscombe's quartet”, “linear regression” is the same for all four data sets. There are four influencing factors in relation to regression detection:

1. Outliers and noise [22-26].
2. Nature of distribution of the clean data, noise, and outliers [27, 28].
3. Amount of outliers and noise [29-32].
4. Missing data [33, 34].

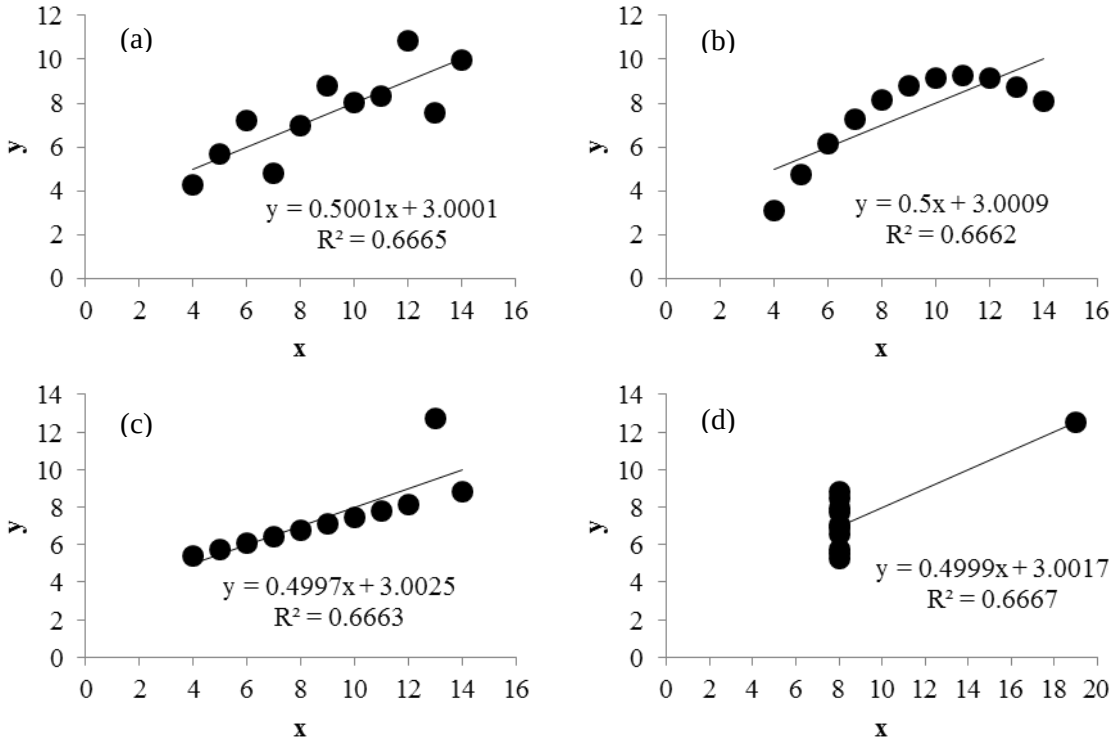


Figure 1.3: Anscombe's quartet. All the plots show nearly identical statistical properties even though they have very clear differences when plotted.

Table 1.1: Statistical properties of plots in Anscombe's quartet. The accuracy of values is to two decimal points.

Property	**Plot (a)**	**Plot (b)**	**Plot (c)**	**Plot (d)**
Mean of x	9.00	9.00	9.00	9.00
Sample variance of x	11.00	11.00	11.00	11.00
Mean of y	7.50	7.50	7.50	7.50
Sample variance of y	4.13	4.13	4.12	4.12
Coefficient of determination (R^2)	0.67	0.67	0.67	0.67
Linear regression	y = 0.50x + 3.00 for all the plots			

The method introduced in this research is a single nonparametric method that is capable of addressing all the afore-mentioned challenges with very high levels of robustness, especially with the data sets that are considered as very extreme situations.

Impact of outliers and noise in the linear fit identification

Least Squares Method" (LSM) is the parametric statistical method that was employed for determining the linear regression in data sets of Anscombe's quartet. LSM is the most common and the most popular linear fit identification method that is used in most popular data analysing packages such as MATLAB, SPSS, Minitab, r, and Mathematica. There are several versions of LSM and the most common version is known as generalized LSM (GLSM). The major drawback of GLSM is that the method demands outliers and noise to be in Gaussian distribution, which cannot be always guaranteed. The results of using parametric statistical methods on such data sets were very clearly elaborated in the Anscombe's quartet (Figure 1.3). As a solution for that, data has been pre-processed and outliers and noise were removed using suitable methods before applying GLSM.

Among the model-based approaches, Kalman filter [35, 36] is one possible method that can be used for linear fit identification. Weighted LSM (WLSM) is another approach for model-based linear fit identification method. When the outliers and

noise are not in Gaussian distribution (heteroscedastic situations), WLSM is applied by using relevant weighted parameters that depend on the considered domain [37, 38]. The accuracy of model-based approach depends on the accuracy of the identified clean data model and error data model. If it is not possible to identify the correct models, the model-based approach is not feasible [17].

Impact of distribution of outliers and noise in the linear fit identification

In any real data capturing, it is usual to observe that sometimes data do not agree with the considered regression. This occurs mainly due to data capturing error or sudden change in the process. Sometimes, these data points are considered as either noise or outliers. In linear regression, when the resultant error of all the data is zero (error in Gaussian distribution), GLSM provides optimal results [39, 40]. Figure 1.4 elaborates the effect of the distribution of outliers on regression identification. In plots (a) and (b) clean data are expected to follow $y = 0$ while in plots (c) and (d) clean data are expected follow $y = x$ model. However, outliers (2 outliers) in plots (a) and (c) are in Gaussian distribution while in the other two plots outliers are not in Gaussian distribution. When the outliers are in Gaussian distribution, detected linear regression is closer to the regression of clean data than the plots with outliers in non-Gaussian distribution. However, in reality it is hard to find data sets consisting of errors that are in Gaussian distribution. Therefore, based on previous results, most of the time, it is assumed that the error distribution is Gaussian.

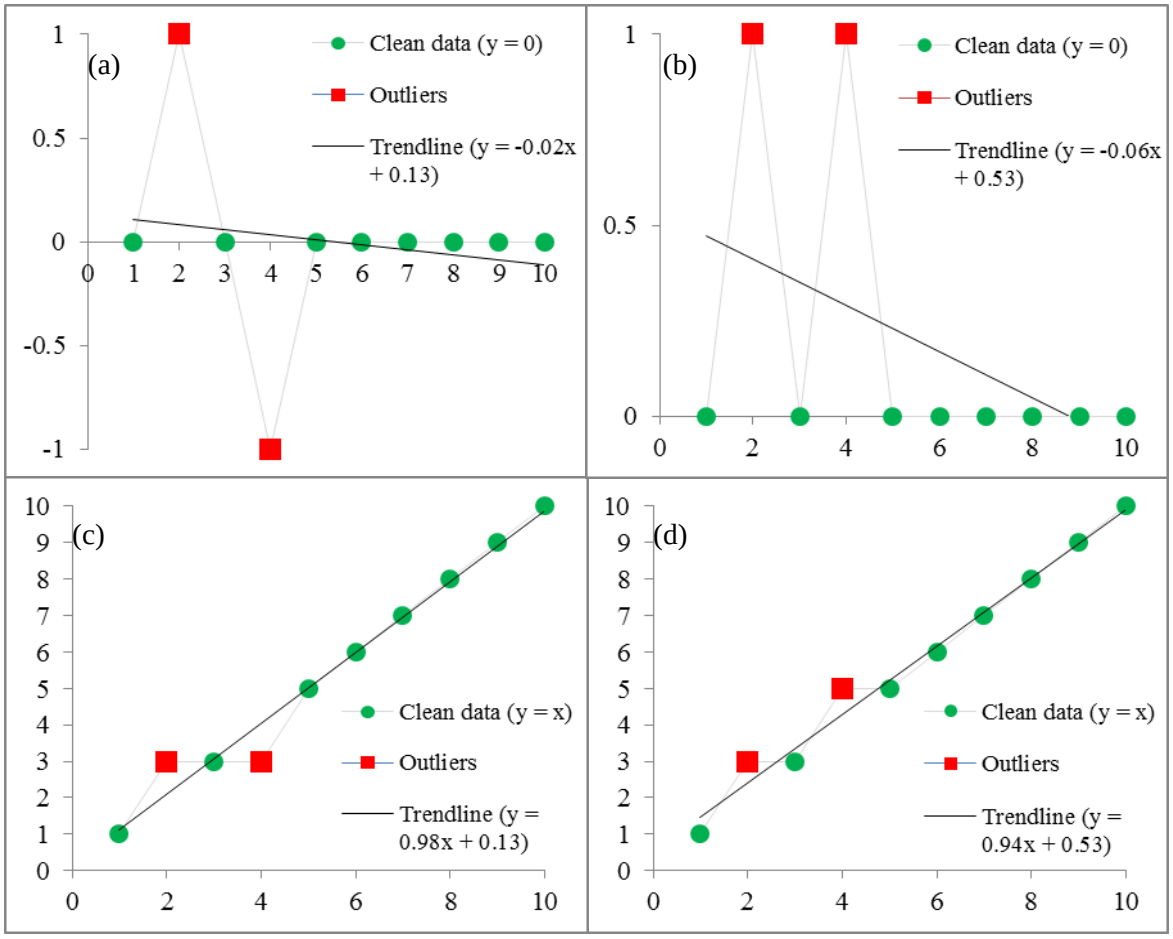


Figure 1.4: Effect of the distribution of outliers. In plots (a) and (b) clean data are expected to follow y = 0 while in plots (c) and (d) clean data are expected follow y = x model. In both plots (a) and (c) outliers are in Gaussian distribution and others are not. When the outliers are in Gaussian distribution, identified linear regression is optimal (plots (a) and (c)).

Impact of number of outliers in the linear fit identification

Theoretically, if the all the data that do not agree with linear fit are in Gaussian distribution (resultant error is zero), the identified linear fit is optimal. However, the degree of optimization depends on the number of outliers. Figure 1.5 shows the same linear regressions shown in Figure 1.4. . Each plot in Figure 1.4 contains 20% outliers and each plot shown in Figure 1.5 contains 40% outliers. The first major observation is that the identified regression in relevant plots in Figure 1.4 and Figure 1.5 are not the same. The reason for this difference is the difference in the

percentage of outliers. This clearly demonstrates that the number of error data causes an impact on detection of the linear fit.

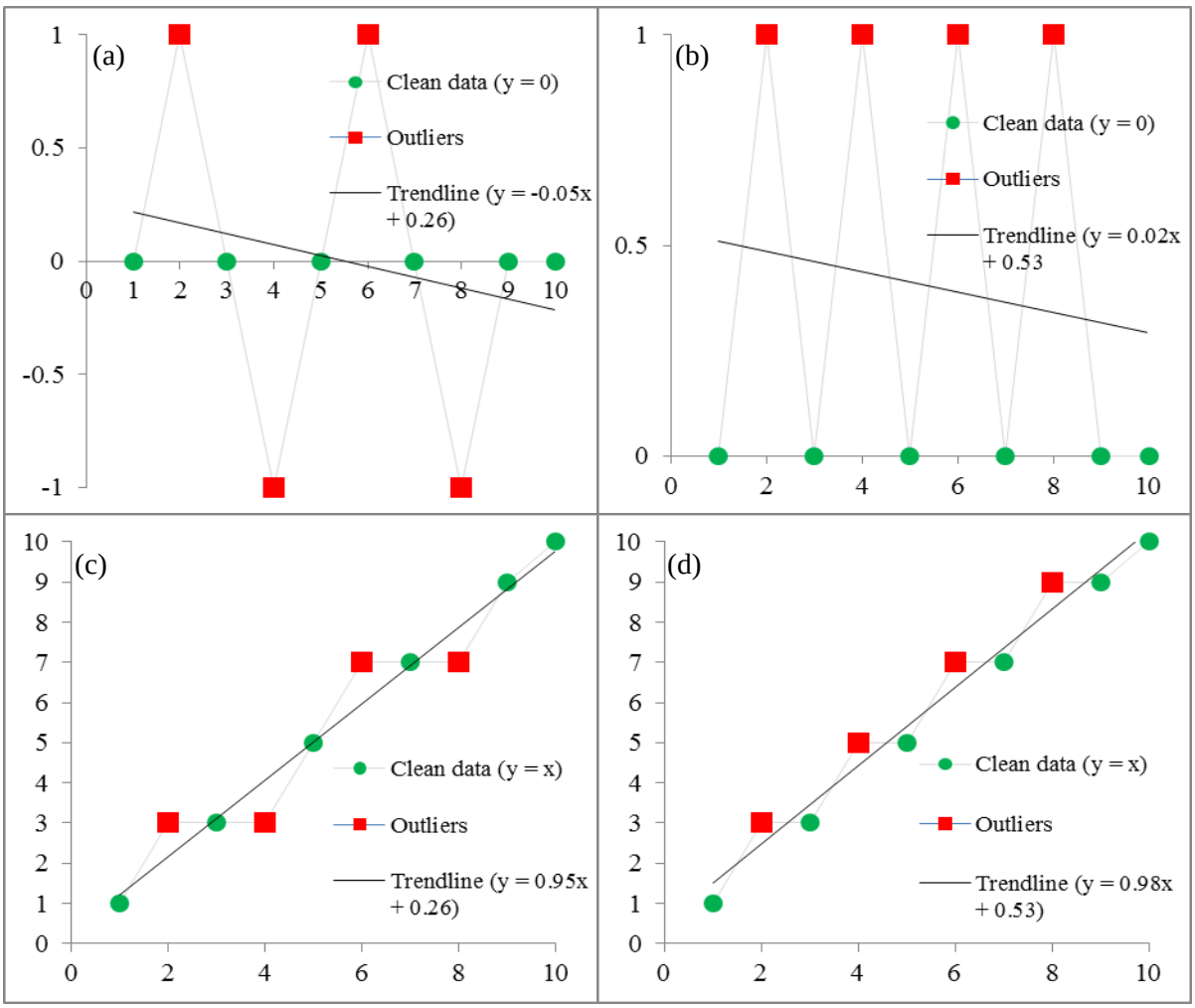


Figure 1.5: Effect of the number of outliers. In plots (a) and (b) clean data are expected to follow y = 0 while in plots (c) and (d) clean data are expected follow y = x model. In both plots (a) and (c) outliers are in Gaussian distribution and others are not. All the plots contain two times of outliers than the plots in Figure 1.4. All the plots show different regression than the equivalent plot in Figure 1.4.

.

Impact of missing data in the linear fit identification

The next challenge in regression analysis and data cleaning is the influence of missing data. Even if the original data set is without missing elements, removing outliers (without replacement) automatically creates a missing data environment. In the literature, considerable numbers of methods were introduced for handling

missing values. Filling and reject missing values are the two techniques used to overcome missing data problems [41, 42]. Among the different missing data filling methods, hot deck, cold deck, mean, median, *k*-nearest neighbours, model-based methods, maximum likelihood methods, and multiple imputations are the most common methods [42-46]. Usually, filling methods derive the filling value from same data set or other known existing data. If there are considerable numbers of outliers, derived data may be biased due to the influence of outliers [47, 48]. Therefore, the most reliable approach is to remove all outliers and replace the outliers with a suitable method, if replacement is necessary.

However, the impact of missing values appears, only when the missing data are clean data and existing data are outliers. Figure 1.6 shows an example for such a situation. Plots (a) and (b) are the data set shown in plots (a) and (b) of Figure 1.4, but without the last five data points. The identified regression do not agree with the regression in the relevant plot in Figure 1.4. This clearly elaborated the impact of missing data on regression identification.

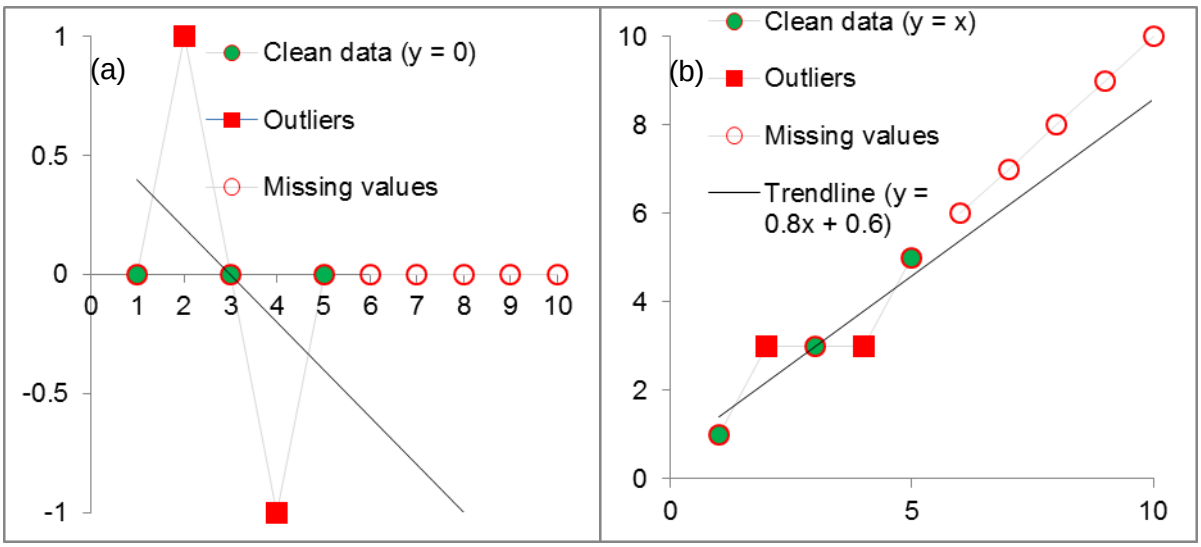


Figure 1.6: Effect of the missing data. Plots (a) and (b) show the data sets in plots (a) and (b) of Figure 1.4 but with the last five data points missing. When the missing data, identified regression does not agree with equivalent plot in Figure 1.4.

Fortunately, GLSM is not affected by missing data when determining linear regression with no outliers. Plots (a) and (b) of Figure 1.7 show such a situation. The identified regression agrees 100% with the regression of clean data. This observation is the one major idea behind the new linear fit identification method. If it

is possible to remove all the data points that do not agree with the regression of clean data, clean linear fit can be achieved. Using this linear fit, very accurate regression can be derived.

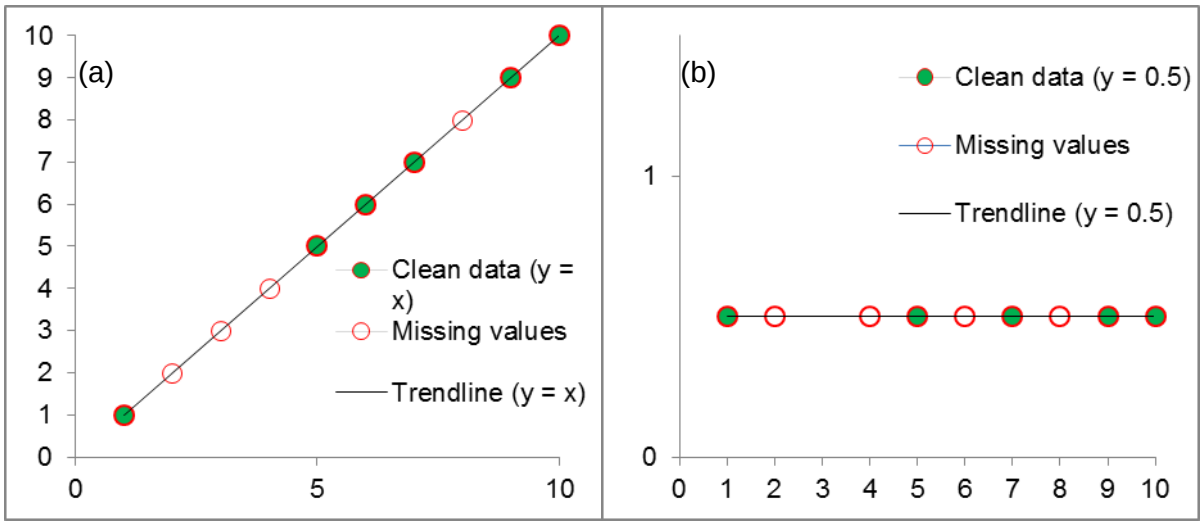


Figure 1.7: Effect of the missing values. If the all remaining data are clean data, there is no effect on linear regression identification with LSM. Plots (a) and (b) contain 40% missing data and all the others are clean data and identified regression agrees 100% with the regression of clean data.

1.4 Extrema detection

In process optimisation, the process of determining extrema plays an important role. Extrema in a signal are used to describe and understand properties of a certain signal. This process is also known as finding out local maxima and minima detection or peaks and valleys detection. Usage of the first and second derivatives is the most common maxima identification method. When the first derivative is zero, it implies that the point has zero gradients (slope). Then this type of point can be a peak, valley or a saddle point. Therefore, only the first derivative is not useful for identifying peaks and valleys. This conflict situation is solved by considering the second derivative. For the second derivative of those points that have zero slopes, there are three situations:

1. If the second derivative is less than zero, the point is considered as a peak.
2. If the second derivative is greater than zero the point is considered as a valley.
3. If the second derivative is equal to zero, the point is considered as a saddle.

Filtering techniques are used to ignore unnecessary peaks and valleys according to the requirements of the considered domain. The magnitude of prominences (height) and the widths at half prominence are two properties of a signal used to filter extrema [49, 50]. Furthermore, baseline correction is another technique used for finding out accurate maxima and minima [51-53]. However, those properties are domain dependent and parametric. Therefore, it is not reliable to define global criteria for filtering extrema.

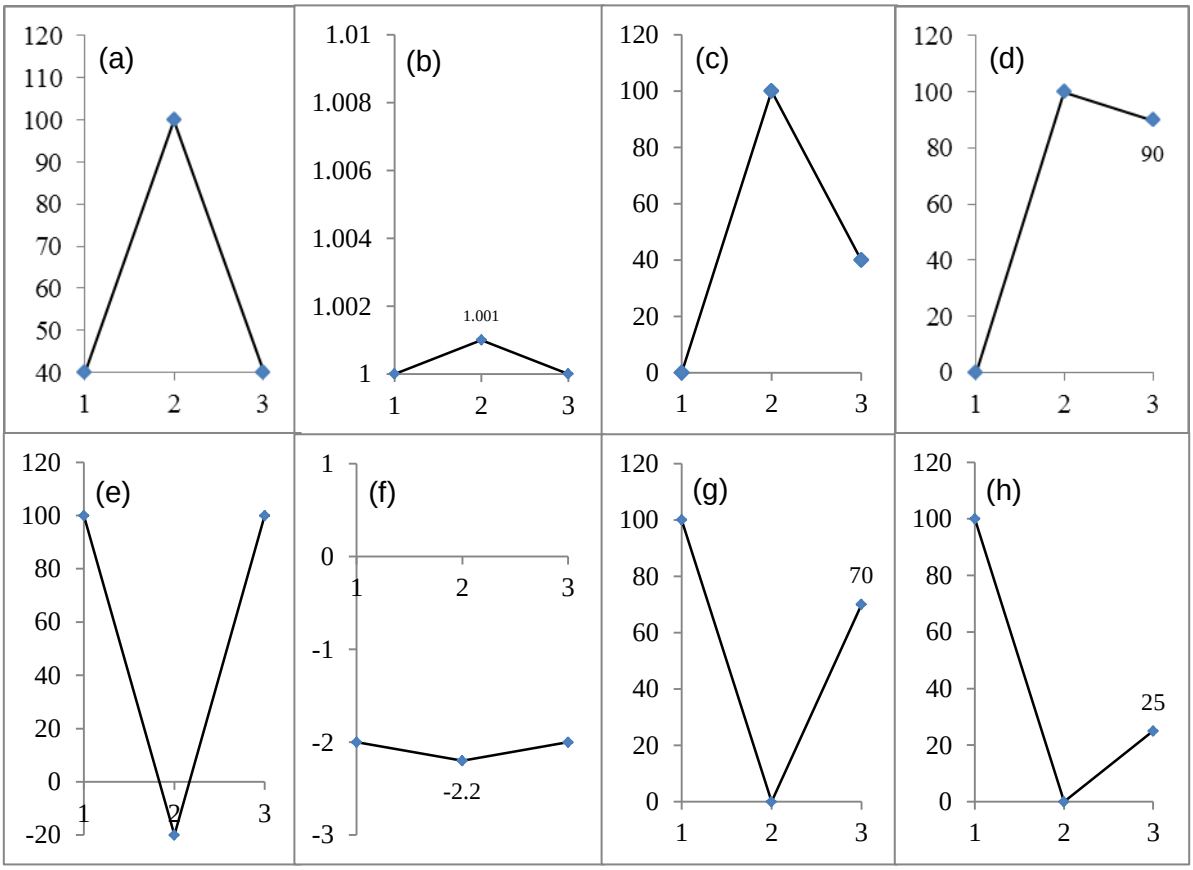


Figure 1.8: Plots (a), (b), (c), and (d) show four different types of peaks and plots (e), (f), (g), and (h) show relevant equivalent types of valleys. Plot (a), (b), (e)and (f) show symatric plots while plots (c), (d), (g) and (h) are asymatric. However, plot (c) and (g)are partialy symatric. The amplitude of plot (a) and (e)are considerably higher than plot (b) and (f).

Plots in Figure 1.8 and Figure 1.9 show different types of extrema that are required to distinguish. Particularly, the height of the extrema (prominence) and the width of the extrema at a certain height of the extrema (e.g.: at half prominence) are the most common properties that used to distinguish those different types of extrema. As

aforementioned these parametric properties are considered as not robust. Therefore, nonparametric methods are necessary, especially for filtering the extrema. The nonparametric method, introduced in this research, is capable of locating and filtering extrema with a high level of robustness.

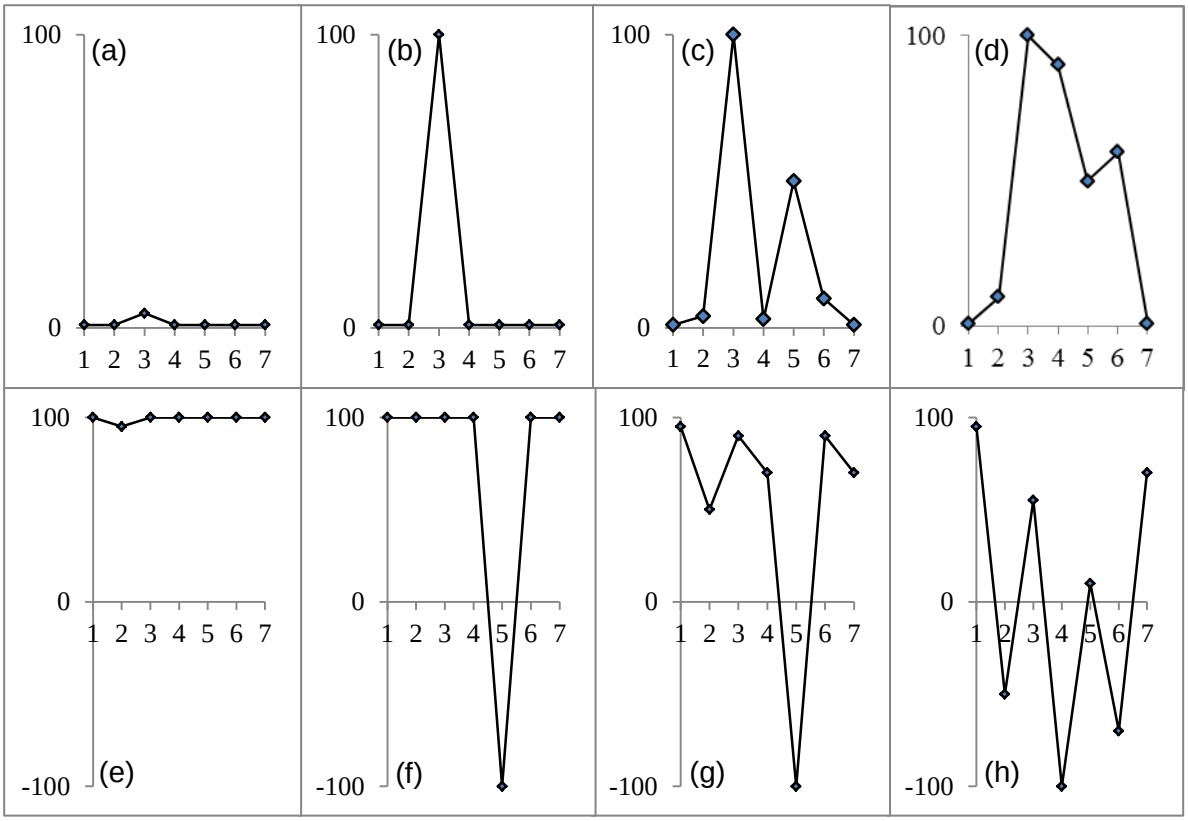


Figure 1.9: Plots (a), (b), (c), and (d) show four different types of peaks and plots (e), (f), (g), and (h) show relevant equivalent types of valleys. Plots (a) and (e) show small extrema while all other plots show higher extrema. However, plots (b) and (f) show sharp extrema while plots (d) and (h) show wide extrema. Combination of the height of the extrema (prominence) and the width of the extrema are the most common approaches used to distinguish those different types of extrema.

1.5 Density clusters in knowledge representation

Knowledge representation (KR) is the key technique applied in the field of artificial intelligence, machine learning [54-57]. Data mining, "Knowledge Discovery in Databases" (KDD), and cluster analysis are some fields that are directly connected with KR. Data mining is the technique that discovers patterns, especially in big data. KDD refers to the basic analysis step of data mining. Representation of those identified information in an understandable structure is the main task of KR. Cluster

analysis or clustering is an unsupervised (*i.e.* it requires no trained data sets) data classification method [58- 60] for identifying homogeneous groups of objects known as clusters [61-63]. Density clustering is one of the most popular techniques, it is a way of identifying and representing already identified knowledge or information [64, 65]. Furthermore, density cluster identification is a tool used to identify the correlation between variables. When dealing with big data clustering, it is used as a clutter reduction technique [66-69]. This technique provides a way of representing big data sets by means of a less number of data points.

In cluster visualisation, first, relevant density clusters were identified by means of a suitable algorithm, usually based on parametric values such as distance or number of neighbours. Then those identified clusters were visualised by means of suitable data visualisation technique. As afore-mentioned, parametric methods are less robust and density cluster analysis methods based on parametric methods suffer from those born drawbacks in parametric approaches. Plot (a) in Figure 1.10 shows scatter plot of 35620 data points with a considerable number of overlaps. The existence of a high number of overlaps implies the existence of high data density areas. Plots (b) and (c) show heat map and contour plot representation of the data shown in plot (a). However, those two methods were unable to provide satisfactory density clusters.

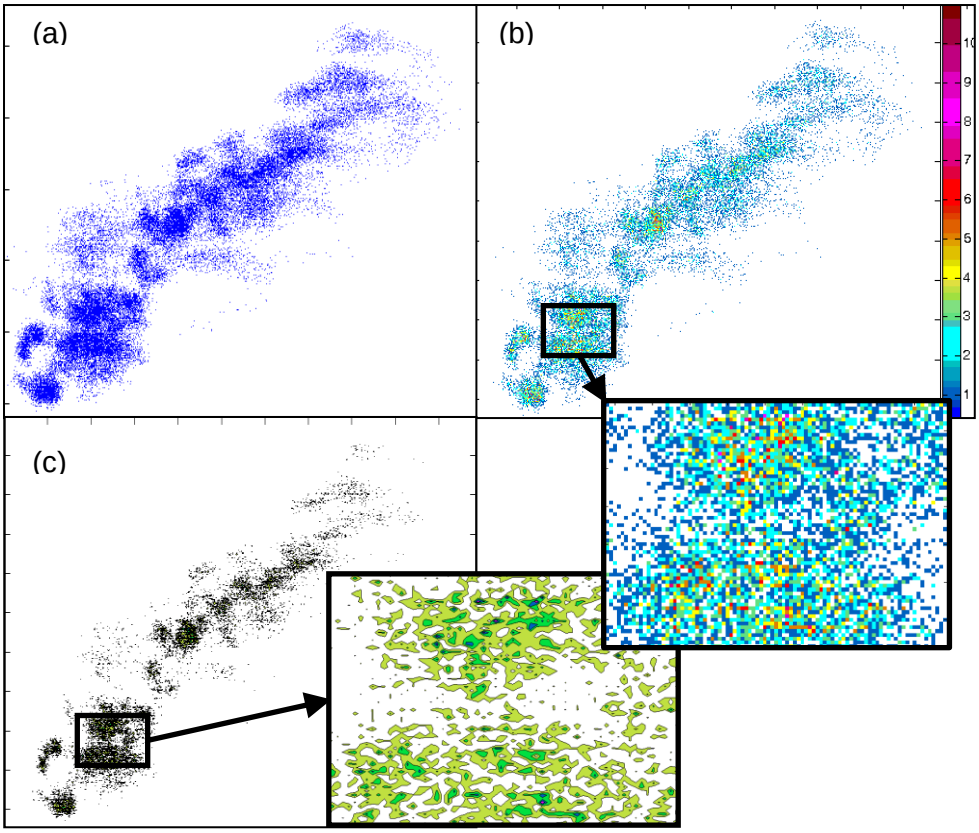


Figure 1.10: Visualization of 35620 data points with (a): scatter plot, (b): heat map and (c): contour plot. The scatter plot shows the distribution of data, whereas the heat map and the contour plot show density clusters. However, compared to the GKU, the heat map and contour plot do not show density clusters.

In the domain of multi-variable, the situation is more complex and representing clusters of multiple variables need special techniques. The multi-variable analysis is a method for depicting the correlation between variables and graphical representation of data, which are two demanding factors in the field of data analysis. Graphical representation of data is a very effective tool for abstracting information in a multi-variable data set. In addition, graphical representation usually conveys the intended information easier than text or numerical values. However, when the number of available data is high, it is difficult to represent all the data points of a scatted data set as a plot due to the high number of overlapping data points; called occlusion or over-plotting. This is one of the main issues in the field of data visualisation, which leads to loss of data in projection [70]. On the other hand, when the number of variables is high, special techniques are required to represent multi-dimensional on a two-dimensional or a three-dimensional plots. The highest challenge is to visualise multi-dimensional big data.

1.6 Thesis outline

The previous chapters gave an introduction to relevant issues in due to usage of parametric methods in the fields of linear fit identification, extrema filtering, and density cluster identification.

In this research, linear fit identification, extrema detection, extrema filtering, and knowledge representation as density clusters were considered. At present, except extrema detection, all other areas are mainly depending on parametric methods. Thus, in this research, novel non-parametric methods were introduced for identifying linear fit, extrema filtering (non-dominating extrema filtering, sharp and gradual extrema filtering, and low and high extrema filtering), and density cluster formation and identification. Furthermore, another novel non-parametric method for extrema identification was introduced. The non-parametric properties used in proposed methods as follows.

1. In the linear fit identification method (UniLiFI: Universal linear fit identification) 2/n is used as the detection criteria, where n is the number of data points. Also, the linear fit identification method is totally independent of data, outlier, and noise distribution models and free of missing/removed data imputation.
2. The extrema detection method is capable of detecting all the local extrema, and the extrema detection is performed after comparing two ratios in relation to maximum, minimum, middle point, and sum.
3. The threshold criteria for methods developed for filtering non-dominating extrema (MMS-Window based filter or MMS-WBF) and sharp and gradual (flat) extrema (MMS-SG filter) are values that based on the number of data points (n). The threshold criterion of the method developed for locating low and high extrema (MMS-LH filter) is a value between 0 and 1.
4. The method named as “Graphical knowledge Unit” (GKU) is a continuous learning knowledge representation method based on properties of the marker (graphical symbol of the data point) such as colour, size, and shape. The GKU is capable of density cluster identification, indicating density of missing data and out of range data, especially with big data. Furthermore, GKU can be used as a data cleaning technique, data visualization technique, and portable database. The GKU is extended for a multivariate version for representing density clusters in multivariate data sets.

2. Summary of results (thesis publications)

2.1 Paper Summary

Paper 1: Universal Linear Fit Identification Method: Independent of Data, Outlier, and Noise Distribution Model and Free of Missing / Removed Data Imputation

Universal Linear Fit Identification (UniLiFI) method is based on the equation of the sum of the elements of finite arithmetic progression (AP) with n elements (Equation (2.1)), which was introduced by Aryabhata [71-73] in 499 CE. Since the date of introduction, Equation (2.1) is used to achieve its original objective. It was unable to find direct applications of the original formula for other objectives. Nevertheless, based on the relation given in Equation (2.1) a method as MMS was developed to locate outliers in linear regression [74].

$$S_n = (n/2) * (a_1 + a_n), \tag{2.1}$$

where S_n is the sum of the elements of finite arithmetic progression with n elements, a_1 is the first element and a_n is the last element of the series.

The UniLiFI method is based on the relation MMS expressed in Equation (2.2) and the transformation method, which is expressed in Equation (2.3). The method MMS, which is shown in Equation (2.2) was developed for outlier detection in data that is expected to follow linear regression [74]. The complete process of deriving Equation (2.2) is shown in Appendix A (*the paper in relation with MMS is not a part of this dissertation*).

In Equation (2.2) there are two ratios as MMS_{max} and MMS_{min} for locating the maximum as the outlier and the minimum as the outlier, respectively.

$$\begin{aligned} MMS_{max} = \frac{a_{max} - a_{min}}{S_n - a_{min} * n} &\begin{cases} > 2/n * (1 + k)\,; Maximum\ is\ the\ outlier \\ \leq 2/n * (1 + k) \end{cases} \\ \text{or} \qquad & \Big\} \ No\ decision \\ MMS_{min} = \frac{a_{max} - a_{min}}{a_{max} * n - S_n} &\begin{cases} \leq 2/n * (1 + k) \\ > 2/n * (1 + k)\,; Minimum\ is\ the\ outlier \end{cases} \end{aligned} \tag{2.2}$$

where a_{max} is the maximum of the series, a_{min} is the minimum of the series, S_n is the sum of all the terms in the series, n is the current number of terms in the series, and k ($k \leq n/2 - 1$) is the weight that determines the level of accuracy.

$$a_{k\mid r}^{TT} = \begin{cases} a_{k\mid r}^{T} - \left(x_{k\mid r}^{T} * \left(Ga_{k\mid r}^{T}/Gx_{k\mid r}^{T}\right)\right); & x_k - x_r \geq 0 \\ -\left(a_{k\mid r}^{T} - \left(x_{k\mid r}^{T} * \left(Ga_{k\mid r}^{T}/Gx_{k\mid r}^{T}\right)\right)\right); & x_k - x_r < 0 \end{cases}, \quad (2.3)$$

where $a_{k\mid r}^{TT}$ is the k^{th} item of the transformed series with reference to the reference point r, $a_{k\mid r}^{T} = a_k - a_r$, $x_{k\mid r}^{T} = x_k - x_r$, x_k is the index of data, a_k is the k^{th} term of the series, (x_r, a_r) is the reference point, k=0,1,…,r,…,n-1, r=0,1,…,n-1, n is the number of elements in current window, r is the index of the reference data point, $Ga_{k\mid r}^{T} = \sum_{k=0}^{n-1} a_{k\mid r}^{T}$,and $Gx_{k\mid r}^{T} = \sum_{k=0}^{n-1} x_{k\mid r}^{T}$.

From Equation (2.2) and Equation (2.3) give the final equation of UniLiFI method as

$$MMS(a^{TT})_{\max\mid r} = \frac{(a_{\max\mid r}^{TT} - a_{\min\mid r}^{TT})}{(S_{n\mid r}^{TT} - a_{\min\mid r}^{TT} * n)} \begin{cases} > 2/n * (1 + k); \\ maximum\ is\ the\ outlier \end{cases}$$

$$\leq 2/n * (1 + k)$$

$$or \qquad \Big\} \ No\ decision \quad (2.4)$$

$$MMS(a^{TT})_{\min\mid r} = \frac{(a_{\max\mid r}^{TT} - a_{\min\mid r}^{TT})}{(a_{\max\mid r}^{TT} * n - S_{n\mid r}^{TT})} \begin{cases} \leq 2/n * (1 + k) \\ > 2/n * (1 + k); \\ minimum\ is\ the\ outlier \end{cases}$$

where $MMS(X)_{\max\mid r}$ is the MMS_{max} with reference to reference point r for data set X, $MMS(X)_{\min\mid r}$ is the MMS_{min} with reference to reference point r for data set X, $S_{n\mid r}^{TT} = \sum_{k=0}^{n-1} a_k^{TT}$; $S_{n\mid r}^{TT} - a_{min}^{TT} * n <> 0$ and $a_{max}^{TT} * n - S_{n\mid r}^{TT} <> 0$.

After applying Equation (2.4) on the transformed data sets with reference to different reference points, produce candidate data sets with different number of data points as clean data (linear fit).

Among those candidate sets, the data set with the highest absolute correlation was selected as the best data set that agree with a linier fit by using Equation (2.5) [75]:

$$r_{xy} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\sqrt{n\sum y_i^2 - (\sum y_i)^2}}, \quad (2.5)$$

where x is the independent and y is the dependent variable.

Paper 2: Derivative Independent, Non-Parametric Local Maxima and Minima Finder with Maxima and Minima Filtering Techniques

The extrema identification method is also in relation with the method MMS that is based on the equation of sum of arithmetic progression [74]. To generate a peak or valley, at least three points are required. For three consecutive positive terms arranging in a line, the ratio (*R*) of the sum of the maximum and the minimum to the sum of three terms is always *2/n*, where *n* is the number of terms and *2/3 ≤ R ≤ 1* when *n=3*. *R>2/3* implies that one term is away from the other two terms. Applying suitable modifications for the Equation (2.2), gives Equation (2.6).

$$MMS_{max} = \frac{a_{max} - a_{min}}{S_3 - a_{min} * 3} \begin{cases} > 2/3 \text{ ; Maximum is away from other two terms.} \\ \leq 2/3 \end{cases}$$

$$\text{or} \qquad \Big\} \text{ No decision} \qquad (2.6)$$

$$MMS_{min} = \frac{a_{max} - a_{min}}{a_{max} * 3 - S_3} \begin{cases} \leq 2/3 \\ > 2/3 \text{ ; Minimum is away from other two terms.} \end{cases}$$

When there is a peak or a valley, for *n=3*, always the middle point is the peak (maximum) or valley (minimum), respectively. Then, if a_{mid} is the middle point of the window, by replacing a_{max} of MMS_{max} and a_{min} of MMS_{min} by a_{mid} gives,

$$MMS_{max|mid} = (a_{mid} - a_{min})/(S_n - a_{min} * n) \text{ and} \qquad (2.7)$$

$$MMS_{min|mid} = (a_{max} - a_{mid})/(a_{max} * n - S_n). \qquad (2.8)$$

When there is a peek,

$$MMS_{max} = MMS_{max|mid} \text{ and} \qquad (2.9)$$

when there is a valley

$$MMS_{min} = MMS_{min|mid} \text{ .} \qquad (2.10)$$

Finally, Equations (2.9) and (2.10) can be used for finding out peaks and valleys, respectively while advancing the window by one data point at a time. The method was named as *MMS max - min finder*.

Furthermore, based on the concept of MMS, another three techniques were developed for filtering non-dominating extrema (MMS-Window based filter or MMS-WBF), sharp and gradual (flat) extrema (MMS-SG filter), and low and high extrema (MMS-LH filter) by considering different ratios of the maximum, the minimum, and the sum of data points in the considered window.

MMS-Window based filter (MMS-WBF) : For filtering non-dominating extrema

For the window sizes greater than three (2.9) and (2.10) identify the most dominating extremum in the considered window. Thus, by increasing the size of the window the usual *MMSmax - min finder* can be used as a technique for filtering non-dominating extrema.

MMS-SG filter: For filtering sharp and gradual (flat) extrema

We showed that the ratios $MMS_{max}/MMS_{min}= n-1$ and $MMS_{min}/MMS_{max}= n-1$. By setting appropriate threshold value t ($0< t \leq n-1$) for the afore-mentioned two ratios sharp and gradual peaks can be filtered.

MMS-LH filter: For filtering low and high extrema

We showed that the ratios $R_{LH\_min} = (a_{min} * n)/ S_n$ and $R_{LH\_max} = n/((a_{max} + 1) * n - S_n)$, where $0<R_{LH\_min} \leq 1$ and $0<R_{LH\_max} \leq 1$. $R_{LH\_min} \rightarrow 1$, implies that the a_{min} has low crater and $R_{LH\_min} \rightarrow 0$ implies that the a_{min} has high crater. Consequently, $R_{LH\_max} \rightarrow 1$ implies that the a_{max} has low prominence and $R_{LH\_max} \rightarrow 0$ implies that the a_{max} has high prominence.

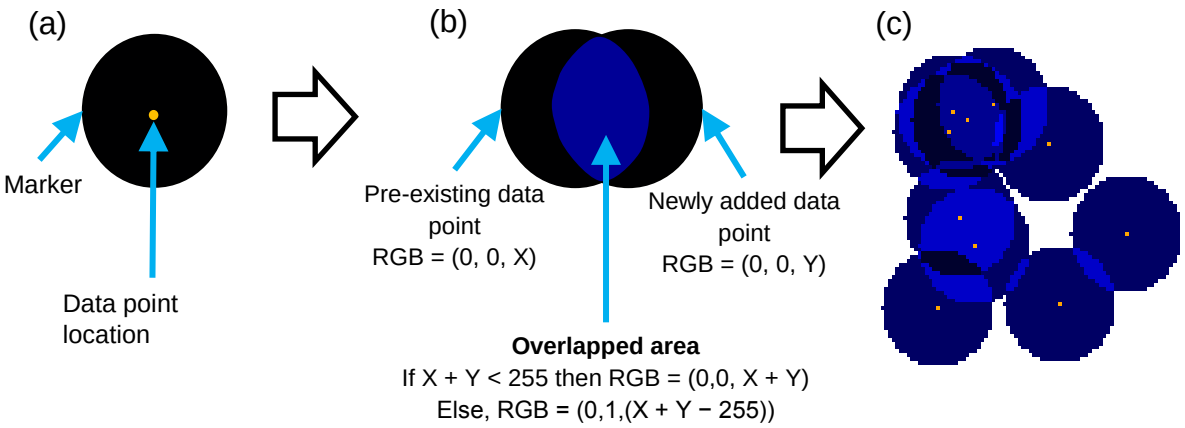
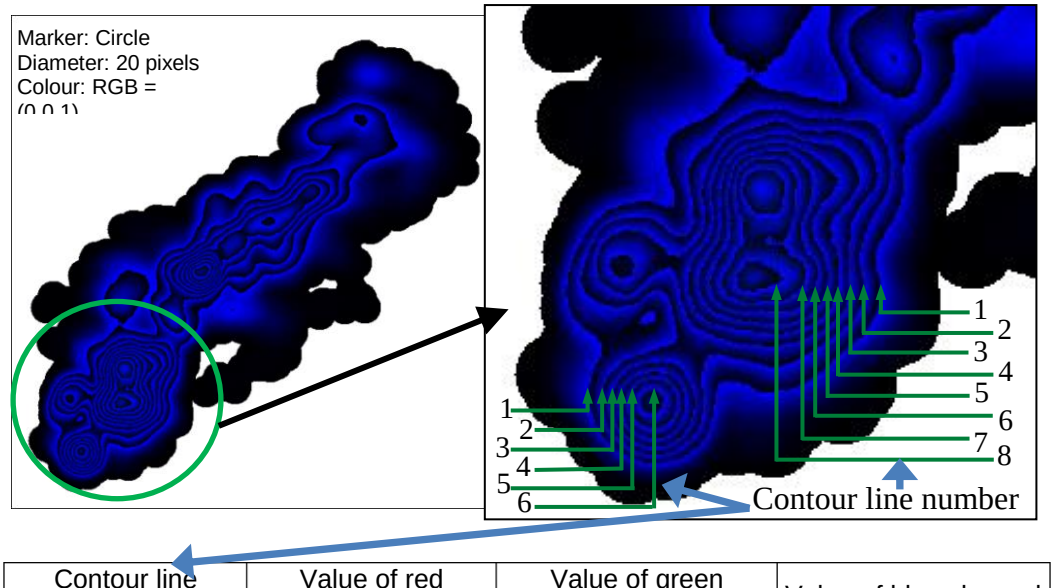


Figure 2.1: Basic idea of the knowledge representation method. (a) The marker is a circle (radius = *n (>1)* pixels), and the RGB colour value of the circle is (0, 0, X), where 0 ≤ X ≤ 255. The center of the circle represents the data point (highlighted), (b) two overlapped markers and (c) several overlapped markers. The data point is represented by the pixel in the center of the marker (the data point is highlighted in orange).

In this paper, a new density cluster formulation was introduced based on the properties of overlapping data points. In the proposed method a data point is represented by a marker (graphical symbol of a data point) that has more than one pixel (e.g.: circle) and by plotting on a bitmap ((a) of Figure 2.1). As the markers overlap, the RGB colour values of pixels in the shared region are added ((b) and (c) of Figure 2.1).This will make the colour of the shared region identical. A higher number of overlap produces the colour of shared region that can be identified by the naked eye ((c) and (d) of Figure 2.1).

The proposed method automatically separates density clusters by automatically generated colour lines that can be considered as contour lines. When the lines were numbered from outside to inside, lines with the same number get the same value for green channel and nearly the same colour value (colour range) for the blue channel (Figure 2.2). This is the way to understand the magnitude of the density in a certain area.



Contour line number	Value of red channel	Value of green channel	Value of blue channel
1	0	0	220–255
2	0	1	220–255
3	0	2	220–255
4	0	3	220–255
5	0	4	220–255
6	0	5	220–255
7	0	6	220–255
8	0	7	220–255

Figure 2.2 : Automatically created contour lines. If the correct maker is selected, GKU automatically creates lines that separate different density areas. These lines are similar to contour lines on a contour map. When the lines were numbered from outside to inside, lines with the same number get the same value for green channel and nearly the same colour value (colour range) for the blue channel.

Paper 4: Multi-Variable, Multi-Layer Graphical Knowledge Unit (MVML-GKU) for Storing and Representing Multi-Dimensional Big Data in Two-Dimensional Plots.

In this paper, a method to represent multiple variables using the concept of GKU is introduced. The usual GKU is capable of representing one dependent variable against one independent variable. In GKU whole bit series was used for a single variable. However, it is possible to split the whole bit series into different sections and assign each section for different variables. Figure 2.3 shows an example of such a situation. Here, the whole series were divided into four 8-bit slots and four variables were assigned for each slot. Now the number of bits per variable is limited to eight bits and number of possible overlaps are 2^8. These numbers of overlaps are not enough. As a solution for this, multi-layers of bitmaps were used (Figure 2.3).

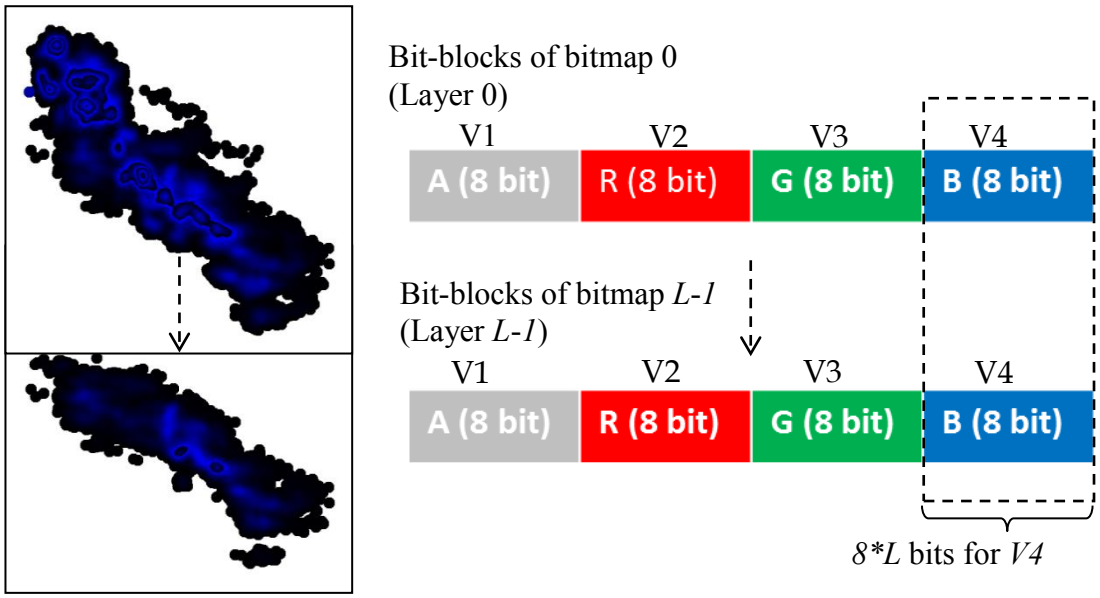


Figure 2.3: Four variables are (V1, V2, V3 and V4) represented using multiple bitmaps of 32-bit RGB pixel format for forming MVML-GKU. Each pixel is divided into four equal portions, where each portion consists of eight bits. When the 32-bit RGB pixel format is divided into four equal parts, each variable represents alpha, red, green and blue sections of the pixel. This provides a vertical array of *k*L* bits for a single variable.

PLOS ONE

RESEARCH ARTICLE

Universal Linear Fit Identification: A Method Independent of Data, Outliers and Noise Distribution Model and Free of Missing or Removed Data Imputation

K. K. L. B. Adikaram[1,2,3]*, M. A. Hussein[1☯], M. Effenberger[2☯], T. Becker[4]

1 Research Group of Bio-Process Analysis Technology, Technische Universität München, Weihenstephaner Steig 20, 85354 Freising, Germany, **2** Institute for Agricultural Engineering and Animal Husbandry, Bavarian State Research Center for Agriculture, Vöttinger Straße 36, 85354 Freising, Germany, **3** Computer Unit, Faculty of Agriculture, University of Ruhuna, Mapalana, Kamburupitiy, Sri Lanka, **4** Lehrstuhl für Brau- und Getränketechnologie, Technische Universität München, Weihenstephaner Steig 20, 85354 Freising, Germany

☯ These authors contributed equally to this work.
* lasantha@daad-alumni.de

OPEN ACCESS

Citation: Adikaram KKLB, Hussein MA, Effenberger M, Becker T (2015) Universal Linear Fit Identification: A Method Independent of Data, Outliers and Noise Distribution Model and Free of Missing or Removed Data Imputation. PLoS ONE 10(11): e0141486. doi:10.1371/journal.pone.0141486

Editor: Xiaosong Hu, University of California Berkeley, UNITED STATES

Received: July 9, 2015

Accepted: October 8, 2015

Published: November 16, 2015

Funding: The authors are grateful to the German Academic Exchange Service (Deutscher Akademischer, DAAD) for providing a scholarship to KKLB Adikaram during the research period.

Competing Interests: The authors have declared that no competing interests exist.

Abstract

Data processing requires a robust linear fit identification method. In this paper, we introduce a non-parametric robust linear fit identification method for time series. The method uses an indicator $2/n$ to identify linear fit, where n is number of terms in a series. The ratio R_{max} of $a_{max} - a_{min}$ and $S_n - a_{min}*n$ and that of R_{min} of $a_{max} - a_{min}$ and $a_{max}*n - S_n$ are always equal to $2/n$, where a_{max} is the maximum element, a_{min} is the minimum element and S_n is the sum of all elements. If any series expected to follow $y = c$ consists of data that do not agree with $y = c$ form, $R_{max} > 2/n$ and $R_{min} > 2/n$ imply that the maximum and minimum elements, respectively, do not agree with linear fit. We define threshold values for outliers and noise detection as $2/n * (1 + k_1)$ and $2/n * (1 + k_2)$, respectively, where $k_1 > k_2$ and $0 \leq k_1 \leq n/2 - 1$. Given this relation and transformation technique, which transforms data into the form $y = c$, we show that removing all data that do not agree with linear fit is possible. Furthermore, the method is independent of the number of data points, missing data, removed data points and nature of distribution (Gaussian or non-Gaussian) of outliers, noise and clean data. These are major advantages over the existing linear fit methods. Since having a perfect linear relation between two variables in the real world is impossible, we used artificial data sets with extreme conditions to verify the method. The method detects the correct linear fit when the percentage of data agreeing with linear fit is less than 50%, and the deviation of data that do not agree with linear fit is very small, of the order of $\pm 10^{-4}$%. The method results in incorrect detections only when numerical accuracy is insufficient in the calculation process.

Introduction

Usage of parametric statistical methods to identify the behaviour of data is a topic for debate. In 1973, Francis Anscombe demonstrated that it is possible to have nearly identical statistical properties even with data sets that have considerable variation when graphed [1]. The four data sets used to show this phenomenon are known as Anscombe's quartet [1]. Furthermore, Anscombe demonstrated the importance of the effect of outliers on statistical properties. Despite the distribution dissimilarities of the data sets of Anscombe's quartet, the linear regression of all four data sets is the same. This implies that the statistical approach might not always identify the correct regression owing to the influence of outliers. There are four factors influencing regression detection: outliers and noise [2–6]; the nature of the distribution of the clean data, noise and outliers [7,8]; the number of outliers and the amount of noise [9–12] and missing data [13,14]. Of these four factors, three factors are related to outliers and noise.

In any domain, clean data are the data that follow the assumed data distribution model [15], while noise is the data that follow the assumed probability distribution [15,16]. Outliers are data that are not in agreement with the assumed clean data and noise models. In this paper, we consider the linear model $y = mx + c$, where m is the gradient and c is the intercept. When the aforementioned definition is applied to the linear model, clean data are the data that agree with the model. Noise can be defined as the data that are within a particular tolerance (e.g. ±x% from the correct value). Outliers are the data that agree with neither clean data nor noise. The most common approach is to remove outliers and noise with reference to the assumed models and then perform the regression analysis. Thus, outlier detection, noise detection and determination of regression of clean data are considered as separate, independent tasks. However, in our method, there is no separate regression analysis for locating linear fit. We first remove outliers and then remove noise using the same method, but with different weight parameters. Finally, the remaining data are the data that agree with linear fit.

Each outlier and noise detection method has a particular level of accuracy. Therefore, it is impossible to guarantee total outlier- and noise-free data. As a consequence, if the cleaned data still contain outliers and/or noise, the detected regression can be incorrect. The accuracy of the outlier detection and noise-removing methods depends on the distribution nature of the outliers and noise, the number of outliers and the amount of noise, which are dependent on the assumed data model. The most common model is the Gaussian distributions. Incorrect determination of the model will cause incorrect detection of outliers and noise. In other words, the accuracy of the selected method is totally dependent on the underlying models. Usually, outliers and noise have been removed, and regression is determined in accordance with the assumed regression model and remaining data that considered as clean data. The major drawback of this approach is that the determined regression is already affected by the influence of outliers and noise models.

The number of outliers and the amount of noise existing in a data set are critical factors when detecting outliers or noise. Especially when detecting outliers, their number plays a critical role. In addition, in a real-world data set, it is very common to have more than one outlier. Therefore, a robust outlier detection method must be capable of detecting multiple outliers. A large number of multiple outlier detection techniques have been proposed to accomplish this aim [17–19]. There are two phenomena, masking and swapping, that have a negative impact on the robustness of the outlier detection process [17]. Masking classifies detection of an outlier as a non-outlier, while swapping classifies detection of a non-outlier as an outlier.

As mentioned above, missing data imputation is another challenge. There are different methods for missing data imputation. These methods are also domain dependent, and there is no guarantee of accuracy when the data are not in accordance with the assumed models. The

method we introduce in this paper is totally independent of missing data or removed data imputation.

Our method is based on the sum of the elements of a finite arithmetic progression (AP), which was introduced by Aryabhata (476–550 CE) [20,21]. Aryabhata was one of the greatest early mathematicians and astronomers [20,21] of India. In 499 CE, he introduced a method for calculating the sum of the elements of a finite AP or arithmetic sequence with n elements [22]. In 2014, we showed that Aryabhata's equation for the sum of an AP can be used as a non-parametric method for detecting outliers in linear regression [23]. The method uses a single point as a reference point, and all detections are conducted with reference to this selected reference point. The method involves two steps, minimum-maximum-sum (MMS) and enhanced MMS (EMMS). MMS is used to remove all significant outliers one by one. Removing an outlier using MMS necessitates recalculating the entire series. After the removal of significant outliers, EMMS is used to remove non-significant outliers. EMMS uses a transformation technique before performing the detection of further outliers. MMS and EMMS are capable of locating outliers correctly when the reference point is not an outlier. When the reference point is an outlier, the method reports incorrect identifications of both outliers and non-outliers. This major drawback resulted in MMS and EMMS being unreliable for identifying outliers. Consequently, using MMS and EMMS jointly did not provide a reliable and robust method for determining linear fit.

Using an improved version of the same methodology, we were able to develop the method presented in this paper for determining linear fit. We expected outlier and noise detection and determination of regression to be possible using a single process that is independent of outlier, noise, data models and data imputation. In the existing literature, there is no such method for identifying a particular linear fit that is independent of models. In this paper, we introduce a single method that is capable of determining linear fit; removes outliers and noise; is independent of the distribution properties of clean data, outliers and noise; is independent of missing or removed data; is resistant to very high rates of outliers and noise (e.g. 50%) and yields no incorrect detections (masking or swapping). The method is suitable for time series or any data series that can be considered as or converted to time series. The most interesting feature of this method is that all five critical factors are addressed in one simple method with a very high level of accuracy. For this reason, we named it the Universal Linear Fit Identification (UniLiFI) method.

Methodology

According to Aryabhata [22], the sum of the elements of an AP or arithmetic sequence with n elements is given by

$$S_n = (n/2) * (a_1 + a_2), \tag{1}$$

where a_1 is the first element, and a_n is the last element of the series.

Eq 1 has been used to achieve its original objective since its introduction. We have been unable to find direct applications of the original formula for other purposes. However, we have been able to use Eq 1 to locate outliers in linear regression [23].

An AP is a sequence of numbers (ascending, descending or constant) such that the difference between the successive terms is constant. The n^{th} term of a finite AP with n elements is given by

$$a_n = d * (n - 1) + a_1, \tag{2}$$

where d is the common difference of successive members, and a_1 is the first element of the series.

Eq 2 is a function of n, represents an AP and fulfils the requirements of a line ($y = mx + c$). A straight line is a series without outliers or noise (if there are outliers or noise, the series is not a line). Therefore, any arithmetic series that fulfils the requirements of an AP can be considered a series without outliers or noise.

Eq 1 can be represented as

$$2/n = (a_1 + a_n)/S_n \; ; \; 2/n \leq 1 \; and \; 2 \leq n < \infty. \tag{3}$$

For any AP, the right-hand side (RHS) of Eq 3 is always $2/n$, which is independent of the terms of the series. In other words, if there are no outliers or noise, the value $(a_1+a_n)/S_n$ will always equal $2/n$. Therefore, the value $2/n$ can be used as a global indicator to identify any AP with outliers or noise. There are four facts in connection with Eq 3: 1. for any AP without outliers or noise, the value $(a_1 + a_n)/S_n$ is always $2/n$, which is independent of the terms of the series; 2. the converse of statement 1 is not always true (*i.e.* if the value $(a_1 + a_n)/S_n$ is $2/n$, this does not imply that the series is free of outliers or noise); 3. if the value $(a_1 + a_n)/S_n$ is not $2/n$, then the series always contains outliers or noise; 4. the converse of statement 3 is not always true (*i.e.* if there are outliers or noise, the value $(a_1+a_n)/S_n$ is not always unequal to $2/n$). However, there are still two situations that are always true (statements 1 and 3), enabling us to use Eq 1 for identifying outlier- and noise-free series. In real-world processes, it is impossible to have noise-free data series. Therefore, we ignore the relation in connection with statement 1 and use the relation in connection with statement 3.

Using statement 3, in 2014 we developed a two-step non-parametric method for identifying outliers in linear regression with reference to a single reference data point [23]. The two steps, MMS and EMMS, and their equations are shown as in Eqs 4 and 5, respectively.

If any series expected to follow $y = c$ form consists of data that do not agree with $y = c$ form,

$$MMS = \begin{cases} MMS_{max} = \dfrac{a_{max} - a_{min}}{S_n - a_{min} * n} = \begin{cases} > (2/n + w) \; ; \\ \text{maximum is the outlier} \\ \leq (2/n + w); - \end{cases} \\ MMS_{min} = \dfrac{a_{max} - a_{min}}{a_{max} * n - S_n} = \begin{cases} \leq (2/n + w); - \\ > (2/n + w) \; ; \\ \text{minimum is the outlier} \end{cases} \end{cases} \tag{4}$$

where a_{max}, a_{min}, S_n, n, and w are the maximum term of the series, the minimum term of the series, the sum of all terms of the series, the number of terms of the series, a weight where

$0 \leq w \leq 1 - 2/n$ and $R_w = 2/n + w$, respectively.

$$EMMS = \begin{cases} EMMS_{max} = \dfrac{(a_{max}^{TT} - a_{min}^{TT})}{(S_n^{TT} - a_{min}^{TT} * n)} = \begin{cases} > (2/n + w); \\ \textit{maximum is the outlier} \\ \leq (2/n + w); - \end{cases} \\ EMMS_{min} = \dfrac{(a_{max}^{TT} - a_{min}^{TT})}{(a_{max}^{TT} * n - S_n^{TT})} = \begin{cases} \leq (2/n + w); - \\ > (2/n + w); \\ \textit{minimum is the outlier} \end{cases} \end{cases} \quad (5)$$

Where $a_k^{TT} = |a_k^T - x_k * (Ga^T / Gx)|$, $a_k^T = a_k - a_0\, x_k$ is the index of data, a_k is the k^{th} term of the series, $k = 0, 1, \ldots, n-1$, n is the number of elements in the current window, $Ga^T = \sum_{k=0}^{n-1} a_k^T$, $Gx = \sum_{k=0}^{n-1} X_k$, $S_n^{TT} = \sum_{k=0}^{n-1} a_k^{TT} <> 0$, $2/n + w = R_w$, and w is the weight, $0 \leq w \leq 1 - 2/n$.

In the abovementioned method, the first value (a_0) is used as the reference point. Therefore, the method gives correct detections when the first point is not an outlier. Furthermore, MMS is used for removing significant outliers, while EMMS is used for removing non-significant outliers. When using MMS, it is possible to obtain incorrect detections of outliers as the result of selecting a small value for w [23]. The recalculation process in MMS and the transformation used in EMMS provide correct transformations only when the reference point is not an outlier [23]. However, it is impossible to determine the nature of a point in advance.

After considering all drawbacks, we introduced a new method based on the same principle. The new method contains a new transformation technique using multiple reference points, shown in Eq 6. The number of reference points can be in the interval [1, n], where n is the total number of data points in the selected data set. However, the process uses each reference point separately as the reference point and transforms the data with

$$a_{k|r}^{TT} = \begin{cases} a_{k|r}^T - (x_{k|r}^T * (Ga_{k|r}^T / Gx_{k|r}^T)), & x_k - x_r \geq 0 \\ -(a_{k|r}^T - (x_{k|r}^T * (Ga_{k|r}^T / Gx_{k|r}^T))), & x_k - x_r < 0 \end{cases} \quad (6)$$

Where $a_{k|r}^{TT}$ is the k^{th} item of the transformed series with reference to the reference point r, $a_{k|r}^T = a_k - a_r$, $x_{k|r}^T = x_k - x_r$, x_k is the index of data, a_k is the k^{th} term of the series, (x_r, a_r) is the reference point, $k = 0, 1, \ldots, r, \ldots, n-1$, $r = 0, 1, \ldots, n-1$, n is the number of elements in the current window, r is the index of the reference data point, $Ga_{k|r}^T = \sum_{k=0}^{n-1} a_{k|r}^T$ and $Gx_{k|r}^T = \sum_{k=0}^{n-1} x_{k|r}^T$.

The transformation in Eq 6 can convert all data to the form $y = c$ if there are no outliers or noise. If the data set consists of outliers or noise, the transformed data do not agree with the form $y = c$ and MMS can locate the outliers or noise. The form $y = c$ is independent of the occurrence sequence of the data [23]. Therefore, there is no effect from missing or removed data on the outlier detection process. In addition, w of Eq 5 can be expressed as $w = 2 * k / n$, where $0 < k \leq (n/2) - 1$ [23]. If $MMS(D)_{\max|r}$ refers to MMS_{max} with reference to reference point r for data set D and if $MMS(D)_{\min|r}$ refers to MMS_{min} with reference to reference point r

for data set D. Then, Eq 7 provides the application of MMS on a^{TT}.

$$MMS(a^{TT}) = \begin{cases} MMS(a^{TT})_{\max \mid r} = \dfrac{(a^{TT}_{\max \mid r} - a^{TT}_{\min \mid r})}{(S^{TT}_{n \mid r} - a^{TT}_{\min \mid r} * n)} = \begin{cases} > 2/n * (1 + k); \\ \textit{maximum is the outlier} \end{cases} \\ \\ MMS(a^{TT})_{\min \mid r} = \dfrac{(a^{TT}_{\max \mid r} - a^{TT}_{\min \mid r})}{(a^{TT}_{\max \mid r} * n - S^{TT}_{n \mid r})} = \begin{cases} \leq 2/n * (1 + k); - \\ \\ > 2/n * (1 + k); \\ \textit{minimum is the outlier} \end{cases} \end{cases} \quad (7)$$

where $S^{TT}_{n|r} = \sum_{k=0}^{n-1} a^{TT}_k$, $S^{TT}_{n|r} - a^{TT}_{min} * n <> 0$, and $a^{TT}_{max} * n - S^{TT}_{n|r} <> 0$.

After transformation, outliers or noise are detected using Eq 7. If an outlier or noise is detected, it is removed from both transformed and original data sets. Then, the transformation is applied again, and outlier detection is performed until one of the termination conditions is reached. In general, there are three termination conditions: 1. $a^{TT}_{\max \mid r} = a^{TT}_{\min \mid r} = 0$; 2. a selected reference point is detected as an outlier; or 3. no more outliers are detected.

Table 1 and S1 File show a complete process cycle for achieving a candidate data set for linear fit, with reference to the second item of the series.

At the end of the process cycle with reference to a particular reference data point, the remaining data set is a candidate data set for linear fit. This process is applied for all selected reference points and yields a candidate data set for linear fit with reference to each reference point. Then, for each candidate data set, the linear correlation is calculated using

$$r_{xy} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\sqrt{n\sum y_i^2 - (\sum y_i)^2}}, \quad (8)$$

where x is the independent variable and y is the dependent variable.

A correlation with $1 \geq |r_{xy}| \geq 0.8$, $0.6 \geq |r_{xy}| > 0.8$, $0.3 \geq |r_{xy}| > 0.6$ or $0.0 \geq |r_{xy}| > 0.3$ is generally described as very strong, moderately strong, fair or poor correlation [24], respectively. The abovementioned intervals are true for the linear relations that have $y = mx + c$ form, only when $m \neq 0$, where c is a constant. When $m = 0$, the linear relation is of the form $y = c$. However, when $y = c$, $n\Sigma x_i y_i - \Sigma x_i \Sigma y_i = 0$ (numerator of (Eq 8)) and $n\sum x_i^2 - (\sum x_i)^2 = 0$ (part of denominator of Eq 8), r_{xy} becomes undefined. This situation prevents identification of linear fits that have the form $y = c$. Therefore, when $\Sigma x_i y_i - \Sigma x_i \Sigma y_i = 0$ and $n\sum x_i^2 - (\sum x_i)^2 = 0$, we stipulate that $r_{xy} = 1$.

When considering the accuracy of Eq 8, a data set with fewer points satisfying the equation will provide better correlation than the best resultant linear fit data set leading to a wrong decision. For example, a data set with two points will give the best correlation ($|r_{xy}| = 1$) despite the best fitting. Therefore, we define a minimum number of data points that must be in the final linear fit. The best linear fit is defined as the data set with the maximum absolute correlation ($|r_{xy}|$) and the minimum number of data points. These two criteria can be used in different ways to determine the best linear fit depending on the requirements. The decision diagrams elaborated in Figs 1 and 2 express two different implementation methods of the new multiple reference point linear fit algorithm.

Table 1. A complete process circle for achieving a candidate data set for linear fit with reference to the second item (30) of the data set. Detection process must be conducted considering each term as a reference point. However, in this example shows calculations only with reference to the second item. In the first iteration $MMS(a^{TT})_{max|2} > 2/n$ and fulfils the detection condition. Thus, in the first iteration $a^{TT}_{max|2}$ is the term that not agrees with the linear fit. Therefore, (8, 41.81) was removed and excluded from the calculations in second iteration. This process was continued until the termination condition ($a^{TT}_{max|2} = 0$ and $a^{TT}_{min|2} = 0$) is reached in fourth iteration. Note that in this example, $k = 0$ and $r = 2$. Also, see S1 File for better understanding on the calculation process.

		Iteration 1			Iteration 2			Iteration 3			Iteration 4		
X	a	$x^T_{k\|2}$	$a^T_{k\|2}$	$a^{TT}_{k\|2}$	$x^T_{k\|2}$	$a^T_{k\|2}$	$a^{TT}_{k\|2}$	$x^T_{k\|2}$	$a^T_{k\|2}$	$a^{TT}_{k\|2}$	$x^T_{k\|2}$	$a^T_{k\|2}$	$a^{TT}_{k\|2}$
6*[2]	22*[2]	-1	-8.000	-2.42	-1	-8.000	-2.25	-	-	-	-	-	-
7	30**	0	0.000	0.00	0	0.000	0.00	0	0.000	0.0E+0	0	0	0
8*[1]	41.81*[1]	1	11.810	1.39	-	-	-	-	-	-	-	-	-
9*[3]	50.001*[3]	2	20.001	-0.85	2	20.001	-0.50	2	20.001	7.8E-4	-	-	-
10	60	3	30.000	-1.27	3	30.000	-0.75	3	30.000	-3.3E-4	3	30	0.0
11	70	4	40.000	-1.69	4	40.000	-1.00	4	40.000	-4.4E-4	4	40	0.0
	Sum	9	93.811		8	82.001		9	90.001		7	70	
			$a^{TT}_{max\|2}$	1.39‡			0.00			7.8E-4‡			0.0
			$a^{TT}_{min\|2}$	-2.42			-2.25‡			-4.4E-4			0.0
			n	6			5			5			4.0
			$S^{TT}_{n\|2}$	-4.85			-4.50			0.00			0.0
			$R_k = 2/n$	0.33			0.40			0.40			0.5
			$MMS(a^{TT})_{max\|2}$	0.39	*(>0.33)*		0.33			0.55	*(>0.40)*		-
			$MMS(a^{TT})_{min\|2}$	0.29			0.50	*(>0.40)*		0.31			-

Legend:

**: Reference data point.

‡: Term identified as the outlier in the relevant iteration.

*[x]: Removed in the relevant iteration and not considered for the next iteration.

doi:10.1371/journal.pone.0141486.t001

Using this method, the data points that do not agree with linear fit can be categorized into several categories using different R_k values based on different k values, in several steps. After identifying different k values, the data with the highest k (or highest R_k) value are first checked, and then the cleaned data are used as input for the next step with the next highest k value. Fig 3 elaborates the implementation of the multi-step multiple reference linear fit algorithm, based on the first method elaborated in Fig 1. The second method elaborated in Fig 2 can be improved for locating linear fit while grouping data that do not agree with linear fit using the same technique.

To check the best linear fit, the algorithm was tested using several synthetic and real data sets based on zero-based numbering (the first term of a series is assigned the index 0). Among the artificial data sets, the first three data sets of Anscombe's quartet [1] can be considered time series. The real data were from biogas plants and were automatically recorded with a frequency of 12 data points per day (*i.e.* every other hour) over a period of seven months. With real data, it is impossible to find a perfect linear relation between two variables. Nevertheless, among the different parameters, we selected the NH_4^+ content measured in g/kg of fresh matter, which we expected to maintain linear behaviour during stable operation. We selected seven segments of different sizes for evaluating the algorithm. In some data sets, there were initial missing elements. Performance of the new method was evaluated using a linear regression model and MMS/EMMS.

Results and Discussion

We used synthetic data sets with different sizes (4 to 1,000 points) and real data sets for evaluating our new linear fit identification method. Here, we include some data sets with extreme

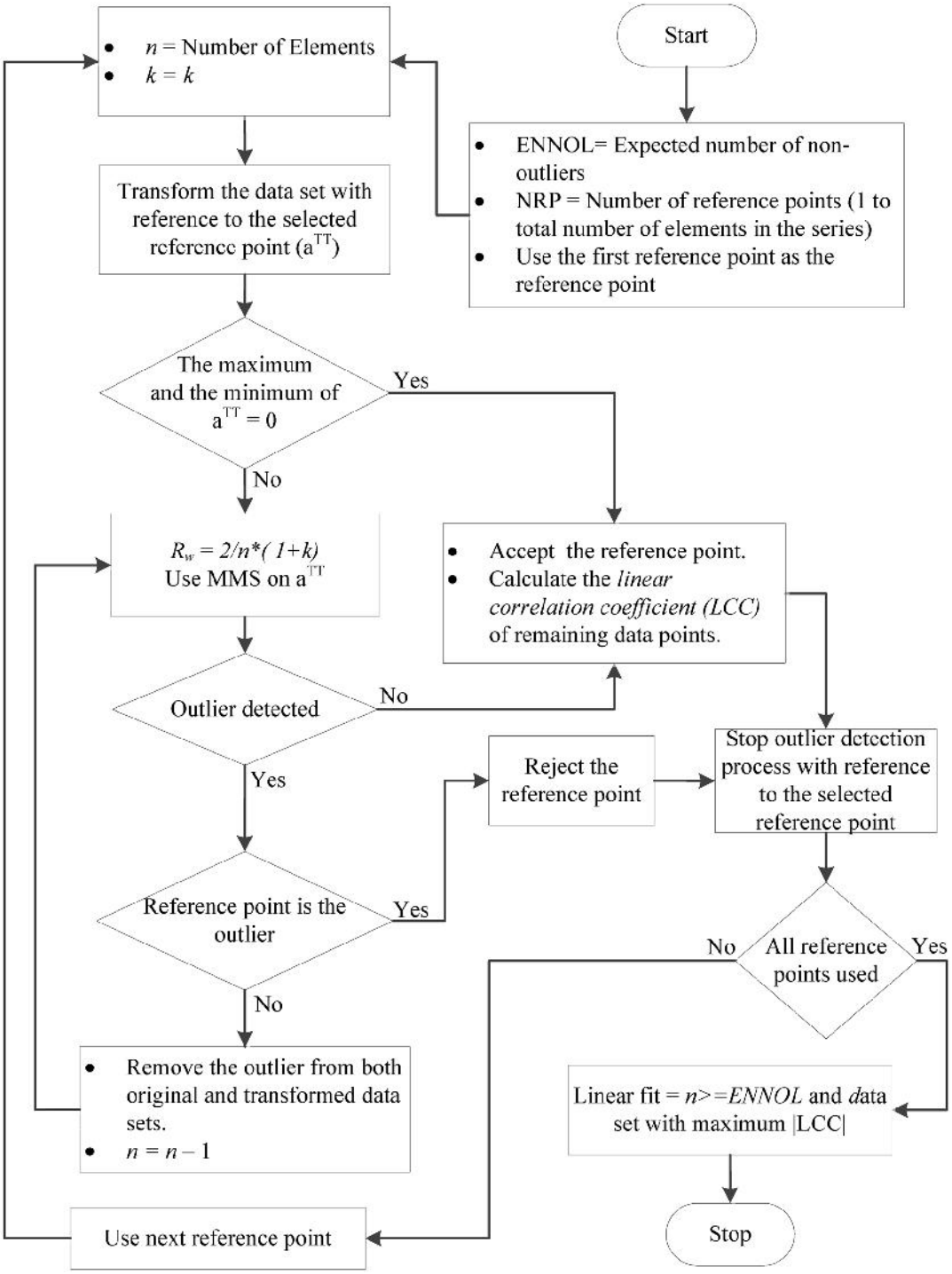


Fig 1. The first method of applying the new multiple reference point linear fit algorithm. When terminating conditions are fulfilled with reference to a particular reference point, outlier detection is terminated. Then, the process continues with the next reference point until all reference points are finished. Among the different candidate linear fits in relation to different successful reference points, the best linear fit is determined by considering the linear correlation coefficient and the number of data points.

doi:10.1371/journal.pone.0141486.g001

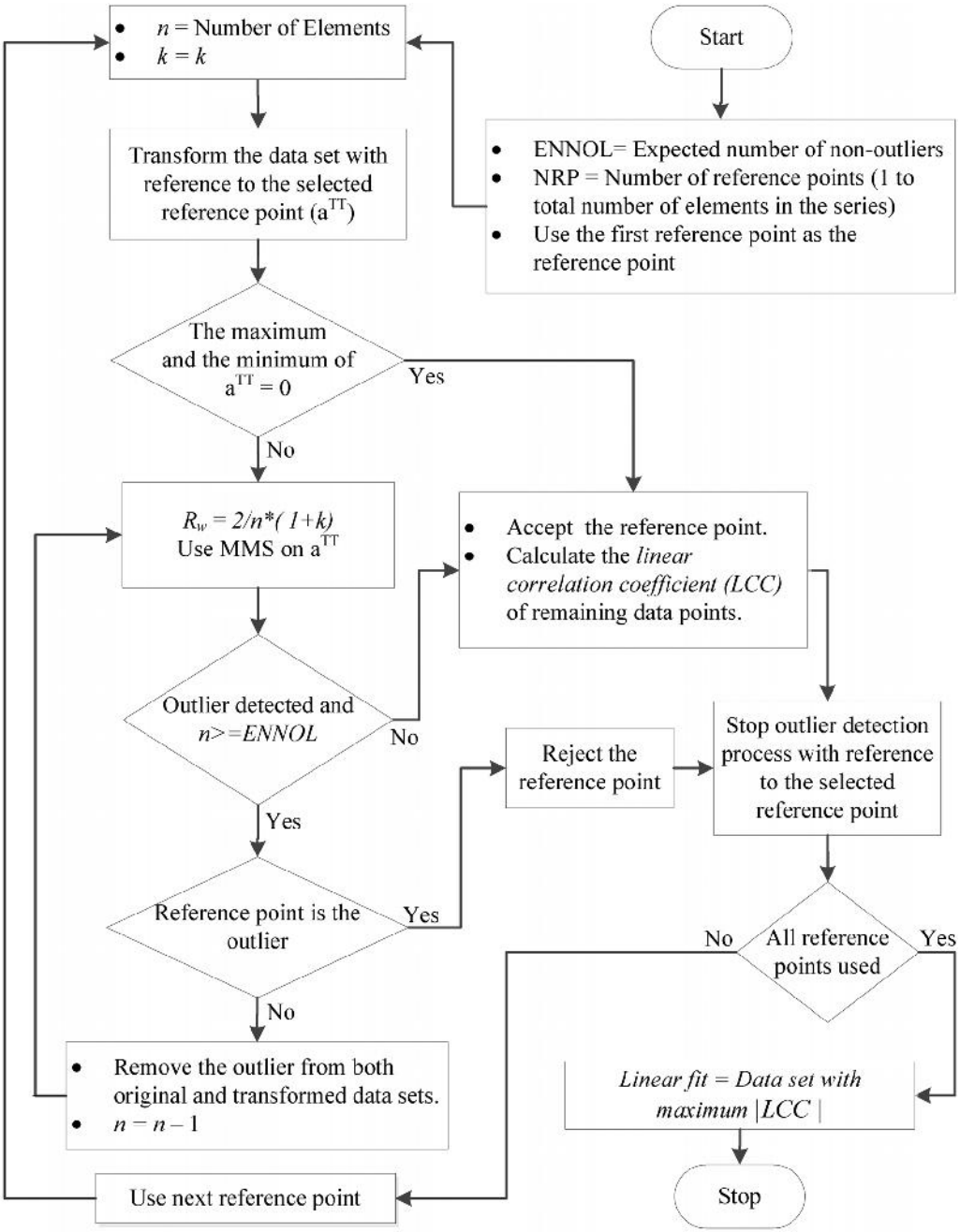


Fig 2. The second method of applying the new multiple reference point linear fit algorithm. In this method, the expected number of non-outliers (ENNOL) is used as a termination condition. When terminating conditions are fulfilled with reference to a particular reference point, outlier detection is terminated. Then, the process continues with the next reference point until all reference points are finished. Among the different candidate linear fits in relation to different successful reference points, the best linear fit is determined by considering the linear correlation coefficient.

doi:10.1371/journal.pone.0141486.g002

conditions. Fig 4 shows six data sets, each consisting of 10 data points, and the data sets that agreed with linear fit have either a positive gradient, a negative gradient or a constant value. In all data sets, fewer than 50% of the data points agreed with linear fit. Some of the data not in agreement with linear fit deviated more than $\pm 10^4$ from the correct value. At the same time,

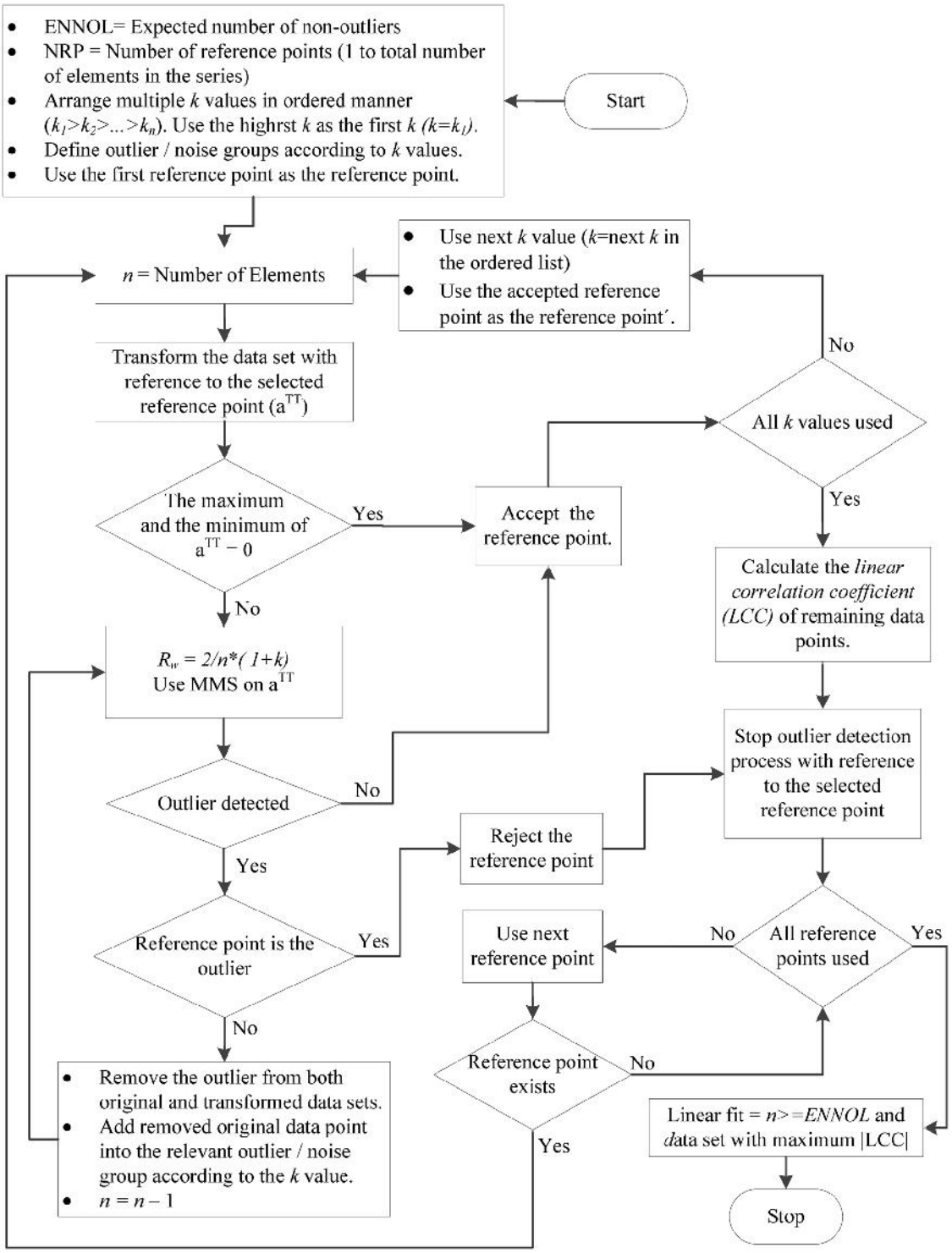


Fig 3. Improved version of the first method shown Fig 1 for grouping outliers or noise into several groups based on different *k* values. The method shown in Fig 2 can also be improved for grouping outliers or noise into several groups in the same manner.

doi:10.1371/journal.pone.0141486.g003

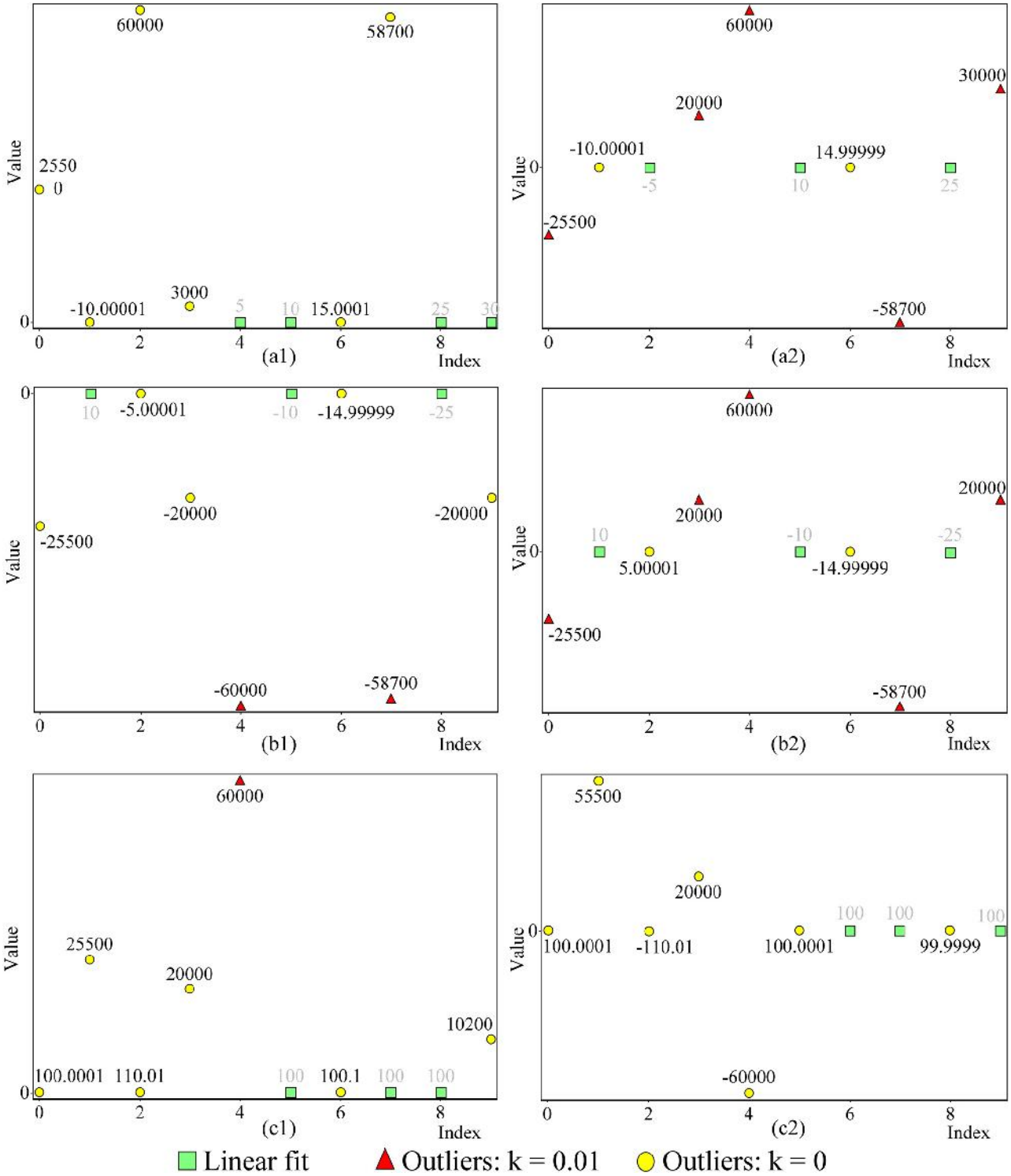


Fig 4. The gradient of linear fits shown in (a1) and (a2), (b1) and (b2) and (c1) and (c2) are ascending, descending and constant, respectively. In data sets (a1), (b1) and (c1), all data points that do not agree with linear fit are located on one side (non-Gaussian) of linear fit. In data sets (a2), (b2) and (c2), all data points that do not agree with linear fit are located on both sides of linear fit. In all data sets, fewer than 50% of the data points agree with linear fit. Some of the data not agreeing with linear fit deviate more than $\pm 10^{4}$ from the correct value. At the same time, there are data points that have very small deviation, as small as $\pm 10^{-4}$, from the correct value. Whatever the condition, the new method was capable of identifying robust linear fit. In all plots, the

reference point is the first data point in linear fit, which was automatically detected during the detection process (all the points were considered as the reference point). For data set of plots in this figure see S2 File. Fig 5 consists of three data sets of Anscombe's quartet [1], which can be considered as APs. As shown in Fig 5, the new method was capable of identifying the nearest data set that agrees with linear fit. We set the number of minimum data points at five for all examples in Fig 5. In Fig 5(c) and 5(d) represent the third data set of Anscombe's quartet and use different *k* values. When the *k* value changes, the reference point and number of non-outliers are not the same for the same ENNOL. Furthermore, no masking or swapping occurred in relation to any *k* value we used for linear fit identification.

doi:10.1371/journal.pone.0141486.g004

there are data points that have very small deviation, as small as $\pm 10^{-4}$, from the correct value. All data points that did not agree with linear fit in data sets (a1), (b1) and (c1) shown in Fig 4 are located on one side (non-Gaussian) of linear fit. Whatever the condition, the new method was capable of identifying a robust linear fit.

The new method showed its ability to identify linear fit with large window sizes as well. Fig 6 shows two data sets consisting of 1,000 data points, each with less than 50% of the data points agreeing with exact linear fit (regression is unknown). The data points that do not agree with linear fit are in the range of $\pm 10^{-2}$ to $\pm 10^{4}$. In Fig 6, plot (a) bears four initial missing data regions, with 50, 100, 100 and 50 data points (total 300 initial missing data), while plot (b) bears a total of 250 initial missing data, with 100 and 150 missing data regions. In Fig 6 plot (a), the data that do not agree with linear fit lie on both sides of linear fit, while in plot (b), all data points that do not agree with linear fit lie on one side of linear fit.

All mentioned properties above are very extreme conditions. However, the new method identified linear fit with a high level of accuracy. Furthermore, in Fig 6 plot (b), there is a set of data that have a nearly linear relation and makes the situation more extreme. All results prove that the new method is capable of locating all data points that agree with linear fit without masking or swapping. This accuracy cannot be achieved with a conventional least squares method or with MMS/EMMS. Nevertheless, when the deviation of the value of a data point was less than $\pm 10^{-2}$ from its correct value, sometimes we observed 0.5% swapping and masking with the new method. However, this is very rare situation and not the result of a failure of the method but of the limited numerical accuracy of the programming language (Visual C++ 2010) [25]. This is more visible when the number of data points is large and their deviation is very small. Therefore, we recommend using a programming platform with high numerical accuracy for better performance with the new method. Fig 7 shows eight data windows of data captured automatically from a biogas plant. Each window consists of 1,000 data points, with results included in relation to two different conditions. The left side of Fig 7 shows identified linear fits of four different windows. The right side of Fig 7 consists of linear fits of four windows corresponding to those shown on the left side with narrower linear fit identification criteria than on the left side. When considering all eight situations, in plots (a1) and (c1), R_k reaches its limit before ENNOL. Furthermore, in all situations, r_{xy} is greater than 0.8 and implies a very strong linear fit [24]. However, plots on the RHS, which have narrower criteria, showed higher correlation than the corresponding plot on the left side. As in the artificial data sets, with these actual data, there is no swapping or masking. This is a major advantage of this method over any other method. The linear fits in relation to the first data set (plots (a1) and (a2)) do not show any exceptionality and no resistance to acceptance. In contrast, in the second data set (plots (b1) and (b2)), there is a minimum that clearly shows two potential regions for linear fit. On the other hand, in the third data set (plots (c1) and (c2)), there are two regions based on the data density.

In both data sets, the new method was able to locate the best linear fit, which can be identified even visually. In the second data set, the new method omitted one potential area and identified linear fit from the longer half. However, in the third data set, the identified linear fit is from both regions. This shows the ability of the new method to identify the best linear fit without influence from data density and other data in the considered window. In the fourth data

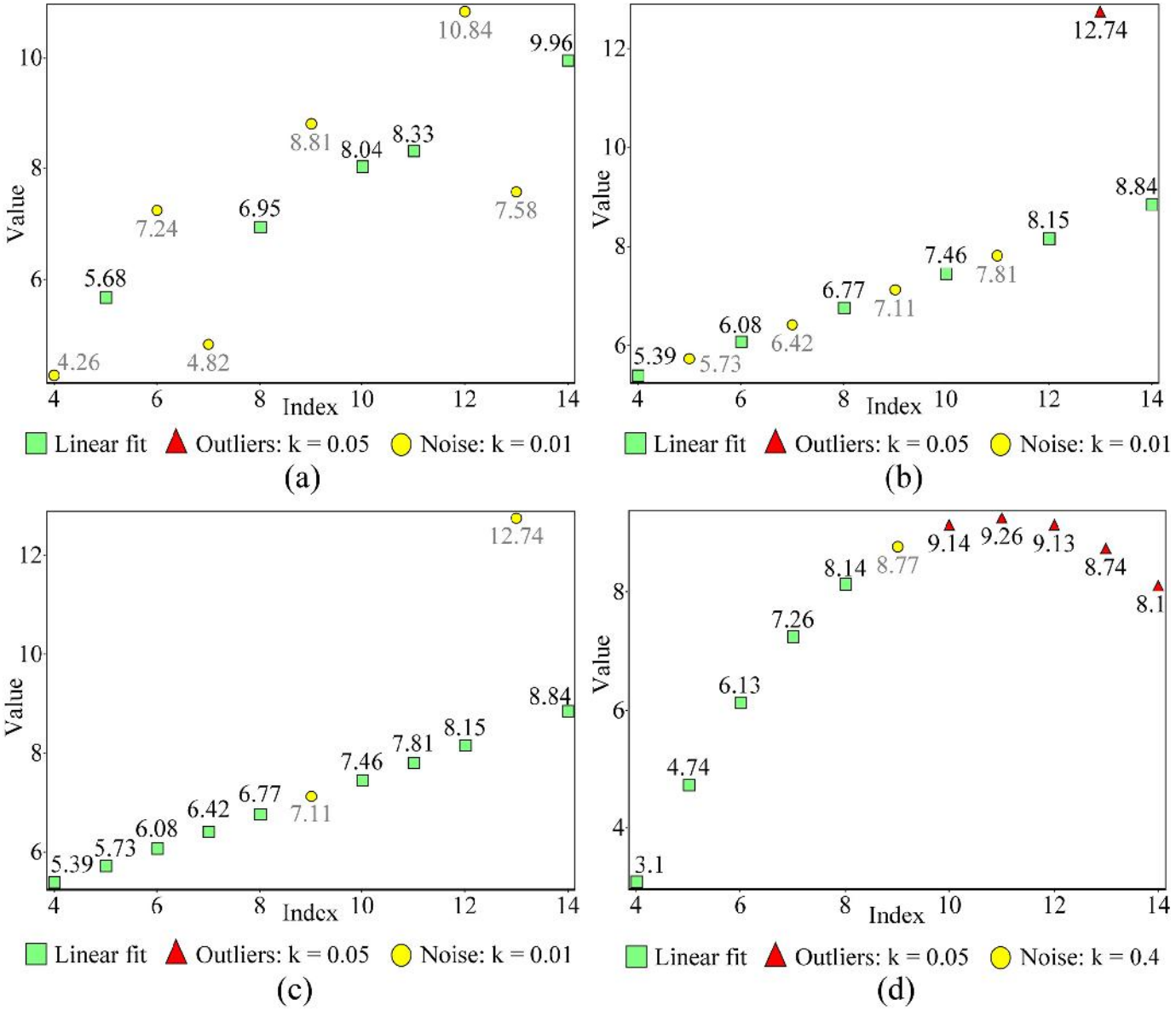


Fig 5. Plots (a) and (b) show the first and second data set of Anscombe's quartet and used the same value of *k*. Plots (c) and (d) represent the third data set of Anscombe's quartet and used different *k* values. In all detections, ENNOL was set to five. When the *k* value changes, the reference point and number of points in linear fit are not the same for the same ENNOL (Plots (c) and (d)). In all plots, the reference point (the first term of the linear fit) was automatically detected during the detection process (all the points were considered as the reference point). For data set of plots in this figure see S3 File.

doi:10.1371/journal.pone.0141486.g005

set, there are a minimum and a maximum that clearly show three potential linear fit regions. Furthermore, region 3 has the highest data density. However, the new method was able to identify linear fit from region 2, which has low data density. Again, this confirms the previous observation. Plots (c2) and (d2) show another feature of the new method: the identified linear fits clearly consist of two segments separated by a no-data area.

According to the most popular least squares method, data points that agree with linear fit are the data points around the trend line. However, our aim is to identify the most potential data sets that agree with the linear fit. Therefore, that detection of least squares method cannot

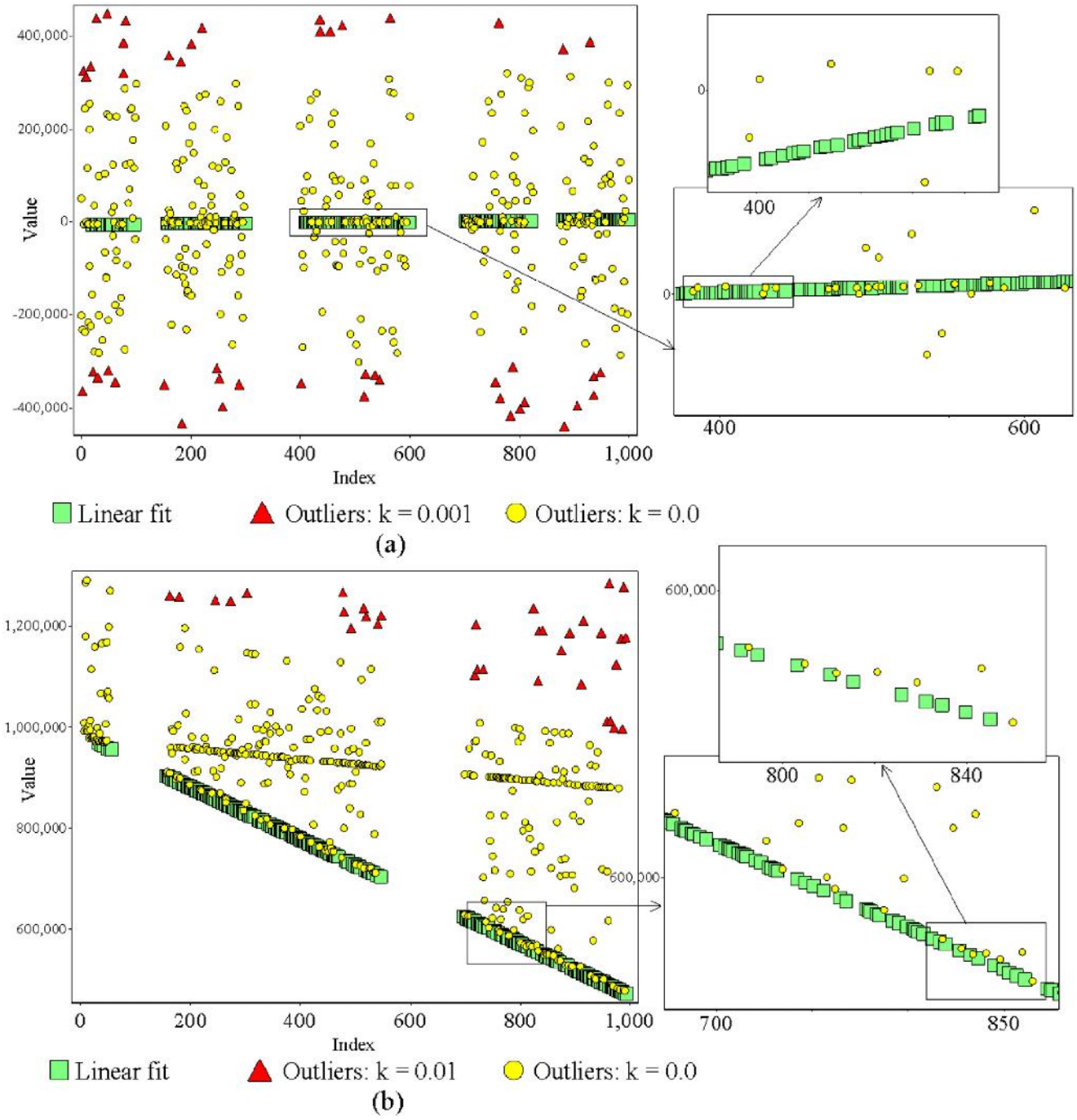


Fig 6. Plots (a) and (b) show two artificial data sets, each consisting of a data set with 100% agreement with unknown linear regression. The number of data points agreeing with linear fit is less than 50% of total existing data points. In plot (a), data points that do not agree with linear fit lie on both sides of linear fit and exhibit four initial missing data regions of 50, 100, 100 and 50 data points (total 300 initial missing data). In plot (b), data points that do not agree with linear fit are located on one side of linear fit and exhibit two initial missing data regions of 100 and 150 data points (total 250 initial missing data). In both plots, data points that do not agree with linear fit are in the range of $\pm 10^{-2}$ to $\pm 10^{4}$. Though both data sets represent very extreme conditions, the method was capable of locating all data points that agreed with linear fit without swapping or masking. Zoomed areas of selected areas that contain very near values to linear fit demonstrate the ability of the proposed method. In plots (a) and (b) the reference points (the first term of the linear fit) were automatically detected during the detection process as 20 and 26, respectively (all the points were considered as the reference point). For data set of plots in this figure see S4 File.

doi:10.1371/journal.pone.0141486.g006

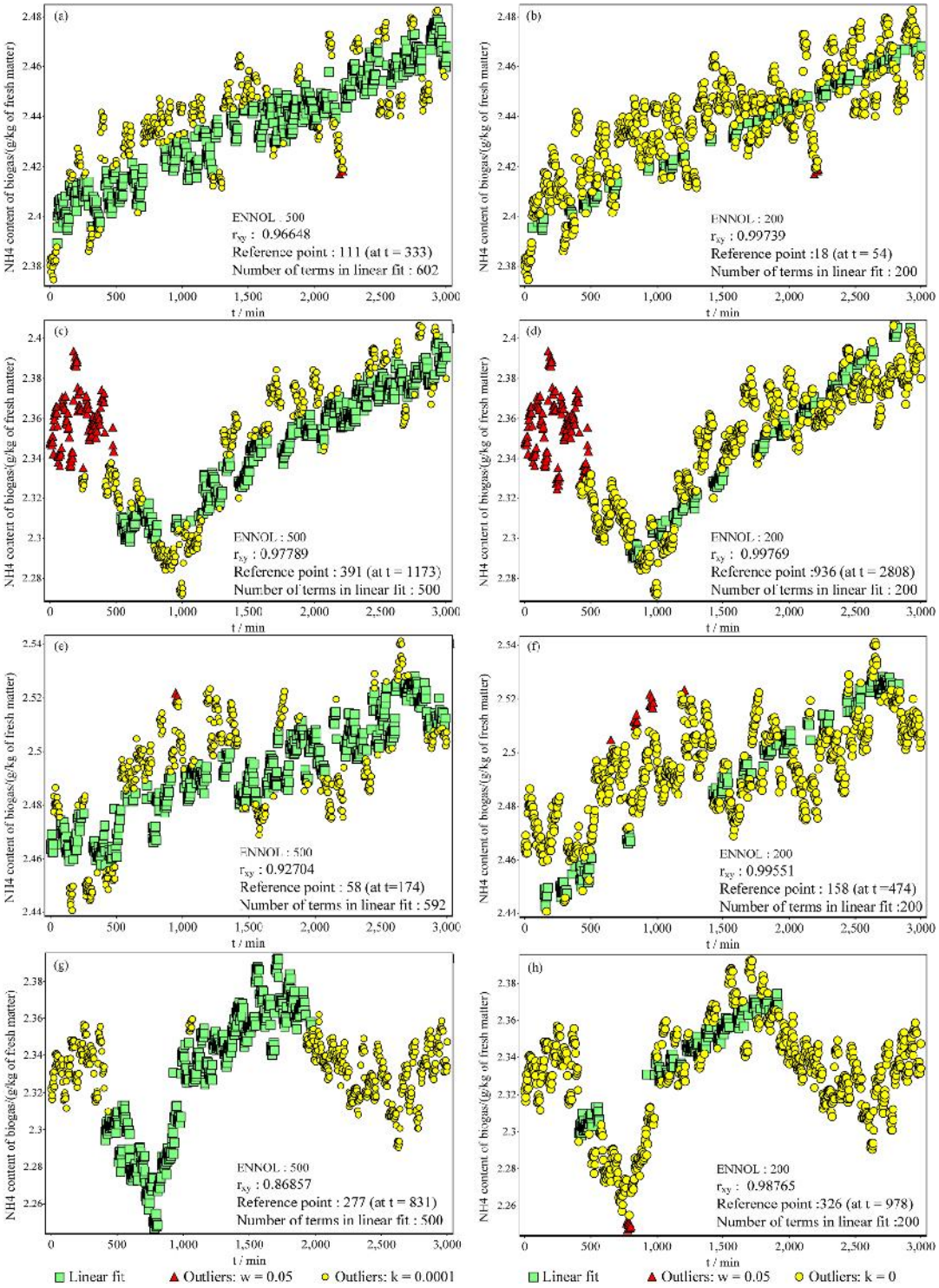


Fig 7. Plots show four selected windows of data captured automatically from a biogas plant in a three-minute interval (each window consists of 1,000 data points). The left side shows the linear fit detection in relation to a particular criterion, while the right side shows the linear fit detection of the relevant left-side data set in relation to narrower criteria than the left-side plot. In all cases, the method identified the most suitable linear fit in relation to the selected window. When the criteria are narrowed, the detection is sharp and there is a sub-set of the linear fit identified in relation to wider criteria. In all plots, the reference point (the first term of the linear fit) was automatically detected during the detection process (all the points were considered as the reference point). For data set of plots in this figure see S5 File.

doi:10.1371/journal.pone.0141486.g007

be considered as good detection according to our requirement. Therefore, the abilities of the new method are in a better position when identifying linear fit because it is capable of identifying the best fit among the several positive candidate linear fits. This type of detection can be performed using an appropriate mask. Sometimes it is necessary to use several masks for identifying different types of linear fit, such as one mask for identifying linear fits with positive gradients, one for identifying linear fits with negative gradients and one for identifying linear fits that are constant. In contrast, the new method is capable of identifying any type of linear relation. Therefore, the new method can be considered as very useful for identifying linear fit.

When considering the theoretical environment of the equations used in the method, there are several situations that must be addressed. Theoretically, there are two situations for which the method could become invalid. The first situation occurs when $Gx^{T}_{k\,|\,r} = 0$ in Eq 6. To overcome this situation, we propose a solution that can be used in normal situations as well. The proposed method in this paper always suspects the maximum and minimum as the data points that do not agree with linear fit. If the suspected data points were removed, it is possible to have better approximation for the gradient as well. However, after removing both suspected values, it is still possible to have the same situation. Therefore, as a standard, when $Gx^{T}_{k\,|\,r} = 0$, removing one suspected point will guarantee the prevention of an undefined situation. In addition, if no undefined situation arises, it is better to exclude both suspected points. As we mentioned earlier, this technique can be applied throughout the process. However, this requires additional computational effort. We used the same technique to improve the outlier detection power of Grubb's test and obtained significant improvement.

The second invalid situation occurs when $S^{TT}_{n} - a^{TT}_{min} * n = 0$ or $a^{TT}_{max} * n - S^{TT}_{n} = 0$. Then, according to (7), $a^{TT}_{\max|\,r} = a^{TT}_{\min|\,r}$ (the maximum and minimum of the transformed series are the same). In addition, the transformed value of the considered reference point is always zero. Therefore, $S^{TT}_{n} - a^{TT}_{min} * n = 0$ or $a^{TT}_{max} * n - S^{TT}_{n} = 0$ represents the status in which all values of the transformed series are zero. This state also represents a totally outlier and noise free series and is a termination condition.

In addition to the two abovementioned undefined situations, there is another situation in which it is not possible to determine the termination point. The situation in which all remaining terms agree with linear fit, with $MMS(a^{TT})_{\max|\,r} = MMS(a^{TT})_{\min|\,r}$, can be considered as the termination point of Eq 7. However, there can be a very rare situation that is in disagreement with the normal situation. For example, if the transformed series is 0, −1.1, −2.1, 2.2, 1, 0.3, then $MMS(a^{TT})_{\max|\,0} = MMS(a^{TT})_{\min|\,0}$ occurs (both values are equal to 0.33). In this case, the transformed series does not agree with linear fit, even though it satisfies the termination condition. Therefore, it is necessary to verify that the situation is a real termination situation. One possible remedy for overcoming this situation is to recalculate the data series by temporarily excluding one data point that is not a suspected point and is not equal to zero. If the same situation still occurs even after removing the data point, it can be considered as the real termination point. Otherwise, conduct the calculation without temporarily removing the term, and add it in the next iteration. If the transformed series 1.1, 1.1, 0, 0, 0, −1.1, −1.1 again satisfies $MMS(a^{TT})_{\max|\,0} = MMS(a^{TT})_{\min|\,0}$, then the aforementioned method cannot be used. Therefore, the only possible solution is to consider all non-zero terms as terms that do not agree with linear fit.

Conclusions and Outlook

The new method shows very promising results in the area of linear fit identification. The method is non-parametric and capable of identifying all data points that agree with linear fit

without swapping or masking. A particular strength of the new method is that it detects the most probable linear fit in the selected window despite the influence of data density, missing data, removed elements, percentage of data agreeing with linear fit and manner of distribution of data points. In other words, the introduced method can be considered as a universal method for linear fit identification. In this paper, we focused on identifying a single linear fit. However, the method could be enhanced for identifying multiple linear fits in the selected window.

Supporting Information

S1 File. Example calculation: A complete process circle for achieving a candidate data set for linear fit with reference to a certain reference point.
(XLSX)

S2 File. Data sets of all the plots in Fig 4.
(XLSX)

S3 File. Data sets of all the plots in Fig 5.
(XLSX)

S4 File. Data sets of all the plots in Fig 6.
(XLSX)

S5 File. Data sets of all the plots in Fig 7.
(XLSX)

Acknowledgments

We are grateful to the German Academic Exchange Service (Deutscher Akademischer Austauschdienst, DAAD) for providing a scholarship to KKLB Adikaram during the research period.

Author Contributions

Conceived and designed the experiments: KKLBA. Performed the experiments: KKLBA. Analyzed the data: KKLBA MAH ME. Contributed reagents/materials/analysis tools: KKLBA MAH ME TB. Wrote the paper: KKLBA.

References

1. Anscombe FJ (1973) Graphs in Statistical Analysis. The American Statistician 27: 17–21.
2. Beckman RJ, Cook RD (1983) Outlier . . . . . s. Technometrics 25: 119–149.
3. Chen Y, Caramanis C. Noisy and missing data regression: Distribution-oblivious support recovery; 2013. pp. 383–391.
4. Sims CA (1974) Seasonality in Regression. Journal of the American Statistical Association 69: 618–626.
5. Choi S-W (2009) The Effect of Outliers on Regression Analysis: Regime Type and Foreign Direct Investment. Quarterly Journal of Political Science 4: 153–165.
6. Stevens JP (1984) Outliers and influential data points in regression analysis. Psychological Bulletin 95: 334.
7. Liu Y, Wu AD, Zumbo BD (2010) The impact of outliers on Cronbach's coefficient alpha estimate of reliability: Ordinal/rating scale item responses. Educational and Psychological Measurement 70: 5–21.
8. Alimohammadi I, Nassiri P, Hosseini MBM (2005) Reliability analysis of traffic noise estimates in highways of Tehran by Monte Carlo simulation method. Iranian journal of environmental health science & engineering 2: 229–236.

9. De Brabanter K, Pelckmans K, De Brabanter J, Debruyne M, Suykens JA, Hubert M, et al. (2009) Robustness of kernel based regression: a comparison of iterative weighting schemes. Artificial Neural Networks—ICANN 2009: Springer. pp. 100–110.
10. Liu Y, Zumbo BD, Wu AD (2012) A demonstration of the impact of outliers on the decisions about the number of factors in exploratory factor analysis. Educational and Psychological Measurement 72: 181–199.
11. Sykes AO (1993) An introduction to regression analysis. 16.
12. Dicker LH (2012) Residual variance and the signal-to-noise ratio in high-dimensional linear models. arXiv preprint arXiv:12090012.
13. Nakai Michikazu K W (2011) Review of the Methods for Handling Missing Data in Longitudinal Data Analysis. Int Journal of Math Analysis 5: 1–13.
14. Stuart EA, Azur M, Frangakis C, Leaf P (2009) Multiple Imputation With Large Data Sets: A Case Study of the Children's Mental Health Initiative. American Journal of Epidemiology 169: 1133–1139. doi: 10.1093/aje/kwp026 PMID: 19318618
15. Gelb A (1974) Applied Optimal Estimation: M.I.T. Press.
16. Liu H, Shah S, Jiang W (2004) On-line outlier detection and data cleaning. Computers & Chemical Engineering 28: 1635–1647.
17. Chiang J-T (2008) The algorithm for multiple outliers detection against masking and swamping effects. Int J Contemp Math Sciences 3: 839–859.
18. Bacon-Shone J, Fung WK (1987) A New Graphical Method for Detecting Single and Multiple Outliers in Univariate and Multivariate Data. Journal of the Royal Statistical Society Series C (Applied Statistics) 36: 153–162.
19. Solak MK (2009) Detection of multiple outliers in univariate data sets. Schering.
20. Yadav BS, Mohan M (2011) Ancient Indian Leaps into Mathematics: Birkhauser.
21. Ray B (2009) Different Types of History. India: Pearson Education.
22. Aryabhata (2006) The Aryabhatiya Of Aryabhata: An Ancient Indian Work On Mathematics And Astronomy. Clark WE, translator. Chicago, Illinois: The University of Chicago Press. 124 p.
23. Adikaram KKLB, Hussein MA, Effenberger M, Becker T (2014) Outlier Detection Method in Linear Regression Based on Sum of Arithmetic Progression. The Scientific World Journal.
24. Chan Y (2003) Biostatistics 104: correlational analysis. Singapore Med J 44: 614–619. PMID: 14770254
25. Bronson G (2012) C++ for Engineers and Scientists: Cengage Learning.

Journal of Signal and Information Processing, 2016, 7, 192-213
http://www.scirp.org/journal/jsip
ISSN Online: 2159-4481
ISSN Print: 2159-4465

Non-Parametric Local Maxima and Minima Finder with Filtering Techniques for Bioprocess

K. K. L. B. Adikaram[1,2,3*], M. A. Hussein[1], M. Effenberger[2], T. Becker[4]

[1]Group Bio-Process Analysis Technology, Technische Universität München, Freising, Germany
[2]Bavarian State Research Center for Agriculture, Institute for Agricultural Engineering and Animal Husbandry, Freising, Germany
[3]Computer Unit, Faculty of Agriculture, University of Ruhuna, Mapalana, Kamburupitiy, Sri Lanka
[4]Lehrstuhl für Brau-und Getränketechnologie, Technische Universität München, Freising, Germany
Email: *lasantha@daad-alumni.de

How to cite this paper: Adikaram, K.K.L.B., Hussein, M.A., Effenberger, M. and Becker, T. (2016) Non-Parametric Local Maxima and Minima Finder with Filtering Techniques for Bioprocess. *Journal of Signal and Information Processing*, **7**, 192-213.
http://dx.doi.org/10.4236/jsip.2016.74018

Received: July 21, 2016
Accepted: October 8, 2016
Published: October 11, 2016

Abstract

Typically extrema filtration techniques are based on non-parametric properties such as magnitude of prominences and the widths at half prominence, which cannot be used with data that possess a dynamic nature. In this work, an extrema identification that is totally independent of derivative-based approaches and independent of quantitative attributes is introduced. For three consecutive positive terms arranged in a line, the ratio (R) of the sum of the maximum and minimum to the sum of the three terms is always 2/n, where n is the number of terms and $2/3 \leq R \leq 1$ when $n = 3$. $R > 2/3$ implies that one term is away from the other two terms. Applying suitable modifications for the above stated hypothesis, the method was developed and the method is capable of identifying peaks and valleys in any signal. Furthermore, three techniques were developed for filtering non-dominating, sharp, gradual, low and high extrema. Especially, all the developed methods are non-parametric and suitable for analyzing processes that have dynamic nature such as biogas data. The methods were evaluated using automatically collected biogas data. Results showed that the extrema identification method was capable of identifying local extrema with 0% error. Furthermore, the non-parametric filtering techniques were able to distinguish dominating, flat, sharp, high, and low extrema in the biogas data with high robustness.

Keywords

Extrema Point, First Derivative, Peak Finder, Peaks and Valleys, Maxima and Minima, Second Derivative

1. Introduction

In process control, the method of determining peaks and valleys of a signal, also known as identification of local maxima and minima, is crucial for describing and capturing

certain signal properties. Identification of local maxima and minima is particularly useful in signal processing, consequently useful in inline/online process control and optimization. Thereby, for reliable feature extraction it is necessary to remove redundant maxima and minima in a processed signal. The issue has been extensively investigated in literature [1]-[4], at which different techniques were reported. Magnitude-based methods and gradient-based methods are the most common two of such techniques. In magnitude based methods, the n^{th} term of a series is x_n; x_n is considered as a peak (maximum) when $x_{n-1} < x_n > x_{n+1}$. In the same time, x_n is considered as a valley (minimum) when $x_{n-1} > x_n < x_{n+1}$. In gradient-based methods, extremum can be located by considering slope (gradient) of a certain point and acts as the most popular method [1]. When the slope is zero (first derivative is zero) at a certain point, the point can be described as a peak, valley or a saddle point. However, additional calculations are necessary to distinguish whether it is a peak, valley or saddle point. This encounter is solved by analysing the sign of the second derivative at the points of zero slopes [1]. The most popular methods of such are Newton Raphson method [2] and Taylor series-based derivatives [3] [4] which evaluate the derivatives numerically for a given data set.

Once the extrema points are identified, a filtration step is unavoidable to identify the dominant or relevant extrema. Magnitude of prominences and the widths at half prominence are two properties of signals that are commonly used to filter extrema [5] [6]. Furthermore, baseline correction is another technique used for finding out accurate maxima and minima [7]-[9]. In addition, there are numbers of methods for filtering unnecessary extrema based on template matching or masks [10], such as Kalman filters [11] [12] and non-linear filters [13]. Nevertheless, all aforementioned approaches are parametric methods [14], which question their robustness.

One of the main classifications existing in data analysis techniques is whether the method is parametric or non-parametric in its nature [15]. As mentioned above, most popular extrema filtering methods suffer from parametric concerns. Particularly, parametric methods use domain dependent value as detection criteria such as average, standard deviation, prominences of an extrema, and the widths at half prominence of an extrema. These criteria are based on domain dependent parameters and are therefore valid only for the considered data model or considered conditions in the domain. Thus, majorly parametric methods' accuracy inherits the variables' ranges and the conditions of the domain [16]. In reality, data capturing, especially within dynamic systems, such as biogas plants, is produced with various alterations. When the model or data range alters, whilst using parametric methods, it is necessary to recalibrate parameters or develop new models for monitoring, controlling, and data analysis, which is not of preference at process line.

Non-Parametric Methods

Non-parametric methods, also known as distribution-free methods, depend on fewer number of underlying assumptions [15] [17] [18], which progress them more as robust methods [16] [19]. In this research a new non-parametric technique for extrema identi-

fication and filtration are developed. The proposed technique determines maxima and minima based on the relation of sum of terms in an arithmetic series. The same relation was used as a non-parametric method (MMS: a method based on maximum, minimum, and sum) for finding outliers in linear relation [20] and non-parametric linear fit identification method [21].

In some situations outliers, peaks and valleys are the same, when a sudden extremum (variation) occurs, additionally extrema can be formed due to gradual increment and gradual decrement. The extrema generated in such situations do not behave as outliers and cannot be identified using the aforementioned outlier detection method based on maximum, minimum, and data series sum (MMS) [20]. Furthermore, MMS can only be used for identifying outliers in liner regression and is not suitable for finding outliers in non-linear series [20]. This work focuses on modifying the methods of MMS for locating extrema in non-linear data series.

The proposed extrema identification method does not involve first or second derivative, but rather compares, within a considered window, two ratios in relation with maximum, minimum, middle point, and the sum of data points. Furthermore, three extrema filtration methods were introduced in this work, which are capable of filtering extrema independent of the prominences or width of an extremum. All the methods introduced in this work are developed for harsh conditions involved in dynamic processes, especially biogas process data, thus handling: non-linear datasets and based upon non-parametric methods.

2. Materials and Methods

As mentioned before, the outlier detection method, also by the same authors [20], will be modified to locate extrema in non-linear series. The method is based upon the theory of the sum of terms of an arithmetic progression. Having two major relations by means of $MMS_{\max}$ and $MMS_{\min}$ and are expressed in Equation (1) and Equation (2). The ratio $2/n$ is used as the detection criteria, where n is the number of terms in the series.

$$MMS_{\max} = \left(a_{\max} - a_{\min}\right) / \left(S_n - a_{\min} * n\right), \text{ and} \tag{1}$$

$$MMS_{\min} = \left(a_{\max} - a_{\min}\right) / \left(a_{\max} * n - S_n\right), \tag{2}$$

where $a_{\min}$ is the minimum element of the series, $a_{\max}$ is the maximum element of the series, n is the number of terms in the series, and S_n is the sum of terms in the series.

The complete expression for outlier detection is given by Equation (3). If any series expected to follow $y = c$ form and contains data that do not agree with $y = c$ form then:

$$MMS = \begin{cases} MMS_{\max} = \dfrac{a_{\max} - a_{\min}}{S_n - a_{\min} * n} = \begin{cases} > \left(2/n + w\right); \text{maximum is the outlier} \\ \leq \left(2/n + w\right); - \end{cases} \\ MMS_{\min} = \dfrac{a_{\max} - a_{\min}}{a_{\max} * n - S_n} = \begin{cases} \leq \left(2/n + w\right); - \\ > \left(2/n + w\right); \text{minimum is the outlier} \end{cases} \end{cases}, \tag{3}$$

where w is the weight.

The method MMS expressed in Equation (3) can be applied on a window with any

number of data points. However, when a window has only three data points it becomes a special situation, since the method generates an extremum when points are not in agreement with a linear fit, thus, if there is an extrema, always the middle point would be the extrema. When the numbers of data points are three ($n=3$) and $w=0$, Equation (3) a special treatment is suggested:

$$MMS=\begin{cases} MMS_{\max}=\dfrac{a_{\max}-a_{\min}}{S_3-a_{\min}*3}=\begin{cases}>2/3;\text{Maximum is away from the other two points}\\ \le 2/3;-\end{cases}\\ MMS_{\min}=\dfrac{a_{\max}-a_{\min}}{a_{\max}*3-S_3}=\begin{cases}\le 2/3;-\\ >2/3;\text{Minimum is away from the other two points}\end{cases}\end{cases} \tag{4}$$

Equation (4) is a simplified version of Equation (3) for handling three data points, where $2/3\le MMS_{\max}\le 1$ and $2/3\le MMS_{\min}\le 1$. According to Equation (4), $MMS_{\max}>2/3$ implies that the maximum of the three points is always considerably apart from the other two points. In the same manner, $MMS_{\min}>2/3$ implies that the minimum of the three points is always considerably apart from the other two points. Plots (a), (b), and (c) of Figure 1 show situations that of $MMS_{\max}>2/3$, where maximum is the peak. Plots (e), (f), and (g) in Figure 1 show situations that of $MMS_{\min}>2/3$, where the minimum is the valley. However, Equation (4) does not always successfully identify extrema, in other words if the identified point is the first or last point of a window, theoretically it cannot be considered as an extrema. Plots (d)

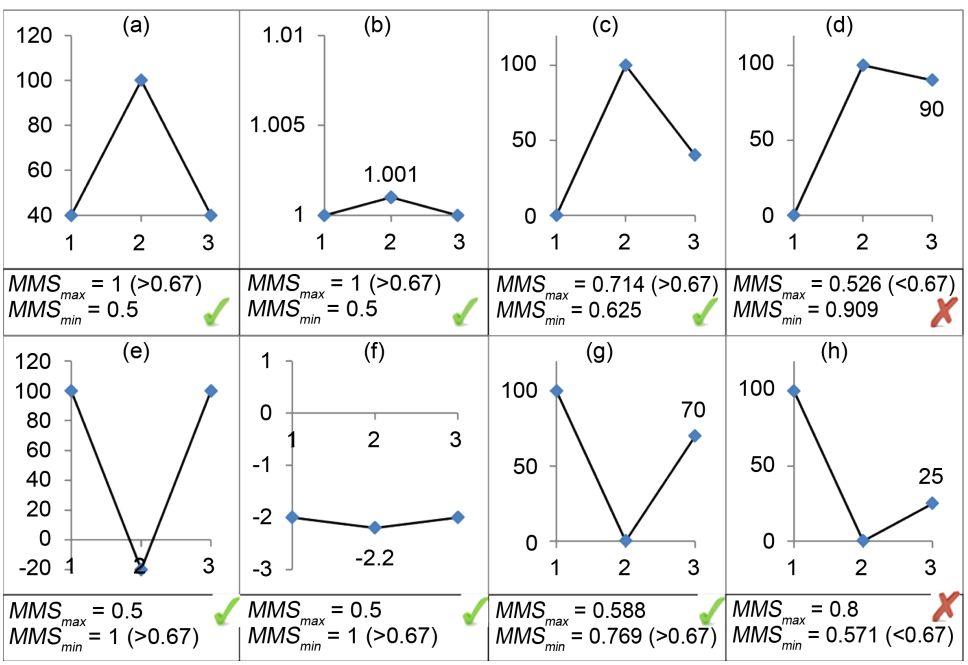


Figure 1. Plots (a), (b), (c), and (d) show different types of peaks and plots (e), (f), (g), and (h) show different types of valleys. For $n=3$, value $2/n$ is 0.67. In all peaks except plot (d) $MMS_{\max}>2/3$ and in all valleys except plot (h) $MMS_{\min}<2/3$. "✓" corresponds to correct detections of extrema when MMS and $2/n$ are used and "×" corresponds to wrong detections of extrema when MMS and $2/n$ are used. Therefore, consideration of MMS and $2/n$ is not a good method for identifying extrema. However, in the concept of outlier detection all the detections are correct.

and (h) of Figure 1 show situations where neither a maximum nor a minimum represents an extremum. Figure 1(d) shows a peak that of $MMS_{\min} > 2/3$ (identification of a valley), this is a contradicting situation. Also, Figure 1(h) shows another failing situation, where the plot shows a valley that of $MMS_{\max} > 2/3$ (identification of a peak). This occurs because in both considered situations, the point has the highest deviation is the first point and not the middle point. Therefore, Equation (4) is not capable of identifying extrema in such cases, thus handling these situations is required.

To address the aforementioned drawback, the MMS method was modified by considering the middle point of the window. To have an exact middle point in a data window the number of considered data points (n) must be odd. When $n = 3$ and a_{mid} is the middle point of the window, substituting $a_{\max}$ from Equation (1) by a_{mid} retrieves:

$$MMS_{\max|\text{mid}} = \left(a_{\text{mid}} - a_{\min}\right) / \left(S_3 - a_{\min} * 3\right). \tag{5}$$

Also, by replacing $a_{\min}$ of Equation (2) by amid gives,

$$MMS_{\min|\text{mid}} = \left(a_{\max} - a_{\text{mid}}\right) / \left(a_{\max} * 3 - S_3\right). \tag{6}$$

Consider the situation,

$$MMS_{\max} = MMS_{\max|\text{mid}}. \tag{7}$$

$$\left(a_{\max} - a_{\min}\right) / \left(S_3 - a_{\min} * 3\right) = \left(a_{\text{mid}} - a_{\min}\right) / \left(S_3 - a_{\min} * 3\right),$$

$$a_{\max} = a_{\text{mid}}. \tag{8}$$

Therefore, Equation (7) denotes the situation of a maximum at the middle point. Thus Equation (7) is a condition, independent of the value of MMS that can be used for identifying a peak.

Consider the situation:

$$MMS_{\min} = MMS_{\min|\text{mid}}. \tag{9}$$

$$\left(a_{\max} - a_{\min}\right) / \left(a_{\max} * 3 - S_3\right) = \left(a_{\max} - a_{\text{mid}}\right) / \left(a_{\max} * 3 - S_3\right),$$

$$a_{\min} = a_{\text{mid}}. \tag{10}$$

Then Equation (9) denotes the situation of a minimum at the middle point. Thus, Equation (9) is a condition, independent of the value of MMS that can be used for identifying a valley.

Therefore, when a window satisfies Equation (7) it implies that the middle point is a maximum and once a window satisfies Equation (9) it alternatively implies that the middle point is a minimum. Advancing the three point window by one data point makes it possible to locate all the extrema in a signal (Figure 2). Table 1 shows sample calculations of extrema detection procedure according to Equation (7) and Equation (9). The first eight value sets shown in Table 1 are the values in relation with the plots shown in Figure 1. Examples a, b, c, and d in Table 1 show calculation in relation with peak identification. In all these examples $MMS_{\max} = MMS_{\max|\text{mid}}$ (Equation (7)) and $MMS_{\min} \neq MMS_{\min|\text{mid}}$ (Equation (9)). Examples e, f, g, and h in Table 1 show calculation in relation with valley identification. In all these examples $MMS_{\min} = MMS_{\min|\text{mid}}$ (Equation (7)) and $MMS_{\max} \neq MMS_{\max|\text{mid}}$ (Equation (9)). The last two examples (i and

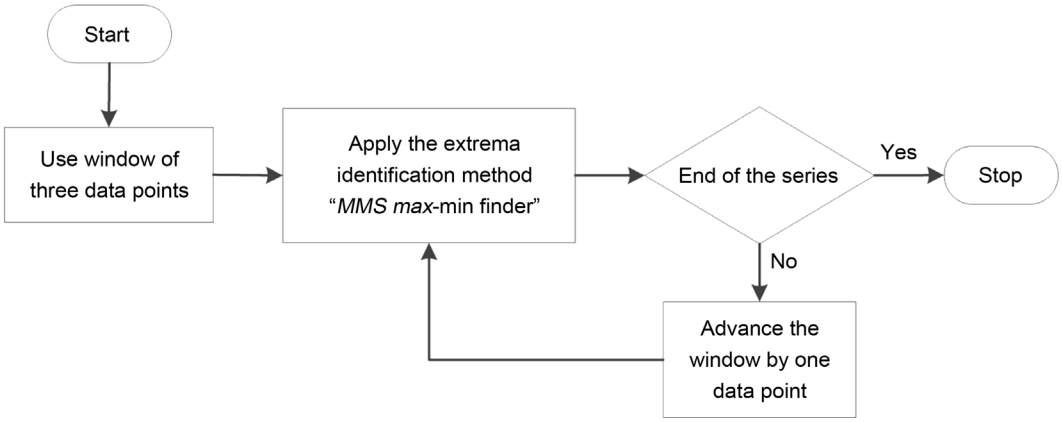


Figure 2. Extrema detection process of proposed extrema identification method named as MMS max-min finder.

Table 1. Sample calculations of peak and valley detection process based new method (MMS max-min finder) for window size of three data points.

Plot	Data set	MMS_{max} (>0.67)	$MMS_{max/mid}$	MMS_{min} (>0.67)	$MMS_{min/mid}$	Peak or Valley	$MMS_{max} = MMS_{max/mid}$	$MMS_{min} = MMS_{min/mid}$
(a)	0, 100, 0	1 (Y)	1	0.5 (N)	0	**Peak**	**Y**	N
(b)	0, 1.001, 0	1 (Y)	1	0.5 (N)	0	**Peak**	**Y**	N
(c)	0, 100, 40	0.714 (Y)	0.714	0.625 (N)	0	**Peak**	**Y**	N
(d)	0, 100, 90	0.526 (N)	0.526	0.909 (Y)	0	**Peak**	**Y**	N
(e)	100, −20, 100	0.5 (N)	0	1 (Y)	1	**Valley**	N	**Y**
(f)	−2, −2.2, −2	0.5 (N)	0	1 (Y)	1	**Valley**	N	**Y**
(g)	100, 0, 70	0.588 (N)	0	0.769 (Y)	0.769	**Valley**	N	**Y**
(h)	100, 0, 25	0.8 (Y)	0	0.571 (N)	0.571	**Valley**	N	**Y**
(i)	0, 100, 50	0.667 (N)	0.667	0.667 (N)	0	**Peak**	**Y**	N
(k)	0, −100, −50	0.667 (N)	0	0.667 (N)	0.667	**Valley**	**N**	Y

k) show very special situations, where MMS_{max}, MMS_{min}, and $2/n$ are equal. In such situations Equation (4) is undefined. However, even then extrema identification is possible with Equation (7) and Equation (9). Since the proposed extrema detection method is based on the maximum, minimum, and sum of the series, the method was named as "*MMS* max-min finder".

2.1. Identifying Dominating Extrema (Primary Filtering of Peaks and Valleys)

As above-mentioned, Equation (7) and Equation (9) are independent of the number of data points and thus valid for the situations where n is greater than three ($n > 3$). However to have an exact middle point, n must be an odd number. When the numbers of data points are higher than three, there can be several peaks and several valleys. However, there is a situation that the highest peak (dominating peak) or lowest valley

(dominating valley) coincides with the middle point of an advancing window. Figure 3 shows an example of detecting dominating peaks in a window with odd number of data points ($n = 7$). When the number of data points per window increases, it allows for the possibility of more than one extremum in the considered window.

The plot in Figure 3 consists of seven data points and contains three peaks named A, B, and C. The peak A is the middle point of window W_n while peak B is the dominating peak. Because of that point A is not recognise as a peak in window W_n. After advancing W_n by two data points, W_{n+2} appears. In the window W_{n+2} the point B is the highest as well as the middle point and the point B is recognized as a peak. Advancing W_{n+2} by two data points W_{n+4} appears, where C becomes the middle point and due to the influence of point B it will not be recognized as a peak. This illustrates that the dominating extrema in a window remains undetected until the middle point of the window coincide with it whilst preventing identification of other small peaks and valleys.

The usage of windows with higher odd number of data points (e.g.: 5, 7, …) makes it possible to filter minor peaks and valleys. In contrast, if the methods in relation with height or width are used, the values are domain dependent and relative. Changing window size (W) is an absolute parameter and can be applied in any condition, especially the situations that the domain conditions are unknown. However, this technique is not capable of filtering absolute small extrema, because the comparison is based on the existing extrema in the considered window. Furthermore, this technique is useful as a filter for removing relative small variations. Since the technique is based on the size of the window, the technique was named as "MMS-Window based filter" or (MMS-WBF).

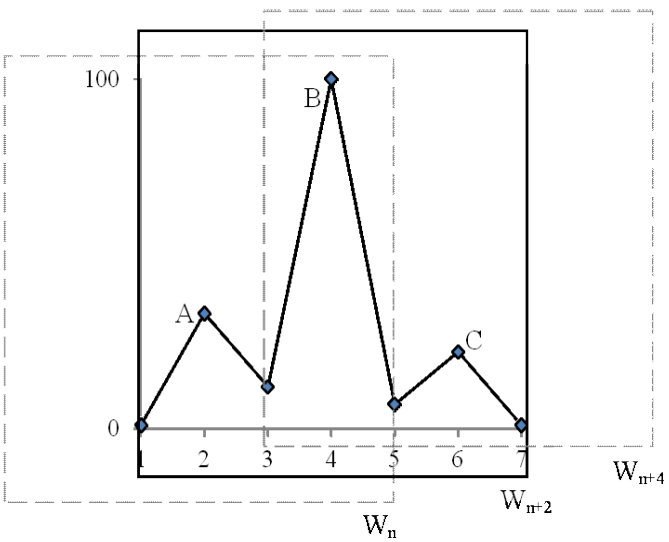


Figure 3. Application of "MMS max-min finder" with window size of seven for locating maximum point. In the window W_n the middle point is "A" and due to existence of point "B" in the considered window, point "A" is not identified as the maximum point. Also, in the W_{n+4} (the window found after advancing by four data points) the point "C" is not identified as an extrema, due to existence of point "B" in the considered the window. In the W_{n+2} the point "B" is the maximum as well as middle point and there is no point larger than point "B" in the considered window. Thus, point "B" is identified as the dominating maximum.

2.2. Sharp and Gradual (Flat) Extrema Filtering

Extrema with starting and end points which are agreeing with $y = c$ and having the middle point as the extremum can be considered as a symmetric extrema case. Plots (a) and (b) of Figure 4 show such symmetric extrema, which can be considered as the simplest symmetric form. Extrema shown in plots (c) and (d) of Figure 4 also fulfil the requirements of a perfect symmetric extrema. All the following equations in this section are based on the perfect extrema.

Consider a perfect maxima situation as shown in plot (c) of Figure 4. Here, all points are equal to $a_{\min}$ ($a_{\min} = c$) except $a_{\max}$. Consider any perfect maximum situation with n points, then $n-1$ points are equal to $a_{\min}$, and $a_{\max} \neq c$. The sum of the terms of such a series can be expressed as:

$$S_n = a_{\min} * (n-1) + a_{\max} \tag{11}$$

$$(S_n - a_{\max}) / a_{\min} = (n-1) \tag{12}$$

Consider a perfect minimum situation as shown in plot (d) of Figure 4. Here, all points of the series are equal to $a_{\max}$ and $a_{\max} = c$ except $a_{\min}$. Consider any perfect maxima situation with n points. Then $n-1$ points are equal to $a_{\max}$, and $a_{\min} \neq c$. The sum of the terms of such a series can be expressed as:

$$S_n = a_{\max} * (n-1) + a_{\min} \tag{13}$$

$$(S_n - a_{\min}) / a_{\max} = (n-1) \tag{14}$$

If $MMS_{\max} / MMS_{\min} = R_{Mm}$, then from (1) and (2),

$$R_{Mm} = (a_{\max} * n - S_n) / (S_n - a_{\min} * n).$$

When the maximum is detected as the peak, substituting in Equation (11) retrieves:

$$R_{Mm} = (a_{\max} * n - (a_{\min} * (n-1) + a_{\max})) / ((a_{\min} * (n-1) + a_{\max}) - a_{\min} * n)$$

$$R_{Mm} = (a_{\max} * n - (a_{\min} * n - a_{\min} + a_{\max})) / ((a_{\min} * n - a_{\min} + a_{\max}) - a_{\min} * n)$$

$$R_{Mm} = ((a_{\max} - a_{\min}) * n - (a_{\max} - a_{\min})) / (a_{\max} - a_{\min})$$

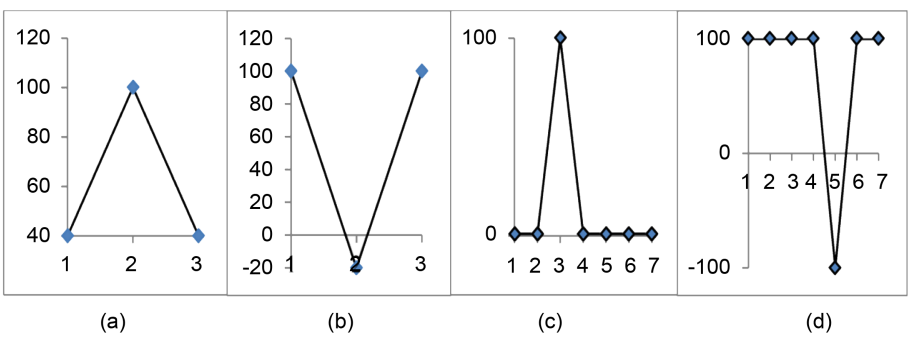


Figure 4. Perfect extrema. A perfect extrema is defined as an extrema that is symmetric extrema. Thus, in a perfect extrema both the starting and end points follow the $y = c$ form. Plots (a) and (b) show the simplest perfect extrema and plots (c) and (d) show perfect extrema that have more points that agree with $y = c$ form.

$$R_{Mm} = \left(\left(a_{\max} - a_{\min}\right) * \left(n-1\right)\right) \Big/ \left(a_{\max} - a_{\min}\right)$$

$$R_{Mm} = \left(n-1\right)$$

$$MMS_{\max} / MMS_{\min} = \left(n-1\right) \tag{15}$$

In the same manner, if $MMS_{\min}/MMS_{\max} = R_{mM}$, then from Equations (1), (2), and (12), the minimum is detected as the valley,

$$R_{mM} = \left(n-1\right)$$

$$MMS_{\min} / MMS_{\max} = \left(n-1\right) \tag{16}$$

The relations of Equation (15) and Equation (16) are crucial findings, which can be used to identify perfect extrema. When the extrema is not perfect, value of Equation (15) and Equation (16) is less than $n-1$. Therefore, Equation (15) and Equation (16) can be used to identify perfect and non-perfect extrema. Also, perfect extrema are sudden (sharp) extrema and non-perfect extrema can be considered as gradual extrema. Thereby, using Equation (15) and Equation (16) it is possible to filter sharp and gradual extrema.

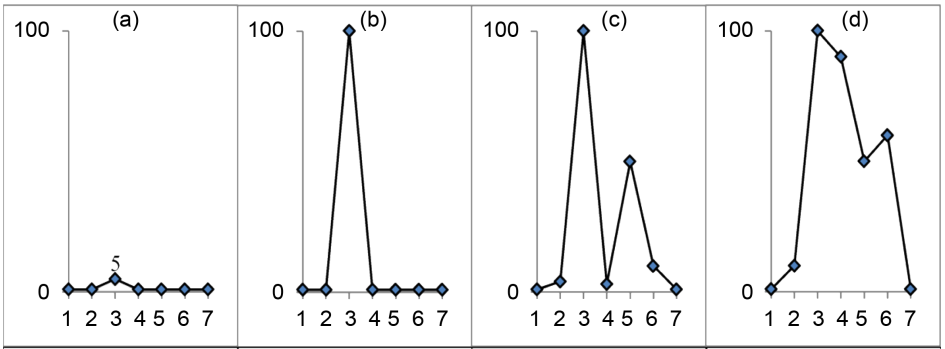
After identifying a peak, by examining the ratio $MMS_{\max}/MMS_{\min}$ it is possible to determine degree of confidence of other points, the same applies for identifying a valley. Assume t_{Mm_mM} is the threshold value for determining sharp and gradual maxima, then t_{Mm_mM} can be expressed as a $k*\left(n-1\right)$, where $0 < k \le 1$. If k is expressed as a function of n (e.g.: $k = 1/\left(n-1\right)$), then t_{Mm_mM} is a function of n. By setting the same threshold value (t_{Mm_mM}) for $MMS_{\max}/MMS_{\min}$ and $MMS_{\min}/MMS_{\max}$, sudden and gradual maxima can be determined. The determination criteria (t_{Mm_mM}) of ratios $MMS_{\max}/MMS_{\min}$ and $MMS_{\min}/MMS_{\max}$ are non-parametric and depend only on the number of data points in the considered window. Since the method is also based on the maximum, the minimum, and the sum, the method was named as MMS-SG filter.

Figure 5 and Figure 6 show examples in relation with Equation (15) and Equation (16), respectively. In plots (a) and (b) of Figure 5, the ratio $MMS_{\max}/MMS_{\min} = 6$, which is exactly equal to $n-1$. This proves the correctness of Equation (15). In the same time, in plots (a) and (b) of Figure 6, the ratio $MMS_{\min}/MMS_{\max} = 6$ and proves the correctness of Equation (16). All these plots exhibit either sudden peak or sudden valley. The corresponding ratios in relation with the plot (c) of Figure 5 and Figure 6 are not equal to $n-1$. However, the corresponding ratios are not very small. Therefore, these extrema can be considered as nearly sharp extrema. Nevertheless, corresponding ratios in relation with, plots (d) of Figure 5 and Figure 6 are very small and these extrema can be considered as gradual extrema.

2.3. High and Low Extrema Filtering

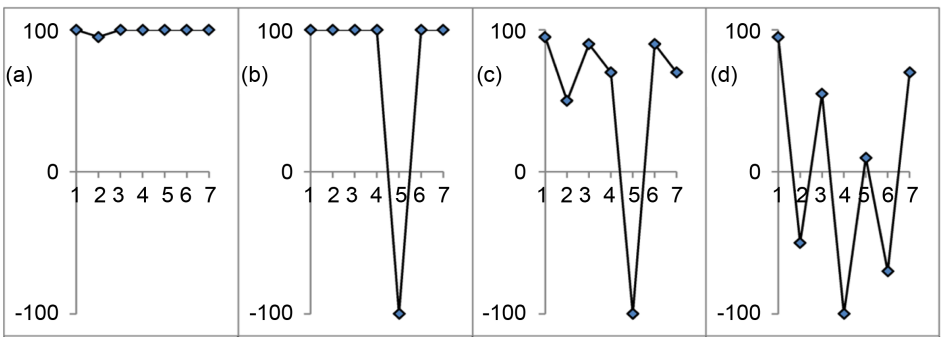
MMS-WBF and MMS-SG introduced in this work are capable identifying dominating, sharp and gradual extrema. However, these techniques are incapable of distinguishing the extrema with very small amplitude as shown in Figure 1(b) and Figure 1(f).

The valley shown in Figure 7 is a general situation of a perfect valley. When a valley has a very small crater, $a_{\min} \approx a_{\max}$.



MMS_{max} = 1 (>0.286) MMS_{min} = 0.167 MMS_{max}/MMS_{min} = 6 MMS_{min}/MMS_{max} = 0.167	MMS_{max} = 1 (>0.286) MMS_{min} = 0.167 MMS_{max}/MMS_{min} = 6 MMS_{min}/MMS_{max} = 0.167	MMS_{max} = 0.611 (>0.286) MMS_{min} = 0.186 MMS_{max}/MMS_{min} = 3.28 MMS_{min}/MMS_{max} = 0.305	MMS_{max} = 0.325 (>0.286) MMS_{min} = 0.255 MMS_{max}/MMS_{min} = 1.27 MMS_{min}/MMS_{max} = 0.786

Figure 5. Plots (a), (b), (c), and (d) show four different types of peaks with window size seven (n = 7) where $2/n$ = 0.286. Ratios MMS_{max}/MMS_{min} and MMS_{min}/MMS_{max} are stated along with each plot. Peaks in plots (a) and (b) are perfect peaks and the ratio $MMS_{max}/MMS_{min} = 6$ (*i.e.* $n-1$). Though, the dominating peak in plot (c) is not a perfect peak, ratio MMS_{max}/MMS_{min} is considerably high. The peak in plot (d) is a gradually developed peak and also not a perfect peak and the ratio MMS_{max}/MMS_{min} is very small. Therefore, consideration of ratio MMS_{max}/MMS_{min} is a good criterion to distinguish sudden and gradual peaks.



MMS_{max} = 0.167 MMS_{min} = 1 (>0.286) MMS_{max}/MMS_{min} = 0.167 MMS_{min}/MMS_{max} = 6	MMS_{max} = 0.167 MMS_{min} = 1 (>0.286) MMS_{max}/MMS_{min} = 0.167 MMS_{min}/MMS_{max} = 6	MMS_{max} = 0.183 MMS_{min} = 0.65 (>0.286) MMS_{max}/MMS_{min} = 0.282 MMS_{min}/MMS_{max} = 3.55	MMS_{max} = 0.275 MMS_{min} = 0.298 (>0.286) MMS_{max}/MMS_{min} = 0.923 MMS_{min}/MMS_{max} = 1.084

Figure 6. Plots (a), (b), (c), and (d) show four different types of valleys with window size seven (n = 7) where $2/n$ = 0.286. Ratios MMS_{max}/MMS_{min} and MMS_{min}/MMS_{max} are stated along with each plot. Valleys in plots (a) and (b) are perfect valleys and the ratio $MMS_{min}/MMS_{max} = 6$ (*i.e.* $n-1$). Though, the valley in plot (c) is not a perfect valley, ratio MMS_{min}/MMS_{max} is considerably high. The valley in plot (d) is a gradually developed valley and also not a perfect valley and the ratio MMS_{min}/MMS_{max} is very small. Therefore, consideration of ratio MMS_{min}/MMS_{max} is a good criterion for distinguishing between sudden and gradual (flat) valleys.

Then, Equation (13) can be expressed as:

$$S_n \approx a_{\min} * (n-1) + a_{\min}$$

$$S_n \approx a_{\min} * n; (< a_{\max} * n)$$

$$R_{LH\_\min} = (a_{\min} * n)/S_n ; 0 < R_{LH\_\min} \le 1 \quad (17)$$

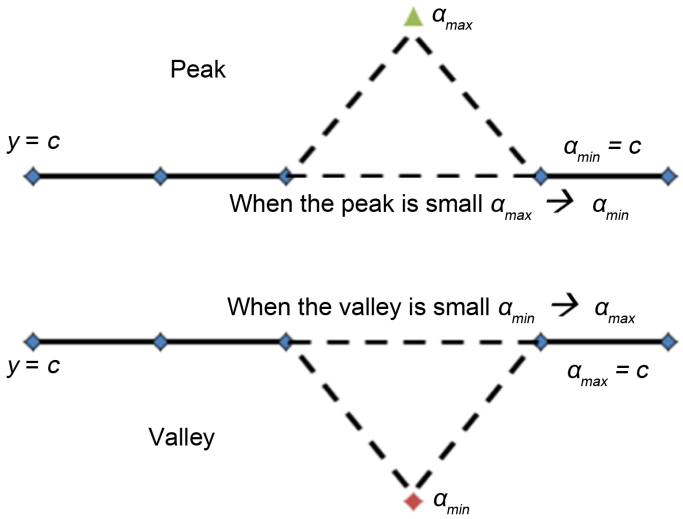


Figure 7. Two possible ways of existence of extremum for a data point; as a peak or as valley. Assume, except the extremum, all the other points are satisfying the $y = c$ relation (perfect extrema). Then, extremum is the peak and all other points are equal to the minimum. In the same manner, when the extremum is a valley, valley is the minimum and all other points are equal to the maximum. If a peak is small it reaches to the minimum and when the valley is small it reaches to the maximum. This is the hypothesis for distinguishing small and high extrema.

If $R_{LH\_\min} \to 1$, it implies that the $a_{\min}$ is very close to the other points (low crater), consequently $R_{LH\_\min} \to 0$ implies that the $a_{\min}$ is apart from the other points (high crater).

In Equation (17), when the term $a_{\min}$ is zero, the ratio $R_{LH\_\min}$ also becomes zero despite of the influence of magnitude valley. Also, due to the influence of negative values S_n can be zero and $R_{LH\_\min}$ becomes invalid. Both these situations inhibit the determination of the real condition of the valley. To overcome the effect of negative values, the minimum value was deducted from all the terms of the data points in the window as expressed in Equation (18).

$$a_{i\_New} = a_i - a_{\min}. \tag{18}$$

Even now it is possible to have a situation of $a_{\min} = 0$. To overcome this situation a constant k, which is greater than zero, was added to each value. This transformation is applied in "Min-Max normalization" process [22] [23]. When $k = 1$ thus Equation (18) becomes:

$$a_{i\_New} = a_i - a_{\min} + 1 \tag{19}$$

From Equation (17) and Equation (19),

$$R_{LH\_\min} = \left(\left(a_{\min} - a_{\min} + 1\right) * n\right) \Big/ \sum_{i=1}^{n} \left(a_i - a_{\min} + 1\right)$$

$$R_{LH\_\min} = n \Big/ \left(\sum_{i=1}^{n} a_i - \sum_{i=1}^{n} a_{\min} + \sum_{i=1}^{n} 1\right)$$

$$R_{LH\_\min} = n \Big/ \left(S_n - a_{\min} * n + n\right); 0 < R_{LH\_\min} \le 1$$

$$R_{LH\_\min} = n/\left(S_n + \left(1 - a_{\min}\right) * n\right); 0 < R_{LH\_\min} \le 1 \tag{20}$$

Then $R_{LH\_\min}$ expressed in Equation (20) can be considered as a robust method for filtering valleys with low crater.

The peak shown in Figure 7 is a general situation of perfect peak. When a peak has a very small prominence, $a_{\max} \approx a_{\min}$. Then Equation (11) can be expressed as:

$$S_n \approx a_{\max} * \left(n-1\right) + a_{\max}$$

$$S_n \approx a_{\max} * n; \left(> a_{\min} * n\right)$$

$$R_{LH\_\max} = \left(a_{\max} * n\right)/S_n; > 0 \tag{21}$$

According to Equation (17), the ratio $R_{LH\_\min}$ has a well-defined upper limit (ceiling) and lower limit (floor) because $0 < R_{LH\_\min} \le 1$. Nevertheless, in Equation (21), $R_{LH\_\max}$ has no upper limit, and subjects only to a lower limit. Therefore, it is difficult to use $R_{LH\_\max}$ as a global criteria as $R_{LH\_\min}$. The peak shown in Figure 7 can be considered as the mirror image of a valley in Figure 7. Thus, it is possible to transform a peak to a valley, for that Equation (17) can be used for determining the peaks with high and low prominence using the same criteria Under the assumption that:

$$a_{i\_New} = \left(a_{\max} + a_{\min}\right) - a_i \tag{22}$$

According to Equation (22), $\left(a_{\max} + a_{\min}\right) - a_{\max} = a_{\min}$ and $\left(a_{\max} + a_{\min}\right) - a_{\min} = a_{\max}$. The expression in Equation (22) transforms the maximum value into the minimum, the minimum value into the maximum and intermediate values into their complements. If the $R_{LH\_\max}$ is the corresponding ratio in relation with high and low peaks identification, then, from Equation (21) and Equation (22), one can reach:

$$R_{LH\_\max} = \left(\left(\left(a_{\max} + a_{\min}\right) - a_{\max}\right) * n\right) \Big/ \sum_{i=1}^{n} \left(\left(a_{\max} + a_{\min}\right) - a_i\right)$$

$$R_{LH\_\max} = \left(a_{\min} * n\right) \Big/ \sum\nolimits_{i=1}^{n} \left(\left(a_{\max} + a_{\min}\right) - a_i\right) \tag{23}$$

Even after the aforementioned transformation, it is still possible to have the influence of negative values. However, it can be resolved by using Equation (19). Then, from Equation (19),

$$R_{LH\_\max} = \left(\left(a_{\min} - a_{\min} + 1\right) * n\right) \Big/ \sum_{i=1}^{n} \left(\left(a_{\max} - a_{\min} + 1 + a_{\min} - a_{\min} + 1\right) - \left(a_i - a_{\min} + 1\right)\right)$$

$$R_{LH\_\max} = n \Big/ \sum_{i=1}^{n} \left(\left(a_{\max} + 1\right) - a_i\right)$$

$$R_{LH\_\max} = n/\left(\left(a_{\max} + 1\right) * n - S_n\right); 0 < R_{LH\_\max} \le 1 \tag{24}$$

$R_{LH\_\max} \to 1$ implies that the $a_{\max}$ is very close to other points (low prominence). Consequently $R_{LH\_\max} \to 0$ implies that the $a_{\max}$ is apart from the other points (high prominence).

Finally, using Equation (17) and Equation (24) it is possible to determine the high and low extrema by defining a threshold value t_{LH} $(0 < t_{LH} \le 1)$ for $R_{LH\_\min}$ and $R_{LH\_\max}$.

Because the method is based on the maximum, minimum and the sum, the method was named as MMS-LH. Figure 8 elaborates the functionality of MMS-LH as a filtering method.

The filtration of sudden, gradual, low, and high extrema are derived based on a data set which satisfies the $y = c$ relation (perfect extrema). However, in reality it is impossible to always have perfect extrema. Therefore, by setting the threshold values in appropriate situations, it is possible to filter the extrema in non-perfect conditions.

Extrema identification is performed after comparing two ratios in relation with maximum, minimum, middle point and sum. The threshold criteria for MMS-WBF and MMS-SG are values that are based on the number of data points (n). The threshold criterion for MMS-LH is a value between 0 and 1. Thus, all the determination criteria are totally non-parametric. However, combination of these methods leads to harvest more robust and reliable output. Figure 9 elaborates one possibility of combining all these methods for achieving reliable output.

All the algorithms were implemented using C++ in Net 2008 platform and tested with biogas data which were collected online form a biogas plant using NIR spectroscopy for a period of seven months with a frequency of twelve data points per day (*i.e.* every second hour). Among the different parameters, the H_2 content measured in ppm was selected, which has considerable amount of variations during the process. Data of each month was considered as a segment, where each segment consists of 350 - 400 data points. The proposed detection methods were applied on each segment with different criteria. Furthermore, another data set of around 4800 data points, concentrations of volatile fatty acid (VFA), was selected for checking segmenting capabilities of the method.

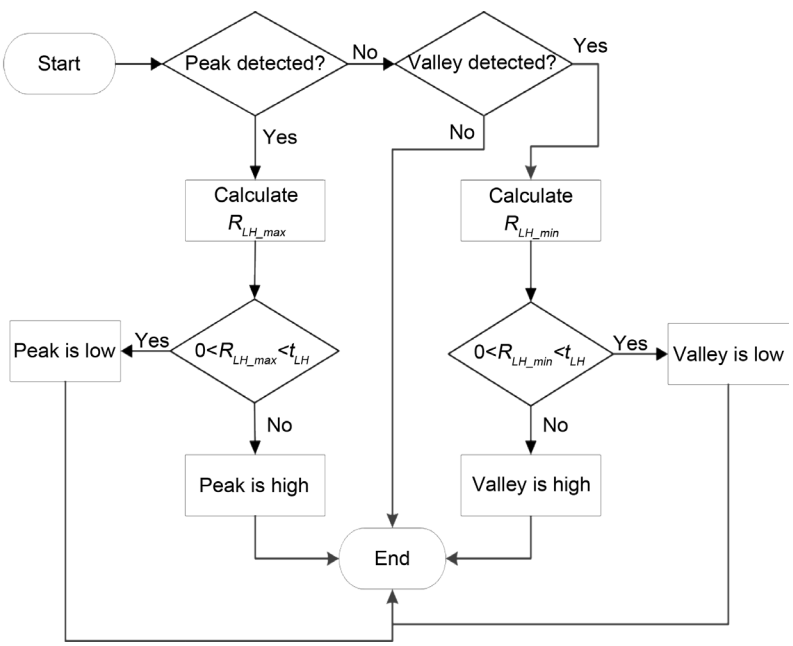


Figure 8. High and low extrema detection algorithm for "MMS-LH filter". Compression of predefined threshold t for R_{LH_max} and R_{LH_min} allows distinguishing low and high extrema.

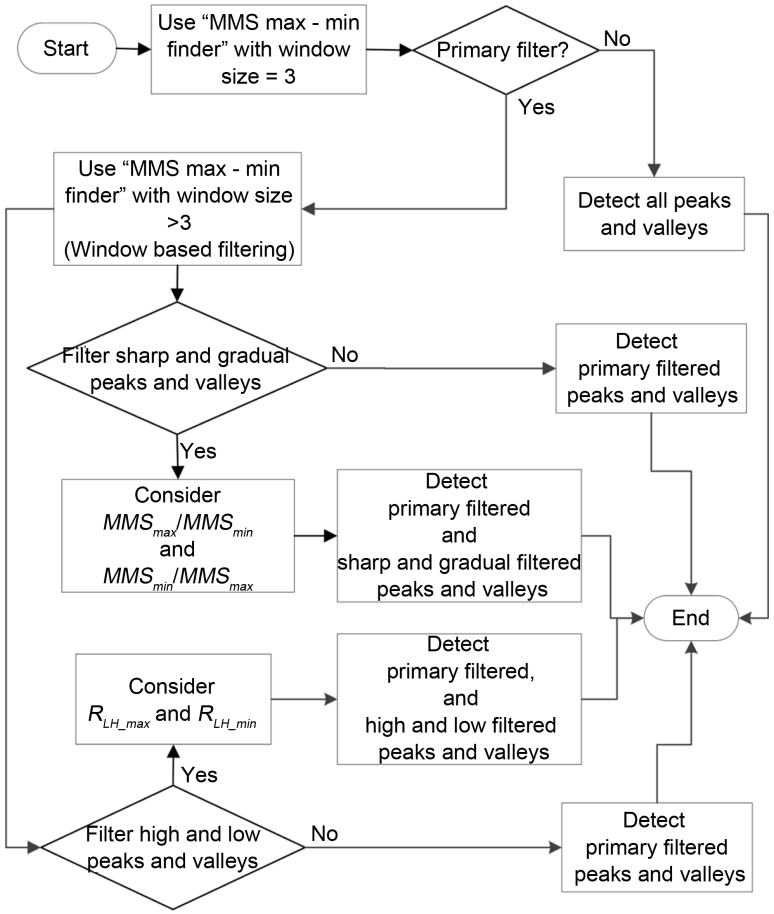


Figure 9. One possible way of combining all the developed methods for harvesting quality output.

3. Results and Discussion

3.1. Identifying Extrema

Each plot (a) and (b) of **Figure 10** contains between 350 and 400 data points and shows the identified extrema using the proposed "MMS max-min finder", which is based on Equation (7) and Equation (9). In both situations all the extrema were detected with a window size of three ($W = 3$), which is the smallest valid size of the window. Results show detection of all the extrema with 0% error. However, there is an interesting feature about detections, which can be sometimes defined as an incorrect detection as seen in **Figures 10(c)-(f)**. Plot (c) and (d) of **Figure 10** show the case where two consecutive maxima with the same value and two consecutive minima with the same value, respectively. When $W = 3$, usually both the adjacent extrema of a certain extremum have opposite extremum type (e.g.: for a maximum, adjacent members are two minima). If one adjacent extremum is with the same type extremum (e.g.: for a maximum, one adjacent member is a maximum) implies that the intermediate points of relevant points have the same value ((d) of **Figure 10**). Using the same criteria these detections can be excluded, if necessary.

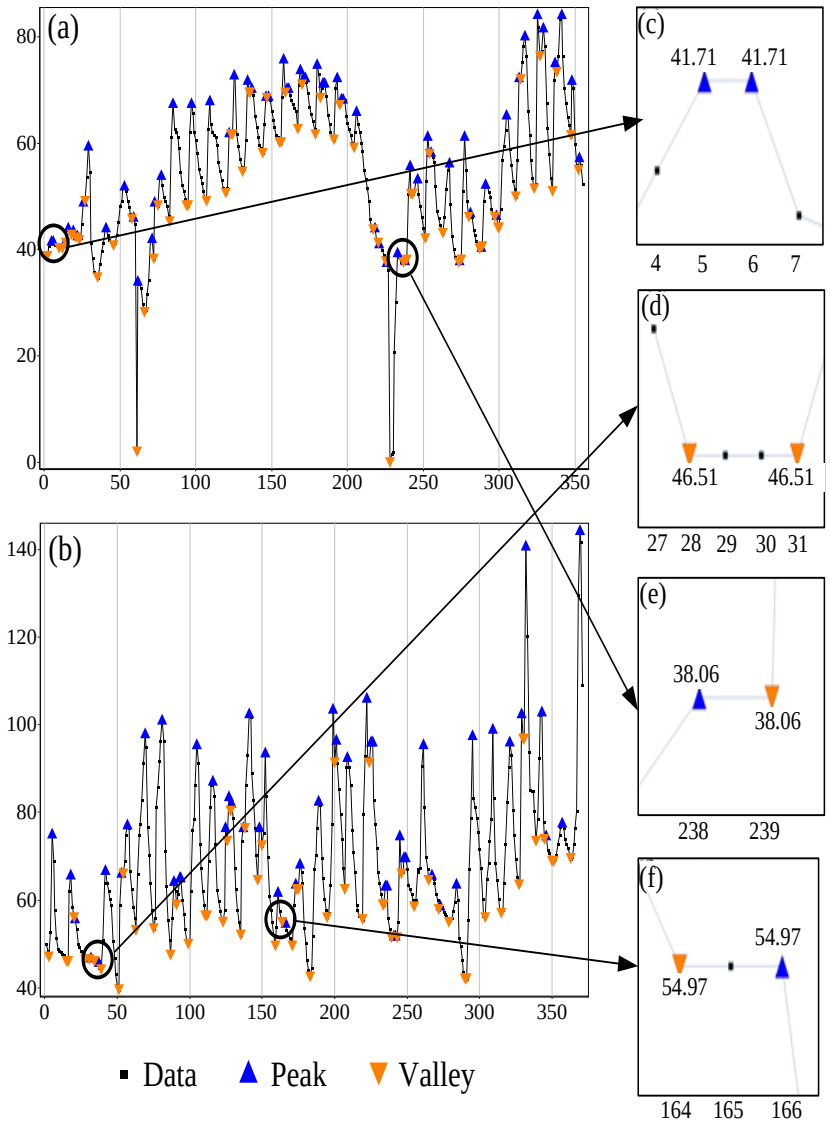


Figure 10. Two plots of H_2 content of biogas data in two different months are presented measured in ppm. All the maxima in both the data sets ((a) and (b)) were identified by the new method with the window side is three ($W = 3$). Plots (c), (d), (e), and (f) show identification of special situations as extrema, even though they are existing derivative methods not consider as extrema situations.

Plot (e) and (f) of **Figure 10** show other different situations, where it has consecutive minima and maxima of the same value. This also implies that the intermediate points have the same value ((f) of **Figure 10**). If consecutive maxima have same values and the order of occurrence is maximum then minimum, it can be considered as a discrete saddle region in an increasing data segment ((e) of **Figure 10**). In the same manner, if the two consecutive extrema have same value and the order of occurrence is minimum then maximum, it can be considered as a discrete saddle region in a decreasing data segment ((f) of **Figure 10**). Using the same criteria these detections can be excluded, if necessary.

3.2. Identifying Dominating Extrema (Primary Filtering of Peaks and Valleys)

The same two data sets shown in Figure 10 were filtered using MMS-WBF (MMS Window based filtering) method for identifying the dominant extrema using a window size of 9 ($W = 9$). Results of the detection process are shown in Figure 11 plots (a) and (b) demonstrate that the MMS-WBF was capable to identify 50% and 59% of all extrema as dominating extrema, respectively. However, out of the identified extrema in plots (a) and (b), there are 0.12% and 0.09% of small peaks which are identified as dominating extrema. These extrema cannot be visually justified as dominating extrema. Nevertheless, numerically they are the dominating extrema in the considered window size. One possible option is to increase the window size, thus covering more data which enhances the capability of removing more non-dominating extrema. However, when $W > 3$, all

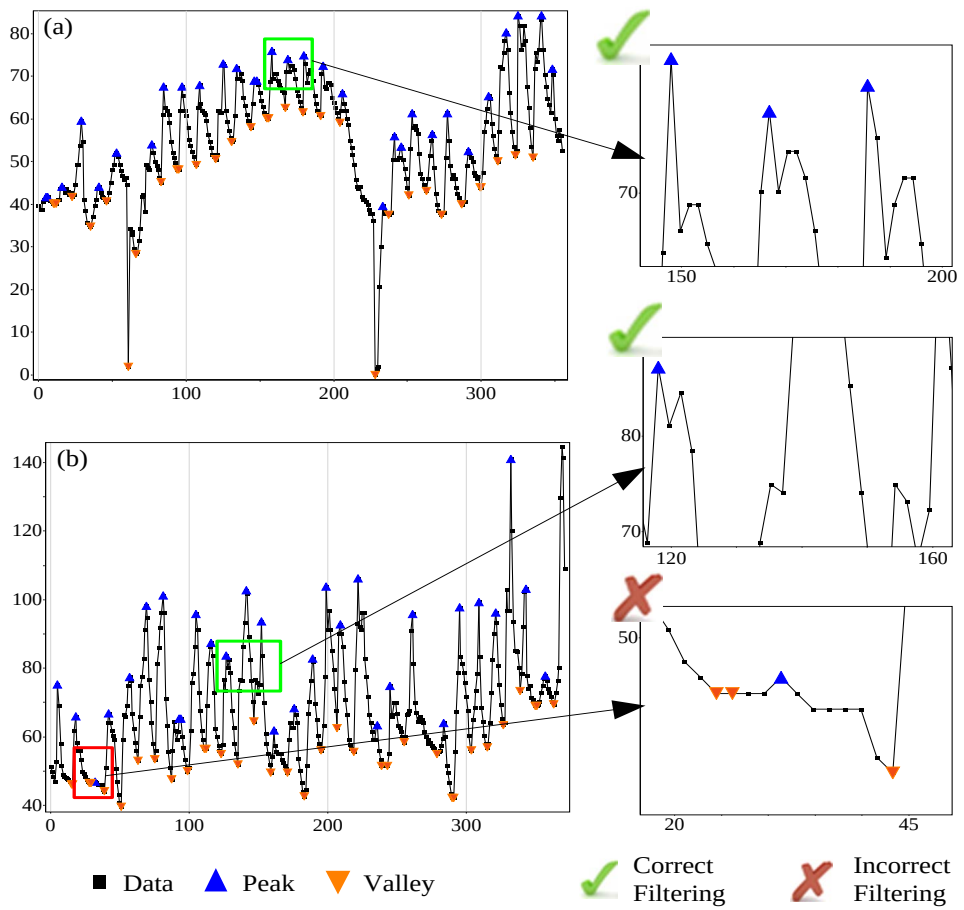


Figure 11. Plots (a) and (b) show the same data as plots (a) and (b) in Figure 10, filtered with MMS-WBF with a window size of nine data points ($W = 9$). MMS-WBF was capable of identifying 53% and 58% of all extrema as dominating extrema. However, MMS-WBF identified 0.12% and 0.09% of extrema in plots (a) and (b) as dominating extrema, which cannot be visually justified as dominating extrema. Though those are cannot be justifies as dominating extrema, mathematically they are the dominating extrema in the considered window size. One possible option is to increase the window size, thus the window would cover more data points. This will remove more non-dominating extrema once a significant dominating extremum exists.

the candidate points have not been checked. This is a disadvantage of increasing the window size for filtering non-dominating extrema. In plot (d) of Figure 11, at the end of the data set shows such an unidentified dominating peak due to $W > 3$ situation.

The combination of MMS max-min finder and MMS-WBF can be used in online data checking. For that, first the window size (W) has to be defined, and then the window accumulates the data, after which the desired detection technique is applied and eventually the extrema are located. Subsequently, window is advanced by one data point and awaits the next data point. After the next point is captured, the extrema-check is performed again. This process is propagated throughout the process for locating extrema in an online environment.

3.3. Sharp and Gradual (Flat) Extrema Filtering

Figure 12 shows the results in relation with sharp and gradual extrema detection performed based upon R_{Mm} and R_{mM} as defined in Equation (15) and Equation (16), respectively. Value of t_{Mm_mM} for R_{Mm} and R_{mM} was set as 1 ($k = 1/(n-1)$). Plot (a) and (b) of Figure 12 show the filtering of extrema, first with MMS-WBF for $W = 3$ and then with MMS-SG filter. Plot (c) and (d) of Figure 12 shows the filtering of extrema with MMS-WBF in the case of a window size of 9 ($W = 9$) and then with MMS-SG filter. When compared, plots (a) and (b) of Figure 12 show 78% and 77% less number of all extrema than number of extrema shown in plots (a) and (b) of Figure 10. When the W is small ($W = 3$) filter excludes some extrema seems to be very high (V_1, P_1, P_2, and P_3 shown in plots (a) and (b) of Figure 12), which can be considered as wrong detection. However, according to Equation (11) and Equation (13), rejections of those points are mathematically correct. This happens due to usage of small window size for extrema detection. Thus, one solution for overcoming this situation is to use lager window size.

Plots (c) and (d) in Figure 12 show identification of V_1, P_1, P_2, and P_3 after increasing the window size to nine ($W = 9$). After applying large W ($W = 9$) almost all the flat extrema have been rejected. Even after increasing the W still extrema such as P_4 are remaining, because W is not big enough to reject such points (*i.e.* in the selected window size, the extremum point is located significantly away from other points). In general, plots (c) and (d) of Figure 12 show 0.46% and 0.75% fewer extrema in comparison with plots (a) and (b) of Figure 12 and all the detections and rejections are agreed with the developed method. Therefore, the ratios MMS_{max}/MMS_{min} and MMS_{min}/MMS_{max} can be considered as filtering criteria and a reliable technique for filtering sharp and gradual (flat) extrema.

3.4. High and Low Extrema Filtering

As per the results shown in Figure 10 and Figure 12 it is very clear that the "primary filtering" and consideration of MMS_{max}/MMS_{min} and MMS_{min}/MMS_{max} are not capable of filtering extrema based on magnitude of their prominence or crater. The results shown in Figure 13 are the results in relation with the method MMS-LH, which is intensively developed focusing on filtering extrema with low prominence or crater.

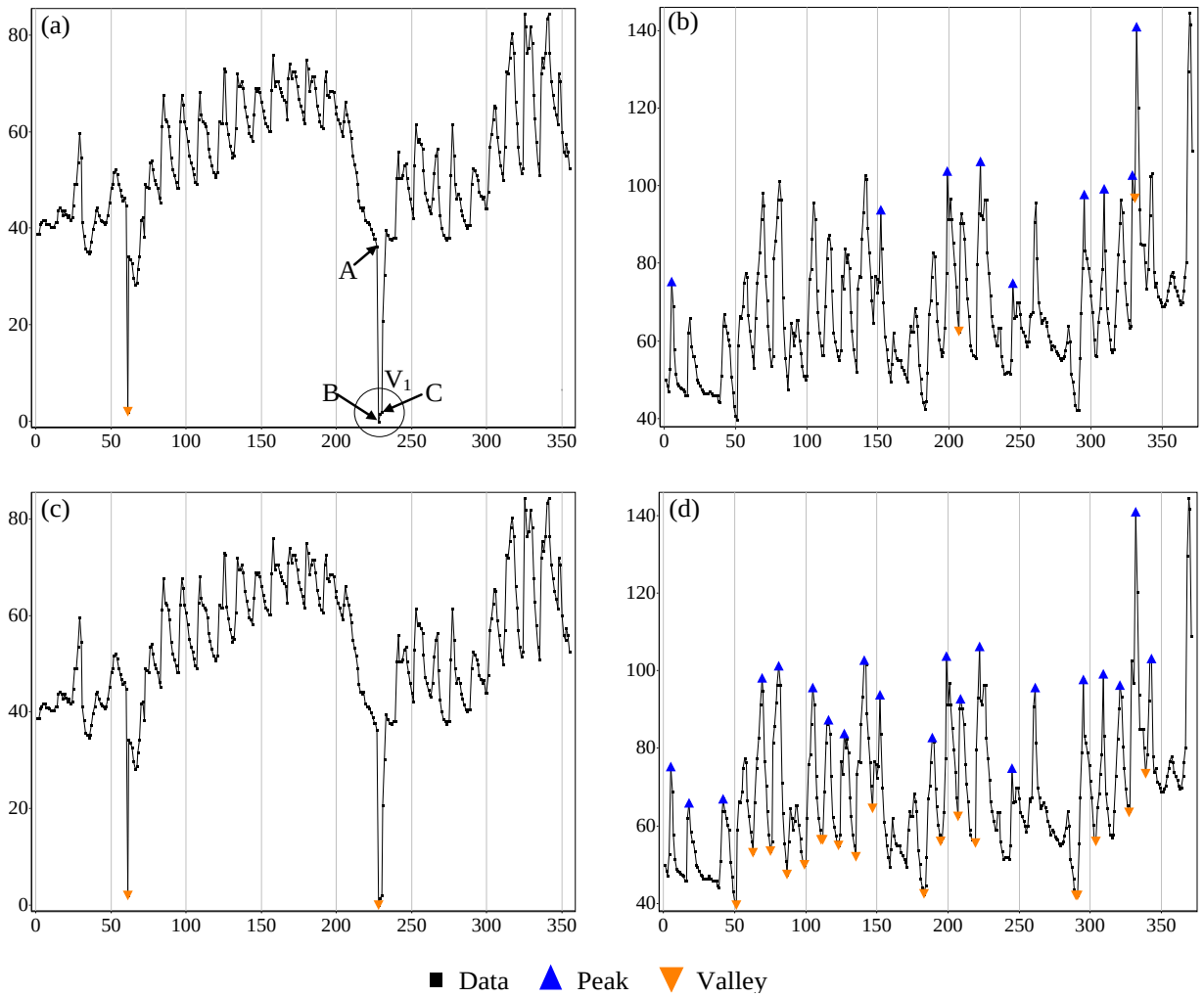


Figure 12. Filtering of sudden and gradually developed (flat) extrema using MMS-SG technique. Data in plots (a) and (b) were first checked for extrema with a window of size three with MMS-WBF. Data in plots (c) and (d) were first checked for extrema with a window of size nine with MMS-WBF. Then ratios MMS_{max}/MMS_{min} and MMS_{min}/MMS_{max} considered and all the plots were checked for sudden and gradually developed extrema with threshold value $t_{Mm\_mM} = 1$. When the window size is small, extrema such as V_1, P_1, P_2, and P_3 remain undetected. However, increasing the window size let those points to be detected (plots (c) and (d)). Even after increasing the window size, points that have very small extrema such as P_4 will de detect as an extrema.

Before applying MMS-LH, data points (plots (a) and (b) of **Figure 13**) were first checked for extrema with a window size three with MMS-WBF and data in plots (c) and (d) of **Figure 13** were first checked for extrema with a window size nine with MMS-WBF. Point V_1 in **Figure 13(a)**, which seems to be a valley with high crater, yet remains as unidentified. To be qualified as an extrema with higher prominence or crater, first, the extremum must be a perfect extremum. However, with $W = 3$, V_1 is not a perfect extremum. Therefore, the rejection is logical as well as mathematically correct. Nevertheless, in **Figure 13(c)**, point V_1 is identified as a valley, because the large window size ($W = 9$) makes V_1 a nearly perfect extremum. Therefore, using $W > 3$ with appropriate filter criteria the method can be used for filtering extrema with low and

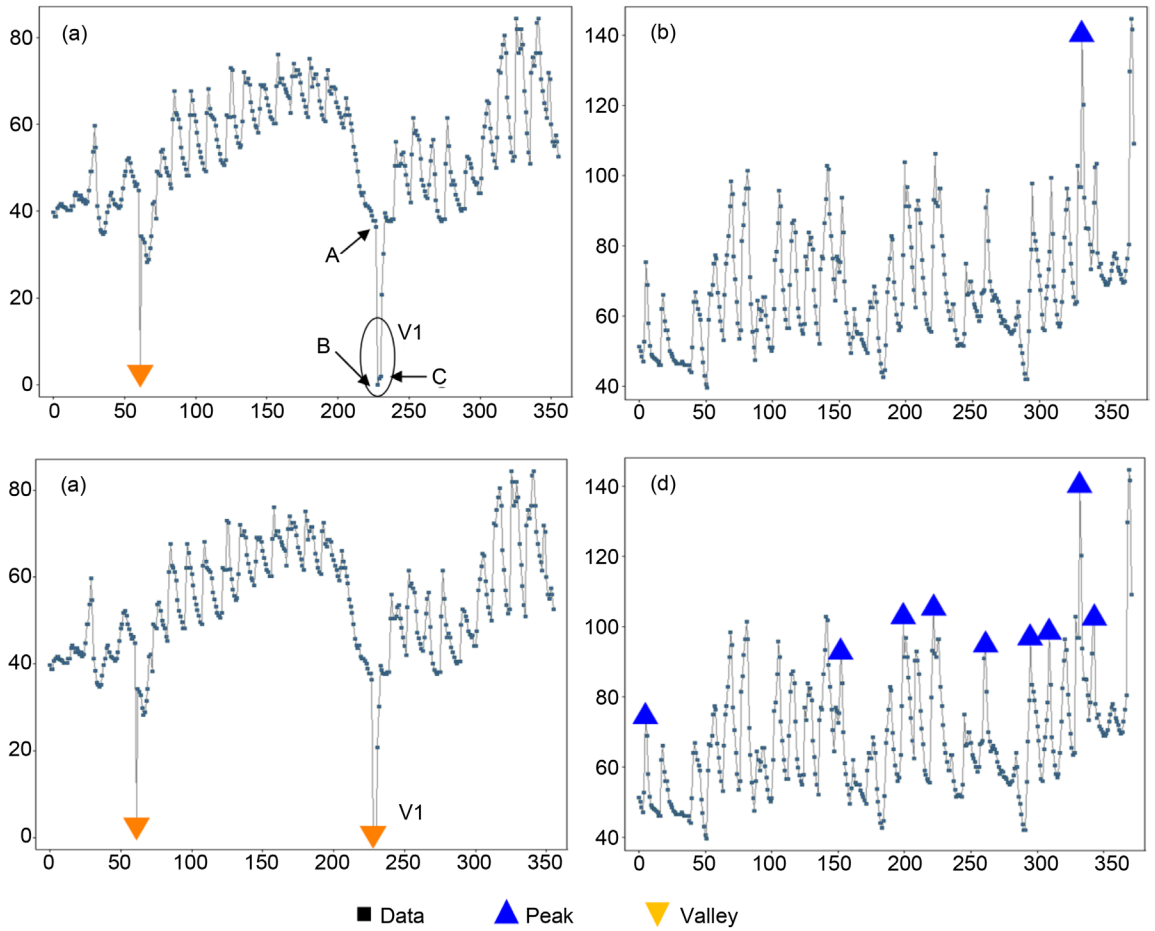


Figure 13. Filtering of low and high extrema using MMS-LH filtering technique. Data in plots (a) and (b) were first checked for extrema with a window of size three with MMS-WBF. Data in plots (c) and (d) were first checked for extrema with a window of size nine with MMS-WBF. Then R_{LH_max} and R_{LH_min} were considered and all the plots were checked for low and high extrema with threshold value $t_{LH} = 0.05$. When the window size is small, extrema such as V_1 remain undetected. The reason is for such detection is that the one point (point C) is located very close to the extremum (extremum is not a perfect extremum). However, increasing the window size ($W = 9$) makes V_1 a nearly perfect extremum and detected in plot (c).

high prominence or crater.

3.5. Drawbacks of Using Large Window Size for Extrema Filtering

In Figure 10, Figure 12, and Figure 13 plots with lager window size, $(W-1)/2$ points from the beginning as well as from the end will not be checked, where W is the window size. If there are matching extrema existing in these regions, they also remain as unidentified (Figure 12 and Figure 13). This is disadvantageous when using large window size, on the other hand if there are enough data points available, the issue is resolved. However, this is a problem for small data sets. Checking unchecked areas with a smaller window is one possibility for resolving this issue. However, results from two different window sizes will lead to violate the homogeneity of the results. The second method is to start the window before a certain number of data points ($w/2$). Then part of the

window is laid on a non-data region. Using a suitable padding, this part can be filled. For example, the entire data in non-data region in the start can be padded with starting value. Also, at the end suitable padding technique can be used to fill the part of the window in the non-data region.

3.6. Possibility of Use as a Data Segmentation Technique

Usually, dominating peaks and the valleys can be considered as turning points of a certain property of a signal, if those dominating extrema are not outliers. Thus, dominating peaks and valleys are good points for segmenting a signal as well as identifying general trends. Figure 14 shows an attempt to accomplish such a segmenting approach using the developed method. Figure 14 contains a data set with around 4600 data points and only the MMS-WBF (dominating extrema identification technique) technique was applied as the filtering technique. For testing segmenting capabilities of the method, considerably large W was used ($W = 155$ in plot (a) and $W = 255$ in plot (b) of Figure 14). In both situations segmentation and general trend identification shows highly promising capabilities. Existences of more than one adjacent similar types of extrema violate the trend identification and segmentation (*i.e.* existence of maximum after a maximum instead of minimum). Circled areas in plot (a) of Figure 14 show two such occurrences. However, removing unnecessary adjacent peaks or valleys while keeping singular important peaks or valleys, is one solution for overcoming this problem. Thereby it is necessary to develop a methodology for removing less important extrema. Increasing the W is another way of overcoming the said drawback. Plot (b) of Figure 14 shows situation of increased W and detection with less adjacent same type of

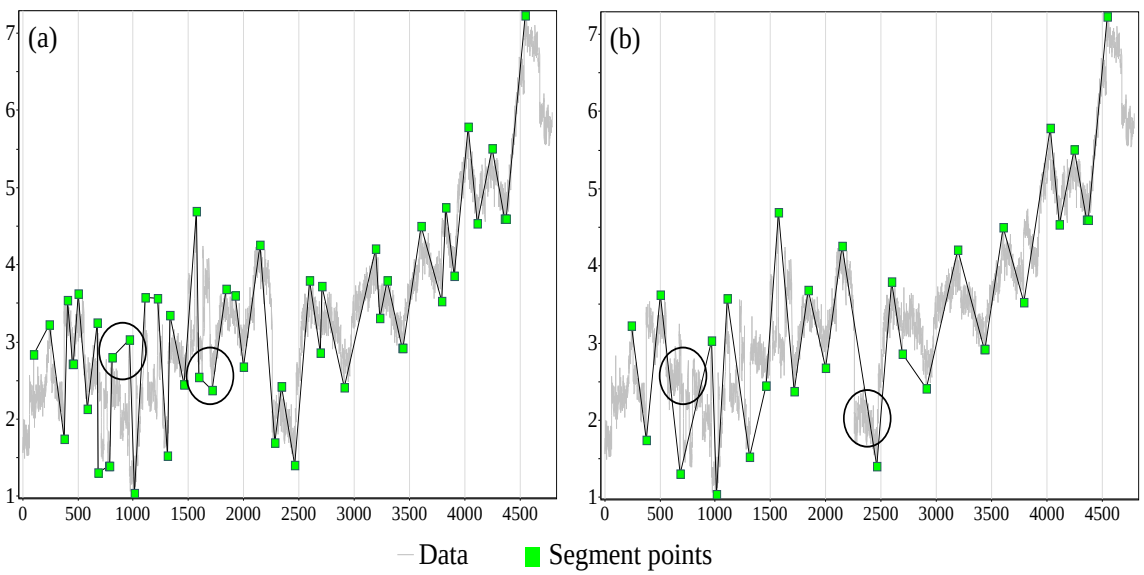


Figure 14. Usage of "*MMSmax-min finder*" as a segmentation technique and trend identification technique. Plot (a) and (b) use window size 155 and 255, respectively. When the window side is low ($W = 155$) segmentation and trend identification is distracted due to occurrence of adjacent same type extrema. In plot (a) such two occurrences were circled. Increasing the window size produces better segmentation as shown in plot (b). However, this leads to ignore some trends as circled in plot (b).

extrema than plot (a) of Figure 14. However, this technique lead to ignorance of some features in the signal as circled in plot (b) of Figure 14. Therefore, determining of proper W is an essential factor for better identification of segments as well as trends. Nevertheless, the method can be used for at least fast segmentation and trend identification method.

4. Conclusion

The introduced extrema finding method named as "MMS Max-Min finder" and three different extrema filtering methods named as MMS-Window Based Filter (MMS-WBF), MMS sharp and gradual extrema filter (MMS-SG), and MMS low high extrema filter (MMS-LH) are non-parametric. Therefore, filtering can be done without considering domain dependent parameters such as height and width of an extremum. Results prove that the detection is capable of identifying all the extrema with 0% error. When the window size is nine ($W = 9$) MMS-WBF reported 0.12% and 0.09% wrong detections. However, a combination of MMS-WBF and MMS-LH filter with window size nine ($W = 9$) was capable of eliminating the error. Despite of the dynamic nature of the data, the results were consistent and robust for the same detection criteria. Thus, using proper window size, it is possible to achieve robust and consistent outcome with dynamic data such as biogas data. Furthermore, MMS-WBF shows promising outcome in the direction of segmenting and trend identification of signals. Hence, MMS-WBF can be enhanced as a segmenting and trend identification technique.

Acknowledgements

This work was supported by the German Research Foundation (DFG) and the Technical University of Munich (TUM) in the framework of the Open Access Publishing Program. Also, we are grateful to the German Academic Exchange Service (Deutscher Akademischer Austauschdienst, DAAD) for providing a scholarship to KKLB Adikaram during the research period.

References

[1] Mavron, V.C. and Phillips, T.N. (2007) Maxima and Minima. In: Mavron, V.C. and Phillips, T.N., Eds., *Elements of Mathematics for Economics and Finance*, Springer, London, 137-158.

[2] Sande, H.V., Henrotte, F. and Hameyer, K. (2004) The Newton-Raphson Method for Solving Non-Linear and Anisotropic Time-Harmonic Problems. *The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, **23**, 950-958. http://dx.doi.org/10.1108/03321640410553373

[3] Chioua, M., Srinivasan, B., Guay, M. and Perrier, M. (2007) Dependence of the Error in the Optimal Solution of Perturbation-Based Extremum Seeking Methods on the Excitation Frequency. *The Canadian Journal of Chemical Engineering*, **85**, 447-453. http://dx.doi.org/10.1002/cjce.5450850407

[4] Khan, I.R. and Ohba, R. (1999) Closed-Form Expressions for the Finite Difference Approximations of First and Higher Derivatives Based on Taylor Series. *Journal of Computational and Applied Mathematics*, **107**, 179-193.

http://dx.doi.org/10.1016/S0377-0427(99)00088-6

[5] Gilgen, H. (2006) Univariate Time Series in Geosciences: Theory and Examples. Springer, Berlin.

[6] Zou, H.-F., Zhang, Y.-K. and Lu, P.-C. (1991) The Prediction of the Peak Width at Half Height in HPLC. *Chinese Journal of Chemistry*, **9**, 237-244. http://dx.doi.org/10.1002/cjoc.19910090307

[7] Antoniadis, A., Bigot, J. and Lambert-Lacroix, S. (2010) Peaks Detection and Alignment for Mass Spectrometry Data. *Journal de la Société Française de Statistique*, **151**, 17-37.

[8] Jeffries, N. (2005) Algorithms for Alignment of Mass Spectrometry Proteomic Data. *Bioinformatics*, **21**, 3066-3073. http://dx.doi.org/10.1093/bioinformatics/bti482

[9] Sauve, A.C. and Speed, T.P. (2004) Normalization, Baseline Correction and Alignment of High-Throughput Mass Spectrometry Data. Proceedings Gensips.

[10] Mtetwa, N. and Smith, L.S. (2006) Smoothing and Thresholding in Neuronal Spike Detection. *Neurocomputing*, **69**, 1366-1370. http://dx.doi.org/10.1016/j.neucom.2005.12.108

[11] Tzallas, A.T., Oikonomou, V.P. and Fotiadis, D. (2006) Epileptic Spike Detection Using a Kalman Filter Based Approach. *IEEE Engineering in Medicine and Biology Society Conference*, **1**, 501-504. http://dx.doi.org/10.1109/iembs.2006.260780

[12] Gelb, A. (1974) Applied Optimal Estimation. MIT Press, Boston.

[13] Shim, B., Min, H. and Yoon, S. (2009) Nonlinear Preprocessing Method for Detecting Peaks from Gas Chromatograms. *BMC Bioinformatics*, **10**, 378. http://dx.doi.org/10.1186/1471-2105-10-378

[14] Scholkmann, F., Boss, J. and Wolf, M. (2012) An Efficient Algorithm for Automatic Peak Detection in Noisy Periodic and Quasi-Periodic Signals. *Algorithms*, **5**, 588-603. http://dx.doi.org/10.3390/a5040588

[15] Györfi, L., Kohler, M., Krzyzak, A. and Walk, H. (2006) A Distribution-Free Theory of Nonparametric Regression. Springer, New York.

[16] Roberts, S.J. (1997) Parametric and Non-Parametric Unsupervised Cluster Analysis. *Pattern Recognition*, **30**, 261-272. http://dx.doi.org/10.1016/S0031-3203(96)00079-9

[17] Wasserman, L. (2006) All of Nonparametric Statistics. Springer, New York.

[18] Kothari, C.R. (2004) Research Methodology: Methods and Techniques. New Age International (P) Limited, Delhi.

[19] Li, J., Ray, S. and Lindsay, B.G. (2007) A Nonparametric Statistical Approach to Clustering via Mode Identification. *Journal of Machine Learning Research*, **8**, 1687-1723.

[20] Adikaram, K.K.L.B., Hussein, M.A., Effenberger, M. and Becker, T. (2014) Outlier Detection Method in Linear Regression Based on Sum of Arithmetic Progression. *The Scientific World Journal*, **2014**, Article ID: 821623. http://dx.doi.org/10.1155/2014/821623

[21] Adikaram, K.K.L.B., Hussein, M.A., Effenberger, M. and Becker, T. (2015) Universal Linear Fit Identification: A Method Independent of Data, Outliers and Noise Distribution Model and Free of Missing or Removed Data Imputation. *PLoS ONE*, **10**, e0141486. http://dx.doi.org/10.1371/journal.pone.0141486

[22] Han, J., Kamber, M. and Pei, J. (2006) Data Mining, Southeast Asia Edition: Concepts and Techniques. Elsevier Science, Amsterdam.

[23] Shalabi, L.A., Shaaban, Z. and Kasasbeh, B. (2006) Data Mining: A Preprocessing Engine. *Journal of Computer Science*, **2**, 735-739. http://dx.doi.org/10.3844/jcssp.2006.735.739

symmetry

MDPI

Article

Continuous Learning Graphical Knowledge Unit for Cluster Identification in High Density Data Sets

K.K.L.B. Adikaram [1,2,3,*], Mohamed A. Hussein [1], Mathias Effenberger [2] and Thomas Becker [4]

1 Group Bio-Process Analysis Technology, Technische Universität München, Weihenstephaner Steig 20, 85354 Freising, Germany; mohamed.hussein@tum.de
2 Bavarian State Research Center for Agriculture, Institute for Agricultural Engineering and Animal Husbandry, Vöttinger Straße 36, 85354 Freising, Germany; mathias.effenberger@lfl.bayern.de
3 Computer Unit, Faculty of Agriculture, University of Ruhuna, Mapalana, Kamburupitiy 81100, Sri Lanka
4 Lehrstuhl für Brau- und Getränketechnologie, Technische Universität München, Weihenstephaner Steig 20, 85354 Freising, Germany; tb@tum.de
* Correspondence: lasantha@daad-alumni.de; Tel.: +94-71-4951248; Fax: +94-41-2292384

Academic Editors: Doo-Soon Park and Shu-Ching Chen
Received: 8 June 2016; Accepted: 5 December 2016; Published: 14 December 2016

Abstract: Big data are visually cluttered by overlapping data points. Rather than removing, reducing or reformulating overlap, we propose a simple, effective and powerful technique for density cluster generation and visualization, where point marker (graphical symbol of a data point) overlap is exploited in an additive fashion in order to obtain bitmap data summaries in which clusters can be identified visually, aided by automatically generated contour lines. In the proposed method, the plotting area is a bitmap and the marker is a shape of more than one pixel. As the markers overlap, the red, green and blue (RGB) colour values of pixels in the shared region are added. Thus, a pixel of a 24-bit RGB bitmap can code up to 2^{24} (over 1.6 million) overlaps. A higher number of overlaps at the same location makes the colour of this area identical, which can be identified by the naked eye. A bitmap is a matrix of colour values that can be represented as integers. The proposed method updates this matrix while adding new points. Thus, this matrix can be considered as an up-to-time knowledge unit of processed data. Results show cluster generation, cluster identification, missing and out-of-range data visualization, and outlier detection capability of the newly proposed method.

Keywords: big data; clustering; contour lines; data and knowledge visualization; knowledge retrieval; mining methods and algorithms; missing data; real-time systems

1. Introduction

Plotted data are visually cluttered by overlapping data points. Reducing, avoiding and reformulating (as a cluster) such overlap are the three major techniques recommended for clutter reduction in the data visualization field [1–5]. However, especially with large numbers of overlap, reducing and avoiding techniques are not feasible [4,6] and reformulation is a complex task [4,7]. In contrast, the method we introduce in this paper incorporates overlaps to generate density clusters without reducing, avoiding or reformulating overlaps. The proposed method requires more overlaps for better cluster formation and better visualization, which contrasts the general practice. Furthermore, the proposed method can be considered as an anytime cluster formation technique (without a separate cluster identification algorithm), which provides faster cluster generation than online methods [8].

Limiting the number of data points on a plot is the most popular technique to avoid overlaps [6]. Typically, when there are few data points, the probability of an overlap occurring is low. Changing the opacity of data points and displacing data points that address the issue of overlapping are among overlap reducing techniques [4]. Changing the opacity of data points enables the identification of small

numbers of underlying or partially overlapping data points [4]. Displacement (agitating or jittering) is a technique that randomly moves a data point over a small distance to overcome the overlap [4,6,9]. However, when too many overlaps occur, limiting the number of data points, changing the opacity or displacing data points does not work very effectively. Reformulating overlapped data points as density clusters is a solution for eliminating the overlap that represents different overlapping density ranges as different clusters, even with big data. The final output depends on the cluster identification algorithm.

In real-world applications, overlapping increases the colour intensity of a redrawn area in contrast to other areas of a picture and hence generates a visually identifiable cluster. This phenomenon indicates that the number of overlaps is proportional to the colour intensity. We have found that step-by-step increase of the colour value of a pixel in a bitmap resembles the phenomenon of overlapping. This motivated us to develop a method to overcome overlapping data points by means of a bitmap. To deal with such a situation, we use bitmaps, clustering techniques, cluster representation techniques and contour lines.

A bitmap is the major component of our proposed method for plotting data and is a raster graphics image comprising a rectangular array of pixels [10]. There are different methods to represent the colour of pixels. The 8-bit RGB and 8-bit red, green, blue, and alpha (RGBA) formats are the most popular methods [6]. The RGB format uses the red, green and blue colour channels, whereas the RGBA format uses the red, green, blue and alpha colour channels, where the alpha channel determines the level of transparency of a colour [11]. In both formats, each colour component has 8 bits representing 256 codes (values of 0–255) for each of the R, G and B channels. The colour value of a pixel is represented as a combination of channel values separated by commas (e.g., dark green in RGB: (0, 100, 0) and in RGBA: (0, 100, 0, 0)). In general, if the total effective bit length of a pixel is n, a pixel can represent 2^n colours. Thus, the 8-bit RGB and 8-bit RGBA formats can represent 2^{24} and 2^{32} different colours, respectively. If the whole bit series of a pixel is considered as a single channel, then each pixel is a number of base 256 (e.g., dark green in RGB: (0, 100, 0) = 25,600). Thus, a bitmap is a matrix that contains numerical values. The visual representations of these numbers indicate different colours. If these numbers are used to represent up-to-time processed data, the bitmap becomes an up-to-time knowledge unit. This is a different usage of bitmaps for representing data.

As already mentioned, clustering is the most popular technique and is capable of eliminating overlaps, especially with big data. Overall, the process of cluster identification is comprised of three components: data (database), algorithm and cluster information (processed data). The processed data are represented in different forms, primarily as data in a database or as visualized output. In almost all clustering methods, clusters are first determined by a separate algorithm and then visualized by a suitable technique. Typically, clusters are represented as a visual output such as a histogram, scatterplot or scatterplot matrix [12]. The graphical output is used only as a visual aid to facilitate the understanding of the clusters. The graphical output cannot be used as a source of processed data for another algorithm. After the cluster analysis process, it is sometimes still possible to find some unidentified clusters. In this case, the general practice is to modify the existing algorithm or introduce a new algorithm to identify new patterns. In contrast, the proposed method does not require a separate algorithm to identify density clusters as it generates clusters on the bitmap while adding data points.

This new approach based on overlapping data points provides a mechanism to access and view information directly without further processing. Our findings are highly relevant for applications in fields of big data visualization, process control, data modelling and bit data representation.

1.1. Related Work

As mentioned above, the proposed method is based on clustering. Therefore, clustering techniques used for clutter reduction are the most related techniques. Clustering or cluster analysis is an unsupervised (i.e., it requires no training data sets) data classification method [13–15] for identifying homogenous groups of objects known as clusters [16–18]. Cluster analysis is used in many fields, such as knowledge discovery in databases (KDD), pattern recognition, image analysis, and machine

learning and data mining [19–24]. Typically, existing clustering techniques used for clutter reduction eliminate overlaps by representing a group of data points or a group of lines by means of a single data point or line [2,3,25,26]. After identifying clusters, these clusters can be used to understand and identify correlations, patterns, features and outliers in the data set.

There are several special methods based on clustering that are intensively related to elimination of overlaps such as heat maps [27–29] variable binned scatter plots [30], hierarchical multi-class sampling [5] and Splatter-plots [31]. A heat map represents data in a matrix using a colour scale that represents the values in the matrix. The quadrat method is a popular technique for creating such a matrix [32–34]. The hierarchical multi-class sampling technique is another effective technique for showing specific feature-clusters by enhancing density contrast by means of different colours. The visual exploration system developed by Haidong et al. is a good example of such an approach [5]. The variable binned scatter plots technique allows visualization of large amounts of data without overlaps [30]. This technique incorporates variable size bins [35] and classifies them into different groups using a colour scheme. The Splatter-plots technique automatically groups dense data points into contours and samples the remaining points. Colour blending is used to reveal the relationship between data subgroups after processing the whole data set. Pre-processing of the original data is a compulsory step for all these cluster visualization techniques to convert the original data into the desired format. In contrast, the proposed method does not require pre-processing of the original data prior to visualization.

A contour line is a different technique used to indicate clusters. The term contour line (also known as isolines, isopleths or isarithms) was originally used in the cartography field. According to Imhof, "Contour lines are lines on the map depicting the metric locations of points on the Earth's surface at the same elevation above sea level" [36]. However, contour lines are also used to map equal values for other properties such as temperature or pressure. In a contour map, contour lines with a certain interval display different values. The main feature of a contour map is that each contour line indicates a certain value, and it is impossible to have crossing contour lines. This technique is used to show the borders of clusters in kernel identification methods [37,38]. The proposed method creates contour lines automatically to separate different density clusters.

2. Methodology

In a bitmap, the coordinates of a pixel specify its location. As in a common plot, these coordinates can be used to represent parameter values (dimensions) that can be considered as data. In addition, the colour value of the pixel can be mapped with information related to the data, e.g., the number of overlaps (or data density) in the proposed method. Furthermore, updating the colour value of the pixel resembles updating the information. An individual pixel is capable of holding 2^n number of different values; thus, a pixel is a memory cell or a knowledge cell. Therefore, we introduce a bitmap that forms a graphical knowledge unit (GKU) out of knowledge cells to represent data and information.

The clustering range of influence is defined as the radius around the cluster centre (the highest density data point of the cluster) [39]. When the clustering range of influence is small, many small clusters are produced. In contrast, when the clustering range of influence is large, a few large clusters are produced. When developing the proposed method, we used the size and shape of a marker (a graphical symbol of a data point) and the position of a data point in the marker to characterize the clustering range of influence. The shape and size of the marker are used to depict the effective clustering range of influence. Typically, the marker is comprised of several pixels. From these pixels, we use one pixel to represent the data point. For example, if the marker is a circle with a diameter of x pixels, the data point is represented by the pixel in the centre of the circle. Figure 1 shows an example of such data point. Here we have highlighted the location of the data point using a different colour, whereas in reality, no distinct colour is used for this.

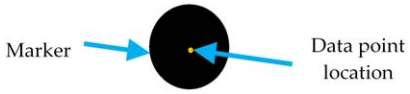


Figure 1. Definition of marker in graphical knowledge unit (GKU): The marker is a circle (radius = 10 pixels), and the RGB colour value of the circle is (0, 0, X), where $0 \leq X \leq 255$. The centre of the circle represents the data point (highlighted).

2.1. Colour Coding Method

When an overlap occurs, there is always a shared area (intersection) between both existing and newly added markers. We then update the colour of the overlapping area by adding the colour values of pre-existing and newly added markers (Figure 2A). When adding colour, the colour of each pixel in the shared area is updated according to the Equations (1) and (2). We first convert the colour of a pre-existing pixel in the shared area and the corresponding pixel of the newly added marker using (1). Subsequently, we add those two values and convert the single value to an RGB value using (2). Finally, we update the considered pixel in the shared area using the new colour derived from (2). Applying this technique for all pixels in the shared area updates the colour of the shared area (Figure 2B). The colour of the new marker is applied if there is no overlap.

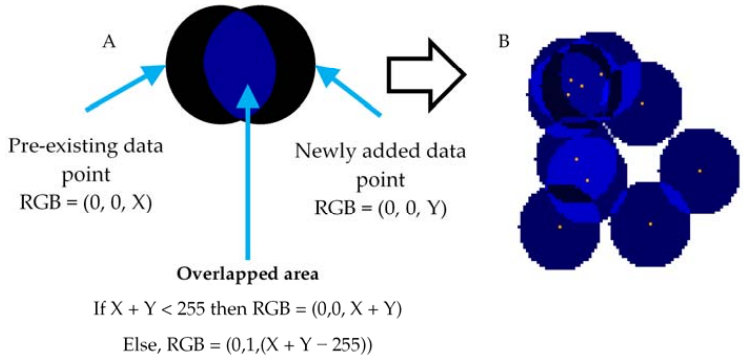


Figure 2. (**A**) Two overlapped markers; and (**B**) overlapped markers. The data point is represented by the pixel in the centre of the marker (the data point is highlighted in orange).

If C_V is the single integer colour value of a pixel, R_V is the red colour value, G_V is the green colour value and B_V is the blue colour value, then

$$C_V = R_V \times 256^2 + G_V \times 256^1 + B_V \times 256^0. \quad (1)$$

The function QUOTIENT(<numerator>, <denominator>) performs division and returns only the integer portion of the result. If $C_2 = R_V$, $C_1 = G_V$ and $C_0 = B_V$, then

$$C_i = \text{QUOTIENT}\left(C_V - \sum_{j=1}^{2} (C_{i+j} \times 2^{i+j}), 256^i\right);\ i \in \{2,1,0\} \text{ and } C_k = 0 \text{ when } k > 2. \quad (2)$$

For example, according to Equation (1), a pixel with RGB colour (1, 2, 3) can be represented as 66,051 ($1 \times 256^2 + 2 \times 256^1 + 3 \times 256^0$). In addition, when C_V = 66,051, according to Equation (2), C_2 = 1, C_1 = 2 and C_0 = 3. Because $C_2 = R_V$, $C_1 = G_V$ and $C_0 = B_V$, the RGB representation of 66,051 is (1, 2, 3). Note that only these two equations are used for density calculation and density cluster formation.

2.2. Data Preparation

Because the location (coordinates) in a bitmap is a positive integer, it is impossible to depict decimal and negative values. In addition, it is impractical to represent a very large range of numbers

with bitmaps, as this would require a very large bitmap. Applying transformation techniques is one of the ways to overcome these challenges. Base line correction is used to eliminate negative values and very large values. This gives a value range that begins from zero. Scaling up or down is applied to overcome very small and very large ranges, respectively. If the data set is a combination of negative, decimal and large values, the respective transformations must be implemented accordingly. Finally, a suitable offset is used to shift the data from the origin. Table 1 shows the basic transformation techniques used for the different value types.

Table 1. Transformation rules used to convert numbers into integers. Depending on the nature of the data, a combination of two or more techniques may be required to achieve a data set suitable to plot on a bitmap.

Value Type	Transformation Technique
Negative integer values	Base line correction. This will convert all negative values to positive values while maintaining the same regression.
Very large values	Base line correction. This will convert large numbers to small numbers while maintaining the same regression.
Decimal values	Multiplication by 10^d ($d \in \{1, 2, 3, \dots\}$). This will convert decimal values to integers (we named d as "decimal to integer factor").
Small or large range	Scale up or down. This will change the range.

2.3. Visualization of Missing and Out of Range Values

A method that can show missing data and data that have unexpected (out of range) values would provide important information for understanding the quality of the data. We have included two border regions in the GKU for recording missing values and data with out of range values. In these borders, different regions are defined to identify the nature of the missing or out of range data (Figure 3). If there are missing or out of range values, these are shown on the relevant border region of the bitmap in the same manner as regular data points (Figure 3).

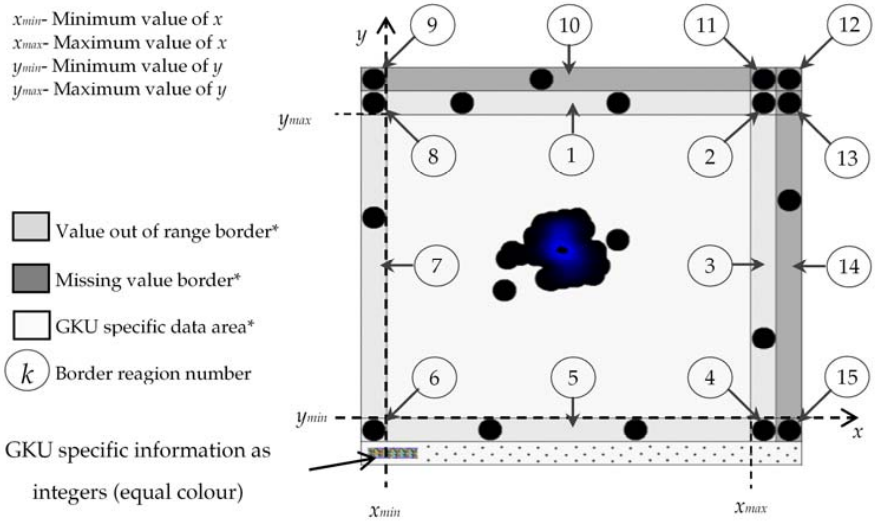


Figure 3. GKU with borders to record missing and out of range values: (1) $y > y_{max}$ at $x = x$; (2) $x > x_{max}$ and $y > y_{max}$; (3) $x > x_{max}$ at $y = y$; (4) $x > x_{max}$ and $y < y_{min}$; (5) $y < y_{min}$ at $x = x$; (6) $x < x_{min}$ and $y < y_{min}$; (7) $x < x_{min}$ at $y = y$; (8) $x < x_{min}$ and $y > y_{max}$; (9) y is missing and $x < x_{min}$; (10) y is missing at $x = x$; (11) y is missing and $x > x_{max}$; (12) both x and y are missing; (13) x is missing and $y > y_{max}$; (14) x is missing at $y = y$; and (15) x is missing and $y < y_{min}$. * Shading is used in the figure to highlight different areas. In the real GKU, there will be no shading.

2.4. Embed GKU Specific Information into Bitmap

The GKU contains several parameters related to data transformation, marker type (circle, square, etc.), marker dimensions, marker colour and border information (width of borders of missing and unexpected values). This GKU information could be stored in a separate file; however, it would be more convenient if this information was embedded in the same bitmap file as integers (=colour). Thus, at the bottom (or top) of the bitmap after (or before) the real data points, the GKU specific data are depicted with the respective colour (Figure 3). The starting location (offset) of the GKU specific data area is stored in the first unused slot of the bitmap header (Table 2).

Table 2. Bitmap header (example of an $m \times n$ pixel bitmap with red, green, and blue (RGB) (24-bit) colour scheme) * this unused slot is used to store the offset for graphical knowledge unit (GKU) specific data. BM: a value in Bitmap Header.

Header Section	Offset	Size/Bytes	Value	Description
Bitmap (BMP) Header (14 Bytes)	0	2	"BM"	Identification (ID) field
	2	4	Size of BMP header, DIB header, and Image	Size of the BMP file
	6	2	Unused *	Application specific
	8	2	Unused	Application specific
	10	4	54 Bytes (14 + 40)	Offset where the pixel array (bitmap data) can be found
Device-independent bitmap (DIB) header	12		40 Bytes	
	...			
	50			
Bitmap data	51		$m \times n \times 4$ Bytes	
	...			
	...			

As discussed above, it is necessary to convert GKU specific data into integers to embed this information into the bitmap. However, certain properties such as marker type or negative values cannot be mapped directly with the colour value of a pixel. Therefore, a protocol is required to map this information. The existing 24-bit pixel RGB structure can represent positive numbers. We refer to this as the unsigned 24-bit pixel format. In addition, using 24 bits, it is possible to represent negative integers, where the first bit is used to indicate the sign of the value (e.g., 1 = negative and 0 = positive). We refer to this as the signed 24-bit pixel format. Furthermore, any number can be represented as a product of an integer and a power of ten (e.g., $-123.45 = -12345 \times 10^{-2}$). Thus, any number can be represented (one for the integer part and the other for power of ten) with two pixels in the signed 24-bit single pixel format. Finally, the structure of the GKU specific data is designed as shown in Table 3 using relevant number representation techniques.

Table 3. Example of GKU specific data layout. The value K is the starting location (offset) of the GKU specific data area. The value of K is stored in the first unused slot of the bitmap header.

GKU Specific Data	Offset of Pixels	No. of Pixels	Content in the Pixels, According to the Order	Pixel Format Used to Store Information	Example
Properties of point marker	K	3	Data point = Circle (1 = circle, 2 = square, ...), radius of the circle, colour of the circle.	unsigned 24-bit pixel format	1, 10, 1
Border widths	K + 1	5	Out of range border, missing value border, GKU specific data border, border padding, offset.	unsigned 24-bit pixel format	10, 10, 10, 1, 10
X value information	K + 2	8	Minimum value, maximum value, decimal to integer factor, scale up/down factor.	two signed 24-bit pixel format	(65, 0), (90, 0), (10, 0), (2, 0)
Y value information	K + 3	8	Minimum value, maximum value, decimal to integer factor, scale up/down factor.	two signed 24-bit pixel format	(223, −2), (9055, −3), (10, 0), (3, 0)

2.5. GKU Evaluation Method

We tested the new method using automatically recorded data from near-infrared (NIR) spectroscopy at a biogas plant over a period of nearly 75 days with a frequency of 20 values per hour (35,620 data points). Volatile solid (VS) and volatile fatty acid (VFA) concentrations were selected as dimensions. The selected data were part of a data set used to develop NIR spectroscopy online calibration for monitoring VS and VFAs as process indicators during anaerobic digestion [40]. In the selected data set, there were some missing data. The missing data were ignored in the offline version of the GKU; however, missing data were considered in the online version. Thus, in the online version, the total number of data points was 35,864.

The creation of a GKU for an online situation was tested by simulating an online environment. An out-of-range data environment was artificially created by replacing some of the missing data with very high and very low values. In the missing data, for some data points, only the x or y parameter was missing, whereas for others, both parameters were missing. The method was implemented with Visual Studio 2008 (Net framework version 3.5 SP1) (Microsoft Cooperation, Way Redmond, WA, USA). MATLAB (Version 7.4.0) (The MathWorks Ins, Natick, MA, USA) was used to create plots that were required to validate the proposed method.

3. Results and Discussion

Determination of the type, size and colour of the marker strongly influences the visual standard and cluster formation of the final GKU output. Therefore, it is very important to select the best combination of those features before creating the real GKU. We determined the best combination after conducting a series of trials. Figures 4 and 5 show several GKUs for the same data set with two different markers (circle and square) and different combinations of features such as size and colour. In Figure 4, the diameter of a circle is equal to the length of one side of a square in the corresponding plot in Figure 5. When comparing corresponding plots in Figures 4 and 5, it can be seen that visual notion of clusters in Figure 5 are more intensive than in Figure 4. A square comprises more pixels than a circle whose diameter is equal to the length of one side of a square. This generates more overlapping when a square is used. Plot D in both figures is a good example.

In both figures, bitmaps A, B and C do not show adequate numbers of visual clusters. In contrast, bitmaps F, H and I show too many clusters, especially visually. Bitmaps D, E and, in particular, G show an appropriate number of clusters. In general, the correct selection of marker size and initial colour will produce clusters with good visual standards. We selected a circle as the marker for generating GKUs, because plots with circles produce better overall visual clarity than squares. Therefore, all results are based on circles with different diameters and colours. We show colours (0, 0, 1) to (0, 0, 10) in this section; however, plots with other colour values are also presented in this paper.

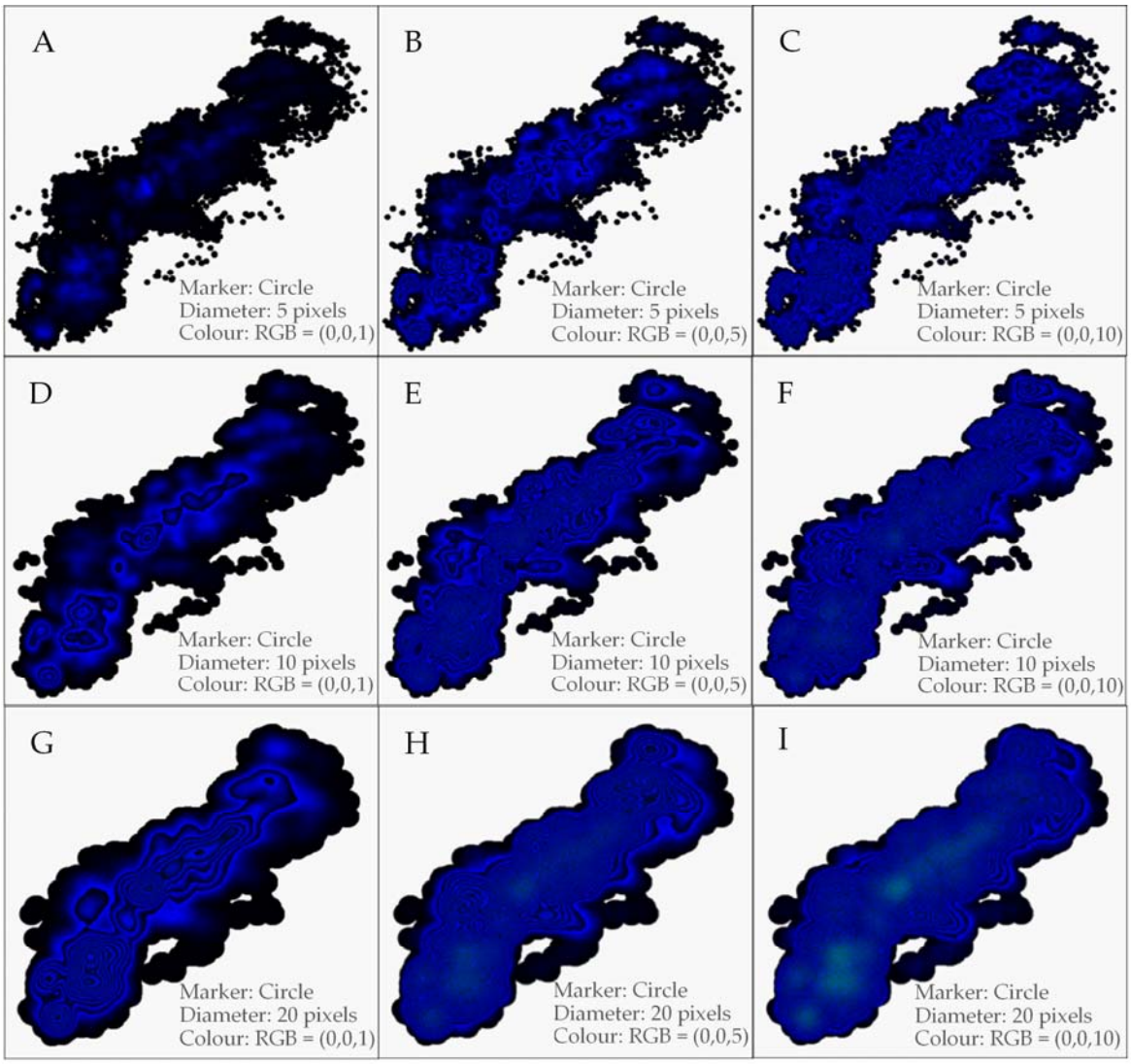


Figure 4. GKUs for the same data set with 35,620 data points using a circle as the marker with different sizes and colours. The correct selection of shape, size and initial colour of the data point will produce clusters that are visually clear and separated by colour borders similar to contour lines. For data set of plots in this figure, see Supplementary Materials, File S1. (**A**): GKU for 35,620 data points generated using a circle as the marker, where diameter is 5 pixels and RGB colour is (0, 0, 1); (**B**): GKU for 35,620 data points generated using a circle as the marker, where diameter is 5 pixels and RGB colour is (0, 0, 5); (**C**): GKU for 35,620 data points generated using a circle as the marker, where diameter is 5 pixels and RGB colour is (0, 0, 10); (**D**): GKU for 35,620 data points generated using a circle as the marker, where diameter is 10 pixels and RGB colour is (0, 0, 1); (**E**): GKU for 35,620 data points generated using a circle as the marker, where diameter is 10 pixels and RGB colour is (0, 0, 5); (**F**): GKU for 35,620 data points generated using a circle as the marker, where diameter is 10 pixels and RGB colour is (0, 0, 10); (**G**): GKU for 35,620 data points generated using a circle as the marker, where diameter is 20 pixels and RGB colour is (0, 0, 1); (**H**): GKU for 35,620 data points generated using a circle as the marker, where diameter is 20 pixels and RGB colour is (0, 0, 5); (**I**): GKU for 35,620 data points generated using a circle as the marker, where diameter is 20 pixels and RGB colour is (0, 0, 10).

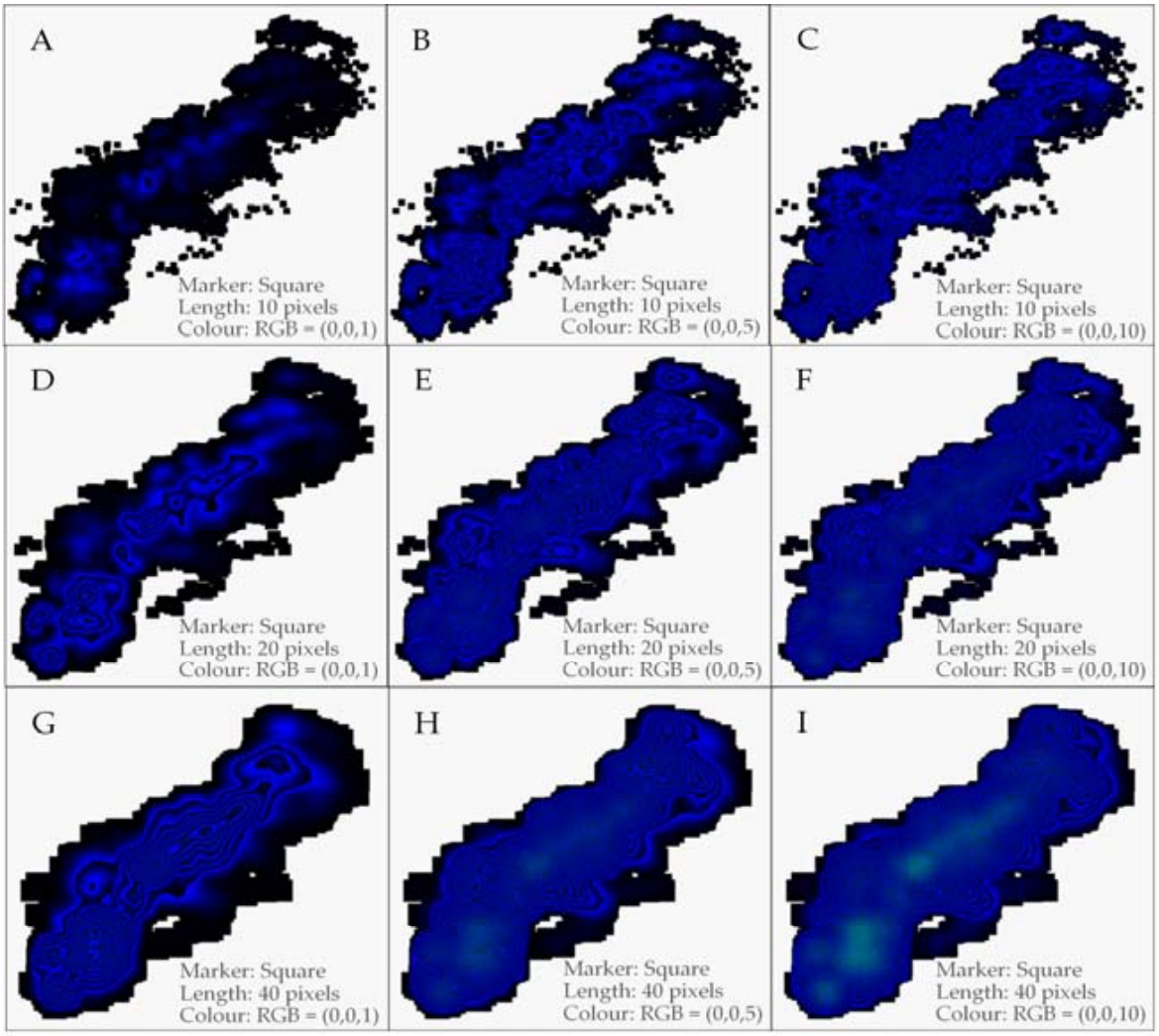
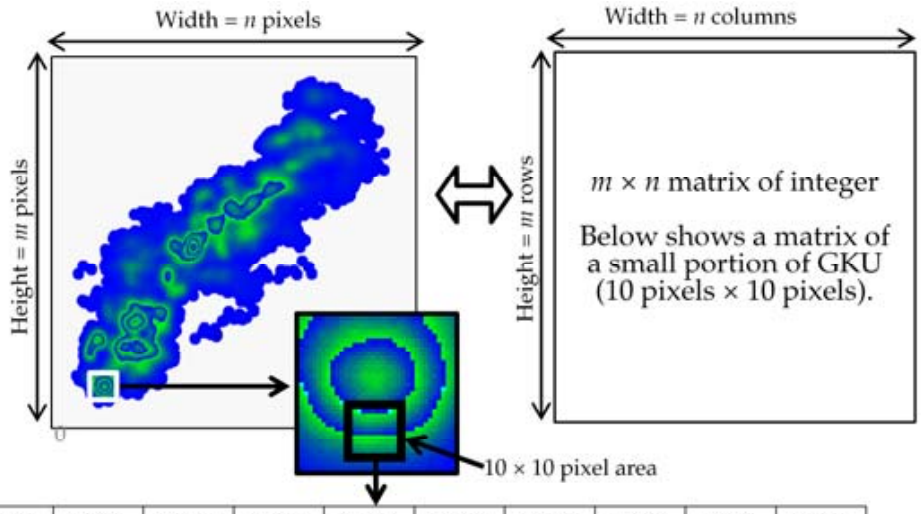


Figure 5. GKUs for the same data set with 35,620 data points using a square as the marker with different sizes and colours. The correct selection of shape, size and initial colour of the data point will produces clusters that are visually clear and separated by colour borders similar to contour lines. For data set of plots in this figure, see Supplementary Materials, File S1. (**A**): GKU for 35,620 data points generated using a square as the marker, where length is 10 pixels and RGB colour is (0, 0, 1); (**B**): GKU for 35,620 data points generated using a square as the marker, where length is 10 pixels and RGB colour is (0, 0, 5); (**C**): GKU for 35,620 data points generated using a square as the marker, where length is 10 pixels and RGB colour is (0, 0, 10); (**D**): GKU for 35,620 data points generated using a square as the marker, where length is 20 pixels and RGB colour is (0, 0, 1); (**E**): GKU for 35,620 data points generated using a square as the marker, where length is 20 pixels and RGB colour is (0, 0, 5); (**F**): GKU for 35,620 data points generated using a square as the marker, where length is 20 pixels and RGB colour is (0, 0, 10); (**G**): GKU for 35,620 data points generated using a square as the marker, where length is 40 pixels and RGB colour is (0, 0, 1); (**H**): GKU for 35,620 data points generated using a square as the marker, where length is 40 pixels and RGB colour is (0, 0, 5); (**I**): GKU for 35,620 data points generated using a square as the marker, where length is 40 pixels and RGB colour is (0, 0, 10).

3.1. Reading GKUs

There are two methods for reading or understanding a GKU. The first method relies on an algorithm (computer-aided method). A GKU is a bitmap and a bitmap is a matrix of pixels; therefore, the content of a GKU can be converted into a matrix of integers (GKU matrix) using (1) (Figure 6). In Figure 6, we used the RGB colour (0, 0, 254) for the marker. These integers in GKU matrix represent

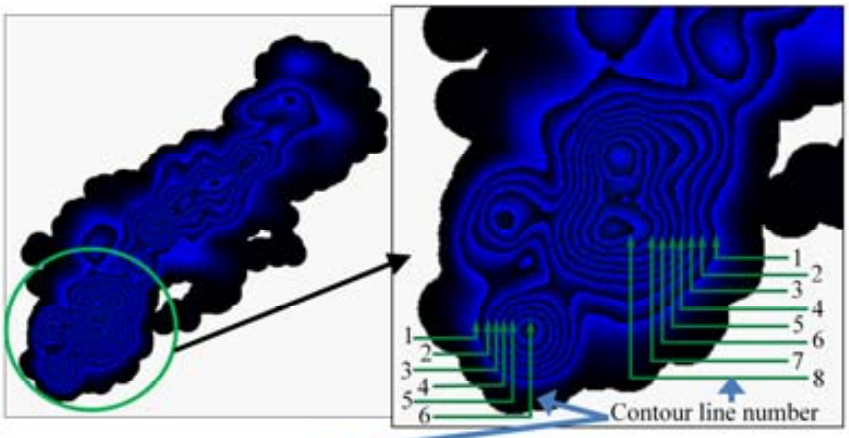
the data density of a particular location and can be used to extract or derive information. This feature is an advantage of the GKU over existing clustering methods.



130330	130330	133874	139484	140249	138719	135659	133109	131068	126989
114749	117809	122399	125714	126224	125459	123929	123168	119849	115514
104294	110159	113219	116534	115259	114239	48958	111179	108629	105514
95879	99704	104039	105824	105314	103019	103019	100214	96899	91544
84149	88233	90269	92163	93077	90014	87209	84404	83129	81599
74714	77009	79304	80324	78794	77774	78029	72299	71909	70889
63494	63494	64259	63749	63749	63494	62729	60689	59414	56864
52019	55844	56099	57119	55589	55589	52784	51509	50234	48449
40799	41564	43604	44879	44114	43859	41564	40544	41819	41309
32348	35699	37229	36464	36719	35399	35189	34424	33659	32384

Figure 6. Relation between bitmap and matrix versions of a GKU. A GKU matrix is a simple way to represent the same GKU. Marker: circle, radius: 10 pixels, marker colour: (0, 0, 254). The table shows the colour values of 10 × 10 pixels in the bitmap, which is a portion of the GKU matrix.

The second method is reading visual information by considering the presented graphical output (observation) as a usual plot. The GKU plot does not include a colour scale or legend to aid the understanding of the clusters. However, clusters and density of clusters can be determined with the aid of colour boarders that are automatically generated due to sudden change of colour values of adjacent pixels in the GKU. These colour boarders are identical to contour lines and maintain the same colour value difference between consecutive borders as in a contour map (Figure 7). For example, consider the RGB colours (0, 0, 255) and (0, 1, 0). According to Equation (1), colour values (C_v) of RGB colour (0, 0, 255) and (0, 1, 0) are 255 and 256, respectively. The RGB colour (0, 0, 255) is blue and RGB colour (0, 1, 0), which, next to (0, 0, 255), is visually black and create sudden change in colour blue to black, even though the difference between colour values is 1. The table in the Figure 7 shows the RGB values of the inner border (blue side) of each contour line. Usually, all colour borders are visually the same. However, the green colour values (G_v) of those lines maintain constant difference of one between adjacent colour borders; which resembles the contour lines (Figure 7). We numbered the contour lines from the outside to the inside (i.e., 1, 2, 3, . . .) and observed that contour lines with the same numbers have nearly the same colour values (same green value + nearly the same blue value) (Figure 7). Thus, it is possible to compare the density of different clusters even without a colour scale or legend. This is another advantage of the GKU over existing clustering methods.



Contour line number	Value of red channel	Value of green channel	Value of blue channel
1	0	0	220–255
2	0	1	220–255
3	0	2	220–255
4	0	3	220–255
5	0	4	220–255
6	0	5	220–255
7	0	6	220–255
8	0	7	220–255

Figure 7. Contour lines in a GKU. Representation of 35,620 data points; marker: circle, radius: 20 pixels, marker colour: (0, 0, 1). This shows clear colour borders that can be considered as contour lines. Contour lines are numbered from the outside to the inside of the cluster. Contour lines with same contour line number have the same green channel value. For such contour lines, blue channel values are in the same range. The higher the number of contour lines, the higher the data density. Therefore, it is possible to understand cluster density without a colour scale or legend. For data set of plots in this figure, see Supplementary Materials, File S1.

3.2. Anytime Cluster Formation

Note that the GKU is not a cluster analysis method; it is an anytime cluster formation method. As above mentioned anytime techniques make an explicit effort to speed up the cluster generation. The related methods discussed in this paper process all existing data to find clusters. If there are new data, the data must be processed again to find new clusters or update existing clusters. In contrast, the GKU shows up-to-time clusters and waits for new data. After adding a new data point, it is not necessary to process the whole data set again to obtain the current state. Adding a new data point to the GKU will update only those pixels that are covered by the marker. All other pixels in the bitmap remain unchanged. This requires a relatively small computational effort. Because the GKU is a matrix that is continuously updated, it can be seen as a continuous learning database of already processed data. Usually, the process of knowledge extraction becomes more difficult when dealing with large data sets because the algorithm needs to check a very large number of data points [41]. In contrast, with the proposed cluster formation technique, the time for updating is independent of the number of data points. The bitmaps in Figure 8 show the development of a GKU over time. All the plots in Figure 8 imply the importance of overlapping in the GKU concept. Initially, the GKU does not show clear clusters (Figure 8A). As overlap increases, the GKU shows clusters that can be easily identified by the naked eye (Figure 8C,D).

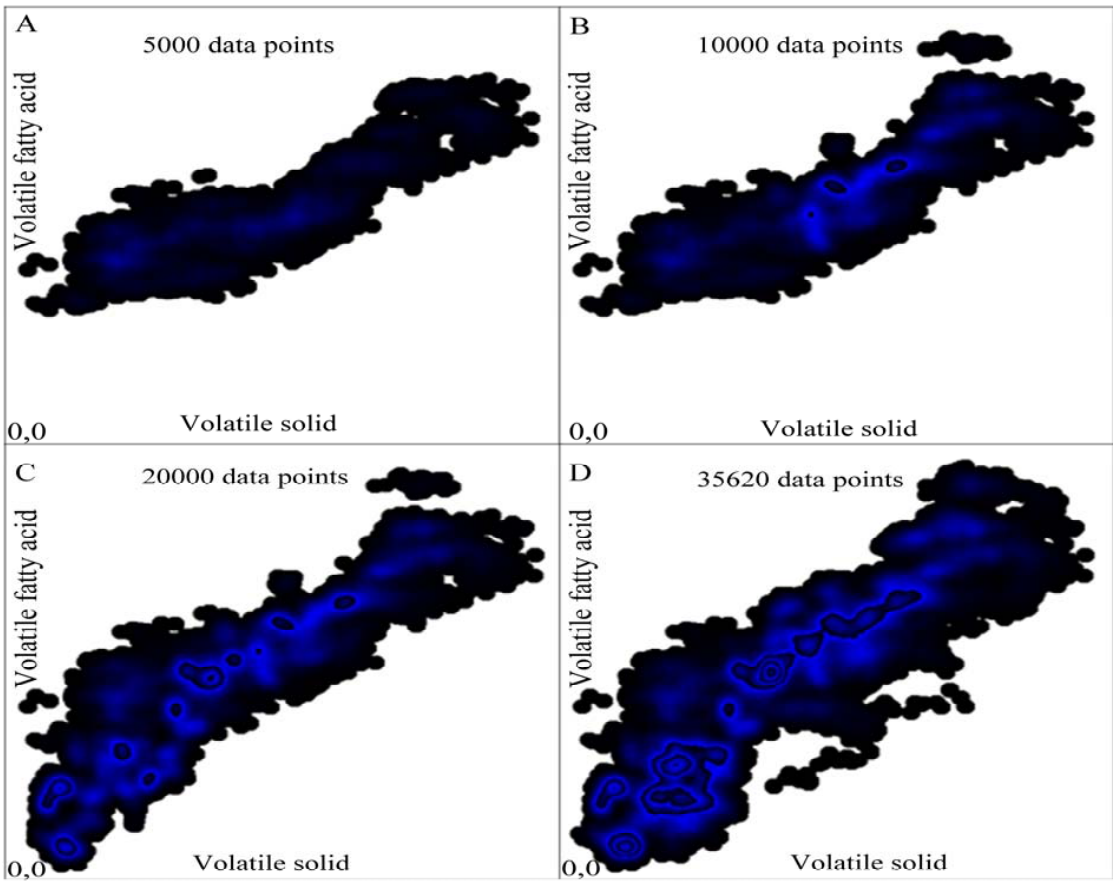


Figure 8. Development of a GKU over time. Bitmaps (**A**–**D**) show GKUs with 5000, 10,000, 20,000 and 35,620 data points, respectively. Marker: circle, radius: 10 pixels, initial colour of the data point: (0, 0, 1). For data set of plots in this figure, see Supplementary Materials, File S1.

3.3. Representation of Missing and Out of Range Values and GKU Specific Data

When considering very large data sets, information about missing and out of range data is vital because it provides a complete overview of the data and its quality. None of the existing clustering methods is capable of visualizing this information. In contrast, the GKU is capable of indicating density of out-of-range values as well as missing values (Figure 9). This makes the GKU a very efficient and effective means of representing big data. The GKU shows the density of out-of-range and missing data categorized in different regions (Figure 3). Figure 9 illustrates the use of a GKU specific data area to save feature information such as marker type (circle, square, etc.), colour and dimensions and border information (width of borders of missing and unexpected values). This information is saved as colours after converting such feature information according to the standards listed in Table 3. Furthermore, the initial row of the GKU specific data is encoded in the bitmap header according to the standards listed in Table 2. Thus, GKU specific data can be identified by reading the bitmap header.

3.4. GKU as an Outlier Detection Method

The GKU can be used to identify outliers in a data set. If data points in low-density areas are outliers, they will always have low colour values. Therefore, it is possible to define a certain colour value in the GKU as a border for outliers. Then, all points with colour values below the colour value of the border can be removed manually or by means of an algorithm. Figure 10 illustrates a very simple way of identifying outliers in a very large data set using a GKU. If the GKU is visually interpretable, it is possible to define a border to identify outliers by checking the colour value of the area. A very simple bitmap reading application can be used to identify the colour values of each pixel and then

manually define a border for outliers. If higher accuracy is required, this can be done by means of an algorithm. Because the method is based on knowledge discovery in databases (KDD), this can be considered an unsupervised outlier detection method [42,43].

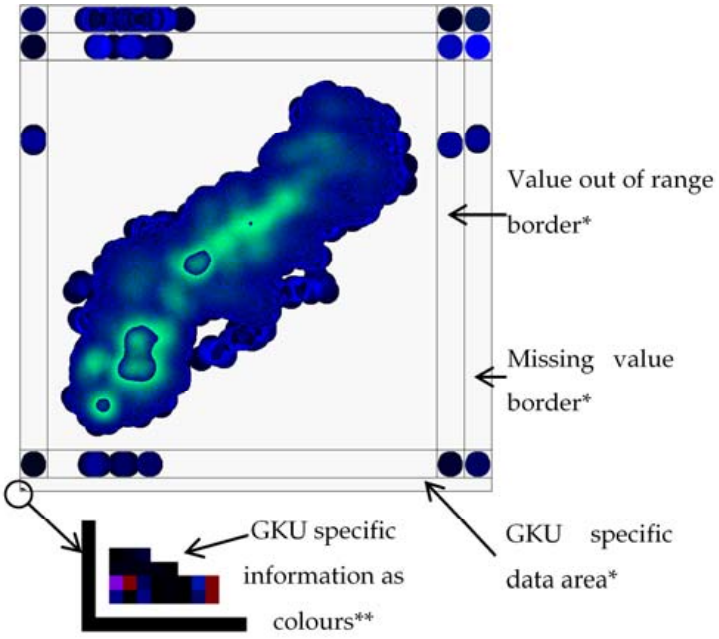


Figure 9. Representation of 35,864 data points in a GKU with borders to record missing values, out of range values and GKU specific information. Marker: circle, radius: 20 pixels, colour of the data point: (0, 0, 50). * Refer to Figure 3 for structure information and usage. ** Refer to Table 3 for structure information about the GKU specific information. For data set of plots in this figure, see Supplementary Materials, File S2.

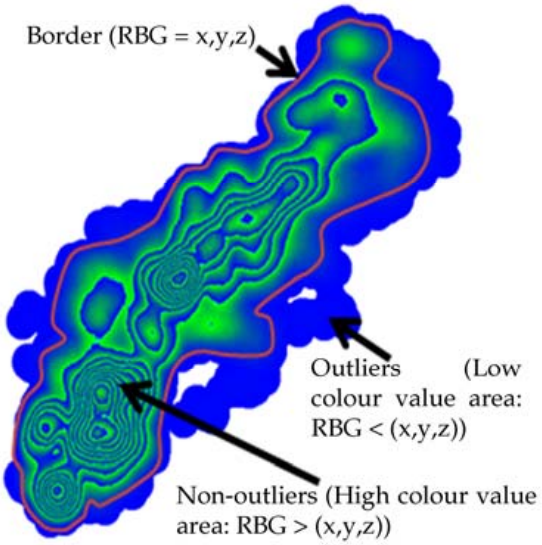


Figure 10. Outlier identification using GKU by defining a border manually. Areas with low colour values are defined as outliers (noise) and vice versa. Shape of the data point: circle, radius: 20 pixels, colour of the data point: (0, 0, 254). For data set of plots in this figure, see Supplementary Materials, File S1.

We compared the visual standards of the proposed GKU method with the three most popular data representation methods: scatter plot, heat map and contour plot. Figure 11A–C show the visualization

of 35,620 data points with a scatter plot, heat map and contour plot, respectively. All plots were generated using MATLAB (Version 7.4.0). According to the Figure 11A, scatter plot does not support for identifying data density in systematic manner. However, scatter plot is useful to illustrate the nature of data distribution and this is the default usage of scatter plot. Nevertheless, heat map and contour plot were employed to illustrate data density. The results show that the heat map was unable to create clusters and the contour plot was unable to generate contour lines. Even after zooming, it is difficult to identify density clusters with heat map and contour map. In contrast, the GKU could generate both clusters and contour lines in the same bitmap which help to illustrate the nature of data distribution in systematic manner, which can be identified by the naked eye (Figure 7). The GKU is capable of indicating density of out-of-range values as well as missing values (Figure 3). However, heat map and contour map have no method for representing out-of-range values and missing values. Therefore, GKU can be considered as a powerful and efficient method for identifying density clusters.

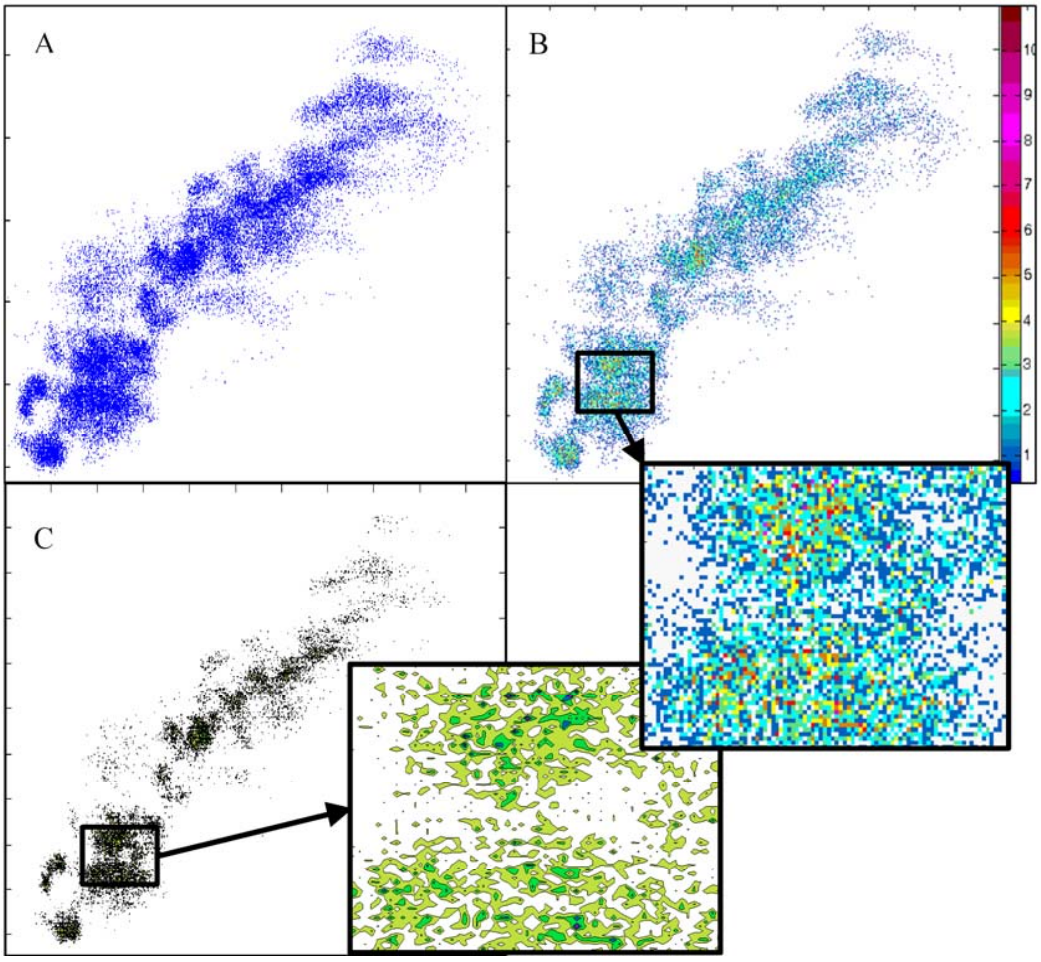


Figure 11. Visualization of 35,620 data points with: (**A**) scatter plot; (**B**) heat map; and (**C**) contour plot. The scatter plot shows the distribution of the data, whereas the heat map and the contour plot show density clusters. However, compared to the GKU, the heat map and contour plot do not show density clusters. For data set of plots in this figure, see Supplementary Materials, File S1.

With the GKU approach, the marker has particular significance compared to a usual plot. In this paper, we used the size of the marker to represent the clustering range of influence and the colour to represent data density. However, the size, type and location of the data point in the marker and the colour of the marker can be mapped with specific properties such as tolerance, error or minimum or maximum distance between two data points. In addition, depending on the domain, these properties

can be mapped with different features such as concentration (chemistry) or the signal strength and coverage area of a transmitter (networking/electronics). In all GKUs shown in this paper, we used the same colour for the pixels in the marker. However, using marker colour as a function of a certain feature will result in different coloured pixels depending on the function. For example, the density of seed propagation of a plant is not linear over distance. In this situation, the colour of the marker can be mapped as a function of seed propagation and the area of propagation can be mapped to the dimensions of the marker. If the propagation covers a certain area such as a sector, then the location of the plant can be represented by the angular point of the sector.

An existing GKU can be used directly in an online environment as a trained set or template. In addition, within an online environment, it is possible to recreate new versions of the GKU repeatedly according to new value ranges. If the number of out of range data points is higher than a certain value (e.g., more than x% of total data), a new GKU can be created according to the new range. Then, the old GKU can be replaced with a new GKU and recording can continue.

The proposed method has three major drawbacks. The first is that the GKU cannot be applied to non-overlapping data. The second is that the GKU is not capable of visualizing high-dimensional data. The third is that the number of overlapping incidents is limited to 2^n, where n is the total bit length of the colour format. This drawback can be overcome by using colour formats with higher bit length.

If a GKU is nearly full, red regions that did not occur in any GKU shown in this paper will appear. This implies that we could handle a much larger number of data points than the maximum number used in this study (35,620). We employed the three-channel RGB colour format with 8 bits for each channel. To create larger GKUs, it is possible to use four channel colour formats (ARGB) and more than 8 bits per channel [44,45]. The 16-bit RGBA format provides a maximum of 2^{64} overlapping incidents.

4. Conclusions

The GKU is a container to process data that is capable of immediately maintaining and displaying a large number of data points in a small area. The GKU can be seen as a combination of the quadrat sampling method with contour lines. It is a very effective method for representing density clusters in offline and online environments. In addition, the GKU is a continuous learning graphical database that can be used as a direct input for another algorithm or as a trained set, template or signature for a certain process. Furthermore, the GKU can be used to identify outliers effectively, particularly in data sets with non-linear relations. In this paper, we presented a GKU with one dependent and one independent variable. However, it is possible to use a GKU with RGB colour scheme to visualize one dependent variable with three independent variables by assigning each colour slot for different independent variables (R for variable 1, G for variable 2, etc.). The major requirement for this is that all independent variables must be in the same value range. This would enable easy identification of correlations between variables and would provide a convenient way to visualize multidimensional data in two dimensions. In addition, compression and integration with swarm intelligence methods such as monarch butterfly optimization (MBO), earthworm optimization algorithm (EWA), and elephant herding optimization (EHO) will enhance the outcome of the GKU.

Supplementary Materials: The following are available online at http://www.mdpi.com/2073-8994/8/12/152/s1, File S1: Data sets of all the plots in Figures 4, 5, 7, 8, 10 and 11, File S2: Data sets of all the plots in Figure 9.

Acknowledgments: This work was supported by the German Research Foundation (DFG) and the Technical University of Munich (TUM) in the framework of the Open Access Publishing Program. In addition, we are grateful to the German Academic Exchange Service (Deutscher Akademischer Austauschdienst, DAAD) for providing a scholarship to K.K.L.B. Adikaram during the research period.

Author Contributions: K.K.L.B.A. conceived and designed the experiments and algorithms; K.K.L.B.A. performed the experiments; K.K.L.B.A., M.A.H., and M.E. analysed the data; K.K.L.B.A., M.A.H., M.E., and T.B. contributed reagents/materials/analysis tools; and K.K.L.B.A. wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Stone, M.C.; Fishkin, K.; Bier, E.A. The Movable Filter as a User Interface Tool. In Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, Boston, MA, USA, 24–28 April 1994; pp. 306–312.
2. Woodruff, A.; Landay, J.; Stonebraker, M. Constant density visualizations of non-uniform distributions of data. In Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology, San Francisco, CA, USA, 1–4 November 1998.
3. Yang, J.; Ward, M.O.; Rundensteiner, E.A. Visual hierarchical dimension reduction for exploration of high dimensional datasets. In Proceedings of the Eurographics/IEEE TCVG Symposium on Visualization, Grenoble, France, 26–28 May 2003.
4. Ellis, G.; Dix, A. A Taxonomy of Clutter Reduction for Information Visualisation. *IEEE Trans. Vis. Comput. Graph.* **2007**, *13*, 1216–1223. [CrossRef] [PubMed]
5. Chen, H.; Chen, W.; Mei, H.; Liu, Z.; Zhou, K.; Chen, W.; Gu, W.; Ma, K.L. Visual Abstraction and Exploration of Multi-class Scatterplots. *IEEE Trans. Vis. Comput. Graph.* **2014**, *20*, 1683–1692. [CrossRef] [PubMed]
6. Cleveland, W.S. *Visualizing Data*; Hobart Press: Hobart, Australia, 1993.
7. Bachthaler, S.; Weiskopf, D. Efficient and Adaptive Rendering of 2-D Continuous Scatterplots. *Comput. Graph. Forum* **2009**, *28*, 743–750. [CrossRef]
8. Mai, S.T.; He, X.; Feng, J.; Plant, C.; Böhm, C. Anytime density-based clustering of complex data. *Knowl. Inform. Syst.* **2015**, *45*, 319–355. [CrossRef]
9. Hoffman, P.; Grinstein, G. Visualizations for High Dimensional Data Mining-Table Visualizations. 1997. Available online: http://web.simmons.edu/~benoit/infovis/MIV-datamining.pdf (accessed on 28 January 2014).
10. Salomon, D. Raster Graphics. In *The Computer Graphics Manual*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 29–131.
11. Salomon, D. Graphics Standards. In *The Computer Graphics Manual*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 947–972.
12. Everitt, B.S.; Landau, S.; Leese, M.; Stahl, D. Index. In *Cluster Analysis*; John Wiley & Sons, Ltd.: New York, NY, USA, 2011; pp. 321–330.
13. Lee, R.C.T. Clustering Analysis and Its Applications. *Adv. Inform. Syst. Sci.* **1981**, *8*, 169–292.
14. Næs, T.; Brockhoff, P.B.; Tomic, O. Cluster Analysis: Unsupervised Classification. In *Statistics for Sensory and Consumer Science*; John Wiley & Sons, Ltd.: New York, NY, USA, 2010; pp. 249–261.
15. Okun, O.; Priisalu, H. Unsupervised data reduction. *Signal Process.* **2007**, *87*, 2260–2267. [CrossRef]
16. Anderberg, M.R. *Cluster Analysis for Applications*; Academic Press: New York, NY, USA, 1973.
17. Chui, C.K.; Filbir, F.; Mhaskar, H.N. Representation of functions on big data: Graphs and trees. *Appl. Comput. Harmon. Anal.* **2015**, *38*, 489–509. [CrossRef]
18. Avramenko, Y.; Ani, E.-C.; Kraslawski, A.; Agachi, P.S. Mining of graphics for information and knowledge retrieval. *Comput. Chem. Eng.* **2009**, *33*, 618–627. [CrossRef]
19. Yu, H.; Yang, J.; Han, J.; Li, X. Making SVMs Scalable to Large Data Sets using Hierarchical Cluster Indexing. *Data Min. Knowl. Discov.* **2005**, *11*, 295–321. [CrossRef]
20. De Vito, E.; Rosasco, L.; Toigo, A. Learning sets with separating kernels. *Appl. Comput. Harmon. Anal.* **2014**, *37*, 185–217. [CrossRef]
21. Galluccio, L.; Michel, O.; Comon, P.; Hero, A.O., III. Graph based k-means clustering. *Signal Process.* **2012**, *92*, 1970–1984. [CrossRef]
22. Sebzalli, Y.M.; Li, R.F.; Chen, F.Z.; Wang, X.Z. Knowledge discovery from process operational data for assessment and monitoring of operator's performance. *Comput. Chem. Eng.* **2000**, *24*, 409–414. [CrossRef]
23. Barbará, D.; Chen, P. Using Self-Similarity to Cluster Large Data Sets. *Data Min. Knowl. Discov.* **2003**, *7*, 123–152. [CrossRef]
24. David, G.; Averbuch, A. Hierarchical data organization, clustering and denoising via localized diffusion folders. *Appl. Comput. Harmon. Anal.* **2012**, *33*, 1–23. [CrossRef]
25. Zhang, L.; Tang, C.; Song, Y.; Zhang, A.; Ramanathan, M. VizCluster and its Application on Classifying Gene Expression Data. *Distrib. Parallel Databases* **2003**, *13*, 73–97. [CrossRef]
26. Johansson, J.; Ljung, P.; Jern, M.; Cooper, M. Revealing structure in visualizations of dense 2D and 3D parallel coordinates. *Inform. Vis.* **2006**, *5*, 125–136. [CrossRef]

27. Wilkinson, L.; Friendly, M. The History of the Cluster Heat Map. *Am. Stat.* **2009**, *63*, 179–184. [CrossRef]
28. Niida, A.; Tremmel, G.; Imoto, S.; Miyano, S. Multilayer Cluster Heat Map Visualizing Biological Tensor Data. In Proceedings of the 2013 8th Brazilian Symposium on Advances in Bioinformatics and Computational Biology, Recife, Brazil, 3–7 November 2013; Setubal, J., Almeida, N., Eds.; pp. 116–125.
29. Weinstein, J.N. A Postgenomic Visual Icon. *Science* **2008**, *319*, 1772–1773. [CrossRef] [PubMed]
30. Hao, M.C.; Dayal, U.; Sharma, R.K.; Keim, D.A.; Janetzko, H. Variable binned scatter plots. *Inform. Vis.* **2010**, *9*, 194–203. [CrossRef]
31. Mayorga, A.; Gleicher, M. Splatterplots: Overcoming Overdraw in Scatter Plots. *IEEE Trans. Vis. Comput. Graph.* **2013**, *19*, 1526–1538. [CrossRef] [PubMed]
32. Nievergelt, J.; Widmayer, P. Spatial data structures: Concepts and design choices. In *Algorithmic Foundations of Geographic Information Systems*; van Kreveld, M., Nievergelt, J., Roos, T., Widmayer, P., Eds.; Springer: Berlin/Heidelberg, Germany, 1997; pp. 153–197.
33. Yoo, J.; Bow, M. Mining spatial colocation patterns: A different framework. *Data Min. Knowl. Discov.* **2012**, *24*, 159–194. [CrossRef]
34. Gross, M.; Pfister, H. *Point-Based Graphics*; Morgan Kaufmann Publishers Inc.: San Mateo, CA, USA, 2007; p. 248.
35. Carr, D.B.; Littlefield, R.J.; Nicholson, W.L.; Littlefield, J.S. Scatterplot Matrix Techniques for Large N. *J. Am. Stat. Assoc.* **1987**, *82*, 424–436. [CrossRef]
36. Imhof, E. *Cartographic Relief Presentation*; ESRI Press: Redlands, CA, USA, 2007; p. 111.
37. Bowman, A.; Foster, P. Density based exploration of bivariate data. *Stat. Comput.* **1993**, *3*, 171–177. [CrossRef]
38. Lampe, O.D.; Hauser, H. Interactive visualization of streaming data with Kernel Density Estimation. In Proceedings of the 2011 IEEE Pacific Visualization Symposium (PacificVis), Hong Kong, China, 1–4 March 2011.
39. George, G.R. New Methods of Mathematical Modeling of Human Behavior in the Manual Tracking Task. Ph.D. Thesis, University of New York, Binghamton, NY, USA, 2008; p. 190.
40. Krapf, L.C.; Heuwinkel, H.; Schmidhalter, U.; Gronauer, A. The potential for online monitoring of short-term process dynamics in anaerobic digestion using near-infrared spectroscopy. *Biomass Bioenergy* **2013**, *48*, 224–230. [CrossRef]
41. Huang, Z. Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values. *Data Min. Knowl. Discov.* **1998**, *2*, 283–304. [CrossRef]
42. Angiulli, F.; Fassetti, F. Exploiting domain knowledge to detect outliers. *Data Min. Knowl. Discov.* **2014**, *28*, 519–568. [CrossRef]
43. Akoglu, L.; Tong, H.; Koutra, D. Graph based anomaly detection and description: A survey. *Data Min. Knowl. Discov.* **2015**, *29*. [CrossRef]
44. Salomon, D. *The Computer Graphics Manual*; Springer: Berlin/Heidelberg, Germany, 2011; p. 967.
45. Van Verth, J.M.; Bishop, L.M. *Essential Mathematics for Games and Interactive Applications: A Programmer's Guide*, 2nd ed.; CRC Press: Boca Raton, FL, USA, 2008; p. 264.

applied sciences

MDPI

Article

Multi-Variable, Multi-Layer Graphical Knowledge Unit for Storing and Representing Density Clusters of Multi-Dimensional Big Data

K. K. L. B. Adikaram [1,2,3,*], Mohamed A. Hussein [1,†], Mathias Effenberger [2,†] and Thomas Becker [4,†]

1 Group Bio-Process Analysis Technology, Technische Universität München, Weihenstephaner Steig 20, Freising 85354, Germany; hussein@wzw.tum.de
2 Bavarian State Research Center for Agriculture, Institute for Agricultural Engineering and Animal Husbandry, Vöttinger Straße 36, Freising 85354, Germany; mathias.effenberger@lfl.bayern.de
3 Computer Unit, Faculty of Agriculture, University of Ruhuna, Mapalana, Kamburupitiy 81100, Sri Lanka
4 Lehrstuhl für Brau- und Getränketechnologie, Technische Universität München, Weihenstephaner Steig 20, Freising 85354, Germany; tb@bgt.wzw.tum.de
* Correspondence: lasantha@daad-alumni.de; Tel.: +94-412292200
† These authors contributed equally to this work.

Academic Editor: Antonio Fernández-Caballero
Received: 25 October 2015; Accepted: 15 March 2016; Published: 5 April 2016

Abstract: A multi-variable visualization technique on a 2D bitmap for big data is introduced. If A and B are two data points that are represented using two similar shapes with m pixels, where each shape is colored with RGB color of (0, 0, k), when $A \cap B \neq \phi$, adding the color of $A \cap B$ gives higher color as (0, 0, 2k) and the highlight as a high density cluster, where RGB stands for Red, Green, Blue and k is the blue color. This is the hypothesis behind the single variable graphical knowledge unit (GKU), which uses the entire bit range of a pixel for a single variable. Instead, the available bit range of a pixel is split, and a pixel can be used for representing multiple variables (multi-variables). However, this will limit the bit block for single variables and limit the amount of overlapping. Using the same size k (>1) bitmaps (multi-layers) will increase the number of bits per variable (BPV), where each (x, y) of an individual layer represents the same data point. Then, one pixel in a four-layer GKU is capable of showing more than four billion overlapping ones when BPV = 8 bits ($2^{(BPV \times number\ of\ layers)}$) Then, the 32-bit pixel format allows the representation of a maximum of up to four dependent variables against one independent variable. Then, a four-layer GKU of w width and h height has the capacity of representing a maximum of ($2^{(BPV \times number\ of\ layers)}$) $\times\ m \times w \times h$ overlapping occurrences.

Keywords: knowledge representation; continuous learning; cluster identification; big data

1. Introduction

Multi-variable analysis and graphical representation of data are two demanding factors in the field of data analysis. Multi-variable analysis is a method for depicting the correlation between variables, while graphical representation of data is a very efficient tool for abstracting information in a multi-variable dataset. In addition, graphical representation usually conveys the intended information more easily than text or numerical values [1]. However, when the numbers of available data are high, it is difficult to represent all of the data points of a scattered dataset as a plot due to the high number of overlapping data points, also called occlusion or over-plotting. This is one of the main issues in the field of data visualization, which leads to the loss of data in projection [2]. On the other hand, when the numbers of variables are high, special techniques are required to represent multi-dimensions on a two-dimensional or three-dimensional space. The biggest challenge is to visualize multi-dimensional big data.

Density cluster identification is a technique that is used in the field of knowledge discovery in databases (KDD) [3–6]. Furthermore, density cluster identification is a tool that is used to identify the correlation between variables. In cluster visualization, first, relevant density clusters were identified by means of a suitable algorithm, and then, those identified clusters were visualized by means of a suitable data visualization technique. Therefore, data visualization is usually a representation method, but not a cluster identification method [7]. The real cluster identification is done by algorithms. Of course, good visualization techniques allow viewers to identify clusters easily. However, this does not imply that these data visualizations are made for identifying clusters. As in multi-dimensional data analysis, cluster visualization also suffers when the numbers of data points are high, especially due to overlapping.

The term "big data" refers to datasets for which the size is beyond the capabilities of current database technology [6,8,9]. According to Karimi, big data is a "collection of databases so large or complex that it becomes difficult to process using regular database management tools or traditional data processing applications" [10]. In this work, big datasets are addressed from the perspectives of processing and representation. In data visualization techniques, it is hard to identify a specific method that is capable of visualizing big data [6]. The major and most practical reason for this is that there is no adequate space in a plot to visualize all of the data points. In the domain of a scattered dataset, overlapping data are another barrier for visualizing big data. On the other hand, according to the definition, big data are data requiring extreme methodologies for processing, as well as storage [11].

2. Related Work

Scatterplot matrices [12], parallel coordinates [13], Andrews' curves [14,15], Radviz [16] and star coordinates [17] are the most popular techniques used for visualization of multi-dimensional data. Except for star coordinates, all of the other mentioned methods display the multi-dimensional data on a lower-dimensional space in a way that additional effort is needed for understanding the original number of dimensions [17]. In contrast, star coordinates directly visualize dimensions as groups in the form of high density clusters in a two-dimensional plot [17]. Overlapping and a higher number of scatter plots in a scatterplot matrix convey poor visualization [18]. Parallel coordinates are not suitable for visualizing a higher number of records [2]. When there are higher numbers of overlapping points, Radviz does not provide the expected output [2,17]. Andrews' curves do not support visualizing clusters of multidimensional space even though they support a large number of data points [2] and require higher computational time for generating curves [19]. The star coordinates suffer from overlapping of clusters of variables [17,20]. These facts imply that all of the above-mentioned techniques are facing the lack of capabilities for visualizing multi-dimensional big data, in most cases due to overlapping data points.

In reality, it is normal to observe missing data [21,22] and out of range data or outliers [23,24] in most data acquisition systems. It is not always necessary to impute missing and out of range data. However, it is good to visualize the amount of missing and out of range data, to understand the magnitude and the nature of the distribution of such data. In the domain of big data, it is very important to know the quantity and the nature of such data for better comprehension of the quality of the data. When dealing with big data, missing and out of range data identification requires considerable computational effort. Nevertheless, in all of the mentioned methods, there is no standard mechanism included for showing missing and out of range data.

3. Concept of the Graphical Knowledge Unit

In 2015, a new way of representing density clusters was introduced for one dependent and one independent variable using a bitmap. The method was named the "graphical knowledge unit" (GKU) [25]. Overlapping data points are the main features that are used for creating a GKU. A higher number of overlapping ones makes a GKU more meaningful and understandable. From its origin, a GKU does not suffer from overlapping data failures. Mainly, a GKU visualizes data in the form of

density clusters. Mostly, it is possible to identify high density visually and by means of an algorithm, depending on the color density. In addition, a GKU provides a mechanism for visualizing the amount and nature of the distribution of both missing and out of range data points along with the density clusters. Thus, the GKU is sought as a reliable approach for big data visualization, especially overlapping types.

Nevertheless, the GKU is not a solution for visualizing multi-variable environments. An extended usage of the GKU for representing multi-variables using the same concept of the overlapping of data points is investigated in this work. As in the GKU, the success of multi-variable multi-layer (MVML)-GKU depends on having high data density for clustering purposes. Figure 1a shows a GKU used for representing 35,864 data points, and Figure 1b, c show the 3D representation and heat map representation of the GKU. Usually, for plotting 3D maps or heat maps, it is necessary to process the data for creating the required matrix. In contrast, while using a GKU, it is not necessary to create such a matrix and when adding data points. A GKU gradually forms the density areas by incrementing the color intensity in overlapped regions.

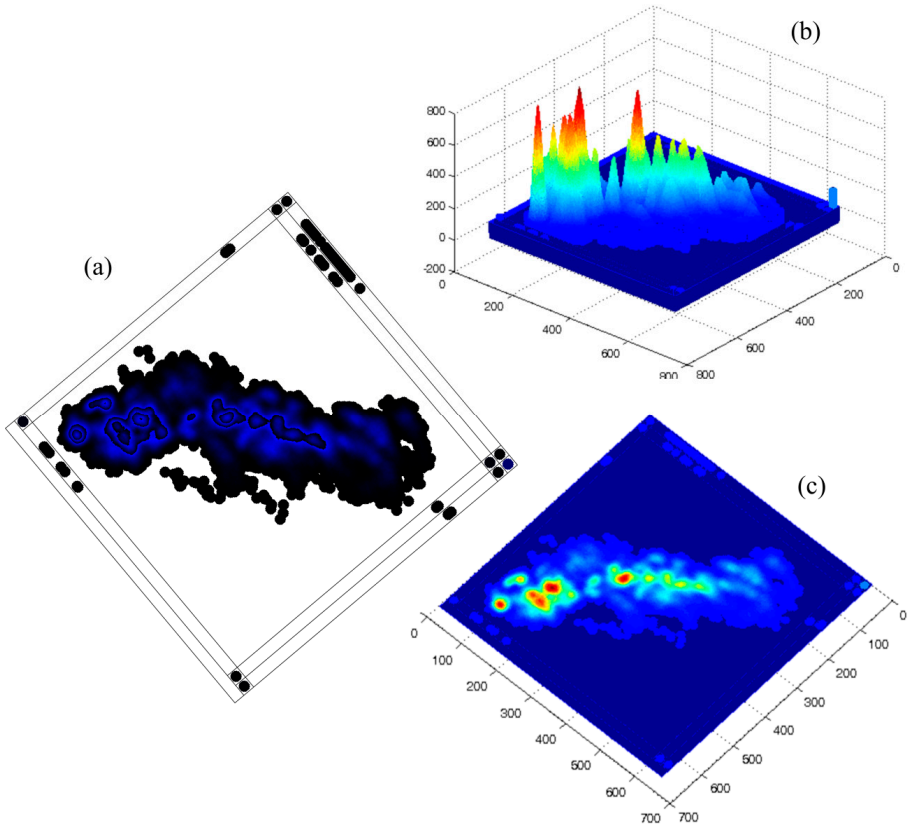


Figure 1. (**a**) Representation of 35,864 data points a graphical knowledge unit (GKU); (**b**) representation of the processed data of the same data in 3D; (**c**) representation of the processed data of the same data in a heat map.

In the proposed method, the concept of the GKU is used to represent a multi-variable big data environment in 2D space. Furthermore, the proposed method generates density clusters of all variables on the same bitmap, which can be identified by the naked eye. Most importantly, the proposed method does not suffer from overlapping data points and requires more overlapping for better visualization of density clusters. Thus, the proposed method is a robust density cluster representation and data visualization technique for multi-variable big data. Due to the very flexible nature of the proposed method, it will help to represent big data captured from dynamic bioprocesses.

4. Methodology

A GKU with a 24-bit bitmap, the RGB pixel format, is used for representing data points. Furthermore, it is possible to use a 32-bit bitmap with the alpha-RGB (ARGB) pixel format for representing the GKU. In both situations, a maximum of 24 bits or 32 bits can be allocated for representing a single variable against another variable.

Instead of allocating the whole bit range of a pixel to a single variable, it is possible to divide the bits among several variables. Figure 2 shows an example of the equal allocation of bits of a 32-bit bitmap with the ARGB pixel format for four variables. Furthermore, unequal allocation is possible depending on the requirements. Here, the equal allocation of bits for the variables was mainly considered. Figure 2 elaborates on the allocation of alpha, red, green and blue portions for representing four variables, where eight bits per variable (BPV) are used. However, when the numbers of variables are high, BPV is low.

Figure 2. A single pixel of a 32-bit RGB format is split into four equal portions for representing four variables (V1, V2, V3 and V4). A, R, G and B represents the alpha, red, green and blue portions of the pixel, respectively.

When the BPV is low, it leads to fewer numbers of overlapping data points. If BPV = 8 bits, it supports the representation of the maximum of 256 overlaps and will not allow the representation of a higher number of data points. This will discourage the usage of multiple variables in the GKU. As a solution, a multi-layer bitmap GKU is proposed, which consists of several equal-sized bitmaps (Figure 3). In the GKU, when one block is full (e.g., blue), it is possible to use the adjacent block (green) to represent overlapping [25]. In contrast, in a multi-layer GKU, when an allocated block is full, it uses the relevant pixel block of another bitmap of the same size (Figure 3), which is named a layer. If the number of layers is L and the number of bits allocated for a variable is k, this technique provides $k \times L$ bits for one variable. In the GKU, the whole bit range of a pixel is allocated for one variable, which can be considered as a horizontal array. In contrast, in the MVML-GKU concept, a single variable is a vertical array with $k \times L$ bits (Figure 3). Depending on the requirement, it is possible to decide the number of layers in the MVML-GKU.

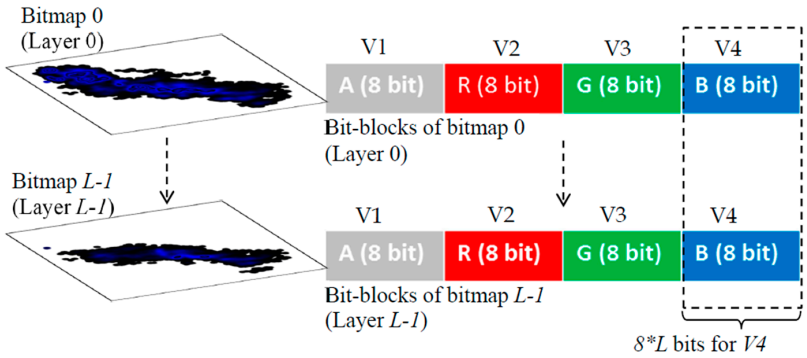


Figure 3. Four variables (V1, V2, V3 and V4) are represented using multiple bitmaps of a 32-bit RGB pixel format for forming a multi-variable multi-layer graphical knowledge unit (MVML-GKU). Each pixel is divided into four equal portions, where each portion consists of eight bits. When the 32-bit RGB pixel format is divided into four equal parts, each variable represents the alpha, red, green and blue sections of the pixel. This provides a vertical array of $k \times L$ bits for one variable.

Furthermore, each layer is numbered starting from 0, and the "place-value" is assigned for each layer according to the layer number. This assigns the same "place-value" for each bit in the relevant layer. The base of the "place-value" is in relation to the number of assigned bits for one variable, as shown in Equation (1).

$$PV^l = \left(2^{(q-p+1)} - 1\right)^l \tag{1}$$

where PV^l is the positional value of a bit in the layer l, p is the index of the starting bit of the bit block and q is the index of the ending bit of the bit block; thus, $p = 0, 1, 2, \ldots$ and $q = 0, 1, 2, \ldots$

If the $c^{(i,j)|l}_{[p,q]|l}$ represents the color value of the variable, it corresponds to bit block $[p, q]$ of the pixel (i, j) of the layer l (Figure 4). Therefore:

$$c^{(i,j)|l}_{[p,q]|l} = \sum_{k=q}^{p} b_k^{(i,j)|l} \times 2^{q-k} \tag{2}$$

where $b_k^{(i,j)|l}$ is the bit value (0 or 1) of the bit k of the bit block of a variable of pixel (i, j) of the layer l.

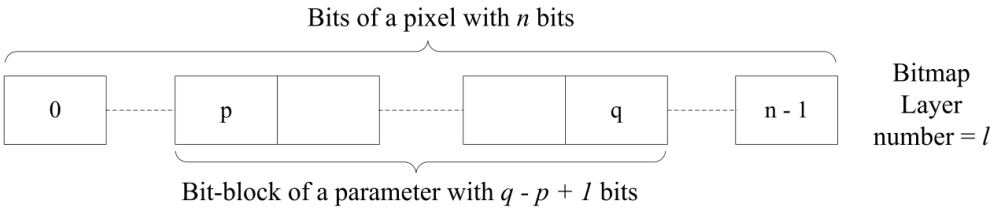


Figure 4. Bit numbering method of a pixel in a layer.

If the total color value in relation to a certain variable corresponding to the bit block $[p, q]$ is $C^{(i,j)}_{[p,q]}$, then:

$$C^{(i,j)}_{[p,q]} = \sum_{l=0}^{L-1} c^{(i,j)|l}_{[p,q]|l} \times PV^l \tag{3}$$

Substituting from Equation (1):

$$C^{(i,j)}_{[p,q]} = \sum_{l=0}^{L-1} c^{(i,j)|l}_{[p,q]|l} \times \left(2^{(q-p+1)} - 1\right)^l \tag{4}$$

We reserved the last number of the bit block for representing the initial color of the bitmap. Therefore, we used $(2^{(q-p+1)} - 1)^l$ instead of using $(2^{(q-p+1)})^l$ as the place-value of a certain layer. For example, in the case of equal bit block allocation, if the BPV are eight, we used 255^l as the place-value of the layer l instead of 256^l. This will reserve the last value "255" for the initial color of the bit block. Whenever the whole pixel is considered, "white" is the initial color of the pixel and the bitmap.

In the concept of the GKU, one data point is represented by a shape that consists of more than one pixel (e.g., a circle) [25] as a data point shape (DPS). While adding a data point, the existing color value of all of the pixels that are overlapped by the new data point need to be updated by adding the color of the newly-added data point. In a multi-variable environment, separate data points are used for each variable. It is possible to use different DPSs for different variables. However, in this paper, the usage of the same DPS (e.g., circle) for all of the variables is discussed. In every new data addition, the color increment of the intersected area needs to be updated with a previously decided value and named the "color increment" (CI) [25]. When there is more than one variable, it is possible to use the same or different CI values for variables. In all of the DPSs, the bit block that is assigned for a certain variable is set to its initial color value while keeping all of the bits that do not belong to the considered

bit block at value 1. For example, consider the situation of representing data points of four variables (V1, V2, V3 and V4) on the MVML-GKU by means of circles (radius = $r > 0$), which has a 32-bit RGB pixel format. Then, in the equal bit allocation, each variable is assigned eight bits as alpha, red, green and blue, respectively. Four separate circles are used to represent V1, V2, V3 and V4 and are colored using different "shape colors" (SC) as (CI, 255, 255, 255), (255, CI, 255, 255), (255, 255, CI, 255) and (255, 255, 255, CI), respectively. Furthermore, using (CI, 0, 0, 0), (0, CI, 0, 0), (0, 0, CI, 0) and (0, 0, 0, CI) is another possible way of implementing the four different variables. If bit block $[p, q]$ represents the CI of variable P, then P can be represented as $P_{[p,q]}$. Then, $P^{(\bar{i},\bar{j})}_{[p,q]}$ represents the pixel color of pixel $(\bar{i}, \bar{j})$ of the DPS, where $(\bar{i}, \bar{j})$ is the pixel of the DPS, which coincides with the pixel *(i, j)* of the MVML-GKU.

When updating the MVML-GKU, first, the $C^{(i,j)}_{[p,q]}$ is calculated using Equation (4), and then, $P^{(\bar{i},\bar{j})}_{[p,q]}$ is added. Therefore:

$$C^{(i,j)}_{[p,q]} = C^{(i,j)}_{[p,q]} + P^{(\bar{i},\bar{j})}_{[p,q]} \tag{5}$$

Finally, the value $C^{(i,j)}_{[p,q]}$ is used to update the layers of the MVML-GKU using Equation (6).

$$c^{(i,j)|l}_{[p,q]|l} = \text{QUOTIENT}\left(C^{(i,j)}_{[p,q]} - \sum_{m=l+1}^{L-1} c^{(i,j)|m}_{[p,q]|m}, \left(2^{(q-p+1)} - 1\right)^{l}\right) \tag{6}$$

where $\sum_{m=l+1}^{L-1} c^{(i,j)|m}_{[p,q]|m} = 0$ when $m > L$

When using Equation (6), it updates the MVML-GKU starting from the last layer and ends with the first layer (layer 0). However, depending on the requirements, it is also possible to update the MVML-GKU starting from Layer 0 with different approaches.

Variables with different scales or different value ranges require a larger bitmap for representing the actual value ranges of the variable values. This is not peaceable, and it is necessary to transform all of the values of different variables to one scale. To accomplish the transformation, a normalization technique known as "min-max normalization" is used. Min-max normalization performs a linear transformation on the original data mapped into a new range using Equation (7) [26,27].

$$v' = ((v - min_A)/(max_A - min_A)) \times (new\_max_A - new\_min_A) + new\_min_A \tag{7}$$

where v' is the transformed value of v, min_A is the minima among all of the data points, max_A is the maxima among all of the data points, new_max_A is the ceiling value of the new range and new_min_A is the floor value of the new range.

If the pixel at coordinates (0, 0) of a bitmap is the origin, then $new\_min_A = 0$. Then, Equation (7) becomes:

$$v' = ((v - min_A)/(max_A - min_A)) \times new\_max_A \tag{8}$$

The width of the bitmap is used for representing the independent variable, while the height of the bitmap is used for representing multiple dependent variables. Thus, the width of the bitmap is proportional to the range of independent variables, and the height is proportional to the maximum range among the ranges of all dependent variables. If the height and the width of the bitmap is h and w, respectively, then:

$$h = R_{maxD} \times F_{maxD} \tag{9}$$

$$w = R_I \times F_I \tag{10}$$

where R_{maxD} is the maximum range among the ranges of all dependent variables, R_I is the range of the independent variable, F_{maxD} is the scaling factor of the dependent variable that has the maximum range among all ranges of all dependent variables and F_I is the scaling factor of the independent variable.

In the concept of the GKU, it always uses the integer part of a number for mapping on the GKU. Because of that, it is a must to scale up data that have a small range and decimal values to minimize the influence due to truncation. Selecting appropriate values for new_max_A for Equation (8) will establish the new set of transformed data. The method of selecting a suitable value for new_max_A is mentioned in the next paragraph. Furthermore, min-max normalization transforms the data series into a positive series, which is another requirement of the GKU. These transformation data are saved in the "GKU specific data area" as a color [25].

With multiple variables, there are two possible ways of scaling. The first method is to scale all of the parameters using a single scaling factor (F). This will change all of the values of the variables while keeping the original ratio between variable values. The name "absolute scaling" will refer to such a scaling technique, from now onwards. In "absolute scaling", new_max_A for all dependent variables is not the same, but F_{iD} is the same for all dependent variables, where F_{iD} is the scaling factor of i-th dependent variable. The scaling factor of the variable with the height range can be considered as the common scaling factor for all of the dependent variables. Define R_{iD} as the range of the i-th dependent variable, and new_max_{Ai} is the new maximum of the i-th dependent variable. Then,

$$F_{iD} = F_{maxD} \tag{11}$$

From Equation (9):

$$F_{iD} = h/R_{maxD} \tag{12}$$

$$new\_max_{Ai} = R_{iD} \times F_{iD} \tag{13}$$

The second method is to scale all of the variables in a manner such that the maximum and minimum of all of the variables coincide with each other. The name "relative scaling" will refer to such scaling techniques, from now onwards. In "relative scaling", new_max_A for all dependent variables is the same, but F_{iD} is not the same for all dependent variables. "h", which is the previously decided height of the image, can be considered as the common value for new_max_{Ai}. Thus, for "relative scaling":

$$new\_max_{Ai} = h \tag{14}$$

Depending on the scaling method, a compatible new_max_A can be calculated using either Equation (13) or Equation (14).

To have a visually meaningful GKU, it is necessary to have a higher number of overlapping clusters. If there are adequate numbers of overlapping clusters, the GKU shows different density areas separated by automatically-generated "contour lines". If the number of data points is smaller, it is still possible to have more overlapping using two techniques: (1) use a bigger shape as the DPS and (2) use a higher number as the CI in SC. Using either technique, it is possible to generate higher color values in pixels and to create visible clusters. However, these techniques are important only for better visualization of the clusters. The absence of visual clusters has no effect on understanding the content of the GKU.

5. Datasets

Two online datasets are used, which are recorded from NIR spectroscopy at a biogas plant. The first dataset is recorded over a period of nearly 25 days with a frequency of 115 values per day, giving a total of 2885 data points. The second dataset is recorded over a period of 75 days with a frequency of 20 values per hour, giving a total of 21,591 data points. In both datasets, the concentration of volatile solids (VS) was selected as the independent variable, and the concentrations of total volatile fatty acids (TVFA), acetic acid (AA) and propionic acid (PA) were selected as the dependent variables. The selected data were part of a dataset that was used to develop a near-infrared (NIR) spectroscopy online calibration for monitoring volatile solids (VS) and volatile fatty acids (VFA) as process indicators during anaerobic digestion [28].

We omitted the alpha portion of the pixel, because it is not visible. Therefore, we used three dependent variables against the independent variable. A circle of a radius of 10 pixels is selected as the DPS for three dependent variables. Three different colors were used as the CI for DPSs. All algorithms in relation to the MVML-GKU were programmed and implemented in the validation process.

6. Results and Discussion

The results included in this section represent different forms of the MLMV-GKU representations, which were obtained using different techniques. Furthermore, this section deliberates the methods and interpretation of the MLMV-GKUs. Figure 5 shows a conventional scatter plot, which presents the 2885 occurrences of three dependent variables (TVFA, AA and PA) against the independent variable (VS). Effective value ranges of TVFA, AA and PA are [0, 8], [0, 5] and [0, 5], respectively. The plot itself does not convey strong evidence on data clustering or data density. On the other hand, due to overlapping, it is difficult to identify some of the data points that may lie under other data points.

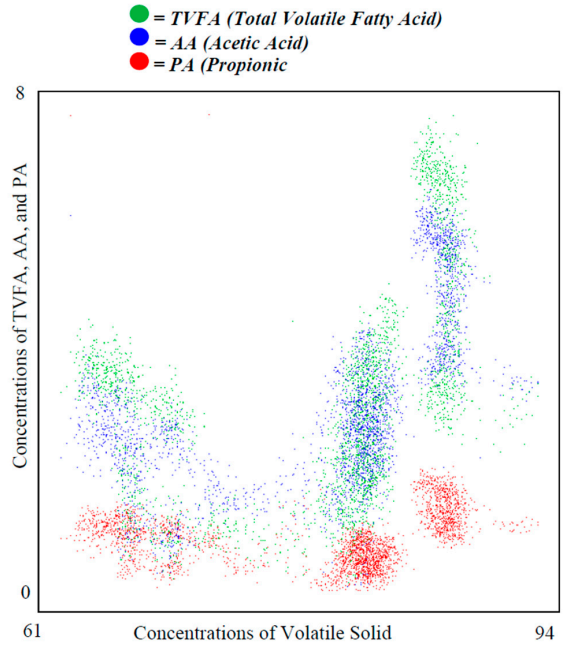


Figure 5. A conventional scatter plot for representing the data of three dependent variables (concentrations of total volatile fatty acids (TVFA), acetic acid (AA) and propionic acid (PA)) against the concentration of volatile solids. Each dependent variable consists of 2885 data points. The ranges of TVFA, AA and PA are [0, 8], [0, 5], and [0, 5], respectively. Due to over-plotting, the large number of data points and the multi-variable environment, this plot itself does not provide direct useful information about data clusters. This is the major drawback of using a scatter plot for representing multi-variable big data.

Plots in Figure 6 show the plotting of the same dataset shown in Figure 7, on the MLMV-GKU of two layers using "relative scaling". The MLMV-GKU was implemented using a circle with a ten-pixel radius DPS and CI = 20. Three different RGB colors, (255, 255, 20), (255, 20, 255) and (20, 255, 255), were used for representing TVFA, AA and PA, respectively. The effective width and height of the bitmap is 400 and 400 pixels, respectively. Since "relative scaling" is used for scaling the values of variables, according to the Equation (14), new_max_A for each dependent variable is 400 pixels. Based on this, all existing ranges of all of the variables ([0, 8], [0, 5] and [0, 5]) were transformed. In Layer 0 of Figure 6, the formation of clusters that belong to different variables can be identified with the naked eye. This is one advantage of using the concept of the GKU, because it creates visibly dense clusters when there is an adequate amount of overlapping.

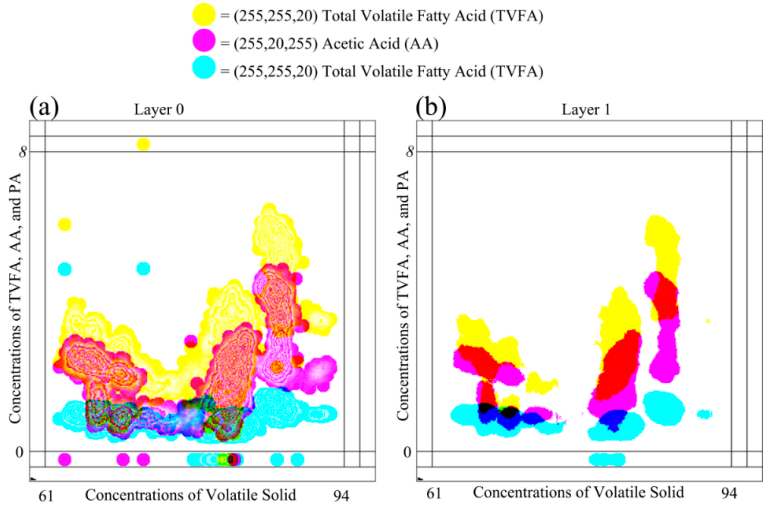


Figure 6. The MLMV-GKU of two layers. (**a**) Layer 0 of the MLMV-GKU; (**b**) Layer 1 of the MLMV-GKU. The effective bitmap width and height are 400 and 400 pixels for representing the data of three dependent variables (concentrations of total volatile fatty acids (TVFA), acetic acid (AA) and propionic acid (PA)) against the concentration of volatile solids where each dependent variable consists of 2885 data points. All of the data are normalized using the max-min normalization method and used 400, 250 and 250 pixels as new_max_A for TVFA, AA and PA, respectively (relative scaling). The initial color of each variable is (255, 255, 20), (255, 20, 255) and (20, 255, 255). A circle of a radius of 10 pixels is used as the data point shape (DPS). The MLMV-GKU shows density clusters of both high and low density areas of each variable by means of automatically-generated contour lines, which can be easily identified by the naked eye.

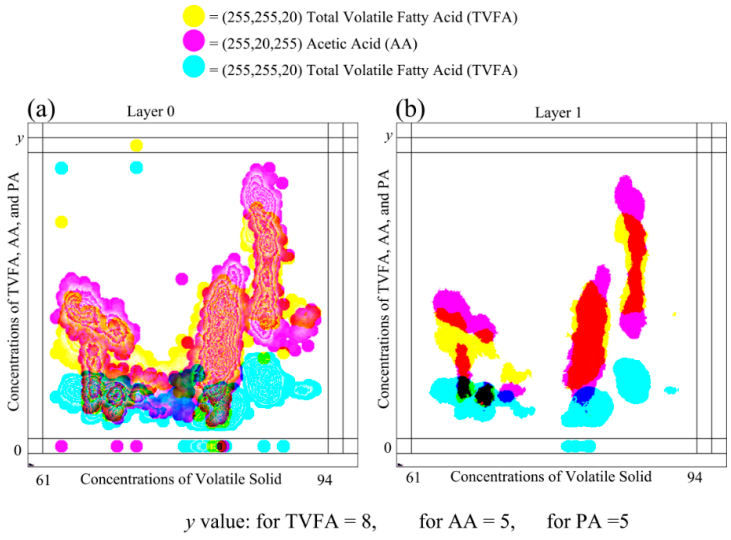


Figure 7. The MLMV-GKU of two layers. (**a**) Layer 0 of the MLMV-GKU; (**b**) Layer 1 of the MLMV-GKU. The effective bitmap width and height are 400 and 400 pixels for representing the data of three dependent variables (concentrations of total volatile fatty acids (TVFA), acetic acid (AA) and propionic acid (PA)) against the concentration of volatile solids where each dependent variable consists of 2885 data points. All of the data are normalized using the max-min normalization method using 400 pixels as new_max_A for all dependent variables (absolute scaling). The initial color of each variable is (255, 255, 20), (255, 20, 255) and (20, 255, 255). A circle of a radius of 10 pixels is used as the data point shape (DPS). The MLMV-GKU shows the density clusters of each variable by means of automatically-generated contour lines, which can be easily identified by the naked eye.

Furthermore, Figure 6 shows a considerable amount of clusters between different variables. If there are clusters due to two or more variables, the color in the shared area is mixed and shows the clusters with a different color. With the basic knowledge of color formation theory, it is possible to identify those areas easily with the naked eye. Additionally, with image processing, it is also possible to identify these areas by analyzing the color values of pixels. When "Layer 1" is considered, there are no contour lines due to less overlapping. However, in Layer 1, clusters due to different variables can be easily identified due to the different color of the shared regions (e.g., the shared area of TVFA and AA). Furthermore, it is possible to distinguish clusters belonging to different variables. Since the place value of Layer 1 is higher than the place value of Layer 0, the presence of color in a higher layer implies the existence of higher density in the position in relation to the respective pixel. Therefore, the examination of higher layers makes higher density area identification much simpler.

Changing parameter values opens new ways of identifying new relations between different variables [27]. This technique is used as a very efficient tool in most data mining techniques. The MLMV-GKU also supports this feature and easily seeks relations by changing the overlapping between different variables. Figure 6 shows layers of the MLMV-GKU of the same dataset (scaled with the relative scaling technique), which is used in Figure 7, and scaled using "absolute scaling". The highest range among all of the dependent variables belongs to TVFA. Since $R_{maxD} = 8$, according to Equation (12), $F_{iD} = 50$ for TVFA, AA and PA. Then, according to Equation (13), new_max_A for TVFA, AA and PA is 400, 250 and 250, respectively.

Both layers in Figures 6 and 7 show overlapping areas of different variables. However, due to different scaling methods, the overlapping areas belong to different variables in Figures 6 and 7 which are not identical. This is a very useful feature when the GKU is used as a knowledge unit. When there is overlapping between variables, it is easy to identify color values of multiple parameters by analyzing a single pixel. This will reduce the overhead of computing. However, to interpret the meaning of a shared area due to one or more variables, sealing factors must be decided after properly investigating the domain requirements. Otherwise, only creating overlap areas between variables may not provide useful information.

Selecting different color schemes for representing the shape color of DPSs has considerable impact on the visual representation ability of GKU. In general, there are two available color schemes for representing three dependent variables using the 24-bit RGB pixel format as (255, 255, CI), (255, CI, 255), (CI, 255, 255) and (0, 0, CI), (0, CI, 0), (CI, 0, 0). Each color in the first version is near to the white color, which is represented as (255, 255, 255), and each color in the second version is near to the black color, which is represented as (0, 0, 0).

Figure 8 elaborates the impact of using the color scheme for the color of DPSs. Figure 8 contains the same data used in Figure 6 with the same scaling parameters. However, (0, 0, 20), (0, 20, 0) and (20, 0, 0) were used as colors of DPSs instead of (255, 255, 20), (255, 20, 255) and (20, 255, 255). The colors (255, 255, 20), (255, 20, 255) and (20, 255, 255) are not identical and are visually identifiable. Though colors (0, 0, 20), (0, 20, 0) and (20, 0, 0) are numerically identical, they are not visually identifiable because all of these colors are equivalent to the black color. The major drawback of this scheme is that the shape colors of DPSs cannot be visually identifiable, though they are different colors. This leads to difficulties in identifying clusters visually, especially in low density areas. When examining Layer 1 of Figure 8, this phenomenon is more visible. In Layer 1, neither clusters between different variables nor density clusters of individual variables are possible to identify. Furthermore, in Layer 0 of Figure 8, except high density areas, all of the other areas are in black or nearly black color, and this prevents visual identification of low density areas in relation to different variables. This phenomenon can be easily identified when considering the points P_A and P_B, as shown in Figure 8. It is not possible to distinguish between P_A and P_B visually due to the colors of the DPSs. However, Figure 6, these two points can be easily identified. Nevertheless, in reality, all black areas contain different colors (e.g., (0, 0, 40), (0, 40, 0) and (40, 0, 0)), which is visible only to an algorithm. Thus, selection of this type of color is not a problem for identifying clusters in low density areas with a computer. As a

recommendation, it is stated that the different colors close to white are the best selection for the MLMV-GKU for visually capturing clusters between different variables, as well as density clusters of individual variables.

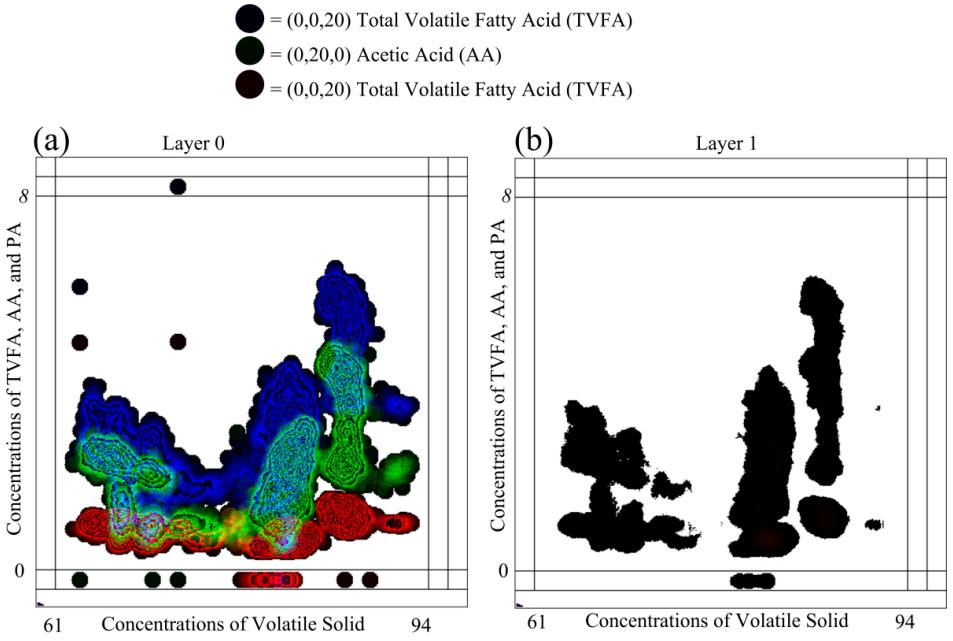


Figure 8. The MLMV-GKU representation of the same data shown in Figure 6 with a different color scheme. (**a**) Layer 0 of the MLMV-GKU; (**b**) Layer 1 of the MLMV-GKU. As in Figure 6, the effective bitmap width and height are 400 and 400 pixels for representing the data of three dependent variables (concentrations of total volatile fatty acids (TVFA), acetic acid (AA) and propionic acid (PA)) against the concentration of volatile solids where each dependent variable consists of 2885 data points. All of the data are normalized using the max-min normalization method and used 400, 250 and 250 pixels as new_max_A for TVFA, AA and PA, respectively. The initial color of each variable is (0, 0, 20), (0, 20, 0) and (20, 0, 0). A circle of a radius of 10 pixels is used as the DPS. The MLMV-GKU shows visual density clusters only in high overlapping areas. When there is low overlapping, it is not possible to distinguish points in relation to variables, such as P_A, P_B and Layer 0.

Plot (a) of Figure 9 shows the compression of the conventional scattered plot for one independent and three dependent variables and plot (b) of Figure 9 the MLMV-GKU of two layers for the same variables. The plots represent 21,951 data points, where the effective bitmap width and height are 400 and 625 pixels, respectively. Plots represent data in relation to three dependent variables (TVFA, AA and PA) against the concentration of volatile solids (VS). All of the data are normalized using the max-min normalization method and used 625, 550 and 225 pixels as new_max_A for TVFA, AA and PA, respectively. In the MLMV-GKU, the initial color of each variable is (255, 255, 10), (255, 10, 255) and (10, 255, 255). A circle of a radius of 10 pixels is used as the DPS. The MLMV-GKU shows density clusters of each variable by means of automatically-generated contour lines, which can be easily identified by the naked eye. This is a special feature of the GKU, which is not possible with a scatter plot.

As mentioned in Section 1, star coordinates directly visualize dimensions as groups in a form of high density clusters in a two-dimensional plot. It is one of the better visualization techniques. However, star coordinates suffer from cluster overlapping of clusters of variables [17,20]. This is the major drawback of the star coordinates. If there are no overlapping regions, star coordinates can be used to visualize high density data clusters. However, the data we used have overlapping regions and cannot be visualized using star coordinates.

The nature of the MLMV-GKU allows the usage of a higher number of layers for representing variables. Unfortunately, this will lead to problems in understanding the whole picture. This is a very common problem in understanding scatterplot matrices [12]. To prevent this problem, it is recommended to use the maximum of four layers where the visual representation is important. Using four layers, it is possible to show 255 × 255 × 255 × 255 (more than four billion) overlaps using eight bits per variable, in the 32-bit pixel format. When using the 32-bit pixel format, basically, it is possible to show four dependent parameters against one independent parameter. This will provide space for more than 16 billion cases of overlapping (four billion × four variables) in a certain pixel coordinate of the four-layer GKU. If the width and height of the data plotting area are w pixels and h pixels, respectively, then it is possible to visualize the maximum of $x \times y \times 16$ billion cases of overlapping in the four-layer GKU. This is a huge amount of overlapping occurrences, and none of the existing graphical visualization techniques are capable of visualizing this amount of overlapping. Although it is recommended to use a maximum of four layers, there is no mathematical restriction to using more than four layers, depending on the requirement.

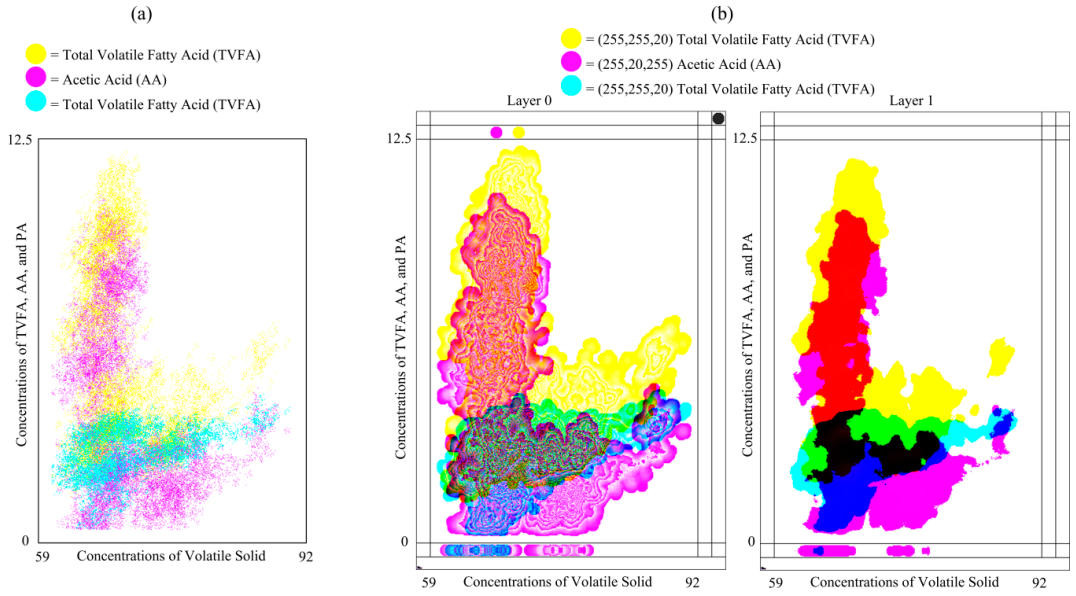


Figure 9. Scatter plot (**a**) and MLMV-GKU of two layers (**b**) representation of 21951 data points where effective bitmap width and height are 400 and 625 pixels. Plots representing the data of three dependent variables (concentrations of total volatile fatty acids (TVFA), acetic acid (AA) and propionic acid (PA)) against the concentration of volatile solids. All of the data points are normalized using the max-min normalization method and used 625, 550 and 225 pixels as new_max_A for TVFA, AA and PA, respectively. In the MLMV-GKU, the initial color of each variable is (255, 255, 10), (255, 10, 255) and (10, 255, 255). A circle of a radius of 10 pixels is used as the DPS. The MLMV-GKU shows density clusters of both high and low density areas of each variable by means of automatically-generated contour lines, which can be easily identified by the naked eye, which is not possible with scatter plots.

The major drawback of MLMV-GKU is the influence of outliers, which leads to artificial high density areas on the GKU. Once these outliers are mixed with non-outlier values of other variables, this leads to incorrect decision making. Furthermore, due to the influence of outliers, the actual scaling cannot be achieved. P_A in Figure 8 can be considered as an example for such an outlier. Because of the influence of this outlier, it was considered as the maximum of the dataset, even though it is an outlier. Therefore, before plotting the MLMV-GKU, it is essential to remove outliers. The other disadvantage of the MLMV-GKU is that it cannot be directly used to find the correlation between variables, though sometimes, it helps to identify some trends between variables. Nevertheless, domain-related correlation can be defined after considering individual situations. For example, in the considered domain, the

concentration of TVFA is proportional to the concentration of AA [29]. In Figure 6, clusters in relation to TVFA and AA have nearly the same shape and reveal the domain conditions more convincingly.

Dividing the 32 bits into small portions will facilitate using more variables in the MLMV-GKU. For example, if four bits are used per variable, it will provide room for eight dependent parameters to be clustered in the 32-bit format. Then, it is necessary to increase the number of layers to accommodate the higher number of data points. As previously mentioned, this is not always a good solution, especially if the aim is to use the MLMV-GKU for visualization. Therefore, the best solution is to use pixel formats that provide a higher number of bit ranges. There are different ARGB color formats that provide up to 128 bits per color channel (total 512 bits) [30,31]. Nevertheless, still, the limit of eight bits per variable and four layers for the MVML-GKU can be maintained.

The GKU is itself a database and knowledge unit that can be used as the input values for an algorithm; thus all of these properties are inherited by the MLMV-GKU, as well. The meaning of the content can be altered by changing three parameters of the DPS: firstly, the type of shape; secondly, the color of the shape; and thirdly, the position of the data point in the shape. The type of shape that is used in all of the plots in this article is circular. Furthermore, it is possible to use a suitable shape (e.g., square, polygon) instead of a circle. Furthermore, using different types of DPSs for different variables, it is possible to identify/show the effect/influence of a variable on other variables in a more meaningful manner. When a circle is selected with a single color as the DPS, this implies that the effect of the data point is equally valid for all of the pixels inside the circle. However, if the color of the circle is selected in a way that the color is proportional to a certain property (e.g., distance to the data point), this can be used to represent the functional influence of a data point. In the examples shown in this article, the data points were located in the center of the circle. By applying those property changes, the meaning of the MLMV-GKU can be enhanced according to different domain requirements.

If there is a dataset scheduled to be collected for a time period of T, at a time t_k ($t_k < T$), all of the data placed on the MLMV-GKU are processed up to time t_{k-1}. Therefore, at any time, the data in the MLMV-GKU can be used to understand the current situation with or without further processing. If further processing is required, a copy of current bitmaps can be used for processing, while the original MLMV-GKU keeps updating. Because each layer is an independent bitmap, each layer can be individually analyzed using a separate process without depending on other layers. This feature enables the MLMV-GKU to be a parallel-processing-ready technique. After analyzing, the final decision can be obtained by summarizing the results of each layer. This could reduce the total processing time to below the usual single process analysis time.

7. Conclusions and Outlook

The MLMV-GKU is a technique that facilitates the visualization of multiple variables with a big amount of data. It does not show each data point individually and shows abstracted information as density clusters. Furthermore, the MLMV-GKU is a highly enhanced version of the GKU, which inherited all of the features of the GKU. The MLMV-GKU is an ideal tool for visualizing big data and fulfils a highly demanding requirement in the field of big data visualization. Using a four-layer GKU, it is possible to show more than four billion instances of overlapping using eight bits per variable, in the 32-bit pixel format. This will provide space for more than 16 billion different overlapping situations in a certain pixel coordinate. Thus, the MLMV-GKU will provide a fast means of decision making, which will be one of the major factors in process control. None of the existing graphical visualization techniques is capable of visualizing this amount of overlapping.

As an outlook for further developing the MLMV-GKU for the online environment, the usage of the 3D bitmap will facilitate the usage of another additional variable, the z-axis, in contrast to the current concept of the 2D bitmap, which facilitates $(1 + n)$ variables.

Acknowledgments: We are grateful to the German Academic Exchange Service (Deutscher Akademischer Austauschdienst) for providing a scholarship to K.K.L.B. Adikaram during the research period.

Author Contributions: Conceived of and designed the experiments and algorithms: K.K.L.B.A. Performed the experiments: K.K.L.B.A. Analyzed the data: K.K.L.B.A., M.A.H. and M.E. Contributed reagents/materials/analysis tools: K.K.L.B.A., M.A.H., M.E. and T.B. Wrote the paper: K.K.L.B.A.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Keim, D.A. Information visualization and visual data mining. *IEEE Trans. Vis. Computer Graph.* **2002**, *8*, 1–8. [CrossRef]
2. Cvek, U.; Trutschl, M.; Stone, R.; Syed, Z.; Clifford, J.L.; Sabichi, A.L. Multidimensional visualization tools for analysis of expression data. *World Acad. Sci. Eng. Technol.* **2009**, *30*, 281–289.
3. Barbará, D.; Chen, P. Using self-similarity to cluster large data sets. *Data Min. Knowl. Discov.* **2003**, *7*, 123–152. [CrossRef]
4. David, G.; Averbuch, A. Hierarchical data organization, clustering and denoising via localized diffusion folders. *Appl. Comput. Harmon Analy.* **2012**, *33*, 1–23. [CrossRef]
5. Galluccio, L.; Michel, O.; Comon, P.; Hero, A.O., III. Graph based K-means clustering. *Signal Process.* **2012**, *92*, 1970–1984. [CrossRef]
6. Chen, F.; Deng, P.; Wan, J.; Zhang, D.; Vasilakos, A.V.; Rong, X. Data mining for the internet of things: Literature review and challenges. *Int. J. Distrib. Sens. Netw.* **2015**, *501*. [CrossRef]
7. Chen, M.; Ebert, D.; Hagen, H.; Laramee, R.S.; van Liere, R.; Ma, K.L.; Ribarsky, W.; Scheuermann, G.; Silver, D. Data, information, and knowledge in visualization. *IEEE Comput. Graph. Appl.* **2009**, *29*, 12–19. [CrossRef] [PubMed]
8. Akerkar, R. *Big Data Computing*; CRC Press: Boca Raton, FL, USA, 2013.
9. Tsai, C.W.; Lai, C.F.; Chao, H.C.; Vasilakos, A. Big data analytics: A survey. *J. Big Data* **2015**, *2*. [CrossRef]
10. Karimi, H.A. *Big Data: Techniques and Technologies in Geoinformatics*; CRC Press: Boca Raton, FL, USA, 2014.
11. Fong, S.; Wong, R.; Vasilakos, A. Accelerated PSO swarm search feature selection for data stream mining big data. *IEEE Trans. Serv. Comput.* **2015**, *9*, 33–45. [CrossRef]
12. Chambers, J.M.; Cleveland, W.S.; Kleiner, B.; Tukey, P.A. *Graphical Methods for Data Analysis*; Wadsworth: Belmont, CA, USA, 1983.
13. Inselberg, A.; Dimsdale, B. Parallel coordinates: A tool for visualizing multi-dimensional geometry. In Proceedings of the 1st Conference on Visualization, San Francisco, CA, USA, 23–26 October 1990; IEEE Computer Society Press: San Francisco, CA, USA, 1990; pp. 361–378.
14. Andrews, D.F. Plots of high-dimensional data. In *Biometrics*; International Biometric Society: Washington, DC, USA, 1972; pp. 125–136.
15. Bartke, K. 2D, 3D and High-Dimensional Data and Information Visualization. Available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.83.3421&rep=rep1&type=pdf (accesed on 28 March 2016).
16. Hoffman, P.; Grinstein, G.; Marx, K.; Grosse, I.; Stanley, E. DNA visual and analytic data mining. In Proceedings of the IEEE Visualization, Phoenix, AZ, USA, 24 October 1997; pp. 437–441.
17. Van Long, T.; Linsen, L. Visualizing high density clusters in multidimensional data using optimized star coordinates. *Comput. Stat.* **2011**, *26*, 655–678. [CrossRef]
18. Hoffman, P.; Grinstein, G. Visualizations for High Dimensional Data Mining-Table Visualizations. Available online: http://web.simmons.edu/~benoit/infovis/MIV-datamining.pdf (accessed on 28 January 2014).
19. Hoffman, P.E.; Grinstein, G.G. A survey of visualizations for high-dimensional data mining. In *Information Visualization in Data Mining and Knowledge Discovery*; Usama, F., Georges, G.G., Andreas, W., Eds.; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2002; pp. 47–82.
20. Danyu, L.; Sprague, A.P.; Gray, J.G. PolyCluster: An interactive visualization approach to construct classification rules. In Proceedings of the 2004 International Conference on Machine Learning and Applications, Louisville, KY, USA, 16–18 December 2004; pp. 280–287.
21. Baraldi, A.N.; Enders, C.K. An introduction to modern missing data analyses. *J. Sch. Psychol.* **2010**, *48*, 5–37. [CrossRef] [PubMed]
22. Schafer, J.; Graham, J. Missing data: Our view of the state of the art. *Psychol. Methods* **2002**, *7*, 147–177. [CrossRef] [PubMed]

23. Nurunnabi, A.A.M.; Imon, A.H.M.R.; Ali, A.B.M.S.; Nasser, M. Outlier Detection in Linear Regression. In *Computational Modeling and Simulation of Intellect: Current State and Future Perspectives*; IGI Global: Hershey, PA, USA, 2011; pp. 510–550.
24. Beckman, R.J.; Cook, R.D. Outlier . . . . . s. *Technometrics* **1983**, *25*, 119–149.
25. Adikaram, K.K.L.B.; Hussein, M.A.; Effenberger, M.; Becker, T. Continuous learning graphical knowledge unit for cluster identification in high density data sets. *IEEE Trans. Vis. Comput. Graph.* under review. **2016**.
26. Han, J.; Kamber, M.; Pei, J. *Data Mining, Southeast Asia Edition: Concepts and Techniques*; Elsevier Science: Amsterdam, The Netherlands, 2006.
27. Shalabi, L.A.; Shaaban, Z.; Kasasbeh, B. Data mining: A preprocessing engine. *J. Comput. Sci.* **2006**, *2*, 735–739. [CrossRef]
28. Krapf, L.C.; Heuwinkel, H.; Schmidhalter, U.; Gronauer, A. The potential for online monitoring of short-term process dynamics in anaerobic digestion using near-infrared spectroscopy. *Biomass Bioenergy* **2013**, *48*, 224–230. [CrossRef]
29. Gronauer, A.; Krapf, L.; Heuwinkel, H.; Schmidhalter, U. Near infrared spectroscopy calibrations for the estimation of process parameters of anaerobic digestion of energy crops and livestock residues. *J. Near Infrared Spectrosc.* **2011**, *19*, 479–493. [CrossRef]
30. Nikiel, S. *Iterated Function Systems for Real-Time Image Synthesis*; Springer: Berlin, Germany; Heidelberg, Germany, 2007.
31. Gelphman, D.; Laden, B. *Programming with Quartz: 2D and PDF Graphics in Mac OS X*; Elsevier Science: Amsterdam, The Netherlands, 2010.

3. Discussion

I have very promising expectations about the positive impact of the developed non-parametric methods for uplifting and opening new doors in the fields of knowledge discovery, data mining, feature identification, and data processing. All the concepts and methods presented in this thesis are totally new to science and have proven accuracy. Therefore, researchers and scientists could apply those methods for uplifting the standards of their research findings.

The Universal Linear Fit Identification (UniLiFI)

The Universal Linear Fit Identification (UniLiFI) method showed ability of identifying all the linear fits (regression is initially unknown) contaminated with all possible maximum combinations of the following conditions:

1. Size of data points (range 4 to 1000)
2. Nature of outliers/noise distribution (Gaussian and non-Gaussian distribution)
3. Amount of outliers/noise (50% - 55%)
4. Range of deviation of outliers / noise ($\pm 10^{4}$ % to $\pm 10^{-4}$%)
5. Existence of initial missing data

Combinations of the above mentioned conditions produce heterogeneous data sets. Particularly, identifying a linear fit in such data sets requires a combination of different methods that is identical to the nature of each data set. However, it may be possible to locate nearly correct linear fit with a certain level of error, which is the maximum. As afore-mentioned, this is the one major disadvantage of parametric methods. In contrast, UniLiFI was able to identify all the linear fits in different forms (with positive gradient, with negative gradient and constant) in all heterogeneous datasets without swapping or masking.

In the real world, certain models can be applied for certain domains, assuming that the domain conditions remain unchanged. When domain conditions change, the model may not give a correct output. The strength of the UniLiFI method is that it is capable of identifying all the linear fits in all the data sets with 0% error, without depending on the domain conditions. It is a very hard task to locate all (or almost all) data points that agree with the linear fit when the number of outliers is

around 50% and deviation is in the range of $\pm 10^{4}$ to $\pm 10^{-4}$. Therefore, there are very promising expectations towards the possible positive gain to science from the new method “UniLiFI”.

One of the main disadvantages of the linear least squares method is that it is not capable of identifying linear regression when the number of data points is high [76]. In contrast, our results showed that the accuracy of Linear Fit Identification method UniLiFI is independent of the number of data points. UniLiFI method showed the same level of accuracy for locating existing linear fit by identifying linear fit from data sets with 4 to 1000 data.

The UniLiFI method introduced in this work is not only a linear fit identification method. The UniLiFI method can also be used as outlier/noise detection method in linear regression. Not only detecting outliers and noise, but also the method is capable of grouping those outliers/noise into several subgroups according to the *k* value. However, it is possible to increase the number of outlier/noise groups by defining several *k* values. For single *k* value there are amaximum of up to three groups as linear fit (clean data) and two outlier (and/or noise) groups; one above the clean data and other below the clean data. For two *k* values, there are maximum of up to five groups as linear fit and four outlier (and/or noise) groups as two above and two below the linear fit. Thus, if there is *p* number of *k* values, maximum of up to *2p* outlier (and/or noise) groups can be achieved (Figure 3.1). Furthermore, borders of regions are nearly parallel (Figure 3.1).

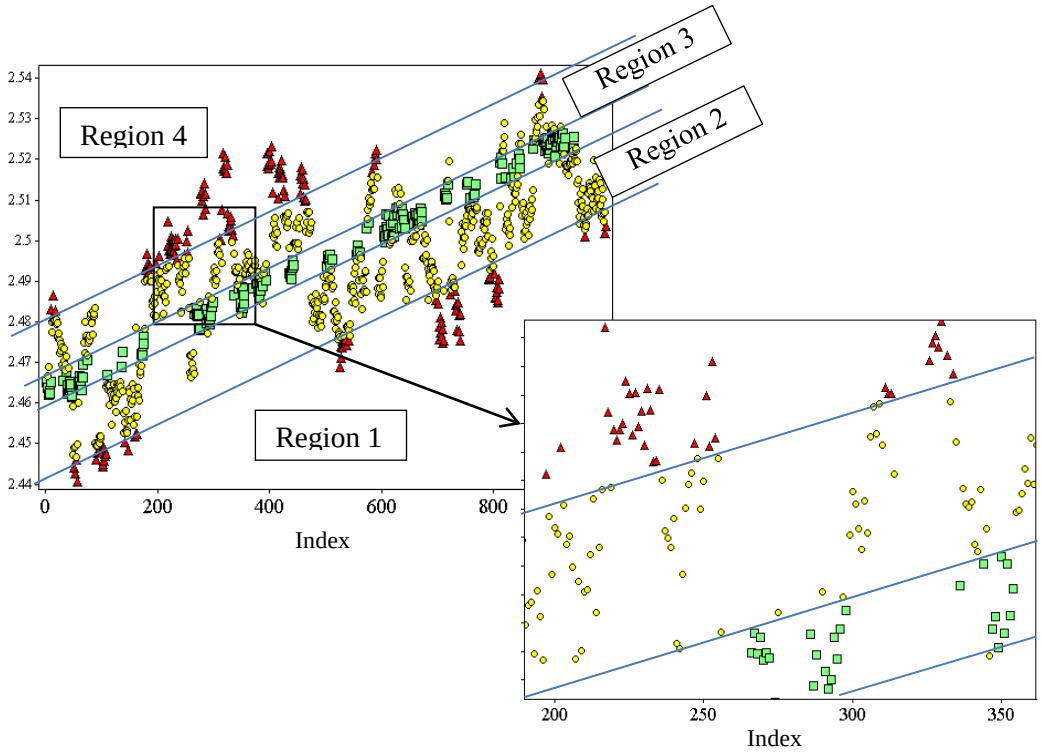


Figure 3.1: If there are *p* numbers of *k* values, a maximum of up to *2p* outliers (and/ or noise) groups can be achieved. Setting two k values gives four regions of outlier/ noise regions (Regions 1 to 4). Also, borders of regions are nearly parallel.

The plot in Figure 3.2 shows a data set of original size 1000 data points, which initially agrees with a linear fit. Data of two regions containing 150 and 100 (total 250) data points were removed for creating initial missing data environment. Out of the rest 750 points, 421 data points (more than 50%) were replaced with data that deviated $\pm 10^{4}$% to $\pm 10^{-4}$% from the original value. The replaced data do not totally agree with the Gaussian distribution. Furthermore, there is a nearly linear fit also existing in the data set. All those conditions made the identification of a correct linear fit very difficult. The trend linein Figure 3.2 shows the identified linear regression using GLSM. Determination of the linear fit based on GLSM will remove the majority of data points belonging to both the near linear fit as well as the exact linear fit.

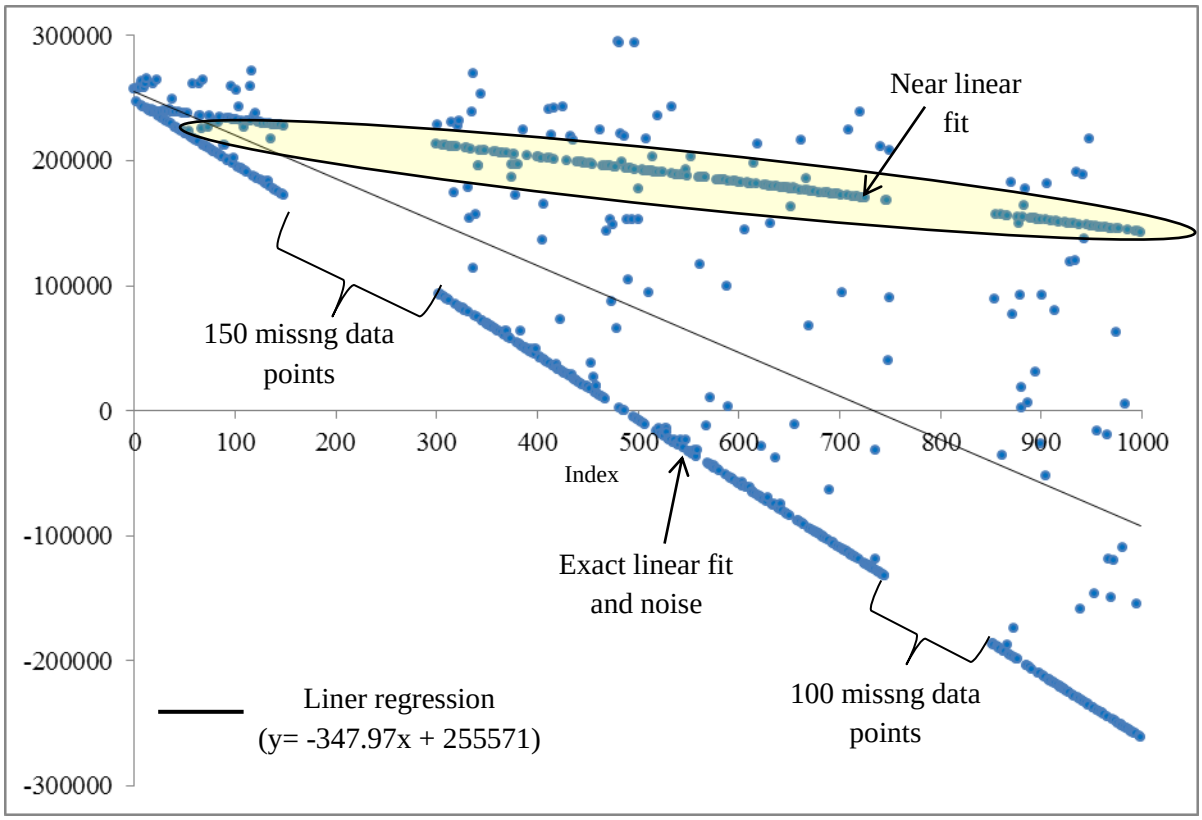


Figure 3.2: Data of two regions containing 150 and 100 (total 250) data points that were removed to create an initial missing data environment. Out of the rest 750 points, 421 data points (more than 50%) were replaced with data that deviated $\pm 10^{4}$% to $\pm 10^{-4}$% from the original value.

Figure 3.3 shows identification of linear fit for the same data set using UniLiFI method. This elaborated the capacity of the UniLiFI method. Firstly, it was capable of identifying all the data points that agreed with the linear fit, without swapping or masking them. Secondly, it divided the data points that do not agree with the linear fit into two groups based on the two *k* values, *k=0* and *k=0.15*. Identifying data deviated by very high percentages is not a challenge. The most interesting factor is that the UniLiFI method is capable of identifying even a data point that is deviated by a very small percentage such as $\pm 10^{-4}$%.

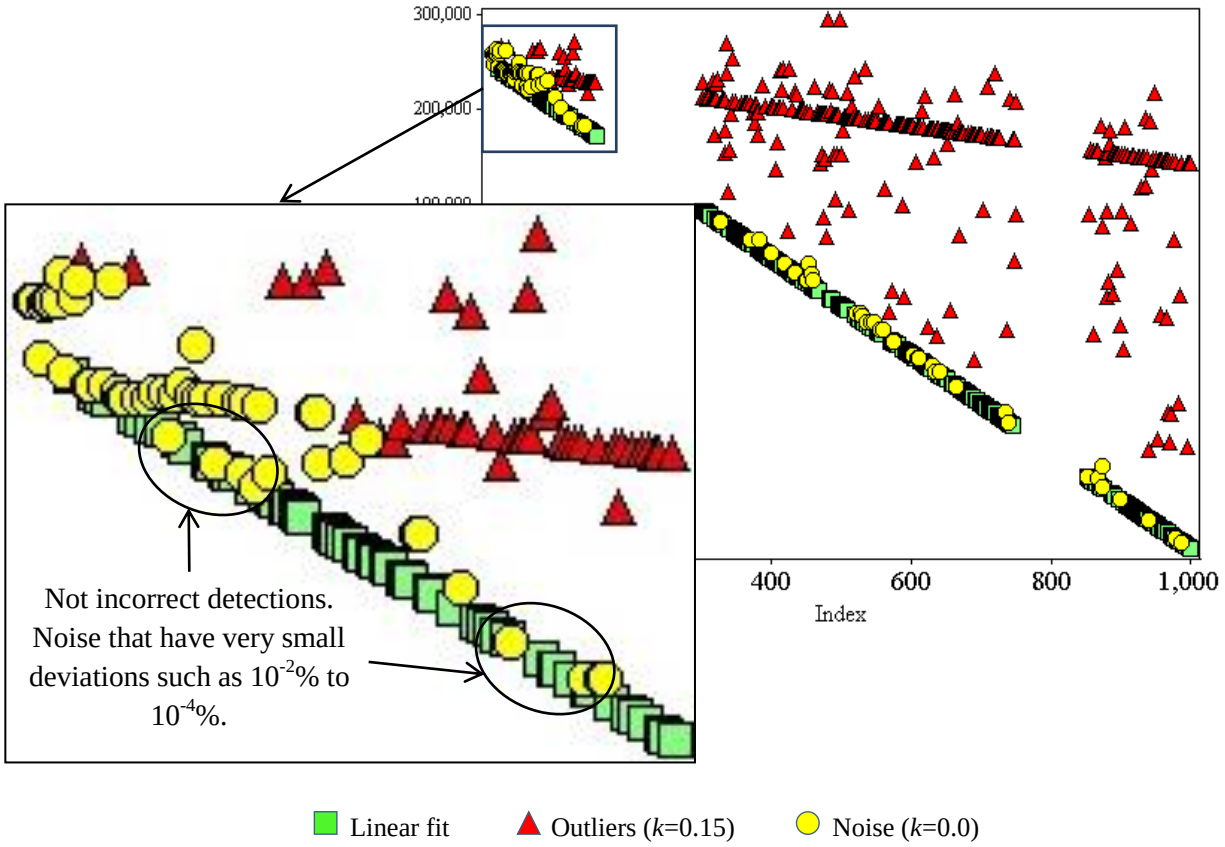


Figure 3.3: Firstly, it was capable of identifying all the data points that agreed with the linear fit, without swapping or masking them. Secondly, it divided the data points that do not agree with linear fit into two groups based on the two *k* values, *k=0* and *k=0.15*. Identifying data deviated by very high percentages is not a challenge. The most interesting factor is that the UniLiFI method is capable of identifying even a data point that is deviated by a very small percentage such as $\pm 10^{-4}\%$.

UniLiFI method is suitable for time series or any data series and can be considered as time series. This feature can be considered as a limitation of UniLiFI. However, most of the data captured in the real world are time series. If not, some situations can be assumed or converted to time series by stamping certain sequence numbers for each data point. The gap between two data points must be equal or must be multiples of the minimum gap. Then, by using the UniLiFI method, it is possible to find the best existing linear fit.

When detecting linear fit, UniLiFI identified the best linear fit in the given window. Most of the time, the best fit is the dominating linear fit in the given window (Figure

3.4). However, sometimes UniLiFI identifies data from two or more linear segments as the linear fit (Figure 3.5). This detection cannot be rejected, because it totally agrees with the calculations and is visually justifiable. On the other hand, this can be considered as a drawback of UniLiFI. If the method is capable of identifying all the possible linear fits in a certain window that would be the best.

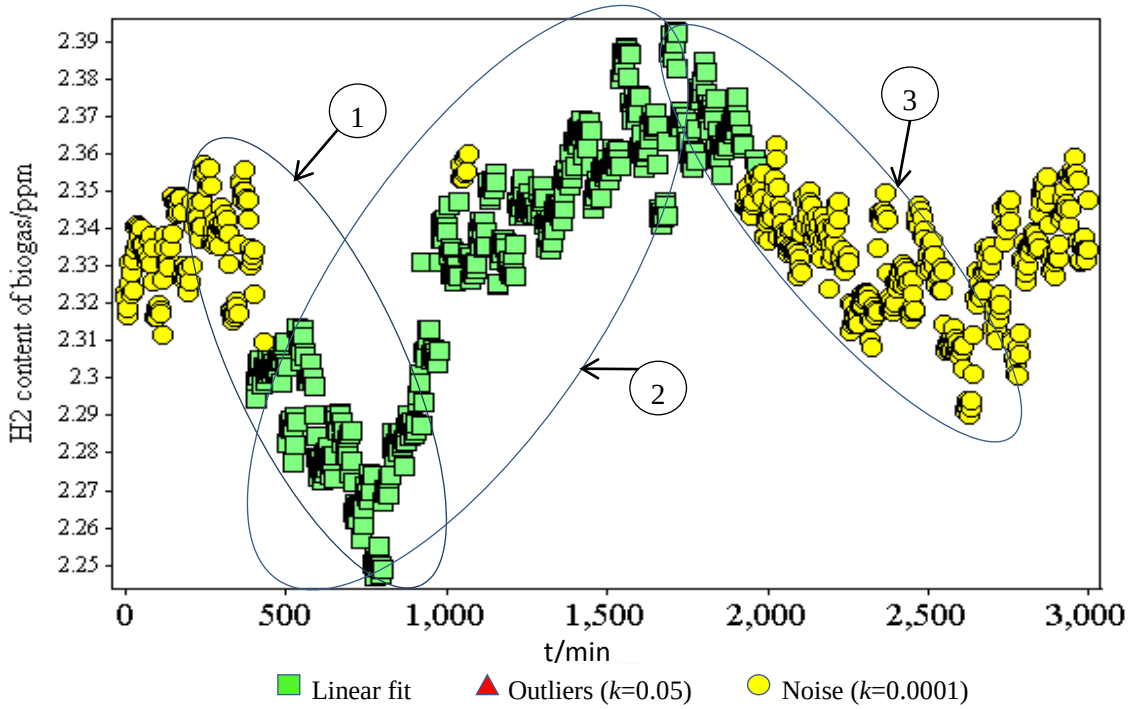


Figure 3.4: There are three potential linear fits as 1, 2, and 3. UniLiFI method identified the most dominating linear fit in the given window, according to the given criteria. Since the dominating linear fit is linear fit 2. UniLiFI identified it as the best fit, according to the given values of the k (k=0.05 and k=0.0001).

One possible remedy for overcoming the aforementioned problem is to manage the nature of input data window by splitting it into several portions from extrema points (Figure 3.6). This will input data window with one potential linear fit and will not cause ambiguous situations. Furthermore, this approach can be considered as applying UniLiFI for cleaning data in non-linear data.

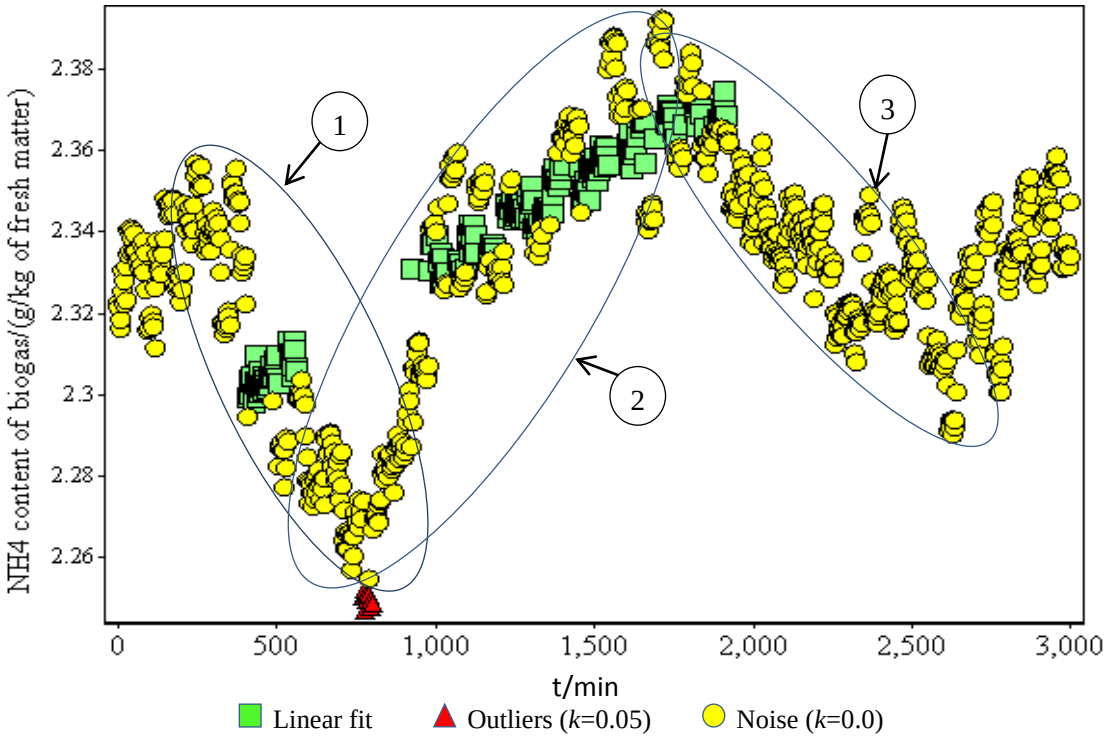


Figure 3.5: There are three potential linear fits as 1, 2, and 3. UniLiFI method identified a line that representing data points belong to fits 1, 2, and 3 as the most dominating linear fit in the given window, according to the given values of the k (k=0.05 and k=0.0). Even though, the dominating linear fit is linear fit 2, UniLiFI did not identify it as the best fit.

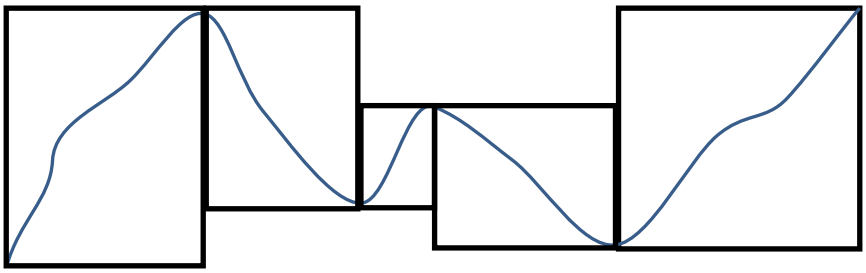


Figure 3.6: If it is possible to split a data set in a manner such that each data window contains only one potential linear fit, UniLiFI can be used to clean non-linear data.

When considering the computation time, the original UniLiFI algorithm is not a one-pass method. For one full circle algorithm, remove only one outlier and again perform the same calculation to detect the next outlier in next iteration. Therefore, this is not an efficient algorithm. There are several places where possible improvements can be done. The original algorithm we introduced and coded, always calculates the sum, checks for the new maximum, and checks for the new minimum after removing an outlier. Instead of that, it is possible to use results of previous

iterations in next coming iterations. This will save a lot of computational time, especially with large amounts of data.

The major challenge in using UniLiFI is the numerical accuracy of the programming language [77]. Visual C++ 2010 version was used for coding the algorithms. Sometimes, calculation errors arise due to numerical accuracy (due to precision error), causing wrong identification of termination point of the algorithm execution. When working with decimal values, it is necessary to use either float or double type. Sometimes, the precision error can be considered as a simple and easily corrected error or a scenario that can be ignored. However, it is not a simple and ignorable scenario; in fact, it is the most critical scenario according to our experience. I spent huge amounts of time verifying the accuracy of UniLiFI due to precision errors. Sometimes very simple calculations did not give the expected results due to a precision error. Table 3.1, Table 3.2, and Table 3.3 show three very simple calculations that elaborate the behaviour of precision errors, when working on Excel 2010 in Windows 7 platform. Note that the real values are visible after formatting the cells to be able to see at least fourteen decimal points. Otherwise, all the values appear to be correct due to round off. In Table 3.1 the sum is influenced by error accumulation. However, in other situations, sums are not influenced by the precision errors. This unpredictable nature causes problems for handing and managing precision errors in a global manner. Figure 3.7 shows a situation that experienced with Visual C++ (Visual Studio 2010) in Windows 7 platform, which was affected by precision errors.

For detecting linear fit in data sets with a large number of data points, numerical accuracy is a very important factor. The linear fit determination criteria are based on *2/n* and *k*, and when *n* (number of data points) is high, *2/n* becomes very small. Thus, the determination is based on decimal values after the 4^{th} decimal point of data sets that have more than 10000 data points is not reliable. Therefore, if the numerical accuracy is not good enough, most of the time the reliability of UniLiFI will be lower. Usage of results contaminated with precision errors for checking termination conditions in “if” clauses or loops give wrong termination points. I faced such situations when determining the exact terminating point after identifying linear fit. Also, sometimes the value of linear correlation coefficient (r_{xy}) is shown incorrect due to precision error (Table 3.1, Table 3.2, and Table 3.3).

Table 3.1: Precision error, when calculating with Excel 2010 in Windows 7 platform. In column $Y_i - Y_1$, lines 3, 4, and 5 contain an error of 10^{-14}. However, finally, this affects the sum. To see the correct values, it is compulsory to format the cells to be able to see at least fourteen decimal points.

X_i	Y_i	$Y_i - Y_1$	Expected Value
1	101.10000000000000	0.00000000000000	0.00
2	102.10000000000000	1.00000000000000	1.00
3	103.01000000000000	1.91000000000001	1.91
4	104.01000000000000	2.91000000000001	2.91
5	105.01000000000000	3.91000000000001	3.91
	Sum	9.73000000000003	9.73

Table 3.2: Precision error, when calculating with Excel 2010 in Windows 7 platform. In column $Y_i - Y_1$, lines 2, 3, 4, and 5 contain an error of $\pm 10^{-14}$. However, finally, this does not affect the sum. To see the correct values it is compulsory to format the cells to be able to see at least fourteen decimal points.

X_i	Y_i	$Y_i - Y_1$	Expected Value
1	101.00000000000000	0.00000000000000	0.00
2	102.10000000000000	1.09999999999999	1.10
3	103.01000000000000	2.01000000000001	2.01
4	104.01000000000000	3.01000000000001	3.01
5	105.01000000000000	4.01000000000001	4.01
	Sum	10.13000000000000	10.13

Table 3.3: Precision error, when calculating with Excel 2010 in Windows 7 platform. In column $Y_i - Y_1$, lines 2, 3, 4, and 5 contain an error of -10^{-14}. However, finally, this does not affect the sum. To see the correct values it is compulsory to format the cells to be able to see at least fourteen decimal points.

X_i	Y_i	$Y_i - Y_1$	Expected Value
1	101.00000000000000	0.00000000000000	0.00
2	102.10000000000000	1.09999999999999	1.10
3	103.10000000000000	2.09999999999999	2.10
4	104.10000000000000	3.09999999999999	3.10
5	105.10000000000000	4.09999999999999	4.10
	Sum	10.40000000000000	10.40

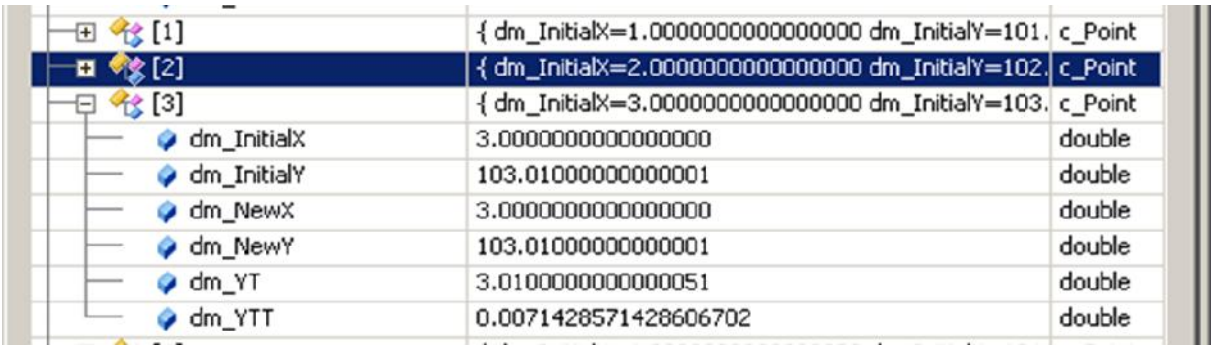


Figure 3.7: Precision error, when programming with Visual C++ (Visual Studio 2010) in Windows 7 platform. The correct value of variable dm_NewY must be 103.01. However, the variable contains 103.01000000000001. The value of variable dm_YT based on dm_NewY and must be 3.01. However, it contains 3.01000000000051.

One solution is to use programming techniques that will help to overcome such situations. One such technique is to use a range instead of using single values for conditional checking. However, this will scale down the computation ability of the algorithm. The best solution is to use a programming platform that gives more numerical accuracy.

Extrema identification

In the research in relation with extrema, the findings of this work be categorised into two groups. The first one is the extrema identification and the second one is the extrema filtering. Both are non-parametric approaches. As a concept, extrema identification is a good alternative method for conventional derivative based approach, which is also a non-parametric method. However, according to my understanding, the most important part is the extrema filtering because all the non-parametric methods introduced in this work fulfilled the needed requirement in the field of data analysing.

The three non-parametric methods introduced in this work for filtering non-dominating extrema, sharp and gradual (flat) extrema, and low and high extrema were named as MMS-Window based filter (MMS-WBF), MMS-SG filter, and MMS-LH filter, respectively. Results showed that applying only one method for extrema filtering may not provide the expected outcome. The reason for that is some extrema have a combination of features (multiple feature extremum) (Figure 3.8).

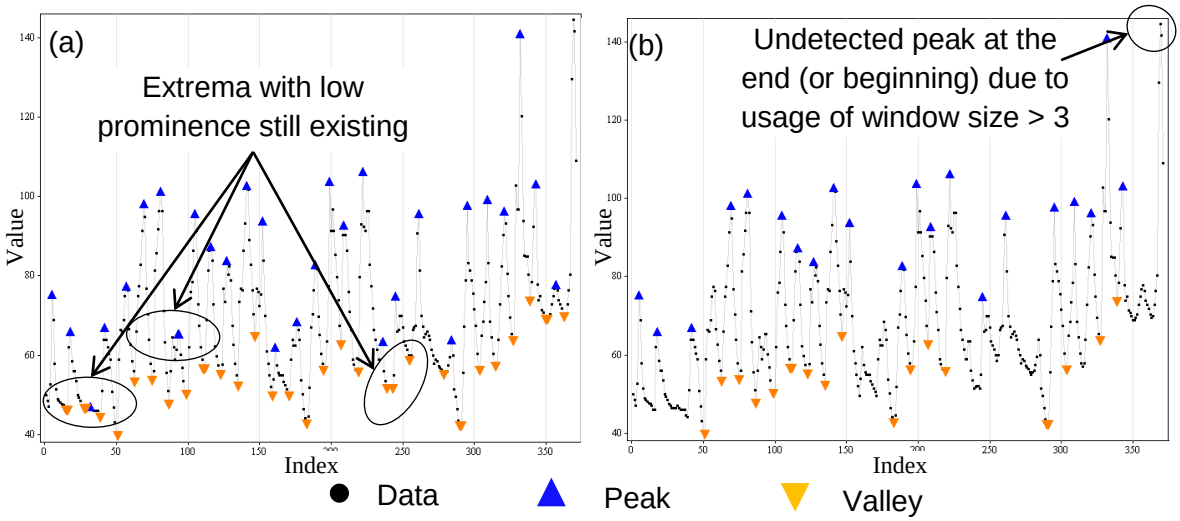


Figure 3.8: Data in plots (a) were first checked for extrema with a window of size nine with MMS-WBF. However, there is still a considerable amount of such extrema with low prominence visible in the plot (a), because in the considered window size they are still dominating extrema. Data in plots (b) were first checked for extrema with a window of size nine. Then R_{LH_max} and R_{LH_min} were considered and the extrema were checked for low and high extrema with threshold value $t_{LH} = 1$. Combination of two methods was capable of totally removing extrema with low prominence. However, due to the usage of window size greater than three, dominating maxima at the beginning as well as at the end of the window will not be detected.

For example, an extremum withvery small prominence can be either a sharp or a gradual extremum as well as the dominant extremum in the region. Therefore, to have a fully filtered data set, it is necessary to apply a combination of filters in a certain order according to the requirements (Figure 3.8). This will guaranty a fully filtered data, because if extremum cannot be identified by one method, it will be identified by another method, when the extremum has multiple features.

The major drawback inthe method is that the method is not capable of detecting dominating maxima at the end as well as at the beginning of the data set, when the advancing window side is greater than three (Figure 3.8). Exactly the dominating extrema between data points first and *(WS + 1)/2* (at the beginning) and between data points *n-(WS + 1)/2* and *n* (at the end), where *WS* is the size of advancing window and *n* is the number of data points in the data set. One possible solution is to add *(WS + 1)/2-2* numbers fordata points as filling at the beginning and at the end of the data set. The value of the filling at the beginning is the value of the first data

point and the value of the filling at the end is the value of the last data point. Usage of such an approach will help to minimise the impact due to large *WS.*

Graphical knowledge unit

Particularly, data processing requires a separate algorithm specified for fulfilling the required outcome. The output of the algorithm is a set of numerical values. These values are represented using a suitable visualisation technique (Figure 3.9). In contrast, the concept Graphical Knowledge Unit (GKU) we introduced is a method that is capable of visualising processed data as well as holding processed data. Not only that, GKU is a combination of many methods, where it is usually required to achieve different algorithms. In the same time, the algorithm of GKU is very easy to understand and has no complex calculations or definitions. Furthermore, GKU can be expressed as a method that is beyond the concept of non-parametric density cluster identification; can be considered as a package of solutions that is has following benefits.

1. Method of generating density clusters without a special complicated algorithm.
2. A good density cluster visualization technique.
3. Suitable for both online and offline applications.
4. One-pass (single-pass) method.
5. Method of visualizing missing data and out of range data density.
6. Automatic contour line generation method.
7. A good multivariable visualization method.
8. A good technique for representing big data.
9. Tool for outlier detecting in non-linear data.
10. Database for processed data / direct input for another algorithm.
11. Processed data and visual representation are in the same place.
12. Easy portability as a bitmap.
13. Bivariate as well as multivariate trained data set.
14. Can be maintained as a bitmap or as a matrix of integers.
15. Visually as well as programmatically readability.

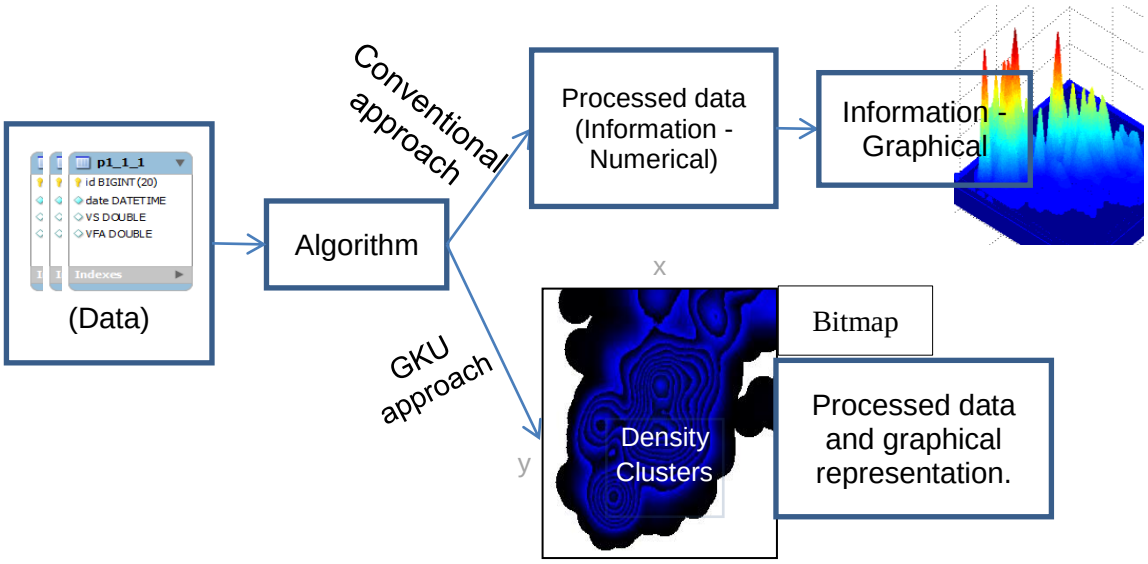


Figure 3.9: Typical data processing and visualising approach and GKU approach. In the conventional approach first process and visualise later using a suitable method. In GKU approach, the density of certain location is placed as a colour-coded value on the respective coordination of a bitmap.

One-pass algorithms are considered as fast algorithms because they read all inputs only once [78, 79]. GKU is also a one pass algorithm because it reads each data point only one time. Therefore, concept GKU is more suitable for online big data. This will allow online systems to represent existing status without further processing and without delay. Furthermore, the ability of processing and visualising multi-variables will be helpful to enhance online process monitoring and controlling environment. In our experiments, we used maximum one independent and three dependent variables with GKU. However, it is possible to use more dependent variables using pixel with a higher number of bits (e.g.: 64-bit,128-bit) and multi-layer concepts.

Findings of this research work showed that the overlapping of data in the multi-variable environment is no longer a problem with the concept of GKU. With the basic knowledge of colour formation theory, the contributed variables in a variable overlapped area can easily be identified, even with the naked eye. The plot in relation with layer 0 in Figure 3.10, most of the areas are overlapped areas of either two or three variables. Also, in layer 1 the overlapped areas can be identified easily. However, in this research I haven't tried to give a meaning for overlapped areas. Nevertheless, depending on the domain environment overlapping areas can be used to convey meaningful messages. In this research,

it was shown that the amount ofoverlapping could be controlled by setting uitable scaling factors for variables. Thus, setting overlapping environment must be done by considering the domain requirements by setting appropriate overlapping environment.

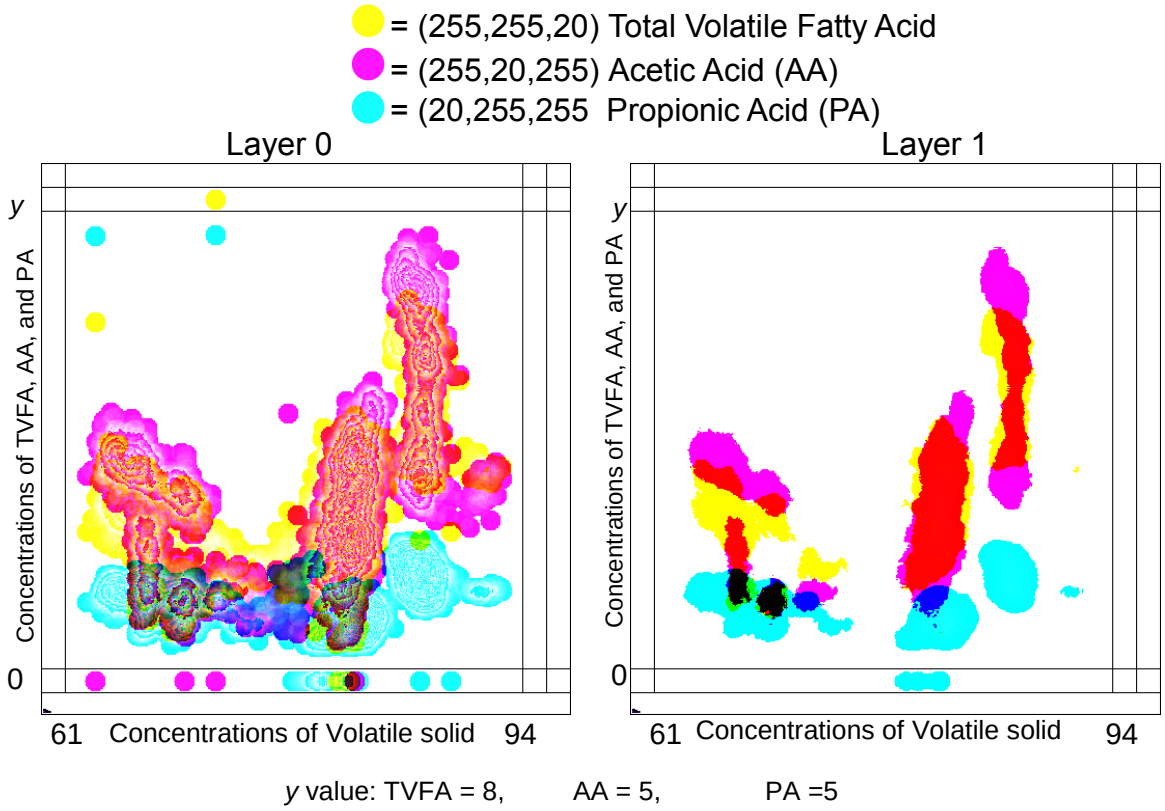


Figure 3.10: MLMV-GKU of two layers where effective bitmap width and height are 400 and 400 pixels for representing data of three dependent variables (Concentrations Total Volatile Fatty Acid (TVFA), Acetic Acid (AA), and Propionic Acid (PA)) against concentration of Volatile Solids, where each dependent variable consists of 2885 data points. A circle of radius 10 pixels is used as a marker. MLMV-GKU shows inter-variable clusters of both high and low-density areas of each variable by means of automatically generated counter lines and overlapped variables with different colours, which can be easily identified by naked eye.

The current research focused only on 2D GKUs that support one independent and one or many dependent variables. However, using 3D GKUs will allow the use of two independent variables and one or many dependent variables. Thus, 3D GKU will bethe next generation of GKU concept.

4. Outlook

I am very confident that our findings will open new doors in the fields of data / signal processing, process controlling, knowledge mining, and data visualisation individually in each field. Our major objective is to deploy those methods as a combined package for process controlling and process monitoring. Figure 4.1 shows diagram of such a system.

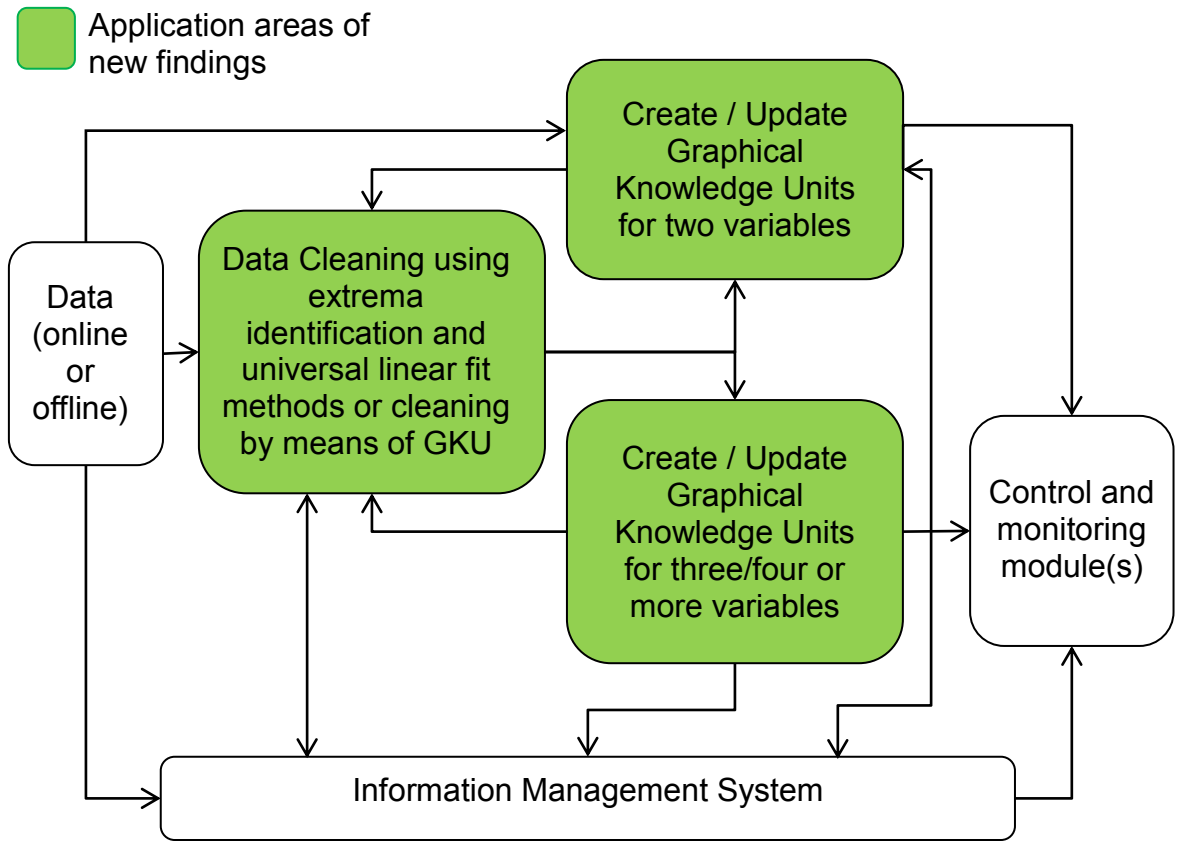


Figure 4.1 : Data / Information flow of proposed system

In real world data capturing, almost all the data in relation to a certain variable can be expressed as time series by plotting such data against time. Then, the regression can be linear or non-linear. If the regression is linear (or expected to be linear), universal linear fit identification method can be used to locate clean data. When the regression is non-linear, any nonlinear relation can be represented as a combination of straight lines. Different techniques can be found for signal segmentation depending on the considered domain, as for example in the field of electroencephalogram (EEG) [80, 81] and speech recognition [82]. If the frequency

of the data is sufficient, segmentation of a signal can be accomplished by using extrema detection method as a segmentation technique (Figure 4.2). This will split any nonlinear signal into a combination of nearly-linear segments. Finally using universal linear fit identification method, a linear fit of each individual segment can be found. This technique will allow cleaning linear or non-linear data by removing unnecessary data.

The data that cannot be represented as time series can be cleaned by means of GKU. After cleaning the data, this data can be sent to an information management system (IMS) or can be represented as GKUs in a suitable number of variables. These GKUs represent knowledge as density clusters. Basically, there are two types of GKUs, two variable and more than two variable GKUs. Using two variable GKU versions, it is possible to represent knowledge of dependent and independent variables. Finally, these totally cleaned data can be transferred into GKUs with one independent and many dependent variables. Well-developed GKUs are real trained sets and can directly be used for monitoring and controlling processes.

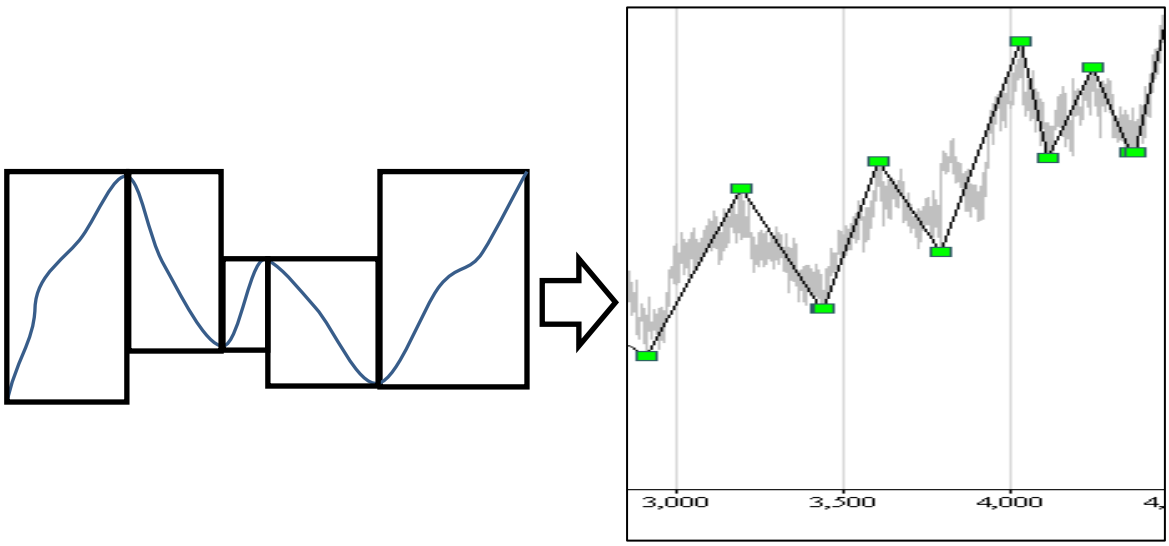


Figure 4.2: Split a nonlinear signal into several linear fits by means of suitable segmentation technique.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 821623, 12 pages
http://dx.doi.org/10.1155/2014/821623

Research Article

Outlier Detection Method in Linear Regression Based on Sum of Arithmetic Progression

K. K. L. B. Adikaram,[1,2,3] M. A. Hussein,[1] M. Effenberger,[2] and T. Becker[1]

[1] *Group Bio-Process Analysis Technology, Technische Universität München, Weihenstephaner Steig 20, 85354 Freising, Germany*
[2] *Institut für Landtechnik und Tierhaltung, Vöttinger Straße 36, 85354 Freising, Germany*
[3] *Computer Unit, Faculty of Agriculture, University of Ruhuna, Mapalana, 81100 Kamburupitiya, Sri Lanka*

Correspondence should be addressed to K. K. L. B. Adikaram; lasantha@daad-alumni.de

Received 25 March 2014; Revised 23 May 2014; Accepted 26 May 2014; Published 10 July 2014

Academic Editor: Zengyou He

We introduce a new nonparametric outlier detection method for linear series, which requires no missing or removed data imputation. For an arithmetic progression (a series without outliers) with n elements, the ratio (R) of the sum of the minimum and the maximum elements and the sum of all elements is always $2/n : (0, 1]$. $R \neq 2/n$ always implies the existence of outliers. Usually, $R < 2/n$ implies that the minimum is an outlier, and $R > 2/n$ implies that the maximum is an outlier. Based upon this, we derived a new method for identifying significant and nonsignificant outliers, separately. Two different techniques were used to manage missing data and removed outliers: (1) recalculate the terms after (or before) the removed or missing element while maintaining the initial angle in relation to a certain point or (2) transform data into a constant value, which is not affected by missing or removed elements. With a reference element, which was not an outlier, the method detected all outliers from data sets with 6 to 1000 elements containing 50% outliers which deviated by a factor of $\pm 1.0e - 2$ to $\pm 1.0e + 2$ from the correct value.

1. Introduction

Outlier detection and management of missing data are the two major steps in the data cleaning/cleansing process [1–3]. For achieving a training set, data mining, and statistical analyses, it is very important to have data sets that have no (or as few as possible) outliers and missing values. Except for model-based approaches, outlier detection and replacing of detected outliers or replacing missing values are two separate processes.

The existing outlier detection methods are based on statistical, distance, density, distribution, depth, clustering, angle, and model approaches [1, 4–7]. The nonparametric outlier detection methods are independent of the model. For the data without prior knowledge, nonparametric methods are known as a better solution than the statistical (parametric) methods [8–10]. The most common nonparametric methods are based on distance, density, depth, cluster, angle, and resolution techniques. Among various methods/techniques are least square method (LSM) [4] and the sigma filter [11] which have been used frequently to remove the outliers of linear regression. These methods require data in Gaussian or near Gaussian distribution, which cannot be always guaranteed. If the correct model can be identified, model-based approaches like the Kalman filter [12–14] are suitable for removing and replacing outliers. However, if it is not possible to identify the correct model, the model-based approach is not feasible [15].

In addition to the noise, missing data is another challenge in the data cleaning/cleansing process. Even if the original data set is without missing elements, removing outliers (without replacement) automatically creates a missing data environment. The most common two techniques to recover this situation are (1) filling the missing data with an estimated value (filling) or (2) using the data without missing values (reject missing values). Complete-case analysis (listwise deletion) and available-case analysis (pairwise deletion) are the most common missing data rejection methods [16–18]. The mentioned methods are under the assumption that they yield unbiased results. Among the different missing data filling methods hot deck, cold deck, mean, median, k-nearest

neighbours, model-based methods, maximum likelihood methods, and multiple imputation are the most common methods [18–22]. Filling methods derive the filling value from the same or other known existing data. If there are a considerable number of outliers, derived data may be biased due to the influence of outliers [23, 24]. Therefore, the best way is to remove all outliers and replace the outliers with a suitable method.

In this paper, we introduce a new nonparametric outlier detection method based on sum of arithmetic progression, which used an indicator $2/n$, where n is the number of terms in the series. The properties used in existing nonparametric methods such as distance, density, depth, cluster, angle, and resolution are domain dependent. In contrast, the value $2/n$, which we used in our new method, is independent of the domain conditions.

Contrary to the existing nonparametric methods mentioned earlier this work addressed identifying outliers in a dataset that is expected to have linear relation. The method is capable of identifying significant and nonsignificant outliers, separately. Moreover, until all the outliers were removed, the new method requires no missing or removed data imputation. This will eliminate the negative influence due to wrongly filled data points. This is an advantage over the methods, which require filling the removed data points. The outlier detection method we introduced showed its best performances when the significant outliers are in non-Gaussian distribution. This is an advantage over existing methods such as LMS and sigma filter. The method uses a single data point as a reference data point. The reference point is assumed to be nonoutlier. Therefore, accuracy of the outcome is depending on the reference point, especially when locating nonsignificant outliers. If the selected reference point is not an outlier, the method was capable of locating outliers from a data set containing very high rate of outliers, such as 50% outliers.

In this work, data from biogas plants were used for evaluating the new method. Since the biogas process is very sensitive, these data contain a considerable amount of noise even during apparently stable conditions. This provides suitable data set for evaluating our method. We were able to get the best outlier-free macroscale data set which agrees with linear (increasing, decreasing, or constant) regression from selected segments of a data set.

2. Methodology

2.1. Arithmetic Progression. An arithmetic progression (AP) or arithmetic sequence is a sequence of numbers (ascending, descending, or constant) such that the difference between the successive terms is constant [25]. The nth term of a finite AP with n elements is given by

$$a_n = d\,(n-1) + a_1, \tag{1}$$

where d is the common difference of successive members and a_1 is the first element of the series. The sum of the elements of a finite AP with n elements is given by

$$S_n = \left(\frac{n}{2}\right) * (a_1 + a_n), \tag{2}$$

Table 1: Sample calculations for illustrating the relation between $2/n$ and $(a_1 + a_n)/S_n$.

a_n	Data set 1	Data set 2	Data set 3	Data set 4	Data set 5
a_1	100	100	100	99.99	1
a_2	101	101	101	101	101
a_3	102	102	102	102	102
a_4	103	103	103	103	103
a_5	104	104.01	204	104	104
$(a_1 + a_5)/S_5$	0.4	0.40001	0.498	0.399	0.255
$2/n$	0.4	0.4	0.4	0.4	0.4
Outlier?	—	Yes-a_5	Yes-a_5	Yes-a_1	Yes-a_1

where a_1 is the first element and a_n is the last element of the series.

Equation (1) is a $f(n)$ and fulfils the requirements of a line. In other words, finite AP is a straight line. In addition, a straight line is a series without outliers. If there are outliers, the series is not a finite AP. Therefore, any arithmetic series that fulfils the requirements of an AP can be considered a series without outliers. Equation (2) can be represented as

$$\frac{2}{n} = \frac{(a_1 + a_n)}{S_n}; \quad \infty > n \geq 2,\ 0 < \frac{2}{n} \leq 1. \tag{3}$$

For any AP, the right-hand side (RHS) of (3) is always $2/n$, which is independent of the terms of the series. In other words, if there are no outliers, the value $(a_1+a_n)/S_n$ will always be equal to $2/n$. If the RHS of (3) is not $2/n$, it always implies that the series contains outliers. Therefore, the value $2/n$ can be used as a global indicator to identify any AP with outliers.

Since we use the relation of AP, we define that elements lying on or between two lines (linear border) are nonoutliers, and others are outliers. When the distance between two lines is zero, they represent a single line. In relation to the method presented in this paper, the term nonoutlier implies an element that lies within a certain linear border, and the term outlier implies an element that does not lie within the linear border.

Primary investigations showed that the method is capable of not only indicating the existence of outliers but also locating the outlier. $(a_1 + a_n)/S_n < 2/n$ indicates that the maximum element is the outlier. $(a_1 + a_n)/S_n < 2/n$ indicates that the minimum element is the outlier. However, $(a_1 + a_n)/S_n = 2/n$ does not imply that the series is free of outliers. Furthermore, primary investigations showed that the method is capable of locating both large and small outliers. Table 1 shows sample calculations for illustrating the relation between $2/n$ and $(a_1 + a_n)/S_n$.

As a principle, the relation of (3) is capable of identifying and locating the outliers. However, we found seven drawbacks, which made relation (3) unusable for identifying outliers in actual data. In Sections 2.1 to 2.7, we address the challenges for making the relation usable.

2.2. Challenge 1: Notation of the Equation. The symbols used in (3), especially a_1, a_n, create a logical barrier. For example, if there are outliers, the minimum and the maximum can be other elements rather than a_1, a_n. Therefore, it is necessary

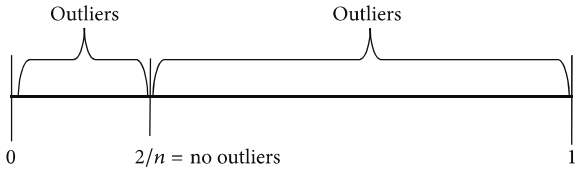


Figure 1: Distribution of criteria range (0, 1].

to use meaningful symbols that reflect the purpose of the method. The first and the last elements are either the minimum or the maximum. Therefore, it is possible to replace a_1 and a_n by the minimum ($a_{\min}$) and the maximum ($a_{\max}$) of the series. Then (3) can be represented as

$$\frac{2}{n} = \frac{(a_{\min} + a_{\max})}{S_n}. \tag{4}$$

Since the RHS of (4) consists of minimum, maximum, and sum of the series, RHS was named MMS with the meaning of minimum, maximum, and sum:

$$\text{MMS} = \frac{(a_{\min} + a_{\max})}{S_n}. \tag{5}$$

2.3. Challenge 2: Set a Range for the Outlier Detection Criterion. According to (3), outlier detection criterion is $2/n$ and can be used to check the elements that exactly agree with a line (Figure 1). To identify elements in a certain range, it is necessary to have a criteria range rather than a single value $2/n$.

The left-hand side of (4) is the ratio $2 : n$ and named as R_w by adding a weight "w" to "R." Then,

$$R_w = \frac{2}{n} + w; \quad 0 \le w \le 1 - \frac{2}{n}. \tag{6}$$

The status $w = 0\,(R_0)$ represents a single line, and $w > 0$ represents a line with a certain width (linear border). The outlier criteria range is a range with both floor (0) and ceiling (1), and standardization is not required. This is an additional advantage over the most common average, variance, and slandered deviation based approaches, which require a separate standardization process.

2.4. Challenge 3: Influence of Negative Values. Due to negative values, the numerator or both the numerator and the denominator of RHS of (5) can be 0 (e.g., −4, −1, 0, 1, 4), even without outliers. When there are outliers, RHS of (5) can be negative, which cannot be accepted as valid values for $2/n, 0 < 2/n \le 1$, must always hold.

Subtracting the first element ($a_{i\_\text{new}} = a_i - a_{\min}$) from each element of any AP creates a new transformed AP where $a_{\min} = 0$ and guarantees a series without negative values. From (5) and $a_{i\_\text{new}} = a_i - a_{\min}$, (7) is derived, which is more robust. Another advantage of (7) is that it performs the transformation, automatically:

$$\text{MMS} = \frac{((a_{\min} - a_{\min}) + (a_{\max} - a_{\min}))}{\sum_{i=1}^{n} (a_i - a_{\min})},$$
$$\text{MMS} = \frac{(a_{\max} - a_{\min})}{(S_n - a_{\min} * n)}. \tag{7}$$

2.5. Challenge 4: Uneven Distribution of Criteria Range. The ranges $(0, 2/n)$ and $(2/n, 1]$ are to identify outliers, which are minimums and maximums, respectively (Figure 1). When $n \to \infty$ and $R_0 \to 0$, then $R_w : (0, 1]$ is not equally distributed, which provides a large range for maximum outliers and a small range for minimum outliers. This is a problem when locating minimum outliers.

To solve this, we used the idea of complement. For any series, this will convert the maximum value into the minimum, the minimum value into the maximum, and intermediate values into their complements. Most importantly, now the minimum value represents the maximum value of the original series and vice versa, while still representing the original series. The complement of an element in a series can be defined as $a_{i\_c} = (a_{\max} + a_{\min}) - a_i$. From (5) and $a_{i\_c} = (a_{\max} + a_{\min}) - a_i$ this gives

$$\text{MMS} = \frac{((a_{\max} + a_{\min} - a_{\max}) + (a_{\max} + a_{\min} - a_{\min}))}{\sum_{i=1}^{n} (a_{\max} + a_{\min} - a_i)},$$
$$\text{MMS} = \frac{((a_{\min}) + (a_{\max}))}{\sum_{i=1}^{n} (a_{\max} + a_{\min} - a_i)}. \tag{8}$$

Apply $a_{i\_\text{new}} = a_i - a_{\min}$ (to remove effect from negative values):

$$\text{MMS} = \frac{((a_{\min} - a_{\min}) + (a_{\max} - a_{\min}))}{\sum_{i=1}^{n} ((a_{\max} - a_{\min}) + (a_{\min} - a_{\min}) - (a_i - a_{\min}))},$$
$$\text{MMS} = \frac{(a_{\max} - a_{\min})}{\sum_{i=1}^{n} (a_{\max} - a_i)},$$
$$\text{MMS} = \frac{(a_{\max} - a_{\min})}{(a_{\max} * n - S_n)}. \tag{9}$$

Consequently, the range $R_0 > 2/n$ represents the range for minimum outliers related to the original series and vice versa (Figure 2), and it is possible to ignore the range $(0, 2/n)$. In addition, (9) automatically performs the transformation.

Now there are two equations for MMS, (7) and (9), to check whether the maximum or the minimum of the series is an outlier. We named the two versions of MMS as $\text{MMS}_{\max}$ (10) and $\text{MMS}_{\min}$ (11)

$$\text{MMS}_{\max} = \frac{(a_{\max} - a_{\min})}{(S_n - a_{\min} * n)}, \tag{10}$$

$$\text{MMS}_{\min} = \frac{(a_{\max} - a_{\min})}{(a_{\max} * n - S_n)}. \tag{11}$$

Table 2: Sample calculations for illustrating the relation between $2/n$ and $\text{MMS}_{\max}$ and $\text{MMS}_{\min}$.

a_n	Data set 1	Data set 2	Data set 3	Data set 4	Data set 5
a_1	100	100	100	99.99	1
a_2	101	101	101	101	101
a_3	102	102	102	102	102
a_4	103	103	103	103	103
a_5	104	104.01	204	104	104
MMS (Max)	0.4	**0.401**	**0.945**	0.399	0.254
MMS (Min)	0.4	0.399	0.254	**0.401**	**0.945**
$2/n$	0.4	0.4	0.4	0.4	0.4
Outlier?	—	**Yes**-Max	**Yes**-Max	**Yes**-Min	**Yes**-Min

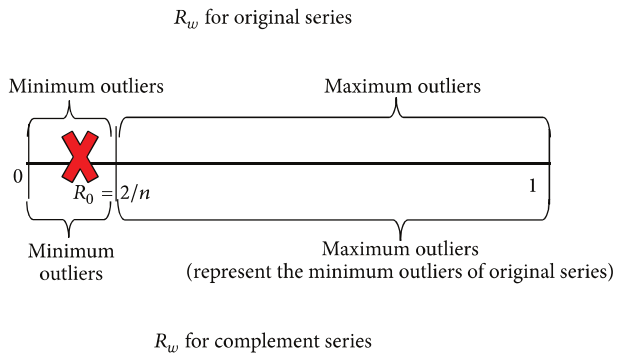


Figure 2: Range of R_w for original series and complement of original series.

The following equation shows the overview of the MMS process:

$$\text{MMS}_{\max} = \frac{a_{\max} - a_{\min}}{S_n - a_{\min} * n} \begin{cases} > \left(\left(\frac{2}{n}\right) + w1\right); \\ \text{maximum is the outlier} \\ \leq \left(\left(\frac{2}{n}\right) + w1\right) \end{cases} \text{or} \quad \text{MMS}_{\min} = \frac{a_{\max} - a_{\min}}{a_{\max} * n - S_n} \begin{cases} \leq \left(\left(\frac{2}{n}\right) + w1\right) \\ > \left(\left(\frac{2}{n}\right) + w1\right); \\ \text{minimum is the outlier} \end{cases} \tag{12}$$

and Table 2 shows sample calculations using (10) and (11) for the same data sets in Table 1.

2.6. Challenge 5: How to Deal with Removed Outliers/Missing Values. In a series, there can be initial missing values. In addition, if there is no replacement after removing an outlier it also creates a missing value environment. If there is no filling, it would transform the elements after the element is removed into another value and destroy the original relationship of elements (Figure 3). These transformed values become outliers in relation to the original data. Therefore, for using the relation of AP, it is compulsory to maintain the original relation of the data even after removing an outlier. Thus, any rejection technique is not feasible. To maintain the original relation, one possible way is replacing the missing value. However, the data we are considering contain a considerable amount of outliers. Therefore, we cannot guarantee that an element derived from existing elements is not an outlier.

To overcome this problem, we considered two different options: (1) recalculate only the data points after (or before) the removed or missing element, thereby maintaining the initial angle in relation to a certain point or (2) transform the elements into a new series where the missing value has no effect.

2.6.1. Recalculate the Data Points after (or before) Removed and Missing Elements. If there is a missing element, the next elements will be shifted horizontally and transformed into wrong values in relation to the current index of the elements (Figure 3). However, angular shifting will not introduce such an error (Figure 3).

In Figure 4, the plot consists of elements a_0 to a_{r+1} ($r \in \mathbb{R}^+$), and element a_r at r needed to be removed. After removing element r, element $r+1$ becomes element r, element $r+2$ becomes element $r+1$, and so on. However, shifting while maintaining the same angle with respect to a certain reference element (e.g., the first element), the same form of the series can be maintained. Equation (13) shows the new value after angular shifting. We used this technique with MMS algorithm to recalculate the series after (or before) missing values or removed elements:

$$B_r T_r = \left(\frac{B_{r+1}C_{r+1}}{AB_{r+1}}\right) * AB_r = \left(\frac{(a_{r+1} - a_0)}{(r+1)}\right) * r, \qquad (a_{r+1})_{\text{new}} = a_o + B_r T_r. \tag{13}$$

2.6.2. Transformation of Data to a Constant Value. A series with a constant value ($y = c$ form, where c is a constant) is a series that has no effect of missing values. Because of that, if it is possible to transform any linear series to $y = c$ form, the transformed series is free of any effect of missing values. After that, the transformed series can be used for outlier detection.

If y^T is a linear series, where $y_k^T = y_k - y_1$, $x_k^T = x_k - x_1$, x_k is the initial index of elements and y_k is the kth element of the series, $k = 1, 2, \ldots, n$. The gradient of the line (m) is given by $\sum_{i=1}^{n} y_k / \sum_{i=1}^{n} x_k$. If one element (e.g., the first element) is $(0, 0)$, this relation is always true even with missing values. The element $(0, 0)$ can be considered as the reference element. The y^T is a series with first element $(0, 0)$ and m that can be calculated even with missing values. Also, it is possible to derive a new series as y' where $y_k = x_k * m$. If there are no outliers, both y^T and y' coincide and $y^T - y' = 0$. If $y^{TT} = y^T - y'$, y^{TT} is in the form of $y = c$ without any influence from missing values. Therefore, this is another method to overcome missing values without replacing them (Figure 5).

2.7. Challenge 6: Locate Outliers That Are Neither the Maximum Nor the Minimum of the Series. When the outlier is

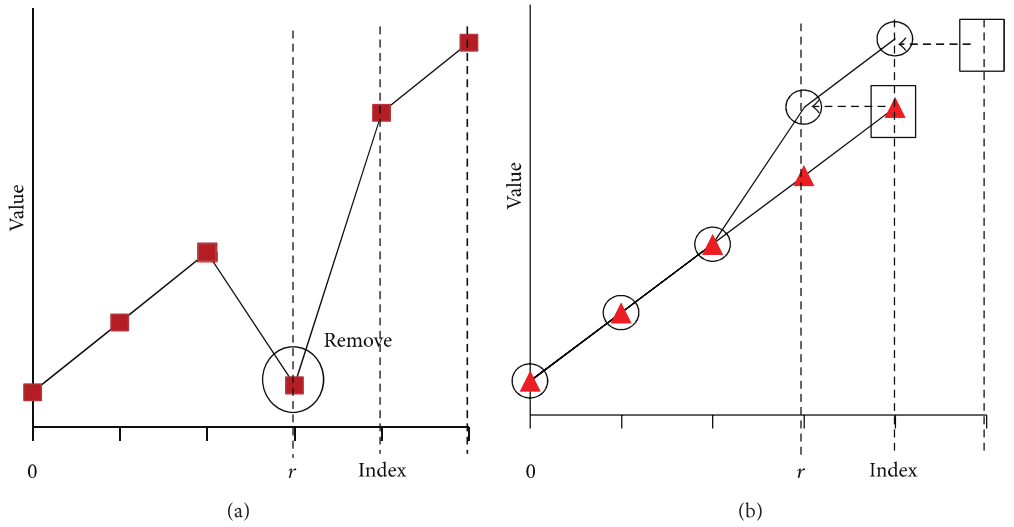


Figure 3: (a) Data set with an outlier at index r. (b) Value autotransformation effect after removing the outlier at index r without replacement, where circle corresponds to elements after removing the outlier, red triangle corresponds to expected (correct) elements, and square corresponds to initial values of the shifted elements after removing the outlier.

Table 3: "Bad Detection" identified wrong (minimum) element as the outlier.

a_n	Data set 6		
a_1	100		
a_2	101	$\text{MMS}_{\max}$	0.377
a_3	102	$\text{MMS}_{\min}$	0.425
a_4	103.6	$2/n$	0.4
a_5	104	Outlier?	Yes-Min

neither the maximum nor the minimum, MMS is unable to locate the outlier (Table 3). We named this phenomenon as "Bad Detection." When R_w reaches "Bad Detection Level," MMS cannot be applied. To overcome this situation, we introduced an improved version of MMS as enhanced MMS (EMMS) based on the missing data imputation technique in Section 2.6.2.

EMMS is expressed as

$$\text{EMMS}_{\max} = \frac{\left(a_{\max}^{TT} - a_{\min}^{TT}\right)}{\left(S_n^{TT} - a_{\min}^{TT} * n\right)}; \quad a_{\max}^{TT} \langle\rangle 0, \tag{14}$$

$$\text{EMMS}_{\min} = \frac{\left(a_{\max}^{TT} - a_{\min}^{TT}\right)}{\left(a_{\max}^{TT} * n - S_n^{TT}\right)}; \quad a_{\max}^{TT} \langle\rangle 0, \tag{15}$$

where $a_k^{TT} = |a_k^T - x_k(Ga^T/Gx)|$, $a_k^T = a_k - a_0$, x_k is the index of data, a_k is the kth term of the series, $k = 0, 1, \ldots, n-1$, n is the number of elements in current window, $Ga^T = \sum_{k=0}^{n-1} a_k^T$, $Gx = \sum_{k=0}^{n-1} X_k$, and $S_n^{TT} = \sum_{k=0}^{n-1} a_k^{TT} \langle\rangle 0$.

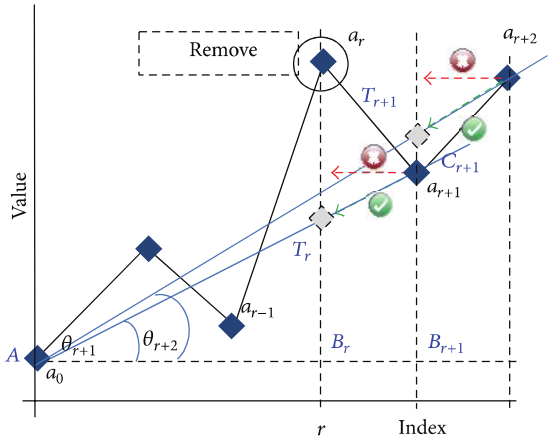


Figure 4: Solution for value autotransformation phenomenon. Use angular shifting instead of horizontal shift, where (×) corresponds to horizontal shift and (√) corresponds to angular shift.

Always the term $a^{TT} > 0$. Thus, the term $a_{\min}^{TT} = 0$. Then (14) and (15) are simplified as

$$\text{EMMS}_{\max} = \frac{a_{\max}^{TT}}{S_n^{TT}}; \quad a_{\max}^{TT} \langle\rangle 0, \tag{16}$$

$$\text{EMMS}_{\min} = \frac{a_{\max}^{TT}}{\left(a_{\max}^{TT} * n - S_n^{TT}\right)}; \quad a_{\max}^{TT} \langle\rangle 0. \tag{17}$$

If there are outliers, $\text{EMMS}_{\min} > 2/n$ or $\text{EMMS}_{\max} > 2/n$ and the greater value represents the outlier. Table 4 shows

Table 4: EMMS for identifying an outlier.

$X(n-1)$	$y(a_n)$	$y^T(a_n - a_1)$	$y^{TT}(\lvert y^T - x_n(G_y^T/G_x)\rvert)$
0	100	0.000	0
1	101	1.000	0.06
2	102	2.000	0.12
3	**103.6**	3.600	**0.42**
4	104	4.000	0.24
$G_x = 2$		$G_y^T = 2.120$	
		EMMS (Max)	**0.500**
		EMMS (Min)	0.333
		$2/n$	0.4
		Outlier?	Yes-Max

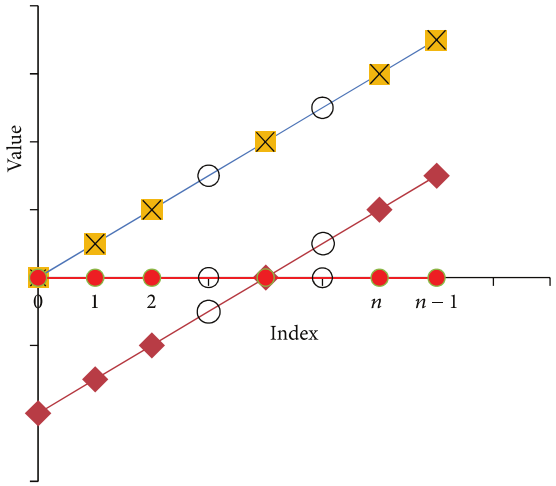


Figure 5: Transformation of data to a constant value to overcome the missing data problem, where red diamond corresponds to $y = f(x)$ form, yellow square corresponds to $y^T = f(x) - f(x_0)$ form, cross corresponds to $y' = m * x_i$ form ($m = \sum_{i=0}^{n} y_i^T / \sum_{i=0}^{n} x_i^T$), red circle corresponds to $y^{TT} = y^T - y'$, and circle corresponds to missing values.

an example calculation of EMMS and the following equation shows an overview of EMMS process:

$$\begin{aligned}\text{EMMS}_{\max} &= \frac{a_{\max}^{TT}}{S_n^{TT}} \begin{cases} > \left(\left(\frac{2}{n}\right) + w2\right); \\ \text{maximum is the outlier} \\ \le \left(\left(\frac{2}{n}\right) + w2\right) \end{cases} \\ &\quad \text{or} \\ \text{EMMS}_{\min} &= \frac{a_{\max}^{TT}}{\left(a_{\max}^{TT} * n - S_n^{TT}\right)} \begin{cases} \le \left(\left(\frac{2}{n}\right) + w2\right) \\ > \left(\left(\frac{2}{n}\right) + w2\right); \\ \text{minimum is the outlier.} \end{cases} \end{aligned} \quad (18)$$

However, EMMS uses derived information from existing data. If there are biased values, it may lead to biased information. Because of that, direct application of EMMS is not a good practice. Hence, significant outliers should be removed first using MMS, before applying EMMS.

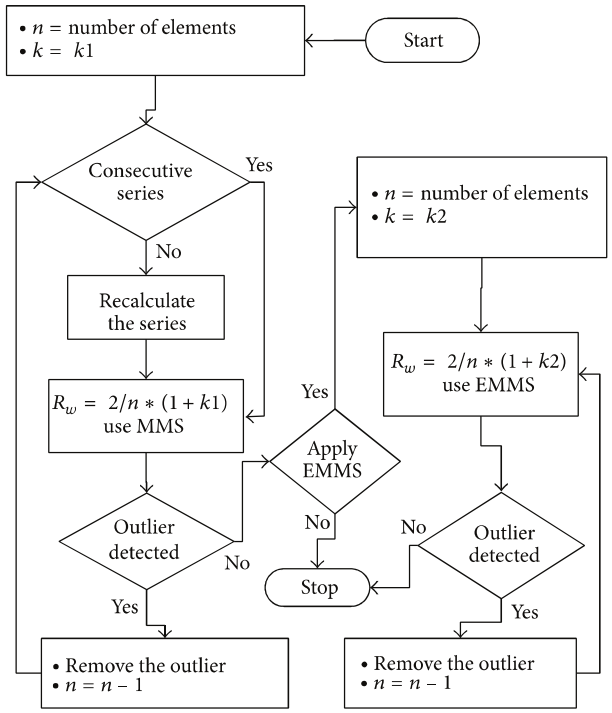


Figure 6: Implementation of MMS and EMMS. Initially algorithm checks for the significant outliers using MMS. After removing all significant outliers, then remove the nonsignificant outliers using EMMS. There is no removed data imputation in relation to both MMS and EMMS.

2.8. Challenge 7: Determining of Outlier Detection Criteria (R_w). The value R_w is the factor that determines the outliers, when $w = 0$ ($R_0 = 2/n$) represents exactly a line and $w > 0$ represents a linear border with certain width. In this section, we propose several possible methods that can be used to determine the outlier detection criteria.

2.8.1. Express the Value "w" as $f(1/n)$. If the value w is $f(1/n)$ then $w = 2 * k/n$; $k \le (n/2) - 1$; and $k \in \mathbb{R}^+$. Then $R_w = 2/n + 2 * k/n$:

$$\begin{aligned} R_w &= \frac{2}{n} * (1 + k), \\ \frac{R_w}{R_0} &= 1 + k \,(= \text{constant}). \end{aligned} \quad (19)$$

When the MMS or the EMMS is greater than R_w of (19), this implies the existence of outliers. Because R_w/R_0 is constant and gives standards to R_w, determination of k still depends on the knowledge of the domain. Figure 6 shows an algorithm based on this technique.

2.8.2. When the First and the Last Items Are Nonoutliers. In the total process, the "Bad Detection level" is the most important criteria. If R_w of MMS is less than the "Bad Detection Level" it is possible to identify nonoutliers as outliers as mentioned in Section 2.7. If there is preknowledge about outliers, it is possible to use a safe value for MMS. Otherwise, there is no 100% guarantee on "Bad Detection Level."

Table 5: Different environments used to validate the new method.

Type of the original dataset	Number of elements	Type of outliers	Reference (first) element is an outlier?	Initial missing values?
Increment, constant, and decrement.	10 to 1000	Non-Gaussian, Gaussian	Yes, no	Yes, no

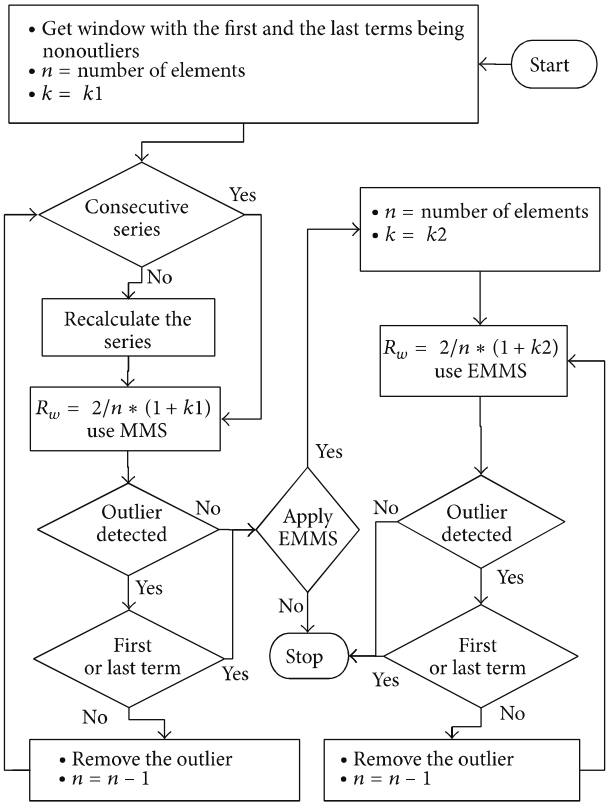


Figure 7: Outlier detection method including the "Bad Detection Level" detection technique. The first and the last data points of the window must be nonoutliers. If the first or the last element was identified as an outlier, it will become a contradictory situation. Thus, this point can be considered as the terminating point of MMS and EMMS.

However, when the first and the last elements are not outliers, the "Bad Detection Level" can be detected automatically. If the first or the last element was identified as an outlier, it will become a contradictory situation. Thus, this point can be considered as the terminating point of MMS and EMMS. The decision diagram elaborated in Figure 7 expresses the new outlier detection method including the "Bad Detection Level" detection technique.

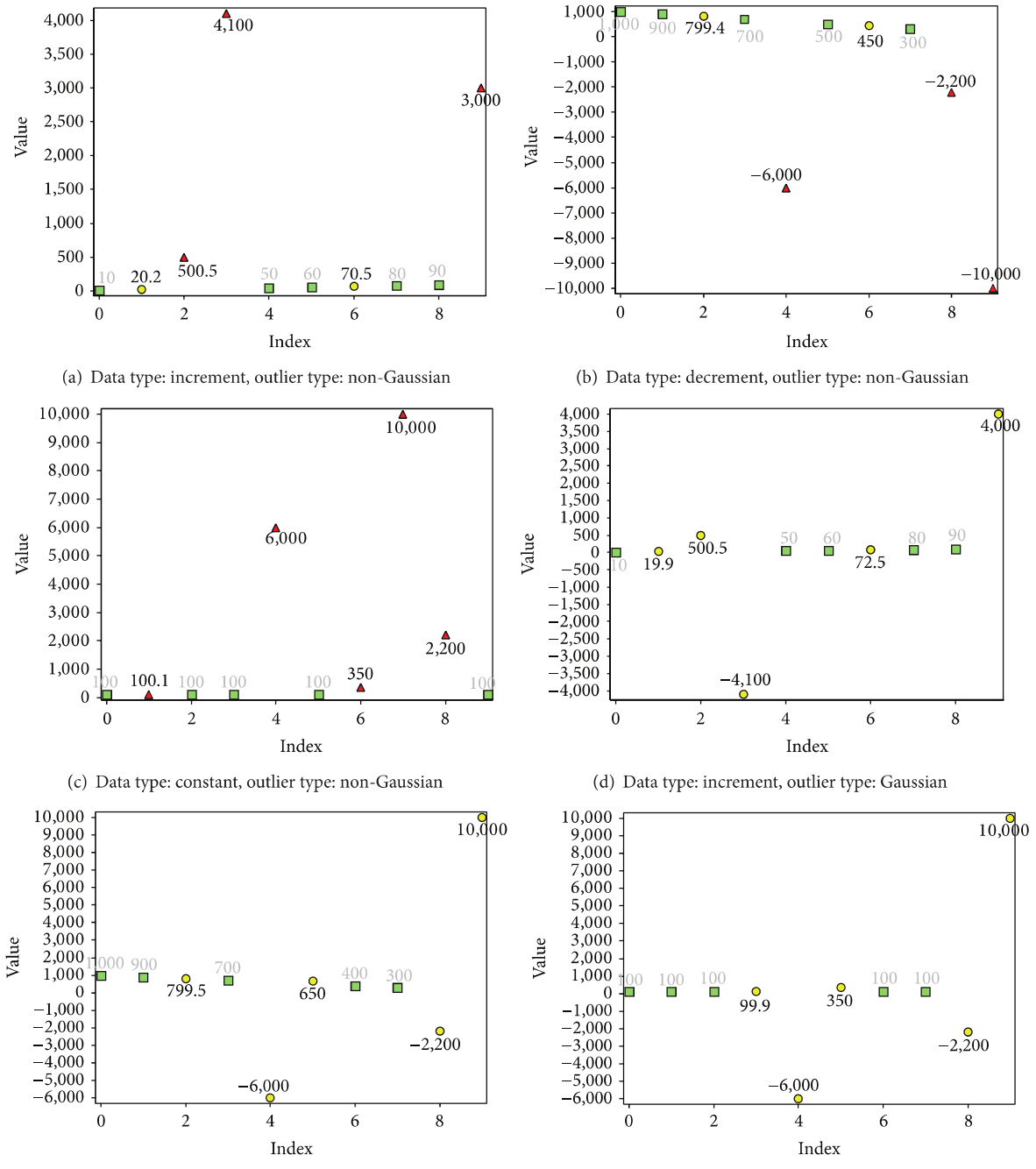
2.9. Validate the Method. We implemented the MMS (with recalculation after an outlier is removed) and EMMS with C++ and conducted the validation process. For the recalculation process, the existing first element of the window was the reference element and always used the original value of the element (not the current updated value of the element). To validate the method, we used artificial data sets of different sizes (10 to 1000) of a line representing increasing, decreasing, and constant line. Then 50% of items of those data sets were replaced with very small and very large outliers ($\pm 1.0e-2$ to $\pm 1.0e+2$ times of correct value). We checked the data sets for all the environment combinations shown in Table 5. The outlier detection criteria were determined based on (19). For all data sets, the same k value was used (for MMS, $k = 0.5$, and for EMMS, $k = 0.01$). Then the percentage of correctly and falsely detected nonoutliers in relation to the number of actual nonoutliers and the percentage of correctly and falsely detected outliers from the total number of outliers (small and large outliers) were determined.

2.10. Evaluation Using Real Data. To check the best linear fitting identification capability, the algorithm was tested using several real data sets which were automatically recorded with a frequency of twelve data points per day (i.e., every other hour) from a biogas plant, over a period of seven months. Among the different parameters, we selected the H_2 content measured in ppm, which we expected to maintain linear behaviour during stable operation. We selected seven segments of different size for evaluating the algorithm. In some data sets, there were initial missing elements. We set the R_w for MMS and EMMS by analysing the first and the third data sets. For the recalculation process, the existing first element of the window was the reference element, and we always used the original value of the elements (not the current updated value of the element). Then the percentage of correctly falsely detected nonoutliers in relation to the total number of nonoutliers and the percentage of correctly and falsely detected outliers from the total number of outliers (small and large outliers) were determined.

We decided to use the LSM, Sigma filter, and Grubb's test [26–29] also known as maximum normed residual test or "extreme studentized deviate" (ESD) test to compare our results. We selected Grubb's test since it has nearly the same formulation as our method. We checked all the biogas data using abovementioned methods. We used each of the data segments as a single window. First, we checked the ability of each method to identify the general trend of the series. Then, we checked the amount of correctly and falsely detected outliers and nonoutliers for each method in relation to the general trend.

3. Results and Discussion

Results related to validation show that when the reference element (the first element) was not an outlier, the algorithm was capable of identifying all outliers with 0% error despite of the type of outliers (Gaussian or non-Gaussian) (Figure 8). If the outliers were Gaussian, there were no significant outliers



(a) Data type: increment, outlier type: non-Gaussian

(b) Data type: decrement, outlier type: non-Gaussian

(c) Data type: constant, outlier type: non-Gaussian

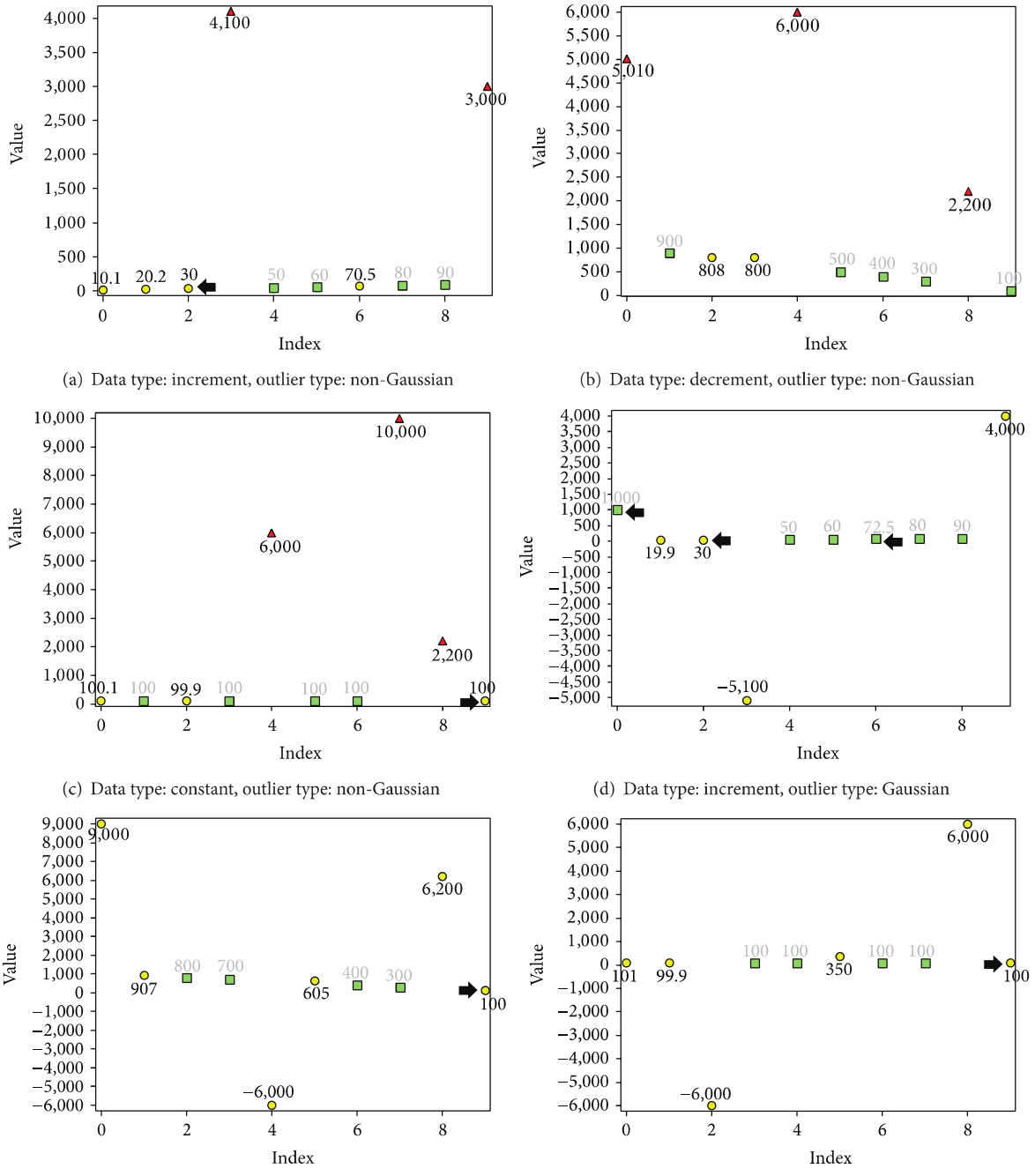
(d) Data type: increment, outlier type: Gaussian

(e) Data type: decrement, outlier type: Gaussian

(f) Data type: constant, outlier type: Gaussian

Figure 8: Outlier detection from data sets with ten elements. The first element is the reference element, which is not an outlier, where red triangle corresponds to outliers detected by MMS, yellow circle corresponds to outliers detected by EMMS, and green square corresponds to nonoutliers. Value of k for MMS and EMMS is 0.5 and 0.01, respectively. When the reference (first) element is not an outlier, the new method is capable of locating all outliers. When the outliers are Gaussian, MMS automatically becomes inactive (now no significant outliers) ((d), (e), (f)).

and MMS automatically became inactive (Figures 8(d), 8(e), and 8(f)). When the first few elements were outliers and outliers were non-Gaussian, MMS detected the significant outliers correctly (Figures 9(a), 9(b), and 9(c)). However, EMMS was unable to locate the nonsignificant outliers, when the first element for EMMS was an outlier (Figures 9(a) and 9(c)). If the reference element for EMMS was not an outlier, it was still possible to achieve correct results (Figure 9(b)). Though it was impossible to locate all nonoutliers, the detected nonoutliers were 100% correct detections. These values can be used to estimate the other values using methods like LSM since now all the existing data are cleaned. In



(a) Data type: increment, outlier type: non-Gaussian

(b) Data type: decrement, outlier type: non-Gaussian

(c) Data type: constant, outlier type: non-Gaussian

(d) Data type: increment, outlier type: Gaussian

(e) Data type: decrement, outlier type: Gaussian

(f) Data type: constant, outlier type: Gaussian

FIGURE 9: Outlier detection from data sets with ten elements. The first element is the reference element, which is an outlier, where red triangle corresponds to outliers detected by MMS, yellow circle corresponds to outliers detected by EMMS, green square corresponds to nonoutliers, and black arrow corresponds to wrong detections. Value of k for MMS and EMMS is 0.5 and 0.01, respectively. When the reference (first) element is an outlier and outliers are non-Gaussian, the new method identifies only the significant outliers ((a), (b), (c)). When the outliers are Gaussian, MMS automatically becomes inactive (now no significant outliers) ((d), (e), (f)).

general, it is fair to state that (1) when the reference element is not an outlier, the method is capable of identifying all outliers and (2) when the first few elements of the series are outliers and the outliers are non-Gaussian, the method is capable of identifying only the significant outliers and part of correct elements.

When the first few elements (reference elements for both MMS and EMMS) were outliers and the outlier distribution was Gaussian, outlier detection was poor (Figures 9(d), 9(e), and 9(f)). Due to the Gaussian distribution of outliers, MMS was inactive and it was not possible to identify the large outliers. Most importantly, the results highlighted

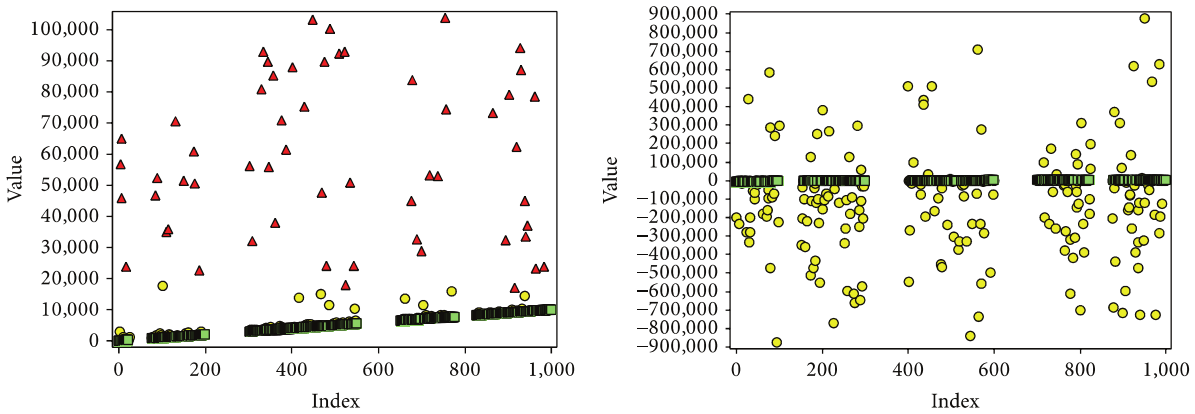


FIGURE 10: Two artificial data samples with 1000 elements each, including 50, 100, 100, and 50 (total 300) missing value regions. The first element is the reference element, which is not an outlier, where (a) corresponds to a data set with outliers in non-Gaussian, (b) corresponds to a data set with outliers in nearly Gaussian, red triangle corresponds to outliers detected by MMS, yellow circle corresponds to outliers detected by EMMS, and green square corresponds to nonoutliers. The value of k for MMS and EMMS is 0.5 and 0.01, respectively. The new method was able to identify all the elements related to the line with 0% error.

the importance of the reference element. If the reference element for MMS and EMMS was not an outlier, it guaranteed good results despite of other factors.

In the methodology, we derived the method based on the first element. However, it is also possible to use any other element as reference point and modify the method. We considered the simplest situation, where the first element is not an outlier. Therefore, if it is possible to segment the data excluding extreme outliers at the beginning, it provides accurate outlier detection. Another possibility is to replace the first element with an already known element. This leads to another possibility for applying the method: if we know only a single correct element, the use of that element as reference element and of the modified method according to the reference element can yield very accurate results.

Some model-based approaches demand a trained data set for correct output. In contrast, this method requires only one correct element to produce a correct output. In addition, it is possible to use multiple reference points and consider the best fitting. For example, (a) consider each point in first x% (e.g., 10%) of data points as reference point and (b) consider all data points as the reference point. Furthermore, it is important to distinguish the purpose of MMS and EMMS. MMS removes only the significant outliers, while EMMS removes nonsignificant outliers. Depending on the requirement, MMS or/and EMMS can be used to remove outliers.

The results show that the new method is a good solution for managing missing values. Figure 10 shows two data sets with 1000 elements each. Each data set consists of 50, 100, 100, and 50 (total 300) missing value regions. When the first element was not an outlier, the new method was able to identify all the elements related to the line with 0% error.

In real world, it is not possible to find nonoutliers that exactly agree with linear regression. Therefore, 100% accuracy is inapplicable. However, it is very important to have a significant outlier-free data set. The new method guaranteed a significant outlier-free data set when the outliers were non-Gaussian. Furthermore, in real world situations, data/outliers are not always in Gaussian distribution. Due to that, we hope the new method can be applied to the majority of outlier detection applications. Our new method is an effective solution for most common LSM and sigma filter need Gaussian outliers. Some methods like sigma filter cannot be applied directly to a certain data segment, and further segmentation (windowing) is required for better results. In contrast, the new method is capable of locating nonoutliers automatically in increment, decrement, or constant form, regardless of the size of the window.

Results related to biogas data proved the abovementioned idea and showed that the algorithm clearly identifies three regions as significant outliers (outliers from MMS), non-significant outliers (outliers from EMMS), and nonoutliers within a data segment (Figure 11). In addition, the results showed that the nonoutliers follow a linear path. Furthermore, the width of the regions can be tuned by changing the relevant R_w values. Figure 11 shows some selected results of biogas data for a k value of 0.2 for MMS and a k value of 0.1 for EMMS.

One of the interesting observations was the ability of the algorithm to continue linear detection even with the noncontinuous clusters (Figures 11(b) and 11(e)). In all data segments, there occurred no false detection (there were no outliers in nonoutlier regions and vice versa). Most importantly, the new method required no further windowing and nonoutliers were detected independent of the window size.

When the general trend was constant and elements were in Gaussian distribution, the Sigma filter and LSM were able to identify the linear trend. However, for series with biased elements, both methods failed to identify the general trend. When the general trend was increment or decrement, the Sigma filter failed to identify the general trend (a further

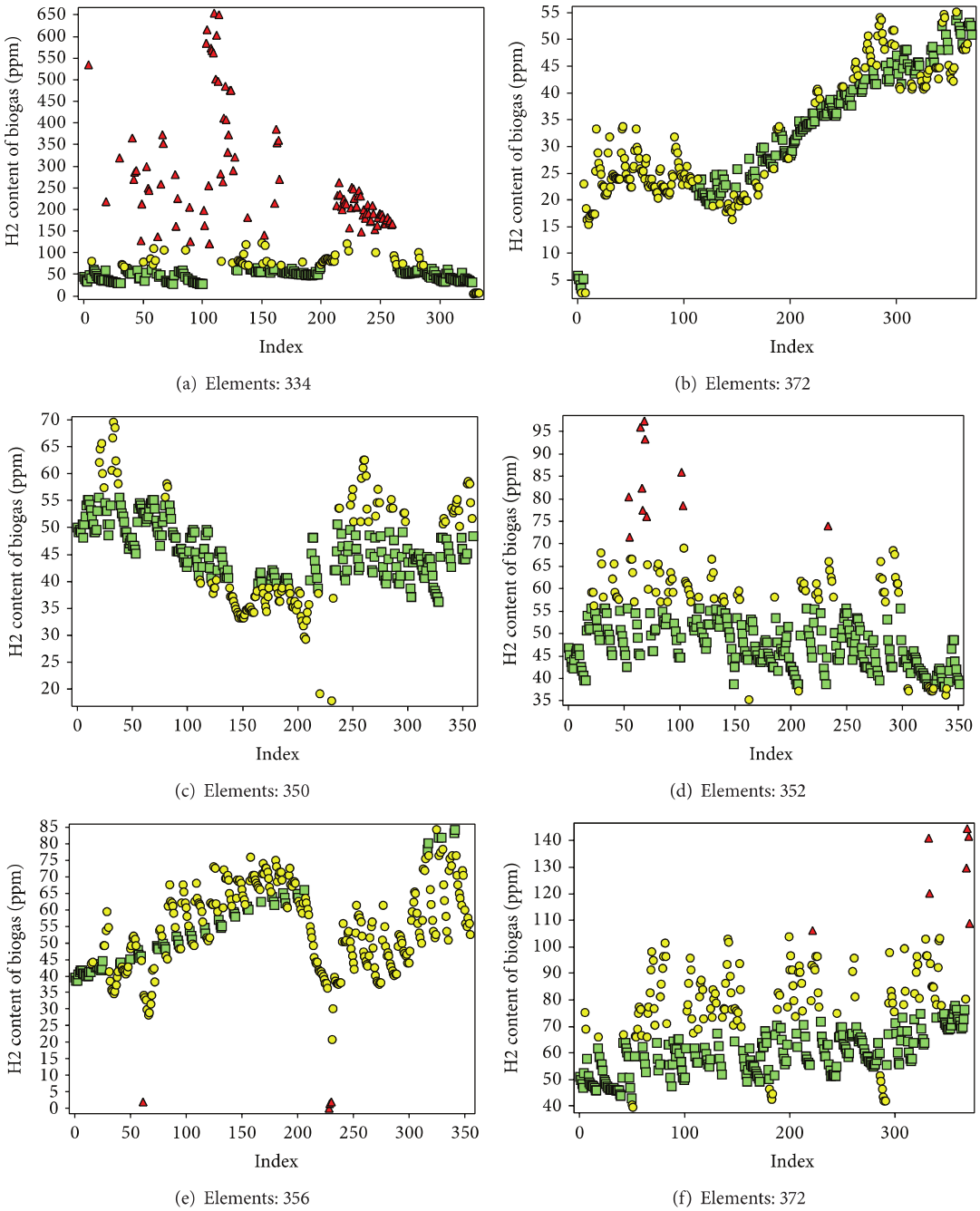


Figure 11: Results related to real biogas data with different size of data sets. The first element is the reference element, which is assumed not to be an outlier. Results showed that the algorithm clearly identifies three regions as significant outliers (outliers from MMS), nonsignificant outliers (outliers from EMMS), and nonoutliers within each data segment. Most importantly, all the nonoutliers lied within a linear border, where red triangle corresponds to outliers detected by MMS, yellow circle corresponds to outliers detected by EMMS, and green square corresponds to nonoutliers. The value of k for MMS and EMMS is 0.2 and 0.1, respectively.

segment would give better result, but we used the whole window). The new method was capable of locating 4% to 45% of elements as outliers with 0% error. Grubbs' test was capable of identifying very small amount of elements as outliers (0%–17%), even with the significance level of 0.05. However, all outliers were significant and no wrong detections were reported.

4. Conclusions and Outlook

This paper introduced a new outlier detection method using the relation of the sum of the elements of an arithmetic progression. The results of this work prove that the new method is a robust solution for outlier detection in a data set with missing elements. The method is capable of identifying

both significant and nonsignificant outliers, when the first value of the data set is not an outlier. Most importantly, the method is a solution for identifying significant outliers in a series with outliers in non-Gaussian distribution. In addition, the outlier detection is nonparametric, has floor and ceiling values, and does not require standardization. When the reference elements are unknown, the method can be used with multiple reference elements to gain optimal output.

If the frequency of the data is sufficient, any nonlinear relation can be represented as a combination of straight lines. Therefore, by using a suitable segmentation technique, it is possible to identify outliers in any data series. This will allow for detecting outliers in a process-oriented data set. Therefore, to bring a data series into a form that is suitable for our method, an intelligent segmentation technique is necessary.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The University of Ruhuna, Sri Lanka, provided the paper processing charges of this paper. The German Academic Exchange Service (German: Deutscher Akademischer Austauschdienst) financed this work.

References

[1] R. J. Beckman and R. D. Cook, "Outlier. . . . . s," *Technometrics*, vol. 25, no. 2, pp. 119–149, 1983.

[2] F. Molinari, "Missing treatments," *Journal of Business and Economic Statistics*, vol. 28, no. 1, pp. 82–95, 2010.

[3] J. Qui, B. Zhang, and D. H. Y. Leung, "Empirical likelihood in missing data problems," *Journal of the American Statistical Association*, vol. 104, no. 488, pp. 1492–1503, 2009.

[4] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85–126, 2004.

[5] Z. X. Niu, S. Shi, J. Sun, and X. He, "A survey of outlier detection methodologies and their applications," in *Artificial Intelligence and Computational Intelligence*, vol. 7002 of *Lecture Notes in Computer Science*, pp. 380–387, 2011.

[6] W. Jin, A. K. H. Tung, and J. Han, "Mining top-n local outliers in large databases," in *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '01)*, pp. 293–298, ACM, San Francisco, Calif, USA, August 2001.

[7] H.-P. Kriegel, M. S. Schubert, and A. Zimek, "Angle-based outlier detection in high-dimensional data," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08)*, pp. 444–452, ACM, Las Vegas, Nev, USA, August 2008.

[8] I. Ben-Gal, "Outlier detection," in *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach, Eds., pp. 131–146, Springer, New York, NY, USA, 2005.

[9] G. Williams, R. Baxter, H. He, S. Hawkins, and L. Gu, "A comparative study of RNN for outlier detection in data mining," in *Proceedings of the IEEE International Conference on Data Mining (ICDM '02)*, pp. 709–712, December 2002.

[10] H. Fan, O. R. Zaïane, A. Foss, and J. Wu, "A nonparametric outlier detection for effectively discovering top-N outliers from engineering data," in *Advances in Knowledge Discovery and Data Mining*, W.-K. Ng, M. Kitsuregawa, J. Li, and K. Chang, Eds., vol. 3918 of *Lecture Notes in Computer Science*, pp. 557–566, Springer, Berlin, Germany, 2006.

[11] J.-S. Lee, "Digital image smoothing and the sigma filter," *Computer Vision, Graphics and Image Processing*, vol. 24, no. 2, pp. 255–269, 1983.

[12] A. Gelb, *Applied Optimal Estimation*, MIT Press, 1974.

[13] D. Sierociuk, I. Tejado, and B. M. Vinagre, "Improved fractional Kalman filter and its application to estimation over lossy networks," *Signal Processing*, vol. 91, no. 3, pp. 542–552, 2011.

[14] P. H. Abreu, J. Xavier, D. C. Silva, L. P. Reis, and M. Petry, "Using Kalman filters to reduce noise from RFID location system," *The Scientific World Journal*, vol. 2014, Article ID 796279, 9 pages, 2014.

[15] H. Liu, S. Shah, and W. Jiang, "On-line outlier detection and data cleaning," *Computers and Chemical Engineering*, vol. 28, no. 9, pp. 1635–1647, 2004.

[16] T. D. Pigott, "A review of methods for missing data," *Educational Research and Evaluation*, vol. 7, no. 4, pp. 353–383, 2001.

[17] M. Nakai and W. Ke, "Review of the methods for handling missing data in longitudinal data analysis," *International Journal of Mathematical Analysis*, vol. 5, no. 1–4, pp. 1–13, 2011.

[18] J. L. Schafer and J. W. Graham, "Missing data: our view of the state of the art," *Psychological Methods*, vol. 7, no. 2, pp. 147–177, 2002.

[19] E. Acuña and C. Rodriguez, "The treatment of missing values and its effect on classifier accuracy," in *Classification, Clustering, and Data Mining Applications*, D. Banks, L. House, F. R. McMorris, P. Arabie, and W. Gaul, Eds., pp. 639–647, Springer, Berlin, Germany, 2004.

[20] A. N. Baraldi and C. K. Enders, "An introduction to modern missing data analyses," *Journal of School Psychology*, vol. 48, no. 1, pp. 5–37, 2010.

[21] J. Tian, B. Yu, D. Yu, and S. Ma, "Clustering-based multiple imputation via gray relational analysis for missing data and its application to aerospace field," *The Scientific World Journal*, vol. 2013, Article ID 720392, 10 pages, 2013.

[22] S. Zhaowei, Z. Lingfeng, M. Shangjun, F. Bin, and Z. Taiping, "Incomplete time series prediction using max-margin classification of data with absent features," *Mathematical Problems in Engineering*, vol. 2010, Article ID 513810, 14 pages, 2010.

[23] J.-F. Cai, Z. Shen, and G.-B. Ye, "Approximation of frame based missing data recovery," *Applied and Computational Harmonic Analysis*, vol. 31, no. 2, pp. 185–204, 2011.

[24] B. L. Wiens and G. K. Rosenkranz, "Missing data in noninferiority trials," *Statistics in Biopharmaceutical Research*, vol. 5, no. 4, pp. 383–393, 2013.

[25] Aryabhata, *The Aryabhatiya of Aryabhata: An Ancient Indian Work on Mathematics and Astronomy*, vol. 1, Kessinger Publishing, 2006.

[26] F. E. Grubbs, "Procedures for detecting outlying observations in samples," *Technometrics*, vol. 11, no. 1, pp. 1–21, 1969.

[27] B. Rosner, "On the detection of many outliers," *Technometrics*, vol. 17, no. 2, pp. 221–227, 1975.

[28] R. B. Jain, "Percentage points of many-outlier detection procedures," *Technometrics*, vol. 23, no. 1, pp. 71–75, 1981.

[29] S. P. Verma, L. Díaz-González, M. Rosales-Rivera, and A. Quiroz-Ruiz, "Comparative performance of four single extreme outlier discordancy tests from Monte Carlo simulations," *The Scientific World Journal*, vol. 2014, Article ID 746451, 27 pages, 2014.

References

1. Bayané, A. and S. Guiot, *Animal digestive strategies versus anaerobic digestion bioprocesses for biogas production from lignocellulosic biomass.* Reviews in Environmental Science and Bio/Technology, 2011. **10**(1): p. 43-62.
2. Pullen, T., *Anaerobic Digestion – Making Biogas – Making Energy: The Earthscan Expert Guide*,2015: Taylor & Francis.
3. Gübitz, G.M., A. Gronauer and H. Oechsner, *Editorial: Biogas science – State of the art and future perspectives.* Engineering in Life Sciences, 2010. **10**(6): p. 491-492.
4. Charaniya, S., W.-S. Hu and G. Karypis, *Mining bioprocess data: opportunities and challenges.* Trends in biotechnology, 2008. **26**(12): p. 690-699.
5. Doran, P.M., *Bioprocess Engineering Principles*,1995: Academic Press. p. 17-47.
6. Alford, J.S., *Bioprocess control: Advances and challenges.* Computers & Chemical Engineering, 2006. **30**(10–12): p. 1464-1475.
7. Krich, K., et al., *Biomethane from dairy waste.* 2005.
8. Gronauer, A., L. Krapf, H. Heuwinkel and U. Schmidhalter, *Near infrared spectroscopy calibrations for the estimation of process parameters of anaerobic digestion of energy crops and livestock residues.* Journal of Near Infrared Spectroscopy, 2011. **19**(6): p. 479-493.
9. Krapf, L.C., H. Heuwinkel, U. Schmidhalter and A. Gronauer, *The potential for online monitoring of short-term process dynamics in anaerobic digestion using near-infrared spectroscopy.* Biomass and Bioenergy, 2013. **48**: p. 224-230.
10. Effenberger, M., et al., *Mesophilicthermophilicmesophilic anaerobic digestion of liquid dairy cattle manure.* Water Science & Technology, 2006. **53**(8): p. 253-261.
11. Györfi, L., M. Kohler, A. Krzyzak and H. Walk, *A Distribution-Free Theory of Nonparametric Regression*,2006: Springer New York. p. 9-14.
12. Roberts, S.J., *Parametric and non-parametric unsupervised cluster analysis.* Pattern Recognition, 1997. **30**(2): p. 261-272.
13. Wasserman, L., *All of Nonparametric Statistics*,2006: Springer New York. p. 1-2.
14. Kothari, C.R., *Research Methodology: Methods and Techniques*,2004: New Age International (P) Limited. p. 283-285.
15. Li, J., S. Ray and B.G. Lindsay, *A Nonparametric Statistical Approach to Clustering via Mode Identification.* Journal of Machine Learning Research, 2007. **8**(8): p. 1687-1723.
16. Gelb, A., *Applied Optimal Estimation*,1974: M.I.T. Press.
17. Liu, H., S. Shah and W. Jiang, *On-line outlier detection and data cleaning.* Computers & Chemical Engineering, 2004. **28**(9): p. 1635-1647.
18. Kriegel, H.-P., M.S. hubert and A. Zimek, *Angle-based outlier detection in high-dimensional data*, in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*2008, ACM: Las Vegas, Nevada, USA. p. 444-452.
19. Foss, A. and O.R. Zaiane. *A parameterless method for efficiently discovering clusters of arbitrary shape in large datasets*. in *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*.2002.
20. Leys, C., et al., *Detecting outliers: do not use standard deviation around the mean, use absolute deviation around the median.* Journal of Experimental Social Psychology, 2013. **49**(4): p. 764-766.
21. Anscombe, F.J., *Graphs in Statistical Analysis.* The American Statistician, 1973. **27**(1): p. 17-21.
22. Beckman, R.J. and R.D. Cook, *Outlier … … …. s.* Technometrics, 1983. **25**(2): p. 119-149.

23. Chen, Y. and C. Caramanis. *Noisy and missing data regression: Distribution-oblivious support recovery*. in *Proceedings of The 30th International Conference on Machine Learning*.2013.
24. Sims, C.A., *Seasonality in Regression.* Journal of the American Statistical Association, 1974. **69**(347): p. 618-626.
25. Choi, S.-W., *The Effect of Outliers on Regression Analysis: Regime Type and Foreign Direct Investment.* Quarterly Journal of Political Science, 2009. **4**(2): p. 153-165.
26. Stevens, J.P., *Outliers and influential data points in regression analysis.* Psychological Bulletin, 1984. **95**(2): p. 334.
27. Liu, Y., A.D. Wu and B.D. Zumbo, *The impact of outliers on Cronbach's coefficient alpha estimate of reliability: Ordinal/rating scale item responses.* Educational and Psychological Measurement, 2010. **70**(1): p. 5-21.
28. Alimohammadi, I., P. Nassiri and M.B.M. Hosseini, *Reliability analysis of traffic noise estimates in highways of Tehran by Monte Carlo simulation method.* Iranian journal of environmental health science & engineering, 2005. **2**(4): p. 229-236.
29. De Brabanter, K., et al., *Robustness of kernel based regression: a comparison of iterative weighting schemes*, in *Artificial Neural Networks–ICANN 2009*,2009, Springer. p. 100-110.
30. Liu, Y., B.D. Zumbo and A.D. Wu, *A demonstration of the impact of outliers on the decisions about the number of factors in exploratory factor analysis.* Educational and Psychological Measurement, 2012. **72**(2): p. 181-199.
31. Sykes, A.O., *An introduction to regression analysis.* 1993: p. 16.
32. Dicker, L.H., *Residual variance and the signal-to-noise ratio in high-dimensional linear models.* arXiv preprint arXiv:1209.0012, 2012.
33. Michikazu Nakai, W.K., *Review of the Methods for Handling Missing Data in Longitudinal Data Analysis.* Int. Journal of Math. Analysis, 2011. **5**(1-4): p. 1-13.
34. Stuart, E.A., M. Azur, C. Frangakis and P. Leaf, *Multiple Imputation With Large Data Sets: A Case Study of the Children's Mental Health Initiative.* American Journal of Epidemiology, 2009. **169**(9): p. 1133-1139.
35. Sierociuk, D., I. Tejado and B.M. Vinagre, *Improved fractional Kalman filter and its application to estimation over lossy networks.* Signal Processing, 2011. **91**(3): p. 542-552.
36. Henriques Abreu, P., et al., *Using Kalman Filters to Reduce Noise from RFID Location System.* The Scientific World Journal, 2014: p. 9.
37. Pronzato, L. and A. Pazman. *Recursively re-weighted least-squares estimation in regression models with parameterized variance*. in *Signal Processing Conference, 2004 12th European*.2004.
38. Xiaorong, Y. and F. Ke-Ang. *Copy number detection using self-weighted least square regression*. in *Systems Biology (ISB), 2011 IEEE International Conference on*.2011.
39. Ramachandran, K.M. and C.P. Tsokos, *Mathematical Statistics with Applications in R*,2014: Elsevier Science. p. 412.
40. Hoschek, J. and P. Kaklis, *Advanced Course on FAIRSHAPE*,2012: Vieweg+Teubner Verlag. p. 257.
41. Pigott, T.D., *A Review of Methods for Missing Data.* Educational Research and Evaluation, 2001. **7**(4): p. 353-383.
42. Schafer, J. and J. Graham, *Missing data: our view of the state of the art.* Psychological methods, 2002. **7**(2): p. 147-177.
43. Acuña, E. and C. Rodriguez, *The Treatment of Missing Values and its Effect on Classifier Accuracy*, in *Classification, Clustering, and Data Mining Applications*, D. Banks, et al., Editors.,2004, Springer Berlin Heidelberg. p. 639-647.

44. Baraldi, A.N. and C.K. Enders, *An introduction to modern missing data analyses.* Journal of School Psychology, 2010. **48**(1): p. 5-37.
45. Tian, J., B. Yu, D. Yu and S. Ma, *Clustering-Based Multiple Imputation via Gray Relational Analysis for Missing Data and Its Application to Aerospace Field.* The Scientific World Journal, 2013 : p. 10.
46. Zhaowei, S., et al., *Incomplete Time Series Prediction Using Max-Margin Classification of Data with Absent Features.* Mathematical Problems in Engineering: p. 14.
47. Cai, J.-F., Z. Shen and G.-B. Ye, *Approximation of frame based missing data recovery.* Applied and Computational Harmonic Analysis, 2011. **31**(2): p. 185-204.
48. Wiens, B.L. and G.K. Rosenkranz, *Missing Data in Noninferiority Trials.* Statistics in Biopharmaceutical Research, 2013. **5**(4): p. 383-393.
49. Gilgen, H., *Univariate Time Series in Geosciences: Theory and Examples*,2006: Springer Berlin Heidelberg. p. 390-393.
50. Zou, H.-F., Y.-K. Zhang and P.-C. Lu, *The prediction of the peak width at half height in HPLC.* Chinese Journal of Chemistry, 1991. **9**(3): p. 237-244.
51. Antoniadis, A., J. Bigot and S. Lambert-Lacroix, *Peaks detection and alignment for mass spectrometry data.* Journal de la Société Française de Statistique, 2010. **151**(1): p. 17-37.
52. Jeffries, N., *Algorithms for alignment of mass spectrometry proteomic data.* Bioinformatics, 2005. **21**(14): p. 3066-3073.
53. Sauve, A.C. and T.P. Speed, *Normalization, baseline correction and alignment of high-throughput mass spectrometry data.* Proceedings Gensips, 2004.
54. Brachman, R. and H. Levesque, *Knowledge Representation and Reasoning*,2004: Elsevier Science. p. 1.
55. Yu, H., J. Yang, J. Han and X. Li, *Making SVMs Scalable to Large Data Sets using Hierarchical Cluster Indexing.* Data Mining and Knowledge Discovery, 2005. **11**(3): p. 295-321.
56. De Vito, E., L. Rosasco and A. Toigo, *Learning sets with separating kernels.* Applied and Computational Harmonic Analysis, 2014. **37**(2): p. 185-217.
57. Galluccio, L., O. Michel, P. Comon and A.O. Hero Iii, *Graph based k-means clustering.* Signal Processing, 2012. **92**(9): p. 1970-1984.
58. Lee, R.C.T., *Clustering Analysis and Its Applications*, in *Advances in Information Systems Science*, J. Tou, Editor,1981, Springer US. p. 169-292.
59. Næs, T., P.B. Brockhoff and O. Tomic, *Cluster Analysis: Unsupervised Classification*, in *Statistics for Sensory and Consumer Science*,2010, John Wiley & Sons, Ltd. p. 249-261.
60. Okun, O. and H. Priisalu, *Unsupervised data reduction.* Signal Processing, 2007. **87**(9): p. 2260-2267.
61. Anderberg, M.R., *Cluster analysis for applications*,1973: Academic Press.
62. Chui, C.K., F. Filbir and H.N. Mhaskar, *Representation of functions on big data: Graphs and trees.* Applied and Computational Harmonic Analysis. http://dx.doi.org/10.1016/j.acha.2014.06.006.
63. Avramenko, Y., E.-C. Ani, A. Kraslawski and P.S. Agachi, *Mining of graphics for information and knowledge retrieval.* Computers & Chemical Engineering, 2009. **33**(3): p. 618-627.
64. Barbará, D. and P. Chen, *Using Self-Similarity to Cluster Large Data Sets.* Data Mining and Knowledge Discovery, 2003. **7**(2): p. 123-152.
65. David, G. and A. Averbuch, *Hierarchical data organization, clustering and denoising via localized diffusion folders.* Applied and Computational Harmonic Analysis, 2012. **33**(1): p. 1-23.

66. Woodruff, A., J. Landay and M. Stonebraker. *Constant density visualizations of non-uniform distributions of data*. in *Proceedings of the 11th annual ACM symposium on User interface software and technology*.1998. ACM.
67. Yang, J., M.O. Ward and E.A. Rundensteiner, *Visual hierarchical dimension reduction for exploration of high dimensional datasets.* 2002.
68. Zhang, L., et al., *VizCluster and its Application on Classifying Gene Expression Data.* Distributed and Parallel Databases, 2003. **13**(1): p. 73-97.
69. Johansson, J., P. Ljung, M. Jern and M. Cooper, *Revealing structure in visualizations of dense 2D and 3D parallel coordinates.* Information Visualization, 2006. **5**(2): p. 125-136.
70. Cvek, U., et al., *Multidimensional visualization tools for analysis of expression data.* World Academy of Science, Engineering and Technology, 2009. **30**: p. 281-289.
71. Yadav, B.S. and M. Mohan, *Ancient Indian Leaps into Mathematics*, 2011: Birkhauser. p. 88.
72. Ray, B., et al., *Different Types of History*,2009: Pearson Longman. p. 95.
73. Aryabhata, *The Aryabhatiya Of Aryabhata: An Ancient Indian Work On Mathematics And Astronomy*. Vol. 1 edition.2006, LLC: Kessinger Publishing.
74. Adikaram, K.K.L.B., M.A. Hussein, M. Effenberger and T. Becker, *Outlier Detection Method in Linear Regression Based on Sum of Arithmetic Progression.* The Scientific World Journal, 2014.
75. Chan, Y., *Biostatistics 104: correlational analysis.* Singapore Med J, 2003. **44**(12): p. 614-9.
76. France, J. and E. Kebreab, *Mathematical Modelling in Animal Nutrition*,2008: CABI. p. 15.
77. Bronson, G., *C++ for Engineers and Scientists*,2012: Cengage Learning. p. 185.
78. Farnstrom, F., J. Lewis and C. Elkan, *Scalability for clustering algorithms revisited.* SIGKDD Explor. Newsl., 2000. **2**(1): p. 51-57.
79. Gupta, C. and R.L. Grossman. *GenIc: A Single-Pass Generalized Incremental Algorithm for Clustering*. in *SDM*.2004. SIAM.
80. Azami, H., A. Khosravi, M. Malekzadeh and S. Sanei, *A New Adaptive Signal Segmentation Approach Based on Hiaguchi's Fractal Dimension*, in *Emerging Intelligent Computing Technology and Applications*, D.-S. Huang, et al., Editors.,2012, Springer Berlin Heidelberg. p. 152-159.
81. Azami, H., K. Mohammadi and B. Bozorgtabar, *An Improved Signal Segmentation Using Moving Average and Savitzky-Golay Filter.* Journal of Signal and Information Processing, 2012. **3** (1): p. 39-44.
82. Chollet, G., A. Esposito, M. Faundez-Zanuy and M. Marinaro, *Nonlinear Speech Modeling and Applications: Advanced Lectures and Revised Selected Papers*,2005: Springer Berlin Heidelberg.