REDCLAN - RElative Density based CLustering and ANomaly detection

Diptarka Saha, *Walmart Labs*
Debanjana Banerjee, *Walmart Labs*
Bodhisattwa Prasad Majumder, *Walmart Labs*

Abstract—Cluster analysis and Anomaly Detection are the primary methods for database mining. However, most of the data in today's world, generated from multifarious sources, don't adhere to the assumption of single distribution as their source — hence the problem of finding clusters in the data becomes arduous as clusters are of widely differing sizes, densities and shapes, along with the presence of noise and outliers.Thus We propose a relative KNN kernel density based clustering algorithm. The un-clustered (noise) points are further classified as anomaly or non-anomaly using a weighted rank based anomaly detection method. This method works particularly well when the clusters are of varying variability and shape, in these cases our algorithm can not only find the "dense" clusters that other clustering algorithms find, it also finds low-density clusters that these approaches fail to identify. This more accurate clustering in turn helps reduce the noise points and makes the anomaly detection more accurate.

Index Terms—Clustering, Relative KNN – kernel density, Varying density clusters, Anomaly Detection,DBSCAN

1 Introduction

In the industry today, categorization could be the single most important problem – categorize people according to annual income, categorize customers according to purchase patterns, categorize items according to price and the list goes on. The underlying data for categorization could have any form whatsoever – structured, unstructured, labeled, unlabeled, adhering to assumptions or, not. Establish the purpose of the categorization is fundamental but most often the purpose of categorization can be established only after successful categorization. In a scenario where the data points are unlabelled, the purpose of categorization would be simply to study the underlying pattern of the data and this class of the problems is typically known as the problem of Clustering [2]. In this paper, we are going to study a novel clustering technology that achieves clustering via anomaly detection or, vice-versa.

One traditional method of finding clusters is meaningfully choosing certain base points at different parts of the data and housing all points which lie 'closer' to those base points based on a certain distance metric and a suitable threshold. Another approach could be distribution based clustering where the method assumes a set of K underlying distributions and every data point to belong to a particular distribution or, a mixture of multiple distributions [5]. The objective is to learn the underlying distributions and the weight vectors giving the mixing proportions in which a data point can belong to a mixture of the underlying distributions. The drawback of this method is that it essentially has a parametric assumption, which is unlikely to hold true for real-life data where the data is rather unruly and unstructured owing to its multifarious origin; especially when they are large in size.

In the case of anomaly detection, the most basic approach uses the method of flagging off the most extreme points, which typically fall beyond a certain threshold, mostly these thresholds being higher quantiles. Even if the approach is non-parametric, it fails to look at any more than what the data has to suggest at its surface. The obvious parametric alternative is checking the distributional properties of the data and replacing the sample quantiles by

theoretical quantiles.

We would take a help of an example to better explain this. Imagine a problem of finding the clusters of watches based on their price. In set of 20 brands, let's also assume there are 10 low priced , 6 moderately priced, 3 high priced and 1 very high priced watches present. Now the lone case of very high-priced watch, it is possible that there is some potential outlier-ish nature to it or it may be just a case of low class presence of very high priced class. Even though many simple anomaly detection method would flag that as anomaly, it is legit to examine the case in the light of other classes. This idea has an intrinsic propensity of including the fundamentals of clustering which lead us to further understanding the problem while complementing both clustering and anomaly detection methods.

Instead of looking for anomalies overall, we look for anomalies with respect to every cluster. That is, a point lying further apart in the price space may be an anomaly or, it could simply be a part of a different cluster that our data falls short of capturing; whereas a point lying close (in the price space) to a densely populated cluster (say, low priced brands) in an absolute sense but not so much in a relative sense, can be a potential outlier. Most clustering techniques determine the efficacy of the exercise by maximizing the Between-Cluster Distance (SSB) & minimizing the Within-Cluster Distance (SSW) [1]. While this may work when the data is large enough and truly representative of the population, for most practical scenarios, using relative distances is a better approach. That tagged with the concept of neighborhood (based on relative distances) is what REDCLAN tries to explore. How this is specifically effective in the case of clusters of varying densities is what we will explain in the subsequent sections.

2 Relative Density Based Clustering and Anomaly Detection

2.1 Motivation

An important property of many real-data sets is that their intrinsic cluster structure cannot be characterized by global density parameters [6]. Very different local densities may be needed to reveal clusters in different regions of the data space. For example in figure 1, it is not possible to detect the clusters $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$ simultaneously using one global density parameter.

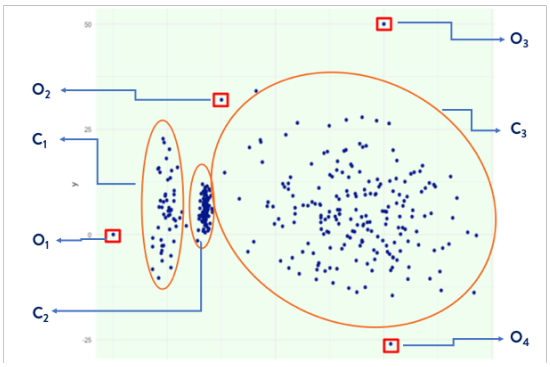


Fig. 1: Data with Clusters of Varying Density

Here we should note that this synthetic 2D dataset will be used several several times for illustration purposes; this contains three Gaussian clusters of varying density $\{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3\}$ and 4 deliberately introduced outliers $\{\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_3, \mathcal{O}_4\}$

The aforementioned drawback of density based clustering techniques such as DBSCAN can be understood as following, if density of $\mathcal{C}_2$ is taken as the global density parameter then $\mathcal{C}_1, \mathcal{C}_2$ will be seen as noise points, on the other hand if density of $\mathcal{C}_3$ is taken as the global density parameter then $\mathcal{C}_2$ will be over-fragmented by algorithms such as DBSCAN [4] as is evident in the following Fig2

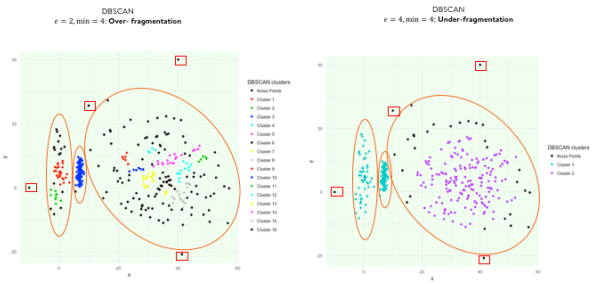


Fig. 2: Data with Clusters of Varying Density: Use of DBSCAN

To overcome this problem, one needs to consider density relative to its neighbour, some-

thing called 'relative density', which will be formally defined later — this essentially means if the density parameter for considering a set of points to be included in a cluster or to be left as noise points, will vary from point to point.

As we understand, after the clustering step we will have clusters and noise points; The natural next step for a data categorization algorithm such as ours would be to find outliers in the data which we will accomplish by using a weighted rank based anomaly detection technique. Since, the performance of outlier detection algorithms depends on how good the clustering algorithm captures the structure of clusters [11] — this algorithms provides significant improvement in data sets comprised of clusters of varying density such as in Fig 1

2.2 Definitions

The following definitions will be used while describing the algorithm
First, consider the set of all d-diemnsional points in the given data to be denoted by $\mathcal{D} = \{X_1, ..., X_n\}$ For what follows, whenever we mention for a point p, it is understood $p \in \mathcal{D}$ Whenever, distance of two points is discussed we assume Euclidean distance

- **kNN—neighbourhood:** If $d_k(p)$ is the distance between p and its k^{th} nearest neighbor, then denote the set of k nearest neighbors of p by
$$N_k(p) = \{q \in D - \{p\} : d(p, q) \leq d_k(p)\}.$$
- **Adaptive Bandwidth:** Suppose, for a point p we have its kNN neighborhood $N_k(p)$, and given fixed $\epsilon > 0$
$$D_k(p) = max(d(p, q) : q \in N_k(p))$$
$$d_k(p) = min(d(p, q) : q \in N_k(p))$$
$$\bar{d}_k(p) = mean(d(p, q) : q \in N_k(p))$$
we form an **adaptive bandwidth** around the point p as following
$$h(p) = (D_k(p) + d_k(p) + \epsilon - \bar{d}_k(p))$$
- **kNN based Relative Density:** following definition 2, a balloon estimator [3] might be defined as
$$\rho(x) = \frac{1}{n(h(x))^d} \sum_1^n K(\frac{x - X_i}{h(x)})$$
on top of this rather dynamic definition of density, we add another layer of local scaling and define our **relative density** as
$$\tilde{\rho}(x) = \frac{\rho(x)}{mean_{X_i \in N_k(x)}(\rho(X_i))}$$
- **Core Points:** A point will be denoted as core point if it has high enough relative density, i.e. for some threshold θ_1 a point p will be denoted a core point iff
$$\tilde{\rho}(x) \geq \theta_1$$
authors typically determine θ_1 using bootstrap on the entire set of relative densities
- **Directly Reachable:** A point p is said to be directly reachable from another point q iff
  - q is a core point
  - $p \in N_k(q)$
- **Reachable:** A point p is said to be reachable from another point q iff
$$\exists\, p_1, p_2, ..., p_n \text{ with } p_1 = q, p_n = p \text{ such that } p_{i+1} \text{ is directly reachable from } p_i \quad \forall i = 1, ..., n-1$$
- **Connected:** A point p is connected to a point q iff there is a point o such that both, p and q are reachable from o
- **Rank:** The rank[12] of p w.r.t. q is defined as
$$rank_q(p) = |X_i \in \mathcal{D} : d(q, X_i) \leq d(p, X_i)|$$
in informal terms this is the order rank of p w.r.t. q is the number of points between q and p plus 1
- **Outlierness:** Outlierness of a point is a function of the weighted sum of its rank w.r.t. its neighbour
for a point p, let q be its neighbor, now if q is part of a cluster C define, $w(q) = \frac{1}{|C|}$, if q is not part of any cluster (noise point) then w(q) = 1. This is to say every cluster has weight 1 which is equally divided among its components
With this weight-age scheme in hand **Outlierness** will be defined as
$$O(p) = \frac{\sum_{q \in N_k(p)} w(q) rank_q(p)}{k}$$

2.3 Methodology

Next, we will discuss the entire algorithm and why/how it works. It might be helpful to demonstrate the anatomy of the algorithm by using it on the synthetic dataset shown in Fig 1.

1) **Core Point Detection:** The very first step would be find the set of core points, this is done with the help of definition 4. Since, the core point is defined based on relative density and not absolute density — we can note (as in Fig 3(a).), the core points will be spread across all the clusters — both dense and sparse ones, this forms the backbone of our algorithm.
2) **Clustering:** Given the set of core points, we will cluster the points into separate clusters. The clustering logic will be the following logic.
Define the clustering function which assigns a cluster number to every point in $\mathcal{D}$:
$\texttt{cl} : \mathcal{D} \rightarrow \mathbb{N}$
Also denote,
$n = |\mathcal{D}|$
$\mathcal{A}(p)$: union of p and the set of points directly reachable from p
`core`: the set of core points from part a

Initialize:
$\texttt{cl}(p) \leftarrow 0 \; \forall \, p \in \mathcal{D}$
$C \leftarrow max(\texttt{cl})$
while p in $\mathcal{D}$ **do**
  if $\texttt{cl}(p) == 0$ **then**
    if $p \in \texttt{core}$ **then**
      $\texttt{cl}(\mathcal{A}(p)) \leftarrow C + 1$
      $C \leftarrow C + 1$
    end if
  end if
end while

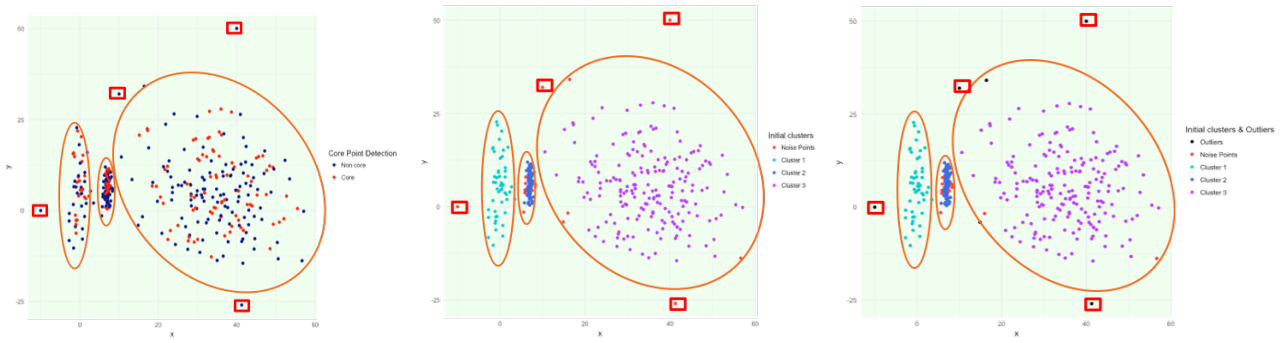
At the end of this step(Fig 3(b)), $\texttt{cl}(p)$ is the cluster number a point is assigned to. If $\texttt{cl}(p) = 0$, this means the point is left as a noise point. This is a one-time breadth-first process and depends on two input parameters, k_1, k_2
k_1 : the k used in determining adaptive bandwidth in step 1 (core point detection), the higher the value of k_1 the lower number of core points will be found, and more and more core points will be concentrated towards the denser cluster
k_2 : the k used in determining the reachability of the points in step 2, the higher the value of k_2 the lower number of clusters will be found
3) **Anomaly Detection:** At the end of step 3, we have clusters and noise points — these noise points maybe either anomalies or just boundary of given clusters, so we will call them potential outliers. Our algorithm, goes the extra mile by finding outliers from the set of potential outliers using the weighted rank based anomaly detection method.
Now, for a suitable threshold, θ_2 we do the following,

while p in $\mathcal{D}$ **do**
  if $\texttt{cl}(p) == 0$ **then**
    $O(p) = \frac{\sum_{q \in N_k(p)} w(q) rank_q(p)}{k}$
    if $O(p) \geq \theta_2$ **then**
      $\texttt{cl}(p) \leftarrow -1$
    end if
  end if
end while

Note, authors have used $k = k_2$ in determining neighborhood while calculating Outlierness.
At the end of this step (Fig 3(c)),
If $\texttt{cl}(p) = -1$, this means the point is assigned an **outlier** status
4) **Cluster Proposal:** In this *optional* final step, we wrap up the process by suggesting a cluster for the set of points $\{p$: $\texttt{cl}(p) = 0\}$, i.e. points which are non-anomalous but noise points. This is a rather easy task as we already have somewhat labelled scenario. Let, $\mathcal{C}_j$ denote the j^{th} cluster — We do the following:

while p in $\mathcal{D}$ **do**
  if $\texttt{cl}(p) == 0$ **then**
    Find the point's average distance from each of the cluster
    for j in 1 to $max(\texttt{cl})$ **do**
      $\Delta_p[j] = \frac{\sum_{q \in \mathcal{C}_j} d(p,q)}{|\mathcal{C}_j|}$
    end for
    if $l == argmin(\Delta_p)$ **then**
      $\texttt{cl}(p) \leftarrow l$
    end if
  end if
end while

As we see (Fig 4.) this clears up the remaining points by assigning them a



(a) Core point detection (b) Clustering (c) Outlier detection

Fig. 3

cluster closest to them. This concludes our algorithm

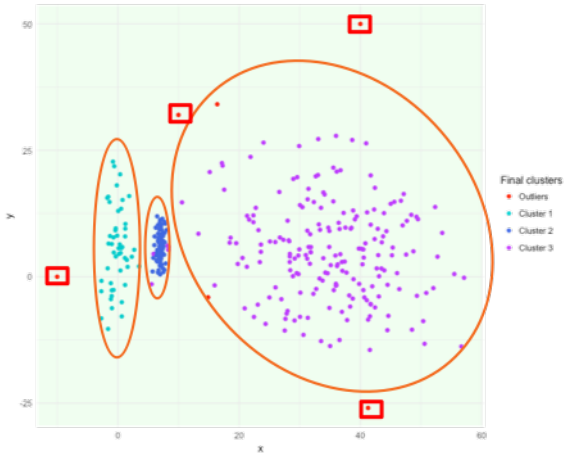


Fig. 4: Final Cluster Proposal

3 Experiments

In this section we show results of the experiments we have performed over two 2D synthetic datasets, only 2D datasets is demonstrated here partly because similar data sets have been used most extensively by other authors [7] and partly because it is easy to evaluate the quality of clustering on 2D data sets by naked eye — hence these are better suited for space constrainted scenarios such as these

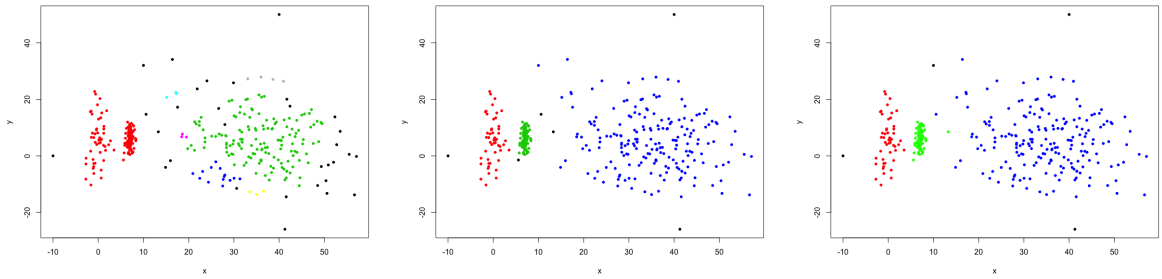
3.1 Benchmarks

For comparison purposes, we will be using two other algorithms. First is DBSCAN, which is probably the most renowned and most used density based clustering algorithm. Second is SNN based clustering proposed in [9], which has shown empirical superiority over similar methods such as k-means, DBSCAN, CURE [8] etc. We have chosen these two algorithms as the anatomy of the these two match with our algorithm — as all three revolve around the idea of identifying core points and building clusters around them. All of these do not require number of clusters to be user defined and works better than other methods when applied on spatial data. However, as mentioned earlier: While DBSCAN can find clusters of arbitrary shapes, it cannot handle data containing clusters of differing densities, since its density based definition of core points cannot identify the core points of varying density clusters; something that SNN seems to alleviate, proving to be superior in terms of identifying clusters of widely different shapes, sizes, and densities.

Authors would like to emphasize, since the parametrization of all three algorithms used here are very fluid and different values of parameters provide vastly different results, we have experimented with a wide array of inputs for all of the algorithms, and will only be sharing the best outcomes for individual algorithms and their corresponding input values.

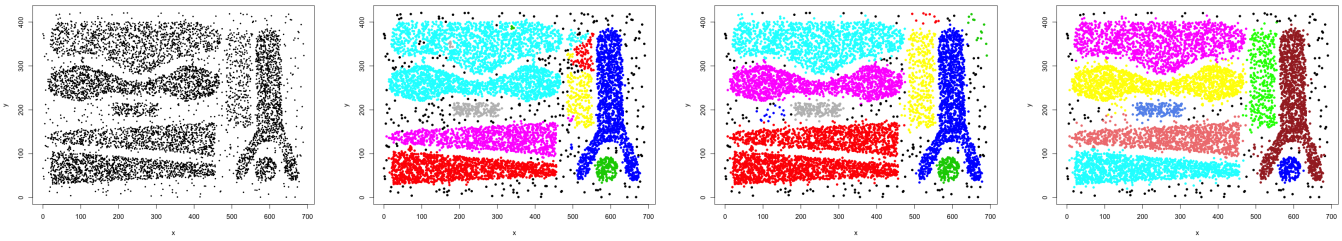
3.2 Synthetic Dataset:1

Coming first is the dataset we have used for illustration purposes throughout the paper; Figures 5(a) & 5(b) show how DBSCAN and SNN



(a) (b) (c)

Fig. 5: On Dataset1: (a) DBSCAN ϵ = 3, Minpt = 3; (b) SNN ϵ = 3, Minpt = 3, k = 12; (c) REDCLAN k_1 = 4, k_2 = 11



(a) (b) (c) (d)

Fig. 6: (a) Dataset2; (b) DBSCAN ϵ = 8, Minpt = 4; (c) SNN ϵ = 5 , Minpt = 10, k = 15; (d) REDCLAN k_1 = 35, k_2 = 14

perform on this dataset respectively.

We can see even at its best, DBSCAN fails miserably— over-fragmenting $\mathcal{C}_3$, which is the low-density cluster and mixing the other two higher-density cluster together all the while creating plenty of noise point for the user to deal with. SNN on the other hand is quite adept at handling clusters with varying density identifying the clusters near perfectly, however, it fails to observe the anomalies, labeling some points as anomalies when they are actually part of clusters and failing to identify 1 out of the 4 outliers. However, one should acknowledge SNN as a huge improvement over more traditional DBSCAN.

Fig 5(c) shows REDCLAN almost perfectly identifies every cluster and also recognises 4 (and only 4) outliers in the dataset, only misclassifying one point which as a boundary point. This datset provides a great case study - on one hand without doubt our algorithm surpasses DBSCAN, it also enjoys a unique edge over algorithms such as SNN which do correct for varying density but don't have any way of differentiating between noise points created and actual outliers. This makes REDCLAN somewhat Swiss army knife for data mining tasks - which is reflected not only in quality of result but also user-friendliness and satisfaction.

3.3 Synthetic Dataset:2

This dataset (Fig 6(a)) was originally part of CHAMELEON study [7] and is publicly available as part of the R package 'seriation' [10]. We can see 8 different clusters of vastly different shape, size and density floating in a pool of noise points.This proves appears to be a comprehensive test of competence for algorithms working on low-dimensional spatial data. The results of the algorithms on this data can be viewed in Fig 6(b), 6(c) and 6(d).

Again, one can note similar results and a clear hierarchy of proficiency among the three

algorithms, DBSCAN when faced with various degrees of densities gives unsatisfactory results - unnecessarily creating smaller clusters in a low-density cluster and merging two higher density cluster just as earlier. SNN performs better than DBSCAN as expected, at least in terms of identifying lower- density clusters correctly. However, it falls short of the accuracy it gained in the previous dataset. In fact, we can see merging of higher density cluster here too - possibly due to inability to adapt to such changes in density in the data. Moreover, the pool of noise points creates problems for SNN; it ends up creating small inconsequential clusters among these points.

The ability of REDCLAN in dealing with all these issues can be demonstrated here, It again outperforms the other two methods by pinpointing the 8 clusters and the surrounding noise points. One can notice however, few noise points are assigned a cluster number - this is due to the fact that they are so close to the cluster spatially, they almost act as boundary points.

4 Conclusion

In this paper, we present a novel technique of clustering and anomaly detection where both work in a complementary fashion. We have established the case for identification of varying density clusters which is the most practical case owing to the multifarious nature of the data. Our methodology shows notable improvements over previous density based clustering methods like DBSCAN and SNN which are popularly used. Even though we have demonstrated the performance on synthetic datasets for the sake of comparison with previous methods, our technique particularly become effective while dealing with various problems in e-commerce and finance. Identifying various minute classes of substitutes or finding database anomalies from a large streaming data or identifying anomalous behaviour in the buyer-seller network are some of the prominent use-cases where our method has seen success.

Bibliography

[1] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[2] A. K. Jain and R. C. Dubes, "Algorithms for clustering data," 1988.

[3] G. R.Terrell and D. W. Scott, "Variable Kernel Density Estimation," *The Annals of Statistics*, pp. 1236–1265, 1992.

[4] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," *AAAI*, p. 3, 1996.

[5] X. Xu, M. Ester, H.-P. Kriegel, and J. Sander, "A distribution-based clustering algorithm for mining in large spatial databases," in *Data Engineering, 1998. Proceedings., 14th International Conference on*, IEEE, 1998, pp. 324–331.

[6] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: Ordering Points To Identify the Clustering Structure," *Int. Conf. on Management of Data*, p. 3, 1999.

[7] G. Karypis, E.-H. (Han, and V. Kumar, *Chameleon: A hierarchical clustering algorithm using dynamic modeling*, 1999.

[8] S. Guha, R. Rastogi, and K. Shim, "Cure: an efficient clustering algorithm for large databases," *Information Systems*, 2001.

[9] L. Ertöz, M. Steinbach, and V. Kumar, "Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data," in *SDM*, 2003.

[10] M. Hahsler, K. Hornik, and C. Buchta, "Getting things in order: An introduction to the r package seriation," *Journal of Statistical Software*, vol. 25, no. 3, pp. 1–34, Mar. 2008, ISSN: 1548-7660.

[11] I. Syarif, A. Prugel-Bennett, and G. Wills, "Unsupervised clustering approach for network anomaly detection," *Communications in Computer and Information Science*, p. 3, 2012.

[12] H. Huang, "Rank Based Anomaly Detection Algorithms," *Electrical Engineering and Computer Science - Dissertations.331*, 2013.