Clustering Data in Stationary Environments with a Local Network Neighborhood Artificial Immune System

A. J. Graaff A. P. Engelbrecht

Abstract

The network theory in immunology inspired the modeling of network based artificial immune system (AIS) models for data clustering. Current network based AIS models determine the network connectivity between artificial lymphocytes (ALCs) by measuring the spatial distance between these ALCs against a distance threshold or by grouping ALCs into sub-networks. This paper discusses alternative network topologies to determine the network connectivity between ALCs and the advantages of using these network topologies. The local network neighborhood AIS model is then proposed as a network based AIS model which uses an index-based ALC neighborhood to determine the network connectivity between ALCs. The proposed model is compared to existing network based AIS models which are applied to data clustering problems. Furthermore, a sensitivity analysis is also done on the proposed model to investigate the influence of the model's parameters on the quality of the clusters. The paper also gives a formal definition of data clustering and discusses the performance measures used to determine the quality of clusters.

1 Introduction

Immunology is the study of the functioning of the natural immune system found within organisms. The natural immune system functions as a defense mechanism against foreign or unknown organisms in the body. The invasion of pathogenic material into the body of an organism triggers an immune response. Artificial immune systems (AIS) model the functioning of the natural immune system. AISs have been successfully applied to classification [12, 14, 15, 35, 36], optimization [4, 5, 7, 13], and data mining and clustering [6, 8, 42, 43, 44]. Different AIS models exist, based on different theories of immunology. The model presented in this paper is inspired by the network theory and is applied to data clustering problems. Thus, the majority of the paper will be on the network theory of the natural immune system and the discussion of AIS models based on the network theory and their application to data clustering problems.

The network interaction or network formation between artificial lymphocytes (ALCs) in existing network based artificial immune systems, is either determined by a proximity matrix of network affinities or the grouping of similar artificial lymphocytes in sub-networks. The former is normalized with a network affinity threshold to determine the network links between the artificial lymphocytes. Therefore the network affinity threshold determines the number of ALC networks and it can be a formidable task to specify the correct network affinity threshold to obtain the correct or required number of clusters, i.e. the ALC network formation is sensitive to the user specified network affinity threshold value. In the latter case where similar artificial lymphocytes are grouped into sub-networks, a hybrid-approach is taken by clustering the ALC population into sub-nets. A potential drawback of the hybrid-approach is that the clusters (sub-nets) might contain ALCs which do not have a good or generic representation of the data. Furthermore, existing network based AIS models for data clustering have many user specified parameters which need to be determined individually for each data set. Some of the existing network based AIS models have a low compression on the data (unable to remove redundant data). The motivation for the research presented in this paper is to propose a network based AIS model for data clustering in stationary and non-stationary environments, and which has less user specified parameters, is independent of a network affinity threshold (or a hybrid-approach) to determine the number of ALC networks, and delivers clusters of high quality with a good compression of the data. Furthermore the proposed AIS model should also be capable to dynamically determine the number of clusters in a data set. The focus of this paper is the clustering of data in stationary environments into a user specified number of clusters. The interested reader is referred to [17] where the proposed AIS model has been applied to the clustering of data in non-stationary environments and

[18] where an enhanced version of the proposed AIS model dynamically determines the number of clusters in a data set.

Focusing on the network interaction or network formation between ALCs, there are alternative and less familiar network topologies to determine the possible interactions in a network of lymphocytes. These are the linear topology introduced by Richter and proposed as a *chain-reaction* between lymphocytes at different levels [38, 39], the cyclic topology which is a conversion of the linear topology and introduced by Hiernaux [21], and the Cayley tree which is a loop-less tree [45]. The node which contains the lymphocyte with the highest affinity with an antigen forms the root node of a Cayley tree [34, 45]. The advantages of a neighborhood topology have been successfully shown for particle swarm optimization (PSO) [26]. The two most common neighborhood topologies used in PSO are the *star* and *ring* topologies [10]. The *star* neighborhood topology is a fully meshed network of particles where every particle is connected to every other particle in the network topology. Each particle can therefore communicate with every other particle. The *ring* topology arranges particles in a ring structure such that each particle has a number of particles to the right and left forming the particle's neighborhood. The *star* topology in PSO is very similar to the proximity matrix of network affinities in AISs where a network affinity threshold is the only difference and controls the connectivity of the ALCs in the population. The *ring* topology in turn is very similar to the cyclic topology in AISs. There is however a major difference between the network topologies in PSO and those in AIS: Neighbors in PSO are not determined by spatial distance as in AIS but by particle indices.

Suganthan [41] introduced the advantages of a neighborhood operator based on spatial information and was illustrated for particle swarm optimization (PSO). A neighborhood is defined as the ratio of the distance between a particle that needs to adapt and all other particles in a swarm. The distance ratio is measured against a threshold to determine a particle's local neighborhood and is similar to the network affinity threshold used with the proximity matrix of network affinities in AISs. An advantage of a neighborhood operator is the initial diverse space coverage (greedy search), gradually converging to a more specific search.

The novelty of the proposed network based AIS model in this paper is the interpretation of the network theory in that a lymphocyte's neighbors are not determined by spatial distance and a network affinity threshold, but by the index in the population of lymphocytes. This results in the formation of local network neighborhoods in the population of lymphocytes. An advantage of local network neighborhoods is increased diversity, which allows lymphocyte networks to adapt to changes in the environment. Furthermore, lymphocytes in a network neighborhood interact and *learn* from one another to have a better local representation of patterns. Graaff and Engelbrecht [16] introduced the concept of an index-based *artificial lymphocyte neighborhood*, as applied to data clustering, with promising results.

The objectives of this paper is to:

- give a formal definition of data clustering,
- introduce a novel network based AIS model which utilizes a different network topology,
- perform empirical analysis on the proposed model, and
- compare the clustering performance of the proposed AIS with existing network based AIS and classical data clustering models.

Section 2 gives a formal definition of data clustering, the performance measures used to measure the quality of clusters and an overview of existing classical data clustering algorithms used in this paper. Section 3 gives an introduction to the natural immune system and provides a detailed explanation of the network theory. An overview of existing network theory based AISs, which are applied to data clustering problems, is given in section 3.2. The proposed local network neighborhood algorithm is discussed in section 4. Experimental results are presented and discussed in section 5 with a sensitivity analysis on the proposed model's parameters. The paper is concluded in section 6 with future work on the proposed AIS model.

2 Data Clustering

Data clustering can be defined as the partitioning of patterns in a data set into different groups in such a way that patterns within the same group are more *similar* to patterns across different groups. Data clustering can formally be defined as follows [2, 22]:

Let P be the data set of patterns in N-dimensional space that needs to be clustered. Thus,

$$P = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_i, \ldots, \mathbf{p}_{I-1}, \mathbf{p}_I\}$$

where $\mathbf{p}_i$ is an N-dimensional *feature vector* (pattern) and I is the number of feature vectors. The partitioning of P into K clusters, $\{C_1, C_2, \ldots, C_K\}$, satisfies the following conditions:

1. $|C_k| \neq 0, k = 1, 2, \ldots, K$, meaning that clusters are not allowed to be empty;
2. $P = \cup_{k=1}^{K} C_k$, meaning that each feature vector is assigned to a cluster;
3. $|C_k \cap C_j| = 0, k \neq j$, meaning that each feature vector is assigned to only one cluster (in the case of *crisp* or *hard* clustering, i.e. *exclusive clustering*); and
4. $|C_k \cap C_j| > 0, k \neq j$, meaning that each feature vector is assigned to all clusters with a certain degree (in the case of *fuzzy* clustering, i.e. *overlapping clustering*).

Each of the formed clusters, $\{C_1, C_2, \ldots, C_K\}$, is respectively represented by a *centroid*, $\{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K\}$ [28]. The most commonly used measure of *similarity* (or *dissimilarity*) between data patterns is the Euclidean distance, defined as

$$\sigma_2(\mathbf{p}_i, \mathbf{p}_j) \quad = \quad \|\mathbf{p}_i - \mathbf{p}_j\|^2 \qquad (1)$$

Partitioning of these patterns maximizes a specific objective function such that the separation between clusters is maximized (*inter-cluster* distance) and the *compactness* of the clusters is minimized (*intra-cluster* distance). The *inter-cluster* and *intra-cluster* distances can be used to measure the quality of the clusters, as explained next.

2.1 Performance measures

The *compactness* of clusters is calculated as the average distance between the centroid of each cluster and the patterns within that cluster. The *compactness* of clusters is measured by the *intra-cluster* distance, calculated as

$$J_{intra} = \frac{\sum_{k=1}^{K} \sum_{\forall \mathbf{p} \in C_k} \sigma\left(\mathbf{p}, \mathbf{c}_k\right)}{|P|} \tag{2}$$

J_{intra} is minimized for more compact clusters. The separation between clusters is calculated as the average distance between the centroids of the clusters. The separation between clusters is measured by the *inter-cluster* distance, calculated as

$$J_{inter} = \frac{2}{K \times (K-1)} \sum_{k=1}^{K-1} \sum_{j=k+1}^{K} \sigma\left(\mathbf{c}_k, \mathbf{c}_j\right) \tag{3}$$

J_{inter} is maximized for more separated clusters. The number of clusters formed and the quality of these clusters needs to be evaluated. An approach to validate the number of clusters formed is to visually present the clustering results. For multidimensional problems where the number of dimensions is greater than three, visualization of the formed clusters becomes difficult [19, 27]. An alternative approach is to validate the clustered data set with a cluster validity index. A validity index which is based on the ratio of *intra-clustering* distance to the minimum *inter-clustering* distance was proposed by Ray and Turi [37] and is calculated as

$$Q_{ratio} = \frac{J_{intra}}{inter_{min}} \tag{4}$$

where $inter_{min}$ is calculated as

$$inter_{min} = \min_{\substack{k=1,\ldots,K-1 \\ j=k+1,\ldots,K}} \left\{\sigma\left(\mathbf{c}_k, \mathbf{c}_j\right)\right\} \tag{5}$$

J_{intra} is defined in eq. (2) and σ is the Euclidean distance as defined in eq. (1). The definition of $inter_{min}$ calculates the smallest distance between the centroids of the clusters to determine the smallest separation between clusters. Q_{ratio} needs to be minimized to have the optimal number of clusters (minimized J_{intra} and maximized $inter_{min}$). The next section discusses two partitional clustering algorithms which respectively optimize a specific objective function to partition a data set.

2.2 Classical data clustering models

This section discusses two *partitional* clustering methods which partitions a data set into a number of non-hierarchical clusters by means of optimizing a specific objective function. The two algorithms discussed next are the classical K-means clustering [11] and Clustering Particle Swarm Optimization (CPSO) [29] algorithms as implemented for the work in this paper.

K-means: K-means initializes K centroids, where K is the number of clusters into which a data set is partitioned. Based on a similarity measure (using eq. (1)), each feature vector in the data set is then assigned to only one of these centroids. A feature vector, $\mathbf{p}$, is assigned to a centroid, $\mathbf{c}$, if $\mathbf{p}$ is most similar to $\mathbf{c}$. Thus the subset of feature vectors assigned to a centroid forms a cluster. After each feature vector in the data set is assigned to a centroid, the centroid of each cluster is recalculated according to the feature vectors assigned to the cluster. The K-means clustering algorithm optimizes the *sum of squared distances* as objective function by minimizing the *inter-cluster* distance [20, 22]. The *stopping criterion* for K-means in this paper is based on a specified number of iterations, t_{max}.

Clustering Particle Swarm Optimization: Clustering Particle Swarm Optimization (CPSO) maintains a population or a swarm of *particles*, S. Each particle in the swarm represents a possible partitioning of the data set [29]. Thus, each particle represents K number of centroids, such that $N = K \times I$ where I is the number of features and N is the number of dimensions of the search space. A particle moves through the search space by adjusting its position towards its own *best* experienced solution and towards the *best* particle in the *neighborhood* (neighborhood with radius d [40]). In addition to the feature vector and *personal best position* contained by a particle, a particle also maintains its *current velocity* (which is based on the *inertia weight*, w, which is a fraction of the previous velocity and acceleration constants, c_1 and c_2). In order to limit the step size with which a particle's position is adjusted, the velocities can be clamped [9]. If a particle's velocity vector exceeds the specified maximum velocity vector, the particle's velocity vector is set to maximum velocity vector. The values of the maximum velocity vector are selected as a fraction, $\delta \in (0, 1]$, of the domain of each dimension of the search space. The quantization error proposed in [29], was used as the *fitness* of a particle.

3 The Natural Immune System

Different theories exist in the study of immunology regarding the functioning and organizational behavior between lymphocytes in response to encountered antigen [3, 24, 34]. One of these theories is the network theory, discussed in the next section.

3.1 The network theory

The network theory was first introduced by Jerne [24, 25]. In brief, when an antigen stimulates a lymphocyte (in this case a B-cell lymphocyte), the lymphocyte not only secretes antibodies to bind to the antigen but also generates mutated clones of itself in an attempt to have a higher binding affinity with the detected antigen. The former is known as affinity maturation and consists of two processes known as *somatic hyper-mutation* and *clonal selection* [3]. *Somatic hyper-mutation* is when a cloned lymphocyte is mutated to have a higher binding affinity with the detected antigen. The theory of lymphocyte cloning is known as *clonal selection* [3]. *Clonal selection* is the process of selecting those lymphocytes with the highest binding affinity with an antigen for cloning. The clonal proliferation of some lymphocytes results in the non-proliferation of other lymphocytes in the body. Lymphocytes (including the generated clones) which are not frequently stimulated by an antigen or which are non-proliferated will eventually be annihilated by the immune system.

The variable region of an antibody can be antigenic and invoke an immune response. Thus, the variable region of an antibody, responsible for binding to an antigen, has an antigenic profile. This antigenic profile is known as the *idiotype* of the antibody. The idiotype of an antibody can invoke an immune response for the creation of anti-idiotypic antibodies by a stimulated B-cell [25]. As illustrated in figure 1, the idiotopic profile of an antibody consists of multiple sites in the variable region of an antibody. These sites are known as *idiotopes.*

The network theory states that a lymphocyte is not only stimulated by an antigen, but can also be stimulated or suppressed by neighboring lymphocytes. Thus, whenever a lymphocyte reacts to the stimulation of an antigen, the secretion of antibodies and generation of mutated clones stim-

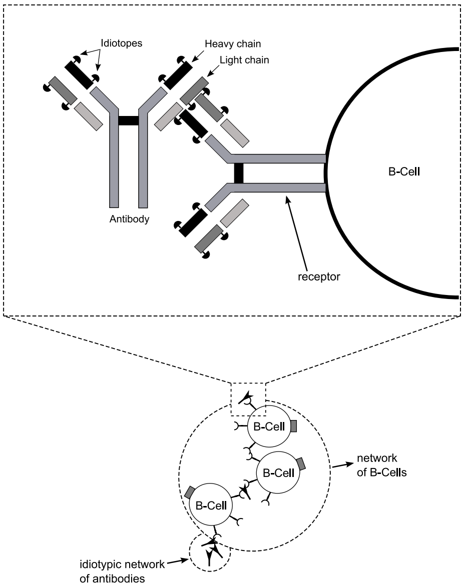


Figure 1: Idiotypic Network of Antibodies and B-cells

ulate the lymphocyte's immediate neighbors if the neighboring B-cells bind to the idiotopes of the produced antibodies or receptor of the stimulated B-cell lymphocyte. A neighboring lymphocyte can then in turn also react to the stimulation of the antigen-stimulated lymphocyte by generating mutated clones, stimulating or suppressing the next group of neighbors, etc.

3.2 Network based artificial immune system models

This section gives a brief introduction to the network based AIS models which are used in this paper. The interested reader is referred to the given references for more detail on the specific models.

Stable Memory Artificial Immune Network AIS: The Stable Memory Artificial Immune Network (SMAIN) AIS contains a population of artificial lymphocytes, $\mathcal{B}$, which is initialized with a cross section, $\mathcal{B}_{init}$, of the training data [33]. Each artificial lymphocyte (ALC) is initialized with R_{init} resources which decay at a rate, R_γ. The maximum resources that can be allocated by an ALC are R_{max}. All ALCs with a resource level less than the mortality threshold, R_Λ, are culled from the network. $R_k \in (0, 1)$ is a constant which is used to calculate an ALC's resource-level. Furthermore, cloning in SMAIN is only performed on the ALC with the closest distance to an antigen pattern if the measured distance is greater than the network affinity threshold (NAT). Whenever an antigen triggers cloning of an ALC, the antigen is initialized as a cloned ALC. Half of the parent ALC's resources is then assigned to the cloned ALC and the clone is integrated with the ALC population. There is no mutation operator on the clone. SMAIN generates stable memory networks which represents structures inherent in complex data sets. The final population of network ALCs is then clustered with a clustering technique such as hierarchical agglomerative clustering to determine the clusters in the data set.

Dynamic Weighted B-cell AIS (DWB): Nasraoui *et al.* proposed the DWB AIS model which can also be applied to the clustering of non-stationary data [30, 31, 32]. An artificial lymphocyte is known as a dynamic weighted B-cell (DWB-cell) since each training pattern is
grouped with all artificial lymphocytes to a certain degree of membership. The ALC population, $\mathcal{B}$, has a maximum of $\mathcal{B}_{max}$ ALCs and is initialized with the first $\mathcal{B}_{max}$ of incoming antigen training patterns. The membership function uses eq. (1) and an initial radius of influence, ϕ_{init}. A parameter, τ, controls the rate at which previously presented antigen patterns contribute to an ALC's membership function. The stimulation of an ALC is determined by weighting the membership with τ_α and τ_β, which respectively weights the co-stimulation and network suppression contribution to the stimulation of an ALC. After every A antigen patterns, the population of ALCs are compressed into
$k_{compress}$ ALC networks using K-means clustering. The DWB AIS model uses a cloning constant, k_{clone}, which determines the number of clones that needs to be generated for an activated ALC. Only mature and activated ALCs in the DWB model are cloned and the clones are mutated with a mutation rate, ς. The minimum threshold for an ALC to activate is m_{min} (membership function value greater than m_{min}) and an ALC is mature if its age is between a_{min} and a_{max}. An ALC's age is incremented if the ALC is not activated by an antigen pattern and reset to zero if activated. Whenever the maximum size, $\mathcal{B}_{max}$, of the network of ALCs has been reached, the ALCs are sorted in ascending order of their stimulation levels and starting from the top, the ALCs with the lowest stimulation levels are removed until

the size of the network is equal to the maximum size, $\mathcal{B}_{max}$. Whenever the immune network of ALCs cannot react to an antigen pattern, the specific antigen pattern is initialized as an ALC and added to the ALC population.

Opt-aiNet: De Castro and Von Zuben proposed a novel network based AIS model known as aiNet. The aiNet model evolves a population of linked memory ALCs through clonal selection [6, 8]. The ALCs are connected by weighted edges to form pairs in an ALC network. The weight value associated with each edge indicates the similarity between two ALCs. The aiNet model was adapted to solve multi-modal function optimization
problems and is known as *opt-aiNet* [4]. The opt-aiNet model maintains a dynamic population size and is capable of locating local optima solutions. The ALC population, $\mathcal{B}$, is initialized with $\mathcal{B}_{init}$ ALCs. The network affinity between two ALCs is calculated as the Euclidean distance, as defined in eq. (1). The fitness of an ALC, $\mathbf{b}$, is calculated using the fitness function, f, which is the objective that needs to be optimized. η ALC clones are generated for each $\mathbf{b}$. Each ALC clone is mutated proportionally to the normalized fitness, f^*, of its parent $\mathbf{b}$ as [4]

$$\mathbf{b}^{'} = \mathbf{b} + \left(\frac{1}{\beta}\right) \exp\left[-\left(1 - f^*\left(\mathbf{b}\right)\right)\right] N\left(0,1\right) \tag{6}$$

where $N\left(0,1\right)$ is a Gaussian random variable with zero mean and standard deviation of one. β controls the decay of the inverse exponential function. Cloning of the ALC population is done until the difference in average fitness of $\mathcal{B}$ is less than a pre-defined threshold $\epsilon_{fitness}$. After cloning, if the calculated network affinities between ALCs are below the network suppression threshold, $\epsilon_{network}$, the ALCs with the lowest fitness are removed from $\mathcal{B}$. A ratio, φ, (of the size of $\mathcal{B}$) of randomly generated ALCs is added to $\mathcal{B}$. In the context of data clustering as an optimization problem, each ALC in the population represents a possible partitioning of the data set (similar to CPSO). Thus, an ALC represents K centroids, one for each cluster. The objective function that needs to be optimized is the quantization error as defined in [29] and is thus the fitness function of the ALCs.

4 A Local Network Neighborhood Artificial Immune System

The co-operation and co-stimulation or suppression between lymphocytes to respond and adapt to invading antigens can result in the formation of lymphocyte network structures in the natural immune system, i.e. the network theory of immunology. An antigen stimulated lymphocyte not only secretes antibodies but also proliferates by generating mutated clones to adapt to the antigen structure. The proliferation of a lymphocyte stimulates the immediate neighboring lymphocytes,
which in turn might also proliferate to adapt to the antigen structure and stimulate neighboring lymphocytes. Thus, a network of lymphocytes *learns* the structure of an antigen by co-stimulating each other. The network topology of co-stimulated lymphocytes inspired the modeling of the local network neighborhood artificial immune system (LNNAIS), which was initially proposed by Graaff and Engelbrecht as a proof of concept [16]. The different parts of the LNNAIS algorithm are discussed in sections 4.1 to 4.4. The differences and similarities between existing network based AIS models and the proposed LNNAIS are discussed in section 4.5.

4.1 The algorithm

The proposed LNNAIS algorithm is given in pseudo code in Algorithm 1 [18] and consists of six high level steps to respond to an antigen/training pattern. These steps are:

1. Present an antigen to each ALC in the population and return the ALC with the highest calculated binding *affinity* with the antigen.

2. The returned highest affinity ALC reacts to the antigen pattern by initializing the antigen pattern as an antigen mutated clone and binds to the clone.

3. If the highest affinity ALC *activates*, the activated ALC spawns a mutated clone.

4. The spawned clone then binds to those antigen mutated clones of the activated ALC with which the spawned clone has a higher binding affinity than the activated ALC.

5. The mutated clone or activated ALC then *co-stimulates* ALCs which is within the *local neighborhood* of the activated ALC.

6. Co-stimulation of neighboring ALCs can result in co-suppression and/or the non-proliferation of other ALCs in the population.

The first three steps simulate the *affinity maturation* of a lymphocyte in the natural immune system. The first step models the *clonal selection* of the natural immune system. The antigen pattern selects the ALC with which the antigen has the highest binding affinity for cloning. The second step models the *proliferation* of a lymphocyte in the natural immune system. When a lymphocyte reaches a certain level of proliferation (clone size), the lymphocyte activates and spawns a mutated clone (*somatic hyper-mutation* in the third step). The fourth and fifth steps simulate the network theory of co-stimulation and/or suppression, and the final step the non-proliferation of other lymphocyte clones due to the proliferation of neighboring lymphocytes. The above high level steps are grouped into three phases, namely *react*, *adapt* and *suppress*. Each of these phases is explained next.

4.2 Reacting to an antigen

The high level steps of the *react* phase are basically the steps responsible for calculating the affinity levels between the ALCs in population $\mathcal{B}$ and an antigen, selecting the ALC with the highest affinity and proliferating the selected ALC. The sections to follow explain and define each of these aspects.

4.2.1 Calculating the affinity

The affinity between an antigen pattern, $\mathbf{a}$, and an ALC, $\mathbf{b}$, is known as the antigen affinity and is calculated as the Euclidean distance between $\mathbf{b}$ and $\mathbf{a}$. Euclidean distance is defined in eq. (1) and is also used to measure the network affinity between two ALCs. The affinity determines the binding strength between an ALC and an antigen pattern or neighboring ALC. Therefore, a lower Euclidean distance implies a higher affinity (stronger binding) between an ALC and an antigen pattern or neighboring ALC, and vice versa.

Algorithm 1 High Level LNNAIS Algorithm

```
Set the maximum size of the ALC population as $\mathcal{B}_{max}$
Initialize an empty set of ALCs as population $\mathcal{B}$
for each antigen, $\mathbf{a}_j \in \mathcal{A}$, at index position $j$ in $\mathcal{A}$ do
    if $|\mathcal{B}| \leq 0$ then
        Initialize a new ALC, $\mathbf{b}$, with the same structure as pattern $\mathbf{a}_j$
        $\mathcal{B} = \mathcal{B} \cup \mathbf{b}$
    end if
    Calculate the antigen affinity between $\mathbf{a}_j$ and each $\mathbf{b}_i \in \mathcal{B}$ using eq. (1)
    Select $\mathbf{b}_h \in \mathcal{B}$, at index $h$, as the ALC with highest calculated antigen
        affinity
    Proliferate $\mathbf{b}_h$ as discussed in section 4.2.2
    if $\mathbf{b}_h$ is activated ($|\mathcal{C}_h| > \epsilon_{clone}$) then
        Generate a mutated clone, $\mathbf{b}_h^{'}$, using eq. (10)
        Secrete an antibody, $\mathbf{b}^*$, as discussed in section 4.2.4
        Determine the local network neighborhood of $\mathbf{b}_h$ using eq. (11)
        Co-stimulate the local network neighborhood of $\mathbf{b}_h$ with $\mathbf{b}^*$, as discussed in section 4.3.3
    end if
end for
```

4.2.2 Proliferating the clonal selected ALC

The ALC with the highest binding affinity with an antigen pattern is selected as $\mathbf{b}_h$, where h is the index position of the selected ALC in $\mathcal{B}$. The antigen pattern $\mathbf{a}$ is then initialized as an antigen mutated clone $\mathbf{a}^{'}$. The antigen mutated clone $\mathbf{a}^{'}$ is grouped with $\mathbf{b}_h$ by inserting $\mathbf{a}^{'}$ at the first index position of the clonal set $\mathcal{C}_h$. Each ALC, $\mathbf{b}_i$, at index position i in $\mathcal{B}$, contains a set of antigen mutated clones, $\mathcal{C}_i$. Inserting an antigen mutated clone into $\mathcal{C}_i$ increases the clonal level of $\mathbf{b}_i$. An ALC activates when the clonal level, $|\mathcal{C}|$, of the ALC exceeds the clonal level threshold, ϵ_{clone}. Once an ALC is activated, the activated ALC generates a mutated ALC clone. When an antigen mutated clone is inserted at the first index of $\mathcal{C}$ and $|\mathcal{C}| > \epsilon_{clone}$, the antigen mutated clone at the last index position $|\mathcal{C}|$, is removed from $\mathcal{C}$. This gives more current antigen mutated clones a higher probability to survive and influence the generation of the mutated ALC clone. The sections to follow discuss different definitions used to generate a mutated ALC clone.

4.2.3 Normalizing the affinity of an antigen mutated clone

The normalized affinity between an antigen mutated clone, $\mathbf{a}^{'} \in \mathcal{C}_i$, and an ALC $\mathbf{b}_i$, is defined as [18]

$$\sigma^{*}\left(\mathbf{b}_i, \mathbf{a}^{'}, \mathcal{C}_i\right) = 1.0 - \frac{\sigma\left(\mathbf{b}_i, \mathbf{a}^{'}\right)}{\sigma_{max} + 1.0} \tag{7}$$

where

$$\sigma_{max} = max_{c=1,\ldots|\mathcal{C}_i|}\left\{\sigma\left(\mathbf{b}_i, \mathbf{a}_c^{'}\right)\right\} \tag{8}$$

and $\mathbf{a}_c^{'}$ is an antigen mutated clone at index position c in clonal set $\mathcal{C}_i$ of ALC $\mathbf{b}_i$. The affinity between an antigen mutated clone, $\mathbf{a}_c^{'} \in \mathcal{C}_i$, and an ALC, $\mathbf{b}_i$, is normalized with respect to the lowest affinity (highest Euclidean distance) in the set of antigen mutated clones, $\mathcal{C}_i$. A lower affinity between an antigen mutated clone and an ALC will result in a lower normalized affinity and vice versa. Thus the higher an ALC's affinity towards an antigen mutated clone, the more the ALC's clone will be mutated towards the antigen mutated clone [18], as explained in the next section.

4.2.4 Generating a mutated clone of an activated ALC

The vector difference between two vectors $\mathbf{q}$ and $\mathbf{r}$ is defined as:

$$\theta\left(\mathbf{r}, \mathbf{q}\right) = \mathbf{q} - \mathbf{r} \tag{9}$$

The above function returns a vector with the same number of attributes (components) as $\mathbf{q}$. These attributes are calculated by subtracting each attribute in $\mathbf{r}$ from the corresponding attribute in $\mathbf{q}$. The set of antigen mutated clones, $\mathcal{C}_i$, which is contained by an ALC $\mathbf{b}_i$ determines the mutated clone which will be generated when an ALC is activated. The mutated clone, $\mathbf{b}_i^{'}$, is calculated using [18]

$$\mathbf{b}_i^{'} = \mathbf{b}_i + \frac{\sum_{c=1}^{|\mathcal{C}_i|} \sigma^*\left(\mathbf{b}_i, \mathbf{a}_c^{'}, \mathcal{C}_i\right) \theta\left(\mathbf{b}_i, \mathbf{a}_c^{'}\right)}{\sum_{c=1}^{|\mathcal{C}_i|} \sigma^*\left(\mathbf{b}_i, \mathbf{a}_c^{'}, \mathcal{C}_i\right)} \tag{10}$$

In the above definition, the ALC clone is mutated more towards higher affinity antigen mutated clones in $\mathcal{C}_i$, since higher binding affinity antigen mutated clones with ALC $\mathbf{b}_i$ have a higher influence on the mutation of the clone in comparison with antigen mutated clones with a lower binding affinity.

4.2.5 Secreting an antibody for co-stimulation

The mutated ALC clone, $\mathbf{b}_i^{'}$, inherits all the antigen mutated clones of the parent ALC, $\mathbf{b}_i$, with which the mutated ALC clone has a higher affinity than the parent ALC. The inherited antigen mutated clones are added to the clonal set of $\mathbf{b}_i^{'}$ (bind to $\mathbf{b}_i^{'}$). If the clonal level of the mutated ALC clone is higher than half of the clonal level of the parent ALC, the parent ALC $\mathbf{b}_i$ is added as an antigen mutated clone to the clonal set of $\mathbf{b}_i^{'}$. The parent ALC is then replaced by the mutated ALC clone in $\mathcal{B}$ and secreted as a co-stimulating antibody to neighboring ALCs. The parent ALC is suppressed if the clonal level of the mutated ALC clone is less than half of the clonal level of the parent ALC. A parent ALC, $\mathbf{b}_i$, is suppressed by removing all of the antigen mutated clones in $\mathcal{C}_i$. Suppression prevents frequently activated ALCs from dominating the population. The mutated ALC clone is then inserted into $\mathcal{C}_i$ of the parent ALC $\mathbf{b}_i$ to co-stimulate the parent ALC and also preserve the memory of the antigen structure. The mutated ALC clone is secreted as a co-stimulating antibody to neighboring ALCs. The following section discusses the co-stimulation of neighboring ALCs within a local network neighborhood.

4.3 Adapting the ALCs in a local network neighborhood

The co-stimulating antibody which is secreted during the activation of a proliferated ALC is presented to the immediate ALC neighbor(s) in the local network neighborhood of the activated ALC. The neighboring ALCs within a local network neighborhood adapt to the antibody as it would react to an antigen (as explained in section 4.2). The following sections discuss the manner in which a local network neighborhood of an activated ALC is determined.

4.3.1 Determining the local network neighborhood of an activated ALC

An ALC's neighborhood, $\mathcal{N}$, is determined by a network neighborhood window of size, ρ, and the highest average network affinity between the potential neighboring ALCs. The neighborhood, $\mathcal{N}_{i,\rho}$, of an ALC, $\mathbf{b}_i \in \mathcal{B}$, is defined as [18]

$$\mathcal{N}_{i,\rho} = \left\{ \forall \mathbf{b}_j \in \mathcal{B} : \min_{j=i-(\rho-1),\ldots,i} \{\mu\left(j, j+(\rho-1)\right)\} \right\} \tag{11}$$

where

$$\rho \leq |\mathcal{B}| \tag{12}$$

$$\mathcal{N}_{i,\rho} \subseteq \mathcal{B} \tag{13}$$

$$\mathbf{b}_i \in \mathcal{N}_{i,\rho} \tag{14}$$

and μ calculates the average network affinity between ALCs in the population from index position i to $i+(\rho-1)$ and is defined in section 4.3.2. The above definition is a network window of size ρ which starts at position $i-(\rho-1)$, sliding over the ALC population in search of the highest average network affinity (minimum average distance) [18]. Figure 2 illustrates a local network neighborhood where $\rho = 5$ and the network with the highest average network affinity starts at index position $h-2$.

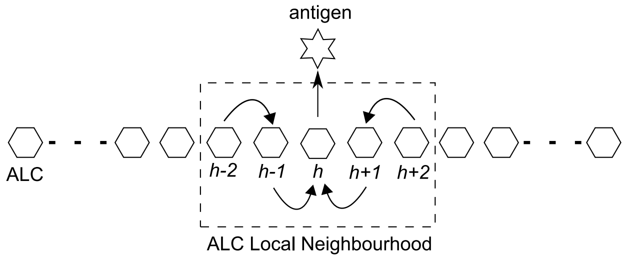


Figure 2: Adapting an ALC network neighborhood

4.3.2 Calculating the average network affinity in a local network neighborhood

The average network affinity level of a network of ALCs starting at index position x to y, is defined as [18]

$$\mu(x, y) = \frac{\sum_{i=x}^{y-1} \sigma(\mathbf{b}_i, \mathbf{b}_{i+1})}{y - x} \tag{15}$$

where σ is the Euclidean distance (as defined in eq. (1)).

4.3.3 Co-stimulating the local network neighborhood

The neighboring ALCs within a local network neighborhood, $\mathcal{N}_{i,\rho}$, adapts to the secreted antibody of its predecessor in the neighborhood. Figure 2 illustrates a local network neighborhood with $\rho = 5$ adapting to an antigen. In this figure, ALC $\mathbf{b}_h$, is selected by the antigen for cloning and proliferation (as explained in section 4.2.2). As a result of proliferating $\mathbf{b}_h$, the ALC became active ($|\mathcal{C}_h| > \epsilon_{clone}$) and secreted an antibody for co-stimulation of the immediate neighbors of $\mathbf{b}_h$. The immediate neighbors of $\mathbf{b}_h$ at indices $h-1$ and $h+1$ react to the secreted antibody by adding the clonal set of the antibody to $\mathcal{C}_{h-1}$ and $\mathcal{C}_{h+1}$, respectively. If either or both of the neighboring ALCs, $\mathbf{b}_{h-1}$ and $\mathbf{b}_{h+1}$ becomes activated, either or both will secrete antibodies (as explained in section 4.2.4), which will co-stimulate their immediate ALC neighbors at indices $h-2$ and $h+2$, respectively. If a neighboring ALC is not activated by the co-stimulation of a predecessor's antibody, the antibody is inserted into the local network at the index of the neighboring ALC, increasing the population size through *clonal expansion* (discussed in section 4.3.4). The neighboring ALCs with the highest network affinity in

$$i^*\left(\mathbf{b}^*,\mathbf{b}_i\right)=\begin{cases} i & \text{if } \frac{\sigma(\mathbf{b}^*,\mathbf{b}_{i-1})+\sigma(\mathbf{b}^*,\mathbf{b}_i)}{2} < \frac{\sigma(\mathbf{b}^*,\mathbf{b}_i)+\sigma(\mathbf{b}^*,\mathbf{b}_{i+1})}{2} \\ i+1 & \text{otherwise} \end{cases} \tag{16}$$

the population, which are not within the local network neighborhood, are merged to stabilize the population size. Merging of ALCs simulate the non-proliferation of other ALC clones in the population (discussed in section 4.4). The process of co-stimulation continues until the ALCs on the boundary of the local network neighborhood are co-stimulated or until a neighboring ALC is not activated by the co-stimulation of a predecessor's antibody. Algorithm 2 lists the pseudo code for adapting the ALCs in a local network neighborhood.

4.3.4 Clonal expansion of a local network neighborhood

A local network neighborhood is clonally expanded
whenever a neighboring ALC, $\mathbf{b}_i$, is not activated by the co-stimulation of a predecessor's secreted antibody. The secreted antibody, $\mathbf{b}^*$, is inserted at position i^* which is defined in eq. (16). The secreted antibody is inserted at the index position where the average network affinity is the highest between the secreted antibody and its potential neighboring ALCs.

4.4 Suppression (Non-proliferation) of the ALC population

The maximum ALC population size, $\mathcal{B}_{max}$, is exceeded whenever clonal expansion occurs in a local network neighborhood. Therefore, the non-proliferation and suppression of other ALCs in the population keeps the size of the ALC population stable. Non-proliferation (suppression) is simulated by merging two ALCs in the population which is not within the clonally expanded local network neighborhood, and which has the highest network affinity in the population.

Algorithm 2 Adapting the Neighborhood, $\mathcal{N}_{h,\rho}$, to an Activated ALC, $\mathbf{b}_h$

```
Let $\mathbf{b}^*$ be the secreted antibody of the activated ALC $\mathbf{b}_h$
$l = h - 1; r = h + 1$
Let $\mathbf{b}^*_l = \mathbf{b}^*$ and $\mathbf{b}^*_r = \mathbf{b}^*$ be the secreted antibodies for co-stimulation of neighboring ALCs $\mathbf{b}_l$ and $\mathbf{b}_r$, respectively
Activated=true
for $\mathbf{b}_l \in \mathcal{N}_{h,\rho}$ and Activated do
    Add antigen mutated clones of $\mathbf{b}^*_l$ to clonal set $\mathcal{C}_l$ of neighboring ALC $\mathbf{b}_l$
    if $\mathbf{b}_l$ is activated (i.e. $|\mathcal{C}_l| > \epsilon_{clone}$) then
        Generate a mutated clone, $\mathbf{b}^{'}_l$, using eq. (10)
        Secrete an antibody $\mathbf{b}^*_l$ from $\mathbf{b}_l$, as discussed in section 4.2.4
        $l = l - 1$
    else
        Activated=false
        Insert $\mathbf{b}^*_l$ into $\mathcal{N}_{h,\rho}$ at position $i^*(\mathbf{b}^*_l, \mathbf{b}_l)$ (as defined in eq. (16))
        Merge two ALCs in the population with the highest network affinity, as discussed in section 4.4
    end if
end for
Activated=true
for $\mathbf{b}_r \in \mathcal{N}_{h,\rho}$ and Activated do
    Add antigen mutated clones of $\mathbf{b}^*_r$ to clonal set $\mathcal{C}_r$ of neighboring ALC $\mathbf{b}_r$
    if $\mathbf{b}_r$ is activated (i.e. $|\mathcal{C}_r| > \epsilon_{clone}$) then
        Generate a mutated clone, $\mathbf{b}^{'}_r$, using eq. (10)
        Secrete an antibody $\mathbf{b}^*_r$ from $\mathbf{b}_r$, as discussed in section 4.2.4
        $r = r + 1$
    else
        Activated=false
        Insert $\mathbf{b}^*_r$ into $\mathcal{N}_{h,\rho}$ at position $i^*(\mathbf{b}^*_r, \mathbf{b}_r)$ (as defined in eq. (16))
        Merge two ALCs in the population with the highest network affinity, as discussed in section 4.4
    end if
end for
```

4.5 Similarities and differences with other network based AIS models

This section discusses some of the differences and similarities between the proposed algorithm and existing network based AIS models.

4.5.1 Training data

Although the proposed LNNAIS model can be trained on normalized data, the normalization of training data is not a prerequisite for LNNAIS. Similar to other network based AIS models, LNNAIS sees all training patterns as antigen patterns.

4.5.2 Population of ALCs

The population of ALCs can be initialized with a number of randomly initialized ALCs or a number of randomly selected training patterns as ALCs, i.e. a cross section of the training data is used to initialize the ALCs. The initial population of ALCs in LNNAIS is an empty set. The first randomly selected training pattern is initialized as an ALC and added to the population of ALCs. This concept is known as *dendritic injection* in the natural immune system. The population of ALCs are grown and pruned in LNNAIS. The *growth* of the population of ALCs in LNNAIS is based on the process of *affinity maturation*. When an activated ALC of a local network neighborhood does not adapt to the presented antigen pattern, the clonal level of the ALC is penalized and a mutated clone of the ALC is inserted into the local network of ALCs.

4.5.3 ALC presentation and affinity measurement

An ALC in LNNAIS is presented by a continuous valued array with the same dimension as the antigen patterns in the training set, as is the case for other network based AIS models. The affinity between an antigen pattern and an ALC is measured using the Euclidean distance as defined in section 4.2.1. The affinity between two ALCs, referred to as network affinity, is also measured using the Euclidean distance. Some of the existing network based AIS models also measure antigen and network affinity using Euclidean distance. The difference between LNNAIS and the existing network based AIS models is that LNNAIS has no threshold to determine whether two

ALCs are linked to form a network. LNNAIS introduces a new concept of an ALC network neighborhood size, as defined in section 4.3.1 and proposed by Graaff and Engelbrecht [16].

4.5.4 Learning the antigen structure

Another similarity between existing network based AIS models and the proposed LNNAIS is that some ALCs are cloned and mutated to adapt to antigen patterns. LNNAIS also models the process of *affinity maturation* to introduce new ALCs into the population as discussed in section 4.3.3. LNNAIS also models the non-proliferation of ALCs, as discussed in section 4.3.3. The difference between LNNAIS and existing network based AIS models is that *expansion* of the ALC population is done on a per local network neighborhood bases. LNNAIS models the *idiotopic network theory* of ALCs. This means that the insertion of new ALCs into a population will be done within a local network neighborhood (as discussed in section 4.3.3). Non-proliferation on the other hand is only done on ALCs which do not form part of the *activated* local network neighborhood. This means that only ALCs outside a network neighborhood will be non-proliferated in the ALC population (as discussed in section 4.3.3). This approach penalizes the population of ALCs by *non-proliferating* the population but also reinforces the network neighborhood by *clonal expansion*.

4.5.5 Determining the number of clusters

The number of ALC networks formed in existing network based AIS models represents potential clusters in the data set. In most of the existing network based AIS models the number of ALC networks in a population is determined by a network affinity threshold or a hybrid-approach is taken by clustering the ALC population into sub-nets (as discussed in section 3.2). The thresholding technique uses a proximity matrix of network affinities between the ALCs in the population. The ALCs with a network affinity below the threshold value are allowed to be linked and form networks. Therefore the specified value of the network affinity threshold determines the number of ALC networks and it can be a formidable task to specify the correct network affinity threshold to obtain the correct or required number of clusters. A potential drawback of the hybrid-approach is that the clusters (sub-nets) might contain ALCs which do not have a *good* or generic representation of the data.

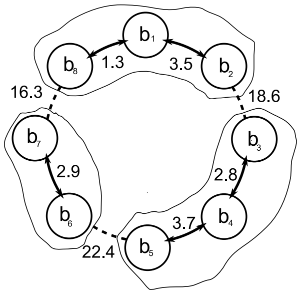


Figure 3: Determining the Number of Clusters in LNNAIS [18]

The proposed LNNAIS model has the advantage that an ALC can only link to its immediate neighbors to form an ALC network. This is due to the network topology and an index-based neighborhood technique. Therefore, there is no need for a network affinity threshold and/or a proximity matrix of network affinities to determine the number of ALC networks in LNNAIS. It is also not necessary to follow a hybrid-approach of clustering the ALC population. Determining the number of clusters in LNNAIS is explained next. In order to obtain a specified number of clusters, K, the network affinities between neighboring ALCs in the population need to be calculated. The boundaries of each cluster are then determined by pruning the network links between the K lowest calculated network affinities. Figure 3 illustrates this technique where $K = 3$ [18]. The edges between ALCs have an associated network affinity. The K edges that form the boundaries between the ALCs (dotted lines) have the lowest network affinity in the ALC population, i.e. highest Euclidean distance. The centroid of each of the formed ALC networks is calculated as the mean vector of the ALCs in the network.

4.5.6 The number of parameters

Focusing on existing network based AIS models which are used in this paper; there is also a significant difference in the number of parameters that need to be specified for each of the models. The DWB model has a total of 12 parameters, SMAIN has a total of seven parameters and Opt-aiNet a total of six parameters. The proposed LNNAIS model has only three parameters which are the maximum population size, B_{max}, the neighboring radius, ρ, and the activation level for ALC cloning, ϵ_{clone}.

5 Experimental Results and Analysis

This section discusses and compares the clustering results obtained by K-means, CPSO, SMAIN, DWB, Opt-aiNet and LNNAIS. Furthermore, a sensitivity analysis of LNNAIS is done on the different data sets.

5.1 Data clustering problems

Table 1 lists the selection of data sets used to benchmark the clustering performance and quality of the proposed LNNAIS model against the clustering quality of existing clustering methods like K-means clustering and CPSO (as discussed in section 2.2) and network based AIS models for data clustering like SMAIN, DWB-AIS and Opt-aiNet (as discussed in section 3.2). The characteristics of each data set are also listed in the table. These are the number of patterns in the data set ($|P|$), the number of features per pattern in the data set (N - number of dimensions), the maximum distance between the patterns in the data set (σ_{max}), the number of clusters selected for partitioning the data set (K) and whether there are any overlapping patterns in the data set. The two spiral, hepta, engytime, chainlink and target data sets are part of a fundamental clustering problems suite [23]. The other data sets were collected from the UCI Machine Learning repository [1].

The data sets in table 1 can be categorized into four groups.

- Group 1 (small number of features / small number of patterns): The data sets within this group have a small number of features and a small number of patterns. The iris data set, two spiral problem and hepta data set form part of this group. All of these data sets have less than 500 patterns and less than five features per pattern.

- Group 2 (small number of features / large number of patterns): The data sets within this group also have a small number of features but a larger number of patterns in comparison to the data sets in group 1. The engytime data set, chainlink data set and the target data set (to a lesser extent) form part of this group. All of these data sets have more than 500 patterns and less than five features per pattern.

- Group 3 (large number of features / small number of patterns): This group contains data sets with a larger number of features in comparison

Table 1: List of Eleven Benchmarking Data Sets for Clustering

Data set name	$\|P\|$	N	σ_{max}	K	Overlap?
Iris	150	4	7.7	3	Y
Two Spiral	190	2	3.045	12	Y
Hepta	212	3	13.383	7	N
Engytime	4096	2	14.806	2	Y
Chainlink	1000	3	4.383	6	Y
Target	770	2	8.627	5	Y (outliers)
Ionosphere	351	34	11.358	2	Y
Glass	214	9	16.449	6	Y
Image Segmentation	2310	19	1775.117	7	Y
Spambase	4601	57	18758.75	2	Y
Letter Recognition	20000	16	60	26	Y

to groups 1 and 2, but a small number of patterns. The ionosphere data set and the glass data set form part of this group and both have less than 500 patterns, with each pattern having more than eight features.

- Group 4 (large number of features / large number of patterns): The last group contains data sets with a larger number of features (compared to groups 1 and 2) and a larger number of patterns (compared to groups 1 and 3). The image segmentation data set, spambase data set and letter recognition data set form part of this group. All of these data sets have more than 500 patterns and more than eight features.

Taken as a whole, the listed data sets in table 1 represent a good distribution of data clustering problems with the number of patterns in the range $[150, 20000]$ and the number of features in the range $[2, 57]$. All the data sets have overlapping patterns except the hepta data set. The target data set also contains outlier patterns.

5.2 Experimental setup and methodology

All experimental results in this paper are averages taken over 50 runs, unless stated otherwise. The stopping criteria for all algorithms was set to 1000 iterations ($t_{max} = 1000$). Populations/Swarms in the respective algorithms were initialized by randomly selecting patterns from the data set. The patterns in a data set were randomly presented to each model. None of the data sets were normalized for training. Tables 2,3,4,5 and 6 summarize the parameter values used by the respective algorithms for each data set. All parameter values for the respective algorithms were found empirically to deliver the best performance for clustering the applicable data set. The Q_{ratio} validity index, intra error distance (J_{intra}) and inter error distance (J_{inter}) are used as performance measures to determine the clustering quality of the different models. These clustering performance measures were discussed in section 2.1.

The following sections investigate whether there is a difference between the clustering quality, Q_{ratio}, of two models for a specific data set or not. The hypothesis is defined as

- *Null* hypothesis, H_0: There is no difference in Q_{ratio}.
- *Alternative* hypothesis, H_1: There is a difference in Q_{ratio}.

The above hypothesis was tested with a non-parametric Mann-Whitney U hypothesis test (0.95 confidence interval, i.e. $\alpha = 0.05$) between the clustering quality of LNNAIS and the clustering quality of each of the other models. The result is statistical significant if the calculated probability (p-value is the probability of H_0 being true) is less than α. The results for each data set group are discussed next.

5.3 Testing for statistical significance - data group 1

Table 7 summarizes the results obtained for data group 1 using the applicable parameter values in tables 2-6 for each of the data sets (results with the lowest average Q_{ratio} are shown in boldface). The corresponding statistical hypothesis tests between LNNAIS and the remaining models for each of the data sets in group 1 are summarized in table 8 (based on the clustering quality, Q_{ratio}).

Table 2: DWB Parameter Values

Data set	K	$\mathcal{B}_{max}$	ϕ_{init}	m_{min}	A	a_{min}	a_{max}	k_{clone}	ς	τ	τ_α	τ_β	$k_{compress}$
Iris	3	39	0.362	0.087	37	49	49	1.688	0.24	14	5	10	6
Two Spiral	12	46	0.668	0.025	55	6	6	2.438	0.913	12	13	13	5
Hepta	7	47	0.959	0.959	78	35	54	1.625	0.592	1	6	15	2
Engytime	2	39	0.485	0.209	24	24	86	3.188	0.852	6	6	1	4
Chainlink	6	40	0.592	0.102	41	72	91	2.125	0.714	7	11	2	2
Target	5	47	0.554	0.982	36	62	62	3.844	0.89	13	12	11	3
Ionosphere	2	17	0.561	0.929	44	7	68	1.25	0.929	5	7	9	3
Glass	6	46	0.845	0.018	13	11	11	4.031	0.569	2	5	9	7
Image Segmentation	7	47	0.201	0.538	16	2	2	2.906	0.477	12	14	7	1
Spambase	2	46	0.27	0.546	71	9	9	4.562	0.025	3	8	4	4
Letter Recognition	26	45	0.148	0.423	9	27	46	3.062	0.148	8	10	13	3

Table 3: CPSO Parameter Values

Data set	K	$\|S\|$	d	w	c_1	c_2	δ
Iris	3	6	3	0.82	1.33	1.218	0.301
Two Spiral	12	9	4	0.558	0.656	1.94	0.326
Hepta	7	44	4	0.697	1.696	0.963	0.62
Engytime	2	63	11	0.641	0.719	0.156	0.359
Chainlink	6	11	5	0.234	0.656	1.969	0.266
Target	5	23	2	0.789	0.422	1.658	0.258
Ionosphere	2	45	8	0.683	1.518	1.207	0.961
Glass	6	13	6	0.914	1.344	1.246	0.115
Image Segmentation	7	10	5	0.77	0.875	1.545	0.312
Spambase	2	42	18	0.812	0.125	1.152	0.938
Letter Recognition	26	49	3	0.836	0.828	1.641	0.055

Table 4: SMAIN Parameter Values

Data set	K	$\mathcal{B}_{init}$	R_γ	R_Λ	NAT	R_k	R_{max}	R_{init}
Iris	3	0.25	0.836	3	1.115	0.422	238	37
Two Spiral	12	0.182	0.516	91	0.039	0.656	975	92
Hepta	7	0.191	0.938	38	0.259	0.375	900	88
Engytime	2	0.019	0.672	35	2.322	0.469	725	36
Chainlink	6	0.2	0.859	23	0.038	0.094	425	91
Target	5	0.049	0.824	22	0.077	0.852	819	31
Ionosphere	2	0.157	0.637	34	0.099	0.727	319	68
Glass	6	0.157	0.637	34	0.015	0.727	319	68
Image Segmentation	7	0.29	0.926	2	24.618	0.898	281	76
Spambase	2	0.123	0.805	33	6.571	0.359	388	43
Letter Recognition	26	0.051	0.93	24	8.595	0.109	988	68

Table 5: Opt-aiNET Parameter Values

Data set	K	$\mathcal{B}_{init}$	η	$\epsilon_{network}$	$\epsilon_{fitness}$	φ	β
Iris	3	44	10	0.186	1.317	0.131	0.356
Two Spiral	12	14	1	0.324	0.902	0.219	0.169
Hepta	7	39	1	0.297	1.54	0.491	0.459
Engytime	2	29	2	0.037	0.412	0.403	0.322
Chainlink	6	7	22	0.178	0.723	0.306	0.283
Target	5	12	3	0.362	1.109	0.338	0.412
Ionosphere	2	28	3	0.477	1.97	0.294	0.144
Glass	6	35	1	0.155	0.961	0.456	0.431
Image Segmentation	7	14	5	0.021	1.184	0.316	0.134
Spambase	2	6	5	0.32	1.985	0.409	0.191
Letter Recognition	26	45	2	0.416	1.258	0.444	0.394

Table 6: LNNAIS Parameter Values

Data set	K	$\mathcal{B}_{max}$	ρ	ϵ_{clone}
Iris	3	14	3	8
Two Spiral	12	39	3	6
Hepta	7	29	3	6
Engytime	2	10	3	22
Chainlink	6	24	3	8
Target	5	28	3	6
Ionosphere	2	10	3	17
Glass	6	24	3	8
Image Segmentation	7	20	2	27
Spambase	2	10	5	22
Letter Recognition	26	104	3	10

Table 7: Descriptive Statistics: Data Group 1

Data set			
Algorithm	J_{intra}	J_{inter}	Q_{ratio}
Iris			
K-means	0.689	3.269	0.509
	(± 0.073)	(± 0.201)	(± 0.268)
CPSO	0.725	2.964	0.658
	(± 0.089)	(± 0.201)	(± 0.354)
SMAIN	0.766	3.705	**0.295**
	(± 0.041)	(± 0.207)	(± 0.021)
DWB	0.753	3.103	0.547
	(± 0.152)	(± 0.282)	(± 0.304)
Opt-aiNet	0.887	2.977	0.882
	(± 0.021)	(± 0.095)	(± 0.168)
LNNAIS	0.738	3.546	0.333
	(± 0.054)	(± 0.309)	(± 0.048)
Two Spiral			
K-means	0.212	1.014	0.521
	(± 0.005)	(± 0.021)	(± 0.102)
CPSO	0.251	0.829	1.648
	(± 0.025)	(± 0.079)	(± 0.978)
SMAIN	0.213	1.096	**0.433**
	(± 0.004)	(± 0.013)	(± 0.015)
DWB	0.241	0.988	1.094

Continued on next page

Algorithm	J_{intra}	J_{inter}	Q_{ratio}
	(± 0.010)	(± 0.065)	(± 0.501)
Opt-aiNet	0.279	0.813	2.740
	(± 0.027)	(± 0.105)	(± 3.020)
LNNAIS	0.233	1.030	0.847
	(± 0.009)	(± 0.041)	(± 0.296)
Hepta			
K-means	0.976	4.041	0.999
	(± 0.232)	(± 0.147)	(± 0.465)
CPSO	0.893	3.930	1.095
	(± 0.355)	(± 0.344)	(± 1.748)
SMAIN	0.641	4.147	**0.219**
	(± 0.001)	(± 0.005)	(± 0.001)
DWB	1.187	3.990	1.254
	(± 0.260)	(± 0.238)	(± 0.618)
Opt-aiNet	1.179	3.681	1.643
	(± 0.462)	(± 0.499)	(± 1.353)
LNNAIS	0.748	4.140	0.345
	(± 0.102)	(± 0.099)	(± 0.206)

The Mann-Whitney U statistical hypothesis test accepts H_0 that the means are the same at a 0.05 level of significance between LNNAIS and Opt-aiNet and between LNNAIS and CPSO for data set hepta. The remainder of the Mann-Whitney U statistical hypothesis tests showed a significant difference in performance between LNNAIS and the other clustering algorithms. LNNAIS tends to deliver clusters of a higher quality when compared to K-means, CPSO, DWB and Opt-aiNet for data sets iris and hepta. Although SMAIN tends to deliver clusters of a higher quality when compared to LNNAIS for all data sets in group 1, LNNAIS delivers more compact clusters for the iris data set. Also, K-means tends to deliver clusters of a higher quality for data set two spiral (refer to table 7). SMAIN tends to find clusters in the data sets of group 1 with a higher quality, followed by LNNAIS.

5.4 Testing for statistical significance - data group 2

The results obtained for data group 2 with the applicable parameter values in tables 2-6 are summarized in table 9. The Mann-Whitney U statistical hypothesis test accepts H_0 that the mean clustering quality, Q_{ratio}, are the

Table 8: Statistical Hypothesis Testing between LNNAIS and Other Models based on Q_{ratio}: Data Group 1 ($\alpha = 0.05$; with continuity correction; unpaired; non-directional)

Data set	Algorithm	z	p	Outcome
Iris	K-means	4.539	< 0.001	Reject H_0
	CPSO	5.958	< 0.001	Reject H_0
	DWB	5.115	< 0.001	Reject H_0
	SMAIN	3.726	< 0.001	Reject H_0
	Opt-aiNet	6.646	< 0.001	Reject H_0
Two Spiral	K-means	5.773	< 0.001	Reject H_0
	CPSO	4.361	< 0.001	Reject H_0
	DWB	2.21	0.027	Reject H_0
	SMAIN	6.646	< 0.001	Reject H_0
	Opt-aiNet	6.246	< 0.001	Reject H_0
Hepta	K-means	3.726	< 0.001	Reject H_0
	CPSO	1.331	0.183	Accept H_0
	DWB	5.892	< 0.001	Reject H_0
	SMAIN	6.646	< 0.001	Reject H_0
	Opt-aiNet	1.804	0.071	Accept H_0

same between LNNAIS and DWB for data set chainlink; and rejects H_0 for all other cases (as summarized in table 10).

Table 9: Descriptive Statistics: Data Group 2

Data set			
Algorithm	J_{intra}	J_{inter}	Q_{ratio}
Engytime			
K-means	1.431	2.998	0.477
	(± 0.000)	(± 0.000)	(± 0.000)
CPSO	1.435	2.935	0.489
	(± 0.001)	(± 0.012)	(± 0.002)
SMAIN	2.097	5.975	**0.355**
	(± 0.103)	(± 0.670)	(± 0.039)
DWB	1.599	3.057	0.540
	(± 0.120)	(± 0.526)	(± 0.115)
Opt-aiNet	1.435	2.932	0.490
	(± 0.001)	(± 0.025)	(± 0.004)
LNNAIS	1.944	4.557	0.438
	(± 0.281)	(± 1.043)	(± 0.069)
Chainlink			
K-means	0.488	1.550	0.517
	(± 0.006)	(± 0.049)	(± 0.031)
CPSO	0.592	1.412	1.092
	(± 0.053)	(± 0.150)	(± 0.667)
SMAIN	0.487	1.643	**0.471**
	(± 0.007)	(± 0.039)	(± 0.023)
DWB	0.538	1.506	0.751

Continued on next page

Algorithm	J_{intra}	J_{inter}	Q_{ratio}
	(± 0.025)	(± 0.074)	(± 0.320)
Opt-aiNet	0.646	1.363	1.352
	(± 0.059)	(± 0.185)	(± 0.554)
LNNAIS	0.535	1.493	0.640
	(± 0.021)	(± 0.116)	(± 0.118)
Target			
K-means	0.544	2.393	0.337
	(± 0.030)	(± 0.244)	(± 0.032)
CPSO	0.749	2.340	1.133
	(± 0.077)	(± 0.556)	(± 0.578)
SMAIN	1.008	5.794	**0.238**
	(± 0.000)	(± 0.000)	(± 0.001)
DWB	0.649	2.058	0.752
	(± 0.059)	(± 0.319)	(± 0.285)
Opt-aiNet	0.792	2.706	1.750
	(± 0.050)	(± 0.494)	(± 1.491)
LNNAIS	0.804	2.985	0.559
	(± 0.124)	(± 0.525)	(± 0.155)

Referring to table 9, LNNAIS tends to deliver clusters of a higher quality when compared to CPSO, DWB and Opt-aiNet for all data sets in group 2. K-means tends to deliver clusters of a higher quality when compared to LNNAIS for data sets chainlink and target but of lower quality for data set engytime. SMAIN also tends to deliver clusters of a higher quality for all data sets in group 2, followed by LNNAIS.

5.5 Testing for statistical significance - data group 3

The results of the Mann-Whitney U statistical hypothesis test accepts H_0 that the mean clustering quality, Q_{ratio}, are the same between LNNAIS and DWB, and LNNAIS and CPSO for data set ionosphere and rejects H_0 for all other cases (as summarized in table 11). LNNAIS tends to deliver clusters of a higher quality for all data sets in group 3 when compared to K-means, CPSO and DWB (refer to table 12). However, SMAIN and Opt-aiNet tend to deliver clusters of a higher quality for data set ionosphere when compared to cluster quality of LNNAIS. SMAIN also tend to deliver clusters of a higher quality for the data sets in group 3, followed by LNNAIS. LNNAIS does

Table 10: Statistical Hypothesis Testing between LNNAIS and Other Models based on Q_{ratio}: Data Group 2 ($\alpha = 0.05$; with continuity correction; unpaired; non-directional)

Data set	Algorithm	z	p	Outcome
Engytime	K-means	3.097	0.002	Reject H_0
	CPSO	3.4	< 0.001	Reject H_0
	DWB	3.888	< 0.001	Reject H_0
	SMAIN	4.931	< 0.001	Reject H_0
	Opt-aiNet	3.4	< 0.001	Reject H_0
Chainlink	K-means	4.886	< 0.001	Reject H_0
	CPSO	3.748	< 0.001	Reject H_0
	DWB	0.85	0.395	Accept H_0
	SMAIN	6.32	< 0.001	Reject H_0
	Opt-aiNet	5.759	< 0.001	Reject H_0
Target	K-means	6.513	< 0.001	Reject H_0
	CPSO	4.517	< 0.001	Reject H_0
	DWB	2.964	0.003	Reject H_0
	SMAIN	6.646	< 0.001	Reject H_0
	Opt-aiNet	4.657	< 0.001	Reject H_0

however deliver more compact clusters than SMAIN for the glass data set.

Table 12: Descriptive Statistics: Data Group 3

Data set			
Algorithm	J_{intra}	J_{inter}	Q_{ratio}
Ionosphere			

Continued on next page

Table 11: Statistical Hypothesis Testing between LNNAIS and Other Models based on Q_{ratio}: Data Group 3 ($\alpha = 0.05$; with continuity correction; unpaired; non-directional)

Data set	Algorithm	z	p	Outcome
Ionosphere	K-means	2.24	0.025	Reject H_0
	CPSO	1.833	0.067	Accept H_0
	DWB	1.582	0.114	Accept H_0
	SMAIN	6.646	< 0.001	Reject H_0
	Opt-aiNet	3.837	< 0.001	Reject H_0
Glass	K-means	4.664	< 0.001	Reject H_0
	CPSO	6.513	< 0.001	Reject H_0
	DWB	6.291	< 0.001	Reject H_0
	SMAIN	4.916	< 0.001	Reject H_0
	Opt-aiNet	6.646	< 0.001	Reject H_0

Algorithm	J_{intra}	J_{inter}	Q_{ratio}
K-means	2.302 (± 0.125)	3.192 (± 0.486)	0.728 (± 0.045)
CPSO	2.806 (± 0.221)	4.197 (± 1.306)	0.778 (± 0.387)
SMAIN	2.767 (± 0.000)	6.047 (± 0.000)	**0.458** (± 0.000)
DWB	2.632 (± 0.168)	3.488 (± 0.888)	0.799 (± 0.195)
Opt-aiNet	2.781 (± 0.068)	4.623 (± 1.086)	0.662 (± 0.275)
LNNAIS	2.807 (± 0.207)	3.962 (± 0.576)	0.725 (± 0.127)
Glass			

Continued on next page

Algorithm	J_{intra}	J_{inter}	Q_{ratio}
K-means	1.035 (± 0.038)	4.557 (± 0.464)	0.901 (± 0.309)
CPSO	1.581 (± 0.120)	3.017 (± 1.121)	1.685 (± 0.674)
SMAIN	1.709 (± 0.003)	7.663 (± 0.038)	**0.381** (± 0.007)
DWB	1.198 (± 0.089)	3.716 (± 0.899)	1.458 (± 0.471)
Opt-aiNet	1.446 (± 0.170)	3.256 (± 1.179)	2.188 (± 0.701)
LNNAIS	1.358 (± 0.149)	5.367 (± 0.423)	0.541 (± 0.113)

5.6 Testing for statistical significance - data group 4

Table 13 summarizes the results obtained for data group 4. The corresponding statistical hypothesis tests between LNNAIS and the remaining models for each of the data sets in group 4 are summarized in table 14. The Mann-Whitney U statistical hypothesis test accepts H_0 that the means are the same between LNNAIS and K-means for data set image segmentation, and between LNNAIS and Opt-aiNet for data set letter recognition. The Mann-Whitney U statistical hypothesis test rejects H_0 for all other cases (as summarized in table 14). In most cases LNNAIS tends to deliver clusters of a higher quality except for data set image segmentation and letter recognition where SMAIN tends to deliver clusters of a higher quality (refer to table 13). Also, K-means tends to deliver clusters of a higher quality for data set letter recognition.

The experimental results show that in general LNNAIS delivers clusters of similar or higher quality than classical data clustering models like K-means and CPSO, and network based AIS models like DWB and Opt-aiNet. Overall, SMAIN tends to deliver clusters of a higher quality for all data sets, followed by LNNAIS. Although SMAIN tends to deliver clusters of a higher quality than LNNAIS, a cursory assessment indicates that SMAIN tends to utilize a larger ALC population than LNNAIS. Since the final population of ALCs in SMAIN is clustered with a hierarchical agglomerative clustering technique to determine the clusters in the data set, the larger ALC population size of SMAIN results in superior clustering quality. Furthermore, LNNAIS has less user specified parameters when compared to SMAIN. The next section compares and discusses the ALC population sizes of SMAIN, DWB and LNNAIS to elaborate on the cursory assessment of the compression of the data. This is then followed by a sensitivity analysis of the LNNAIS parameters on the clustering quality of the model.

Table 13: Descriptive Statistics: Data Group 4

Data set			
Algorithm	J_{intra}	J_{inter}	Q_{ratio}
Image Segmentation			
K-means	65.274 (± 0.523)	356.964 (± 32.396)	0.694 (± 0.033)
CPSO	77.522 (± 7.161)	177.950 (± 24.600)	1.493 (± 0.598)
SMAIN	126.990 (± 0.283)	787.028 (± 1.906)	**0.400** (± 0.001)
DWB	71.657 (± 3.074)	245.495 (± 133.903)	1.060 (± 0.301)
Opt-aiNet	74.457 (± 6.321)	174.931 (± 28.219)	1.621 (± 0.990)
LNNAIS	87.984 (± 9.635)	597.456 (± 116.260)	0.989 (± 1.015)
Spambase			
K-means	216.058 (± 0.000)	2003.26 (± 0.000)	0.108 (± 0.000)
CPSO	301.660 (± 19.617)	136.613 (± 30.941)	2.301 (± 0.452)

Continued on next page

Algorithm	J_{intra}	J_{inter}	Q_{ratio}
SMAIN	239.369 (± 27.139)	1599.79 (± 831.832)	0.194 (± 0.096)
DWB	185.926 (± 22.246)	1216.17 (± 1509.06)	0.236 (± 0.120)
Opt-aiNet	247.833 (± 18.812)	71.578 (± 37.181)	5.586 (± 6.421)
LNNAIS	432.734 (± 221.003)	6720.66 (± 2691.21)	**0.074** (± 0.046)
Letter Recognition			
K-means	5.383 (± 0.012)	11.121 (± 0.157)	1.090 (± 0.043)
CPSO	6.571 (± 0.121)	11.028 (± 0.764)	1.480 (± 0.225)
SMAIN	7.297 (± 0.238)	17.299 (± 0.455)	**0.751** (± 0.029)
DWB	6.562 (± 0.108)	12.268 (± 0.704)	1.758 (± 0.662)
Opt-aiNet	6.419 (± 0.108)	11.778 (± 0.630)	1.367 (± 0.179)
LNNAIS	6.072 (± 0.080)	12.601 (± 0.331)	1.351 (± 0.202)

5.7 ALC Population Size - Compression of the Data

This section investigates the ALC population sizes between SMAIN, DWB and LNNAIS to indicate compression of the data (ability to remove redundant data). Low compression of the data could result in superior clustering quality of a specific model when compared to other models which utilize a smaller ALC population size. Figure 4 illustrates a histogram of the ALC population size of SMAIN, DWB and LNNAIS to cluster the data sets. The size of the ALC population is expressed as a ratio of the applicable data set size. Therefore, an ALC population size ratio closer to 1.0 indicates a lower compression of the applicable data set. The figure illustrates that LNNAIS has a population size ratio of less than 0.2 for all of the data sets. On the contrary, SMAIN has a population size ratio of more than 0.4 for six of the data sets (two-spiral, hepta, chainlink, target, ionosphere and glass). For data sets glass and ionosphere, the ALC population size of SMAIN is almost

Table 14: Statistical Hypothesis Testing between LNNAIS and Other Models based on Q_{ratio}: Data Group 4 ($\alpha = 0.05$; with continuity correction; unpaired; non-directional)

Data set	Algorithm	z	p	Outcome
Image Segmentation	K-means	1.922	0.055	Accept H_0
	CPSO	5.093	< 0.001	Reject H_0
	DWB	3.6	< 0.001	Reject H_0
	SMAIN	6.646	< 0.001	Reject H_0
	Opt-aiNet	5.064	< 0.001	Reject H_0
Spambase	K-means	3.984	< 0.001	Reject H_0
	CPSO	6.646	< 0.001	Reject H_0
	DWB	5.603	< 0.001	Reject H_0
	SMAIN	5.5	< 0.001	Reject H_0
	Opt-aiNet	6.646	< 0.001	Reject H_0
Letter Recognition	K-means	5.404	< 0.001	Reject H_0
	CPSO	2.144	0.032	Reject H_0
	DWB	3.053	0.002	Reject H_0
	SMAIN	6.646	< 0.001	Reject H_0
	Opt-aiNet	0.288	0.773	Accept H_0

equal to the size of the data sets (ratio close to 1.0). In general, SMAIN utilizes a larger ALC population to cluster the data than DWB and LNNAIS. This not only explains the superior clustering quality of SMAIN in the previous section but also a drawback of SMAIN which has a low compression of the data (unable to remove redundant data). Compared to SMAIN in view of these findings, LNNAIS delivers clusters of high quality with a good compression of the data.

5.8 Influence of LNNAIS parameters

This section investigates the influence of the LNNAIS parameters on the clustering quality of the model with reference to Q_{ratio}, J_{intra}, J_{inter} and the number of obtained clusters K. These parameters are the maximum population size, $\mathcal{B}_{max}$, the neighborhood size, ρ, and the clonal level threshold, ϵ_{clone}. Compared to the network based AIS models which are used in this

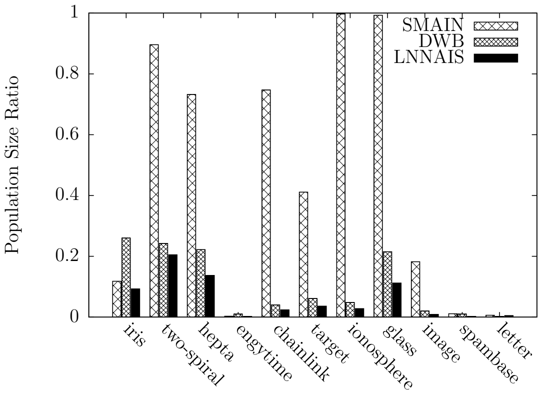
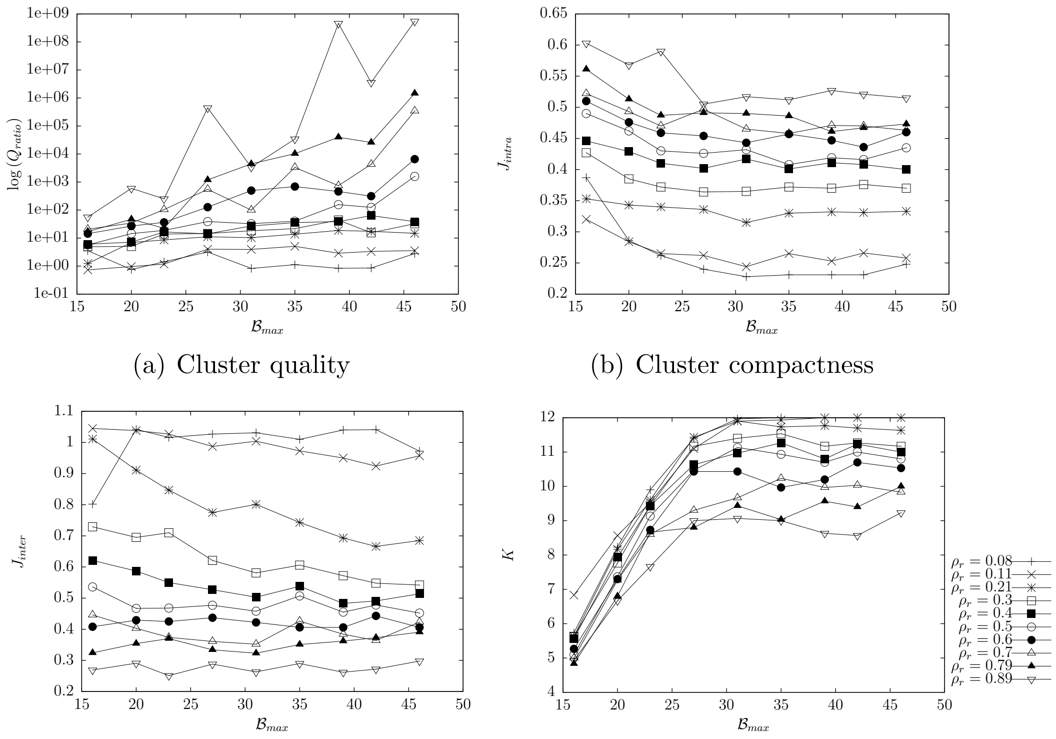


Figure 4: ALC Population Size Ratios of SMAIN, DWB and LNNAIS

paper, LNNAIS has significantly less parameters. The clustering results of a representative data set were selected from each of the defined data groups for the discussion. All of the other clustering results of the remaining data sets within the same data group, followed similar trends unless stated otherwise. The identified data sets include two spiral from group 1, chainlink from group 2, glass from group 3 and image segmentation from group 4. The LNNAIS model has been executed with population sizes of 10 to 50 ALCs, clonal level threshold values of 6 to 27 and neighborhood sizes which are calculated as a ratio of the population size. Neighborhood size ratios from 0.05 to 0.9 were used to calculate the neighborhood size ρ using $\rho = \rho_r \times \mathcal{B}_{max}$ (ρ_r is the neighborhood size ratio). In cases where a parameter was kept constant, the parameter was set to the value as listed in table 6 for each of the applicable data sets.

Population Size: Figures 5 to 8 show the effect of different ALC population sizes, $\mathcal{B}_{max}$, at different neighborhood size ratios, ρ_r, and a constant clonal level threshold, ϵ_{clone}. These figures show that for small neighborhood sizes an increase in the ALC population size has less significant influence on the clustering quality, Q_{ratio}, when compared to larger neighborhood sizes. The cluster compactness and separation do however tend to decrease at low neighborhood sizes with an increase in the ALC population size (increasing J_{intra} and decreasing J_{inter}). Furthermore, figures 5 to 8 also show that for all the different neighborhood sizes, no significant improvement is achieved



(a) Cluster quality

(b) Cluster compactness

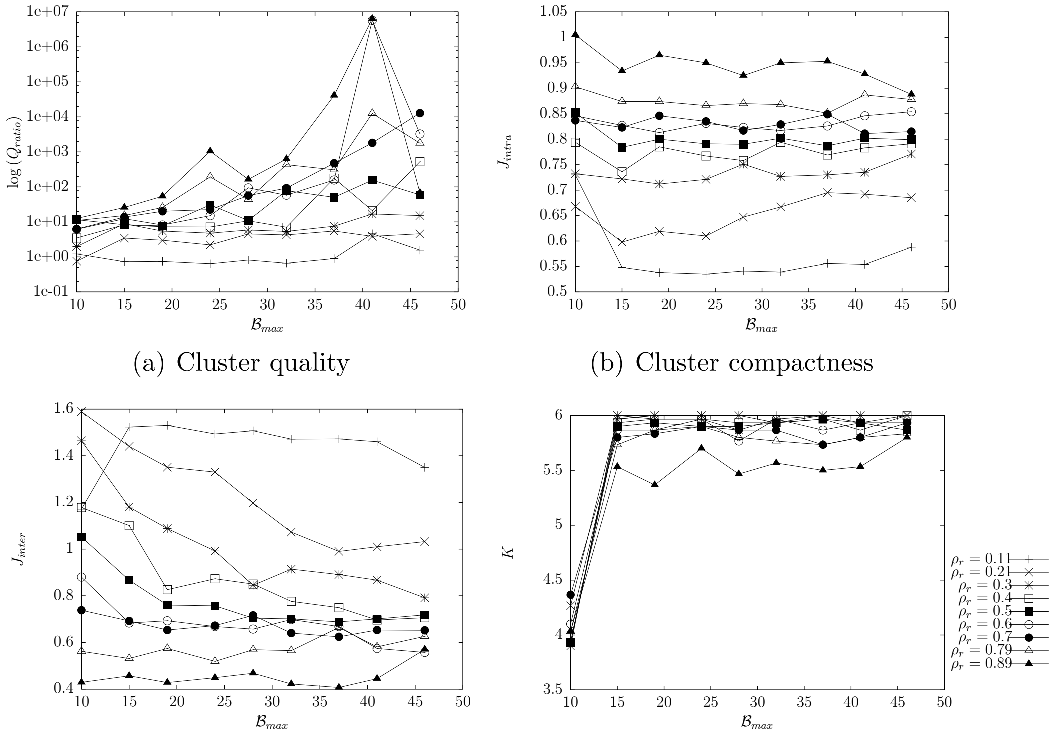
(c) Cluster separation

(d) Number of obtained clusters

Figure 5: Two Spiral data set ($\epsilon_{clone} = 6$): Effect of the ALC population size with a constant clonal level threshold

in the number of obtained clusters for ALC population sizes larger than a specific optimal value (which is problem dependent). This can also be observed in figures 13 to 16. Figures 13 to 16 show that for different clonal level threshold values with a low constant neighborhood size, an increase in the ALC population size increases the cluster compactness and separation (decreasing J_{intra} and increasing J_{inter}). Therefore, an increase in the ALC population size increases diversity which obtains the required number of clusters and improves the clustering quality.

Neighborhood Size: Figures 9 to 12 show the effect of different neighborhood size ratios, ρ_r, at different clonal level threshold values, ϵ_{clone}, and a constant ALC population size, $\mathcal{B}_{max}$. An increase in the neighborhood



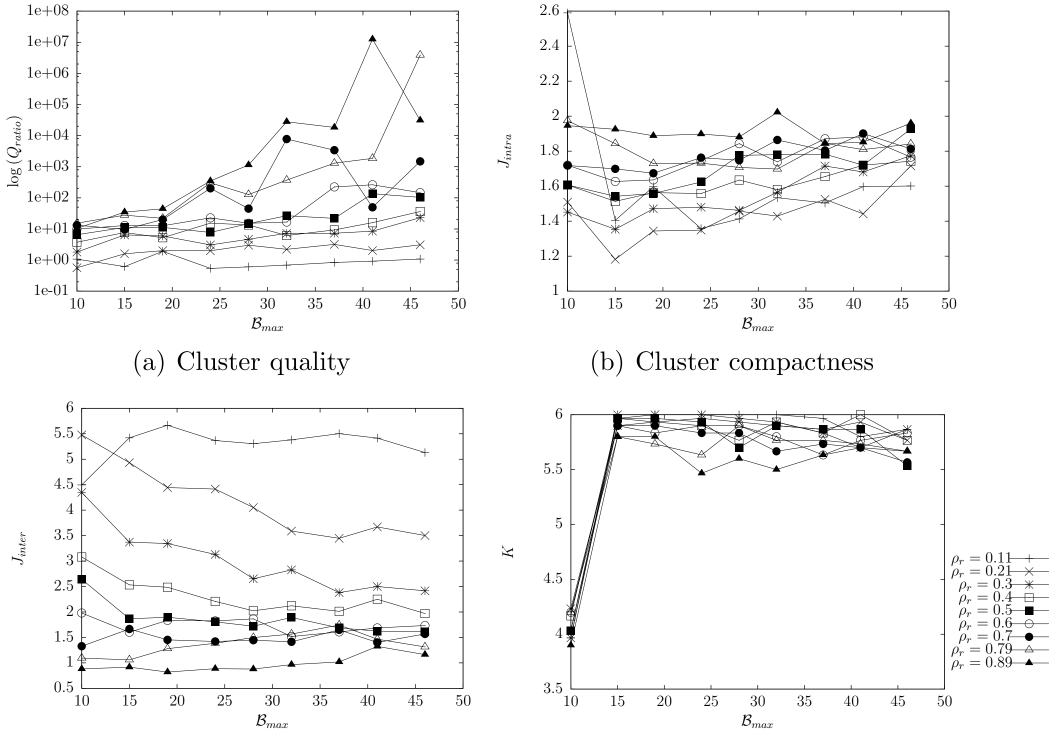
(a) Cluster quality

(b) Cluster compactness

(c) Cluster separation

(d) Number of obtained clusters

Figure 6: Chainlink data set ($\epsilon_{clone} = 8$): Effect of the ALC population size with a constant clonal level threshold



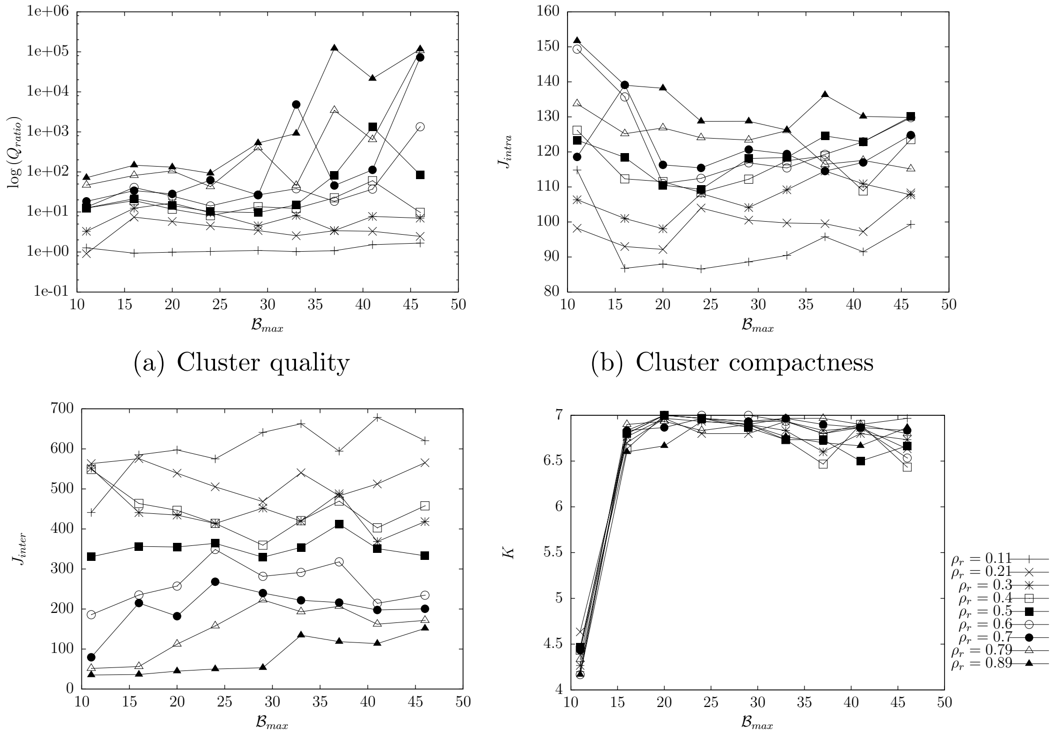
(a) Cluster quality

(b) Cluster compactness

(c) Cluster separation

(d) Number of obtained clusters

Figure 7: Glass data set ($\epsilon_{clone} = 8$): Effect of the ALC population size with a constant clonal level threshold



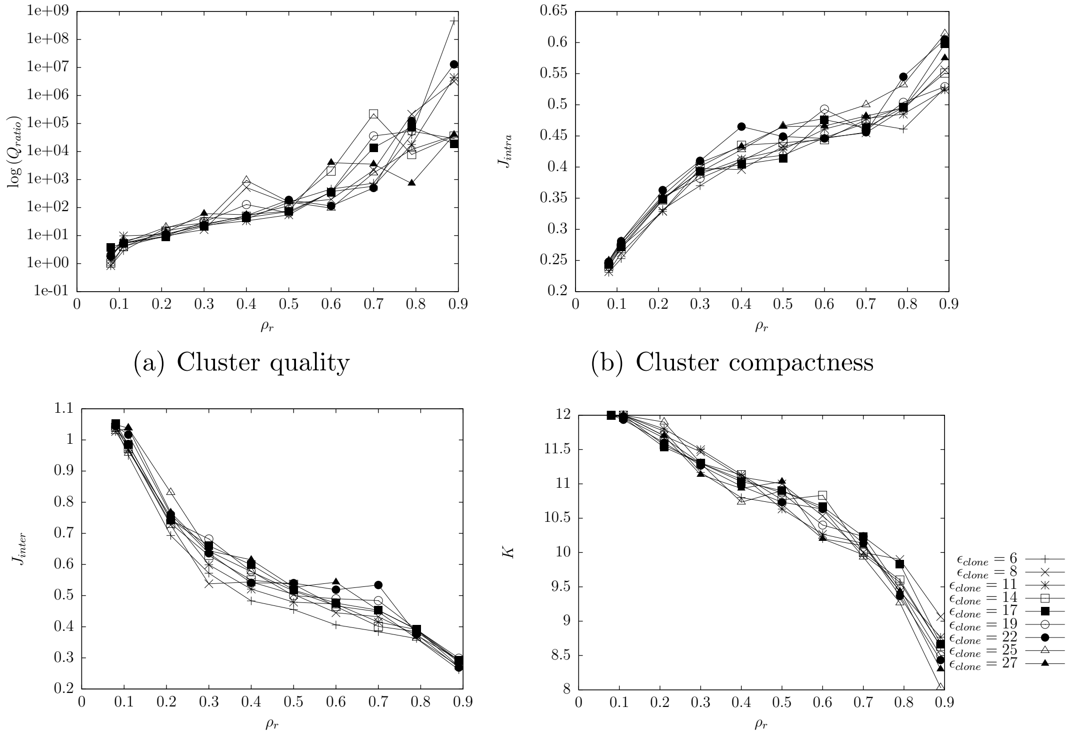
(a) Cluster quality

(b) Cluster compactness

(c) Cluster separation

(d) Number of obtained clusters

Figure 8: Image Segmentation data set ($\epsilon_{clone} = 27$): Effect of the ALC population size with a constant clonal level threshold



(a) Cluster quality

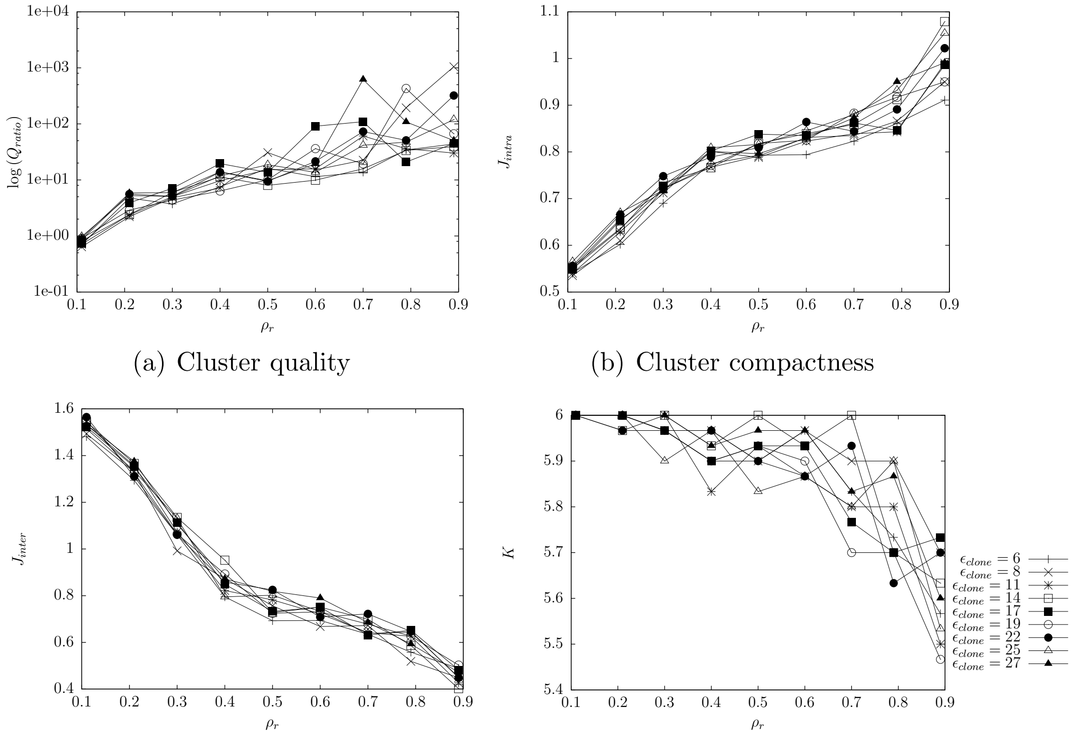
(b) Cluster compactness

(c) Cluster separation

(d) Number of obtained clusters

Figure 9: Two Spiral data set ($\mathcal{B}_{max} = 39$): Effect of the neighborhood size with a constant ALC population size

size decreases the cluster compactness and separation for all of the different clonal level threshold values, resulting in clusters of a lower quality (increasing Q_{ratio} and J_{intra} with a decreasing J_{inter}). This effect is also shown in figures 5 to 8 where an increase in the neighborhood size ratio decreases the cluster compactness (increases J_{intra}) and decreases the cluster separation (decreases J_{inter}) for all values of $\mathcal{B}_{max}$. From these observations it can be concluded that small values of ρ_r deliver more compact and more separated clusters (lower J_{intra}, higher J_{inter}) and therefore clusters of higher quality (lower Q_{ratio}) when compared to higher values of ρ_r. From the above mentioned figures, lower neighborhood sizes also tend to obtain the required number of clusters.



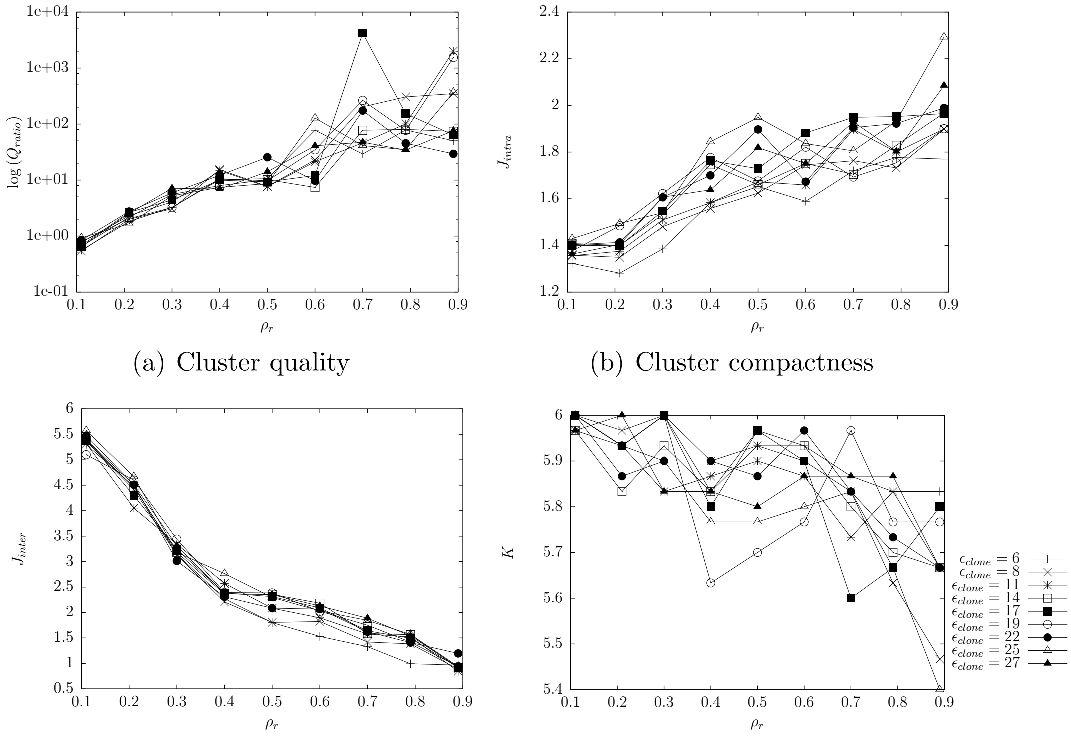
(a) Cluster quality

(b) Cluster compactness

(c) Cluster separation

(d) Number of obtained clusters

Figure 10: Chainlink data set ($\mathcal{B}_{max} = 24$): Effect of the neighborhood size with a constant ALC population size



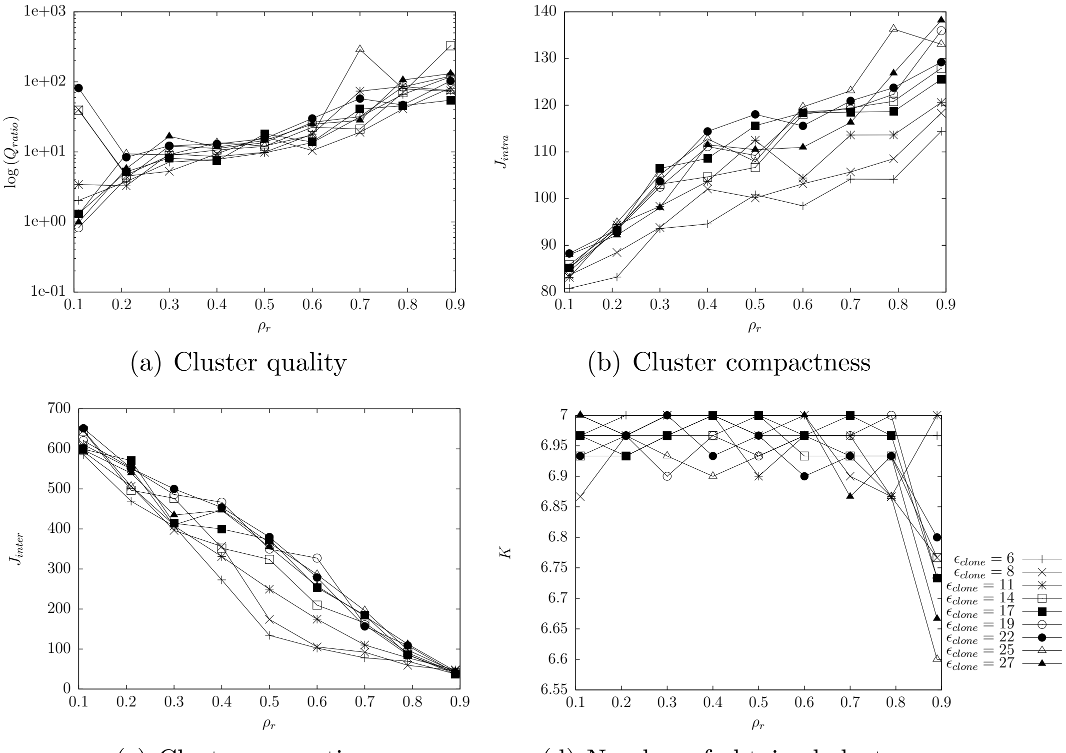
(a) Cluster quality

(b) Cluster compactness

(c) Cluster separation

(d) Number of obtained clusters

Figure 11: Glass data set ($\mathcal{B}_{max} = 24$): Effect of the neighborhood size with a constant ALC population size



(a) Cluster quality

(b) Cluster compactness

(c) Cluster separation

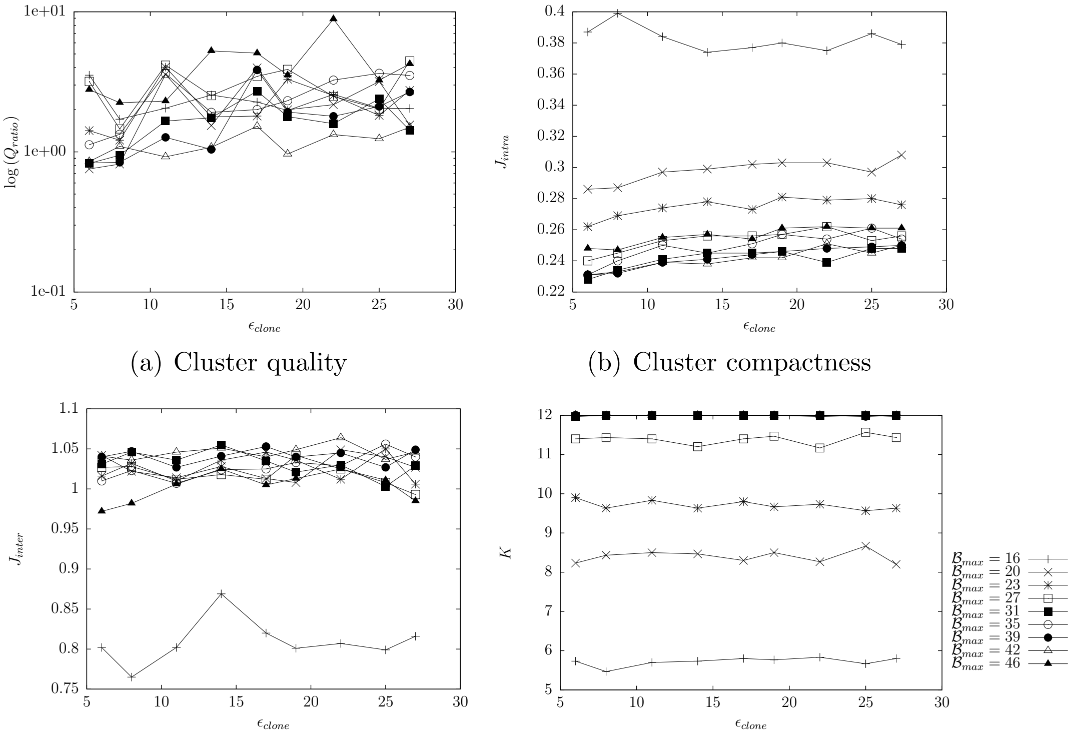
(d) Number of obtained clusters

Figure 12: Image Segmentation data set ($\mathcal{B}_{max} = 20$): Effect of the neighborhood size with a constant ALC population size

Clonal Level Threshold: Figures 13 to 16 show the effect of different clonal level threshold values, ϵ_{clone}, at different ALC population sizes, $\mathcal{B}_{max}$, and a constant neighborhood size, ρ. An increase in the clonal level threshold has no significant improvement in the number of obtained clusters at different ALC population sizes (as illustrated in figures 13 to 16) and also not at different neighborhood sizes (as illustrated in figures 9 to 12). Furthermore, the different clonal level threshold values follow similar trends with reference to the quality, compactness and separation of the clusters when the neighborhood size increases (as illustrated in figures 9 to 12 and explained in the previous paragraph). In the case of the chainlink and image segmentation data sets, increasing the clonal level threshold also results in less compact clusters at different ALC population sizes (as illustrated in figures 14 and 16), whereas for the two spiral and glass data sets there is no significant change in the compactness of the clusters (as illustrated in figures 13 and 15). Therefore, the clonal level threshold influences the cluster compactness and is problem specific. In summary, the clustering performance of LNNAIS is sensitive to the values of the ALC population size and neighborhood size. The ALC population size is problem specific and in general low neighborhood size values deliver clusters of higher quality. The clustering performance of LNNAIS is generally insensitive to the value of the clonal level threshold.

6 Conclusion and Future Work

A formal definition of data clustering was given with different performance measures to determine the quality of clusters. The network interaction and formation between artificial lymphocytes (ALCs) in existing network based AIS models were discussed as well as alternative and less familiar ALC network topologies. A new network based AIS model (LNNAIS) was proposed for data clustering. LNNAIS utilizes a different network topology, which is an index-based ALC neighborhood topology to determine the network connectivity between ALCs. The clustering performance of the LNNAIS model was compared against classical clustering algorithms (K-means clustering and CPSO) and existing network based AIS models (SMAIN, DWB and Opt-aiNet). In most cases, LNNAIS produced better or similar results with reference to the quality, compactness and separation of the clusters. Although SMAIN tends to deliver clusters of a higher quality than the proposed LNNAIS, a cursory assessment indicates that SMAIN tends to utilize a



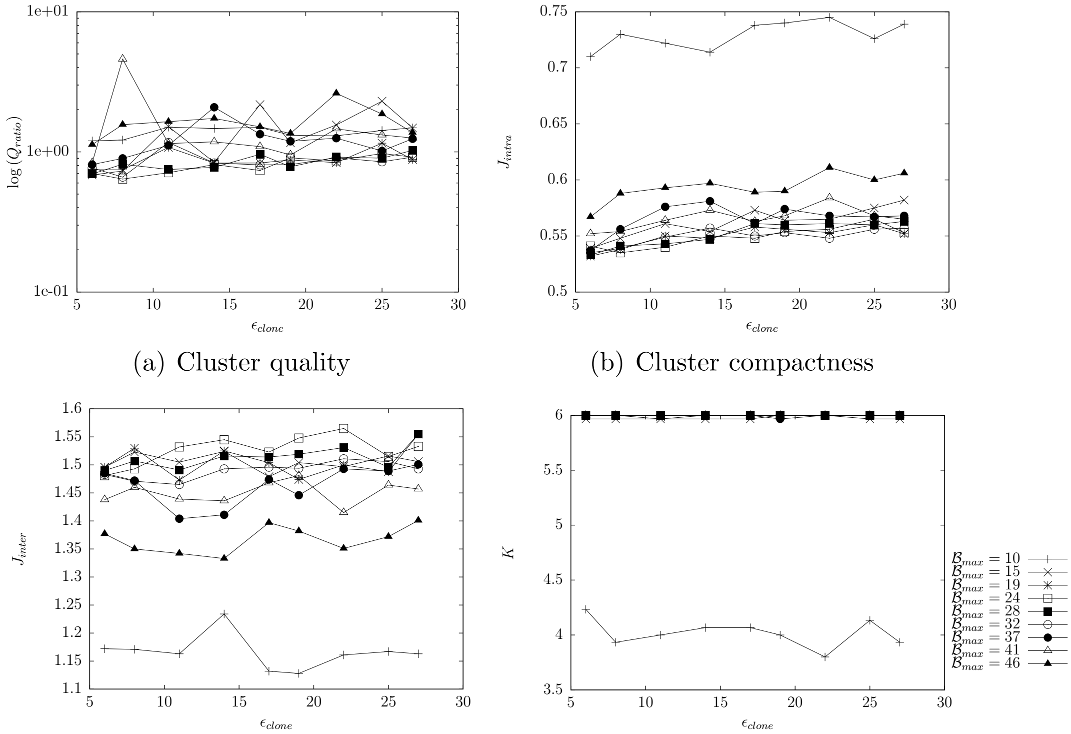
(a) Cluster quality

(b) Cluster compactness

(c) Cluster separation

(d) Number of obtained clusters

Figure 13: Two Spiral data set ($\rho = 3$): Effect of the clonal level threshold with a constant neighborhood size



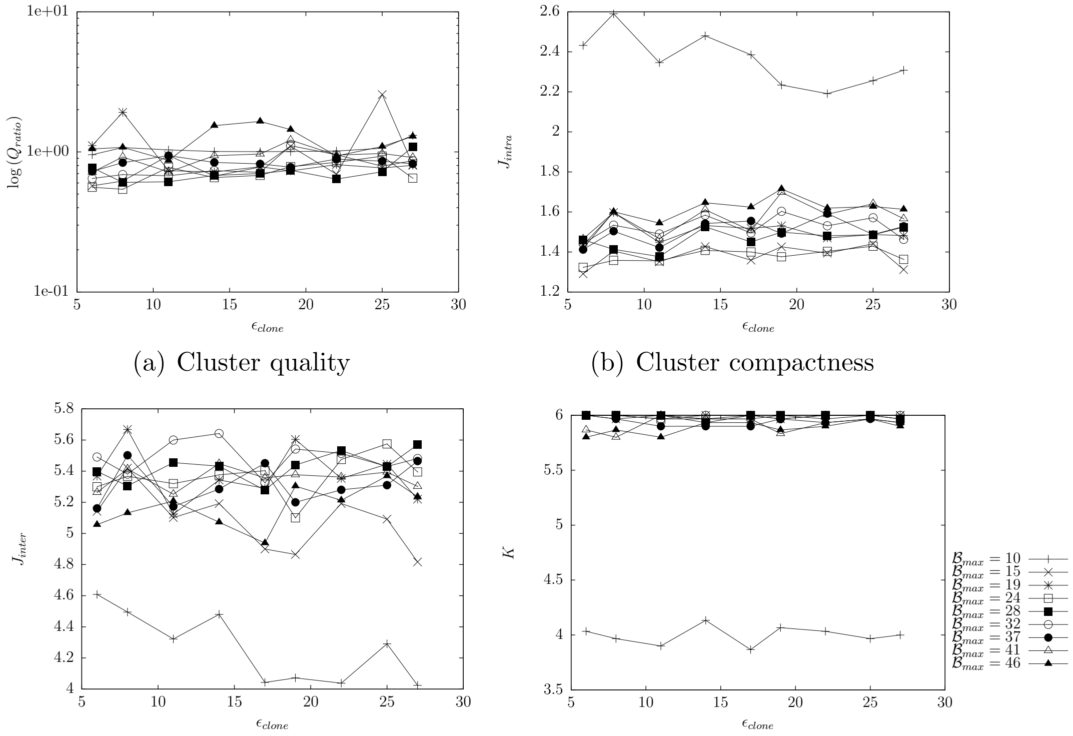
(a) Cluster quality

(b) Cluster compactness

(c) Cluster separation

(d) Number of obtained clusters

Figure 14: Chainlink data set ($\rho = 3$): Effect of the clonal level threshold with a constant neighborhood size



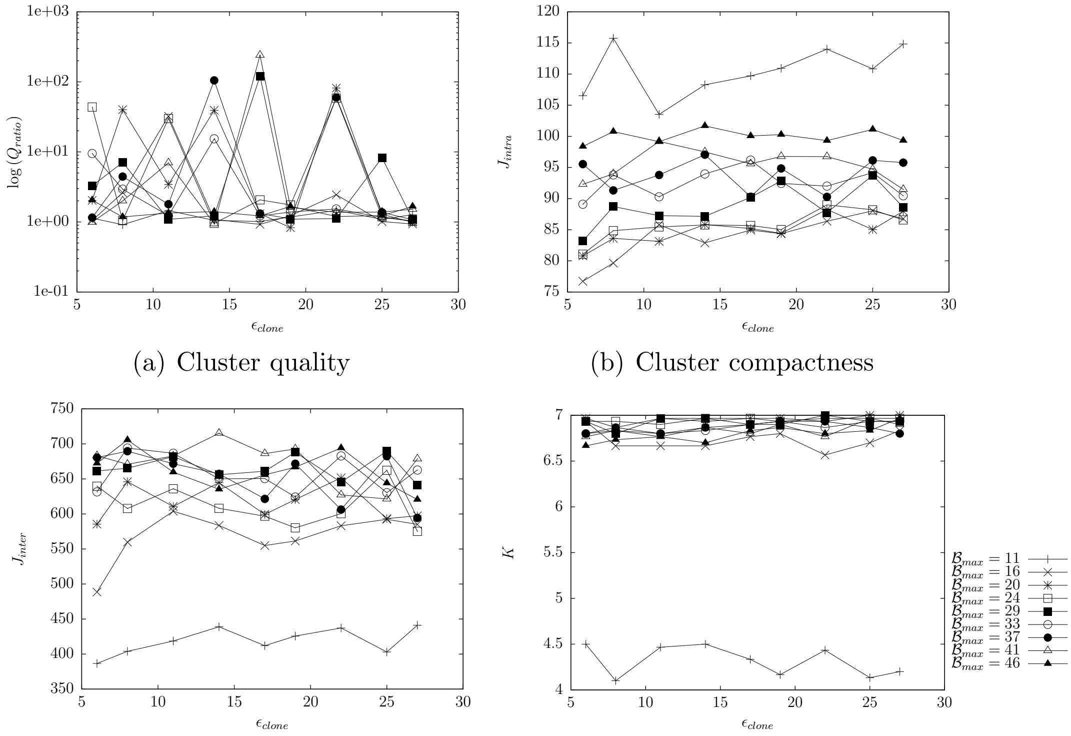
(a) Cluster quality

(b) Cluster compactness

(c) Cluster separation

(d) Number of obtained clusters

Figure 15: Glass data set ($\rho = 3$): Effect of the clonal level threshold with a constant neighborhood size



(a) Cluster quality

(b) Cluster compactness

(c) Cluster separation

(d) Number of obtained clusters

Figure 16: Image Segmentation data set ($\rho = 2$): Effect of the clonal level threshold with a constant neighborhood size

larger ALC population than LNNAIS and therefore has a lower compression of the data. Further investigation is needed to qualify this assessment. A sensitivity analysis was done on the parameters of LNNAIS to investigate the effect of the parameters on the clustering quality. An increase in the ALC population size increases diversity which obtains the required number of clusters and improves the clustering quality. Smaller neighborhood sizes deliver more compact and more separated clusters when compared to larger neighborhood sizes, and tend to obtain the required number of clusters. Therefore small neighborhood sizes deliver clusters of a higher quality. The clonal level threshold influences the compactness of the clusters and is problem specific. Although none of the discussed AIS models in this paper require any user specified parameter of the number of required clusters to cluster the data, the techniques used by these models to determine the number of ALC networks do, however. Therefore, future work also needs to investigate alternative techniques that can be used with LNNAIS to dynamically determine the number of clusters in a data set [18]. Furthermore, as proposed in [17], future work also includes the application of LNNAIS to the problem of image segmentation and classification problems. In the case of image segmentation, the pixels of an image are seen as antigen patterns which are clustered by LNNAIS. Each cluster represents a segment of the image. In the context of classification problems, an amended LNNAIS model labels the ALCs with the same class labels with which patterns in the antigen set are labeled. This is a semi-supervised approach of the amended LNNAIS model, where ALCs only adapt to antigen patterns of the same class and the final ALC networks represent different classes in the antigen set.

References

[1] Asuncion, A., Newman, D.: UCI machine learning repository (2007). URL http://www.ics.uci.edu/∼mlearn/MLRepository.html

[2] Berkhin, P.: Survey of clustering data mining techniques, accrue software. Inc. TR, San Jose, USA (2002)

[3] Burnet, F.M.: The Clonal Selection Theory of Acquired Immunity. Nashville: Vanderbilt University Press (1959)

[4] de Castro, L., Timmis, J.: An artificial immune network for multimodal function optimization. pp. 699–704. IEEE Press (2002)

[5] de Castro, L., Zuben, F.V.: The clonal selection algorithm with engineering applications. pp. 36–37 (2000)

[6] de Castro, L., Zuben, F.V.: AiNet: An Artificial Immune Network for Data Analysis, pp. 231–259. Idea Group Publishing, USA (2001)

[7] de Castro, L., Zuben, F.V.: Learning and optimization using the clonal selection principle. IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems **6**, 239–251 (2002)

[8] de Castro, L.N., Zuben, F.J.V.: An evolutionary immune network for data clustering. In: IEEE SBRN, pp. 84–89. Rio de Janeiro (2000)

[9] Eberhart, R., Simpson, P., Dobbins, R.: Computational intelligence PC tools. Academic Press Professional, Inc. San Diego, CA, USA (1996)

[10] Engelbrecht, A.P.: Fundamentals of Computational Swarm Intelligence. Wiley (2005)

[11] Forgy, E.: Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications. Biometrics **21**, 768 (1965)

[12] Forrest, S., Perelson, A.S., Allen, L., Cherukuri, R.: Self-nonself discrimination in a computer. pp. 202–212. IEEE Computer Society Press, Oakland, CA (1994). URL http://citeseer.ist.psu.edu/forrest94selfnonself.html

[13] Fukuda, T., Mori, K., Tsukiyama, M.: Parallel search for multi-modal function optimization with diversity and learning of immune algorithm. Springer (1998)

[14] Gonzalez, F., Dasgupta, D., Kozma, R.: Combining negative selection and classification techniques for anomaly detection. Hawaii (2002)

[15] Graaff, A.J., Engelbrecht, A.P.: Optimised coverage of non-self with evolved lymphocytes in an artificial immune system. International Journal of Computational Intelligence Research **2**, 127–150 (2006)

[16] Graaff, A.J., Engelbrecht, A.P.: A local network neighbourhood artificial immune system for data clustering. pp. –. IEEE Press, Singapore (2007)

[17] Graaff, A.J., Engelbrecht, A.P.: Clustering data in an uncertain environment using an artificial immune system. Pattern Recognition Letters **32**(2), 342–351 (2011). DOI DOI: 10.1016/j.patrec.2010.09.013

[18] Graaff, A.J., Engelbrecht, A.P.: Using sequential deviation to dynamically determine the number of clusters found by a local network neighbourhood artificial immune system. Applied Soft Computing **11**, 2698–2713 (2011). DOI http://dx.doi.org/10.1016/j.asoc.2010.10.017. ACM ID: 1930600

[19] Halkidi, M., Batistakis, Y., Vazirgiannis, M.: Cluster validity methods: part i. ACM SIGMOD Record **31**, 40–45 (2002)

[20] Hamerly, G., Elkan, C.: Alternatives to the k-means algorithm that find better clusterings. Proceedings of the eleventh international conference on Information and knowledge management pp. 600–607 (2002)

[21] Hiernaux, J.: Some remarks on the stability of the idiotypic network. Immunochemistry **14**, 733–739 (1977)

[22] Jain, A., Murty, M., Flynn, P.: Data clustering: a review. ACM Comput. Surv. **31**, 264–323 (1999). URL http://portal.acm.org/citation.cfm?id=331504

[23] Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice-Hall, Englewood Cliffs, New Jersey (1988)

[24] Jerne, N.K.: Towards a network theory of the immune system. Annals of Immunology (Inst. Pasteur) **125C**, 373–89 (1974). PMID: 4142565

[25] Jerne, N.K.: The generative grammar of the immune system. The European Molecular Biology Organization journal **4**, 847–52 (1985). PMID: 2410261

[26] Kennedy, J.F., Eberhart, R.C., Shi, Y.: Swarm Intelligence. NetLibrary, Incorporated (2001)

[27] Kovács, F., Legány, C., Babos, A.: Cluster validity measurement techniques. In: 6th International Symposium of Hungarian Researchers on Computational Intelligence. Budapest (2005)

[28] Lee, C.Y., Antonsson, E.K.: Dynamic partitional clustering using evolutionary strategies. Proc. of the 3rd Asia?Pacific Conference on Simulated Evolution and Learning (2000)

[29] van der Merwe, D.W., Engelbrecht, A.P.: Data clustering using particle swarm optimization. Evolutionary Computation, 2003. CEC'03. The 2003 Congress on **1** (2003)

[30] Nasraoui, O., Cardona, C., Rojas, C., Gonzalez, F.: Mining evolving user profiles in noisy web clickstream data with a scalable immune system clustering algorithm. pp. 71–81 (2003). URL http://citeseer.ist.psu.edu/nasraoui03mining.html

[31] Nasraoui, O., Cardona-Uribe, C., Rojas-Coronel, C.: Tecno-streams: Tracking evolving clusters in noisy data streams with a scalable immune system learning model. Melbourne, Florida (2003). URL http://citeseer.ist.psu.edu/nasraoui03tecnostreams.html

[32] Nasraoui, O., Gonzalez, F., Cardona, C., Rojas, C., Dasgupta, D.: A scalable artificial immune system model for dynamic unsupervised learning. pp. 219–230. Springer-Verlag, Chicago (2003)

[33] Neal, M.: Meta-stable memory in an artificial immune network. In Artificial Immune Systems: Proceedings of ICARIS 2003 pp. 168–180 (2003). DOI 10.1.1.95.6647

[34] Perelson, A.S., Weisbuch, G.: Immunology for physicists. Reviews of Modern Physics **69**, 1219 (1997). URL http://link.aps.org/abstract/RMP/v69/p1219

[35] Potter, M., Jong, K.D.: The coevolution of antibodies for concept learning. pp. 530–539 (1998)

[36] Pramanik, S., Kozma, R., Dasgupta, D.: Dynamical neuro-representation of an immune model and its application for data classification. pp. 130–135. Honolulu, HI, USA (2002)

[37] Ray, S., Turi, R.H.: Determination of number of clusters in k-means clustering and application in colour image segmentation. The 4th International Conference on Advances in Pattern Recognition and Digital Techniques?, Calcuta (1999)

[38] Richter, P.: A network theory of the immune system. European Journal of Immunology **5**, 350–354 (1975)

[39] Richter, P.: The network idea and the immune response, pp. 539–569. Marcel Dekker, New York (1978)

[40] Shi, Eberhart: Parameter selection in particle swarm optimization, pp. 591–600 (1998). URL http://dx.doi.org/10.1007/BFb0040810

[41] Suganthan, P.: Particle Swarm Optimiser with Neighbourhood Operator. In: P.J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, A. Zalzala (eds.) Proceedings of the Congress of Evolutionary Computation, vol. 3, pp. 1958–1962. IEEE Press, Mayflower Hotel, Washington D.C., USA (1999)

[42] Timmis, J.: Artificial immune systems: A novel data analysis technique inspired by the immune network theory. Ph.D. thesis, University of Wales (2000). URL http://www.cs.kent.ac.uk/pubs/2000/1102/index.html

[43] Timmis, J., Neal, M.: A resource limited artificial immune system for data analysis. pp. 19–32. Springer, Cambridge, UK. (2000)

[44] Timmis, J., Neal, M., Hunt, J.: Data analysis using artificial immune systems, cluster analysis and kohonen networks: Some comparisons. pp. 922–927. Tokyo, Japan (1999)

[45] Weisbuch, G., Boer, R.J.D., Perelson, A.S.: Localized memories in idiotypic networks. Journal of Theoretical Biology **146**, 483–499 (1990). DOI 10.1016/S0022-5193(05)80374-1