Exploring Island Genetic Algorithms and Unbounded Recognition Regions for Artificial Immune Systems

1st Storm Jan Anton Visser
Department of Computer Science
Norwegian University of Science and Technology
Trondheim, Norway
sjvisser@stud.ntnu.no

2nd Pauline Catriona Haddow
Department of Computer Science
Norwegian University of Science and Technology
Trondheim, Norway
pauline@ntnu.no

***Abstract*—Artificial Immune Systems (AIS) have been extensively applied to the single-class anomaly detection problem, in optimization and adaptive control system applications. However, recent works are are finding ways to adapt AISs to address the needs of multi-class problems, at times achieving competitive advantages over traditional machine learning. Further, for higher dimensional problems such as Fake News classification, further extensions are needed. One such extension is Unbounded Recognition regions – the area of the search space in which an antibody may detect an antigen (data). Exciting hybrid advances are also present in the field, combining AIS with, for example, Island Genetic Algorithms or Quantum computing.**

The work herein introduces the AISIGA model - a hybrid AIS and Island Genetic Algorithm, adapted from the MAIM model. Results exploring the extension of unbounded recognition regions, contributes to an improved understanding of this newer AIS mechanism. Preliminary results comparing AISIGA with the state of the art, show promise but a need for further work, with respect to accuracy whilst highlighting the significant efficiency benefits of AISIGA compared to the state of the art.

***Index Terms*—artificial immune systems, recognition regions, island genetic algorithms, classification**

I. Introduction

Artificial Immune systems (AIS) have demonstrated significant potential, addressing complex challenges including anomaly detection, adaptive systems control, and optimization. Anomaly detection may be considered a one-class classification problem. In recent years, AIS algorithms have been successfully adapted to higher dimensional feature selection and flexible job scheduling problems for type-2 fuzzy logic systems [1], [2]. Further, a hybrid AIS algorithm and multi-class classification tasks have been presented [3], [4], where [3] showed promising results combining AIS with Island Genetic Algorithms (IGA) for multi-class classification. The work herein, proposes a refined classification algorithm AISIGA, extending the work in MAIM [3], exploring alternative mechanisms within AIS and IGA through experiments conducted on six traditional classification datasets.

The work herein is structured as follows; Sec. II presents the background theory and discusses related work, Sec. III describes the proposed model, Sec. IV describes the experiments and results, and Sec. V presents the contributions and future work. The project data/code is available at [5].

II. Background and Related work

The natural immune system (IS) is a complex system tasked with protecting an organism from harmful changes. The IS classifies threats based on antigens (AG) that are found on the surface of the threats. Antibodies (AB) are then created and optimized to recognize the antigen (class) to fight off the threat if needed – see [6] for a more detailed description.

AISs draw inspiration from different elements of the IS. Although many variants exist, at their core is a population of ABs, tuned to detect historical and current AGs. For classification, data is represented as the AGs. To ensure all feature values of the AG are considered evenly, the data is first normalized. The AIS then creates a population of ABs that evolves to classify the AGs of the dataset. In contrast to other population-based optimization algorithms, every individual AB in the population is part of the solution, as in [7]–[10].

Variants of AIS are many, including the CLONALG algorithm [7] - one of the first AIS algorithms that applied a cloning approach for creating ABs; and the AIRS algorithm [8], [9] - introducing affinity maturation that clones stimulated ABs more often. Both CLONALG and AIRS classify AGs based on the class of the closest AB, therein only utilizing one single AB for an AG classification. To allow multiple ABs to influence the class of the AG, VALIS introduced a voting process [10].

When creating ABs in an AIS, a common approach to apply is random initialisation [2], [8], [9]. However, it was proposed in [11] to ensure that such random initialisation was constrained to consider the class distribution in the AG population i.e. in the data. Thus Class ratio locking (CRL), a concept from

machine learning, was implemented. However, surprisingly CRL did not have an impact in the results presented.

A further challenge with random initialisation is low initial coverage - depending on the distribution of the AGs in the search space. One solution is introducing a per class limiting window (see [3], [10]) where ABs are randomly created within the bounds of the class window, ensuring that ABs of a certain class are initialised in proximity of AGs of the same class.

A common strategy applied to the evolving AB population is to apply cloning, relying on mutation to change ABs [1], [2], [7]–[9]. A downside of the cloning strategy is that having one clone per parent limits exploration. Alternatively, multiple clones can be created per parent [12]. Only the best clones from each parent are selected to survive. In addition to mutation, crossover has also been utilized in combination with mutation to evolve ABs [10].

Traditional hybridization of IGAs with Genetic Algorithms have shown much success, as in [13]–[16]. However, hybridization with AIS was only recently proposed in [3] and with this success, applied herein. A common topology applied to connect the islands of the IGA is the unidirectional ring topology [3], [13], [16]–[18]. The master slave topology is another commonly applied topology [3], [19]. A number of slave islands are connected to one central master island, which controls migration to and from the slave islands. The master island allows for a higher level of control, while maintaining the isolation inherent in unidirectional ring topologies. Other topologies applied include the 2D grid, 3D grid, and hypercube topologies [15], as well as dynamic topologies [20].

When splitting the population of ABs over multiple islands, each island contains $\frac{1}{N}$ of the population, where N is the number of islands. The role of the master may vary between algorithms. However, in [3], its role is simplified to the collection of the AB subpopulations of ABs so as to create and evaluate the AB population i.e. the solution. The simplification allows the algorithm to utilize a ring topology for evolution, while leveraging the central positioning of the master island for collection.

An important challenge in AIS is the recognition region (RR) of the ABs, which represents the region around the ABs where AGs are detected. The most common RR is the hypersphere (HS) [3], [21], which can be defined by a middle point and a radius. However, HS lack the flexibility to adapt to data structures, resulting in the shrinking effect in higher dimensions, which causes HS to struggle to adapt to higher dimensional data [22], [23]. Hyper-Ellipsoids (HE) are an alternative that provide more flexibility, and can be defined by a middle point, a radius and a vector of multipliers (applied to the radius in each dimension) as proposed in [24]. HE showed increased performance on higher dimensional datasets, while lower dimensional data shows no significant improvements. A further solution to address high dimensionality utilizes local feature selection (LFS) [25] where, for example, a 3D problem may be simplified to a 2D problem. Simplifying the problem not only showed improved performance but the amount of training data required was also less [25].

A recent approach to high dimensionality is by applying Unbounded (UB) RRs to open an RR in a certain direction [11]. Making a HS (circular RR) unbounded changes it into a parabolic line, detecting infinitely in that direction — see Sec. III-C. As such, UB RRs are an extension of bounded RRs, such as the UB HS described. The results in [11] showed that UB RRs outperform the LFS solution. However, the opening of the bounded RR, was limited to only the upward direction (upward) rather than being able to open in either direction (upward or downwards).

III. Model

The AISIGA model extends the base model MAIM [3], enabling further exploitation of the integrated island model through the possibility to split the AG population across the individual islands. Further, it allows for the exploration of different RRs, including HE [24] and UB RRs [11], and different locking schemes during survivor selection.

The flow of the AISIGA algorithm is shown in Fig. 1. The AG population is created from the dataset. As stated, the model then allows for splitting the AGs among the islands or alternatively the AG population can be distributed in its entirety to each island — see Sec. III-A. During initialisation, the topology is chosen, herein according to a unidirectional ring topology with a central master island — see Sec. III-B. Further, the RR is chosen. For the experiments herein, the following RRs are included: HS; HE; HS with UB and HE with UB — described further in Sec. III-C.

Each island synchronously runs the same island process shown in Fig. 1. The process starts with the creation of ABs that reflect the dataset (AG population) allowing for a split to be invoked and applying the limited window mechanism, discussed in Sec. II. The model allows for the application of class ratio locking (CRL), and introduces RR Type locking (RRTL) — see Sec. III-E.

During the evolutionary process, parent selection, crossover, and mutation are applied to create new ABs until the child population is complete. The fitness of the individuals is then calculated (evaluation) by detecting the AGs and applying the fitness functions to score the individual ABs – see Sec. III-D. As shown, the population (ABs_{new}) from the previous iteration is combined with the new child population and survivor selection is applied, involving CRL and RRTL — see Sec. III-E.

After the new population is created (ABs_{new}), it is determined whether to migrate or to terminate. During the slave migration (migrate_S), each island simultaneously sends and receives R best individuals to/from its neighbours, where R is the migration rate. Once every F slave migrations, where F is migration frequency, the master island initiates master migration (Migrate_M), collecting all AB populations from the slave islands. The master island updates (replaces) the best population (combined population) found so far. When the algorithm terminates, after a maximum number of iterations, the master migration process is triggered and the best solution is updated and presented.

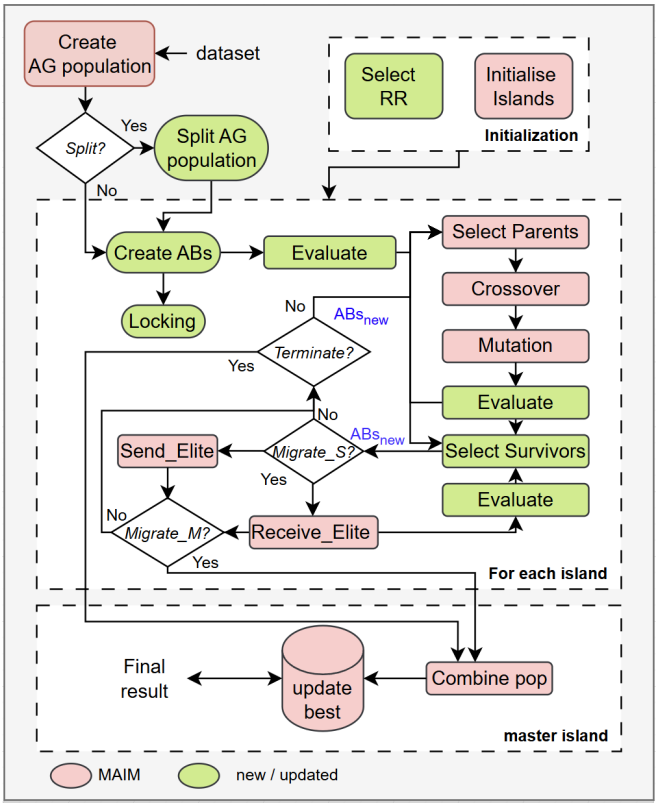

Fig. 1. The flow of the proposed algorithm

A. Antigen Splitting

The proposed splitting of the AGs across the islands enables each island to focus on different parts of the solution in parallel, aiming to increase efficiency. The AGs are divided based on a uniform distribution. Consequently, the ABs that are trained on one island get specialized to the subset of the problem, with migration moving the ABs to different subsets on the different islands. Furthermore, the AG splitting allows for a more informed initialisation, as less points will have to be considered when determining the bounds of the limiting window — see Sec.II.

B. Topology

The topology applied herein consists of four slave islands connected in a unidirectional ring, combined with one master island, as in [3]. During slave migration, elite migration is utilized for its proven use in other IGAs [26]. The master migration from MAIM [3] has been applied, which allows the master island to collect the sub populations to form the new population. The new population (solution) is then evaluated based on correct classifications divided by the total number of AGs.

C. Chromosome Structure and different RRs

The AB chromosome structure from [11] is utilized as it allowed for the adaptation of both HE and UB RRs into MAIM [3]. Fig. 2 illustrates the concept of open dimensions where

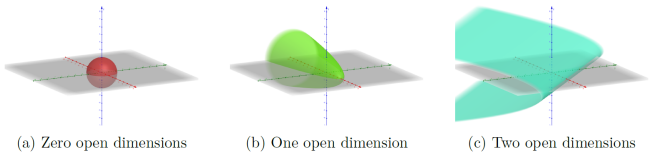

Fig. 2. AB RR with zero, one, and two open dimensions in 3d space [11]

Fig. 3. The chromosome structure of an AB

(a) shows a bounded HS, (b) shows an HS with one open dimension and (c) shows an HS with two open dimensions. UB has been adapted to allow the direction to change, solving the previous limitation of having only upward facing UB RRs. The AB chromosome consists of a class, a feature vector (F), a radius (R), a vector of multipliers (M), and an RR type vector (T) — see Fig. 3. The feature vector represents the central point of the AB. The radius value is used to determine whether an AG lies within an AB. The vector of multipliers consists of multipliers for each dimension of the AB that are applied to the radius e.g. an AB with R=1, and M=[1,2] would become an ellipsoid with radius of 1 in the first dimension, and radius of 2 in the second dimension. The RR type vector defines whether a dimension is bounded (0), UB upwards (1) or UB downwards (2).

The chromosome structure of an AG consists of its class and its feature vector (F) — see Fig. 3.

D. Evolution

As in the MAIM algorithm [3], tournament selection is used to select parents, uniform crossover is used to create children, and uniform mutation is applied to the children. The uniform mutation multiplies the different chromosome values with random numbers in the bounds of 0.1 and 2.

To evaluate a population of ABs, first, the distance between the ABs and AGs needs to be calculated. For a given instance of an AG and an AB, (1) calculates the distance for a a single dimension (^{d}) where F^d_{AG} and F^d_{AB} are the respective center points, M^d is the multiplier for that dimension, and RR type (T^d) is the type of UB in that dimension. Further, (2) sums the distance in each dimension and compares it to the radius value (r). A negative result implies the AG is within the RR of the AB implying the AB detects the AG.

$$Df(F^d_{AB}, F^d_{AG}) = \begin{cases} ((F^d_{AG} - F^d_{AB})/M^d)^2 & \text{if } T^d = 0 \\ (F^d_{AG} - F^d_{AB})/M^d & \text{if } T^d = 1 \\ (F^d_{AB} - F^d_{AG})/M^d & \text{if } T^d = 2 \end{cases} \quad (1)$$

$$Detect(ab, ag) = (\sum_{n=1}^{d} Df(F^d_{AB}, F^d_{AG})) - r^2 \quad (2)$$

The fitness in [11] is adapted to work with the new RRs and the changes in chromosome structure. The fitness function (3) consists of four components: Correctness (Cor), Coverage (Cov), Uniqueness (Uni), and Invalid Avidity (InAv) along with weights: $\alpha, \beta, \gamma, \epsilon$ that are provided as hyperparameters. One component, the Valid Avidity, has been removed, as [11] found it to have no impact. A complete description of the components of the fitness function can be found in [11].

$$Fitness = \alpha * Cor + \beta * Cov + \gamma * Uni + \epsilon * InAv \quad (3)$$

E. Survivor selection

The model allows pure elitism and partial elitism during survivor selection, after which further concepts are applied to preserve ratios in the population. First, the model allows for CRL, which is important for preservation of the original class ratios i.e. the ratios of population generated during create ABs — see Fig. 1. Further, to prevent RR types from homogenizing, an RR type locking (RRTL) was also introduced, inspired by CRL, where the ratio of unbounded regions is locked instead of the class. The model allows for 3 types of locking: CRL; RRTL and RRTL with class ratio encouragement. In the latter, ABs are selected by forcing the ratio of RR types whilst ensuring the best class ratio possible.

IV. Experiments and Results

A. Experimental setup

The experiments have been divided into four phases: the first investigates the impact of splitting the AGs across the islands, the second explores the different RRs, the third investigates the impact of different locking schemes, and the fourth compares the proposed model to the state of the art. A subset of the hyperparameters and their values is shown in Tab. I. A complete set of parameters is shown in [27].

TABLE I
Hyperparameters and their values

Parameter	Value	Parameter	Value
NumberOfGenerations	200	MigrationFrequency	0.1
PopulationSizeFraction	1	TournamentSize	2
PercentageOfParents	0.2	BaseRadius	1
NumberOfIslands	4	aScoreMultiplier (α)	3.5
MutationRate	0.4	bScoreMultiplier (β)	1.0
MutationFrequency	0.8	cScoreMultiplier (γ)	1.2
MasterMigrationFreq	1	dScoreMultiplier (ϵ)	2.0
CrossoverRate	0.8	ElitismPercentage	0.1
MigrationRate	0.1	RateOfUnboundedRegions	0.2

TABLE II
Datasets and their properties

Dataset	Size	Dims.	Classes	Class Dist.
Wine	178	13	3	0.40/0.33/0.27
Diabetes	768	8	2	0.35/0.65
Ionosphere	351	34	2	0.36/0.64
Sonar	208	60	2	0.47/0.53
Iris	150	4	3	0.33/0.33/0.33
Glass	214	9	6	0.33/0.35/0.08/0.06/0.04/0.14

TABLE III
Phase 1 - Antigen splitting (Split) with Hypersphere RRs (w/o Split is MAIM [3] executed on the AISIGA simulator)

	Test accuracy (%)		Time per Fold (secs)	
Data	**w Split**	**w/o Split**	**w Split**	**w/o Split**
Wine	92.88(4.97)	**94.69(3.79)**	**0.38(0.08)**	5.82(0.68)
Diabetes	**66.42(1.41)**	65.36(0.55)	**5.36(0.38)**	57.56(0.69)
Ionosphere	**72.98(4.10)**	64.65(1.32)	**3.06(0.29)**	44.14(0.50)
Sonar	**62.43(7.00)**	62.36(7.48)	**2.31(0.21)**	27.18(3.19)
Iris	92.63(4.35)	**93.83(4.58)**	**0.12(0.06)**	1.15(0.06)
Glass	**51.22(7.50)**	50.09(5.73)	**0.41(0.08)**	4.64(0.11)

Six datasets from the UCI Machine Learning Repository (see Tab. II) were selected, providing a variation in the number of features and classes as applied in [3], [11], [24], [25]. The experiments were run on an Intel(R) Core(TM) I7-8750H CPU @2.20GHz with 2×8 GB DDR4 @2400 MHz.

5-fold cross validation was applied. The cross validation process ran 20 times, utilizing the average accuracy and efficiency in the results, including results of the individual folds. T-tests were applied to all results and a detailed analysis may be found in [27]. Results described as significant herein, reflect a P value below 0.05. Further, standard deviation over the 20 runs is shown in parenthesis.

B. Results

The results of the preliminary experiments involved determining the best parameters for the parameters not explored in the later phases of the experiments. The final set of parameters is shown in [27].

*1) **Phase 1 - Splitting the AGs**:* The proposed AG population spread over the islands (w Div.) was compared to the MAIM implementation where the population of AGs is present on each island (w/o Div) – see Tab. III; with the MAIM HS type applied in both cases. A significant increase in efficiency may be observed, confirming the beneficial exploitation of the

TABLE IV
Phase 1 - Antigen splitting (Split) applying HE RRs

	Test accuracy (%)		Time per Fold (secs)	
Data	**w Split**	**Split**	**w Split**	**w/o Split**
Wine	94.72(3.88)	**96.27(3.41)**	**0.60(0.09)**	13.19(1.07)
Diabetes	65.10(0.21)	65.10(0.21)	**15.56(0.79)**	471.80(18.88)
Ionosphere	**68.06(4.43)**	66.55(3.52)	**7.67(0.58)**	193.80(9.63)
Sonar	**63.71(7.81)**	54.59(2.26)	**3.85(0.27)**	82.05(4.01)
Iris	**96.80(2.83)**	96.37(3.30)	**0.16(0.05)**	3.12(0.27)
Glass	55.58(6.42)	**63.64(5.57)**	**0.60(0.09)**	10.84(0.93)

TABLE V
Phase 1 - Antigen splitting (Split) applying HE with UB RRs

	Test accuracy (%)		Time per Fold (secs)	
Data	**w Split**	**w/o Split**	**w Split**	**w/o Split**
Wine	93.18(4.98)	**93.44(5.05)**	**0.65(0.09)**	18.84(1.62)
Diabetes	65.10(0.21)	65.10(0.21)	**16.71(0.64)**	603.88(19.89)
Ionosphere	**66.04(2.77)**	64.42(1.24)	**7.98(0.50)**	229.96(11.10)
Sonar	**62.87(7.06)**	53.64(0.96)	**3.77(0.30)**	87.31(3.07)
Iris	**96.37(3.24)**	96.03(3.32)	**0.19(0.05)**	4.23(0.19)
Glass	57.47(6.00)	**60.15(6.64)**	**0.63(0.09)**	14.42(0.71)

island model sought in AISIGA. The only significant changes in accuracy appear on the Diabetes and Ionosphere datasets, which both show better results for AG splitting.

Tab. IV and Tab. V further investigate the effect of splitting the AGs in the light of HE and HE with UB. Again a significant efficiency benefit may be observed in all datasets for both cases. Although the challenge in the unbalanced datasets of Diabetes and Ionosphere is reflected in all the results, the efficiency of the split is in fact up to 28 times faster, with lower deviation.

When it comes to accuracy there are significant differences in some datasets for HE (Tab. IV), particularly Sonar and Glass, where the split is beneficial in sonar but not in Glass. Higher STD may also be noted in these results too. Similar patterns in the results may be noted for HE with UB. However, it is interesting to note that for Sonar, better but less robust accuracy results can be observed when applying the split.
The class distribution in Glass may be affecting the accuracy results of the split. Glass has 214 data entries, and the smallest class only has 4% of the data points. When the data is split into 5 folds, each fold has 1 or 2 data points of the smallest class, and when it is then further split among 4 islands, not every island will have a data point of the smallest class. As a result, the algorithm could struggle to adapt to the data as there not enough data to train after splitting.

*2) **Phase 2 - RRs with AG Splitting**:* The second phase extends the unbounded RR investigation explored in [11], assuming the AG split, proven in Phase 1 and considering the affects on HE and HS types with an unbounded RR that can expand in either direction — see Sec. III-C.

Comparing HS, as applied in MAIM, with HE in Tab. VI; a significant drop in efficiency may be seen across all datasets. Further, the introduction of HE shows a significant improvement in accuracy, except for the cases of Diabetes and Ionosphere where HS shows a significant improvement. The instability of the glass and sonar results are present in both cases.

When introducing unbounded to HS results, a significant decrease in efficiency across all datasets may be seen, as shown in Tab. VII). However, accuracy shows significant improvement in Iris and Glass, and less significant improvements in other datasets, apart from Diabetes and Ionosphere, where a significant decrease may be observed. Again the unbalanced class distributions of Diabetes and Ionosphere may have affected the results compared to Iris which is balanced.

TABLE VII
PHASE 2 - APPLYING UNBOUNDED RRS TO HYPERSPHERES

	Test accuracy (%)		Time per Fold (secs)	
Data	**HS**	**UB + HS**	**HS**	**UB + HS**
Wine	92.88(4.97)	**93.91(4.56)**	**0.38(0.08)**	0.68(0.10)
Diabetes	**66.42(1.41)**	65.10(0.21)	**5.36(0.38)**	16.28(0.68)
Ionosphere	**72.98(4.10)**	66.39(2.93)	**3.06(0.29)**	8.00(0.52)
Sonar	**62.43(7.00)**	62.28(6.23)	**2.31(0.21)**	3.82(0.46)
Iris	92.63(4.35)	**95.70(3.84)**	**0.12(0.06)**	0.19(0.06)
Glass	51.22(7.50)	**58.21(6.30)**	**0.41(0.08)**	0.62(0.09)

TABLE VIII
PHASE 2 - APPLYING UNBOUNDED RRS TO HYPER-ELLIPSOIDS

	Test accuracy (%)		Time per Fold (secs)	
Data	**HE**	**UB + HE**	**HE**	**UB + HE**
Wine	**94.72(3.88)**	93.18(4.98)	**0.60(0.09)**	0.65(0.09)
Diabetes	65.10(0.21)	65.10(0.21)	**15.56(0.79)**	16.71(0.64)
Ionosphere	**68.06(4.43)**	66.04(2.77)	**7.67(0.58)**	7.98(0.50)
Sonar	**63.71(7.81)**	62.87(7.06)	3.85(0.27)	**3.77(0.30)**
Iris	**96.80(2.83)**	96.37(3.24)	**0.16(0.05)**	0.19(0.05)
Glass	55.58(6.42)	**57.47(6.00)**	**0.60(0.09)**	0.63(0.09)

However, although some classes within Glass are balanced, there are many small classes that are unbalanced — see Tab. II. It is interesting to note the accuracy score between 55% and 60% which is close to the 68% of the balanced classes within the dataset i.e. the algorithm may be ignoring the less balanced classes. These results imply that unbounded does not have the desired effect on accuracy for the more challenging datasets, the datasets that show poor accuracy performance, but shows benefits for the less challenging datasets.

A similar significant decrease in efficiency may be seen when introducing UB to HE — see Tab. VIII; except in the case of Sonar, which shows a small but significant increase in efficiency, with standard deviations being generally low. While it was hypothesized that UB+HE would outperform HE, there was no significant change in accuracy.

Since the focus of the second phase is the introduction of the unbounded types, the unbounded versions are compared in Tab. IX). Although HE was able to outperform HS in terms of accuracy while being significantly less efficient — see Tab. VI, the comparison of the unbounded types showed no clear advantage in terms of efficiency or accuracy. This highlights a need for further investigation into the unbounded types.

TABLE VI
PHASE 2 - HYPERSPHERE RRS COMPARED TO HYPER-ELLIPSOID RRS

	Test accuracy (%)		Time per Fold (secs)	
Data	**HS**	**HE**	**HS**	**HE**
Wine	92.88(4.97)	**94.72(3.88)**	**0.38(0.08)**	0.60(0.09)
Diabetes	**66.42(1.41)**	65.10(0.21)	**5.36(0.38)**	15.56(0.79)
Ionosphere	**72.98(4.10)**	68.06(4.43)	**3.06(0.29)**	7.67(0.58)
Sonar	62.43(7.00)	**63.71(7.81)**	**2.31(0.21)**	3.85(0.27)
Iris	92.63(4.35)	**96.80(2.83)**	**0.12(0.06)**	0.16(0.05)
Glass	51.22(7.50)	**55.58(6.42)**	**0.41(0.08)**	0.60(0.09)

TABLE IX
PHASE 2 - COMPARING UNBOUNDED RRS COMBINED WITH HS AND HE

	Test accuracy (%)		Time per Fold (secs)	
Data	**UB + HS**	**UB + HE**	**UB + HS**	**UB + HE**
Wine	**93.91(4.56)**	93.18(4.98)	0.68(0.10)	**0.65(0.09)**
Diabetes	65.10(0.21)	65.10(0.21)	**16.28(0.68)**	16.71(0.64)
Ionosphere	**66.39(2.93)**	66.04(2.77))	8.00(0.52)	**7.98(0.50)**
Sonar	62.28(6.23)	**62.87(7.06)**	3.82(0.46)	**3.77(0.30)**
Iris	95.70(3.84)	**96.37(3.24)**	0.19(0.06)	0.19(0.05)
Glass	**58.21(6.30)**	57.47(6.00)	**0.62(0.09)**	0.63(0.09)

TABLE X
PHASE 3 - COMPARING RRTL AND CRL APPLIED TO HE WITH UB RRS

	Test accuracy (%)		Time per Fold (secs)	
Data	**RRTL**	**CRL**	**RRTL**	**CRL**
Wine	89.47(6.79)	**93.18(4.98)**	0.86(0.09)	**0.65(0.09)**
Diabetes	**65.12(0.24)**	65.10(0.21)	22.19(0.67)	**16.71(0.64)**
Ionosphere	**67.02(3.23)**	66.04(2.77)	11.80(0.62)	**7.98(0.50)**
Sonar	53.49(0.82)	**62.87(7.06)**	4.95(0.30)	**3.77(0.30)**
Iris	92.77(5.40)	**96.37(3.24)**	0.21(0.05)	**0.19(0.05)**
Glass	54.07(6.63)	**57.47(6.00)**	0.78(0.08)	**0.63(0.09)**

TABLE XI
PHASE 3 - APPLYING RRTL WITH SOFT CLASS RATIO ENCOURAGEMENT (SCRE) TO HE WITH UB RRS

	Test accuracy (%)		Time per Fold (secs)	
Data	**w SCRE**	**w/o SCRE**	**w SCRE**	**w/o SCRE**
Wine	**93.73(4.49)**	89.47(6.79)	0.87(0.09)	**0.86(0.09)**
Diabetes	**65.14(0.29)**	65.12(0.24)	**22.08(0.66)**	22.19(0.67)
Ionosphere	**75.19(3.97)**	67.02(3.23)	**11.70(0.52)**	11.80(0.62)
Sonar	**67.70(6.45)**	53.49(0.82)	5.09(0.33)	**4.95(0.30)**
Iris	**94.47(4.38)**	92.77(5.40)	0.21(0.05)	0.21(0.05)
Glass	**55.33(5.75)**	54.07(6.63)	0.78(0.09)	0.78(0.08)

*3) **Phase 3 - Locking schemes**:* A preliminary investigation into a potential source for the unconvincing results in Tab. IX) involved monitoring the number of unbounded types in the ABs during a single run of the algorithm, where mutation is applied to the RR Types. The results showed that the percentage of ABs with at least 1 UB region increased from 19% to 92% in the evolved AB population. Similarly, the average amount of UB regions of a single AB showed a considerable increase from 19% to 54%. However, interestingly, the introduction of the added direction of the UB regions stayed relatively balanced, starting at 50% each and ending at 55% for upwards and 44% for downwards. As such, there seems to be no evolutionary advantage tied to a particular direction.

To address the large number of evolved UB types, the proposed RRTL enhancement was enforced. However, introducing RRTL, replacing CRL, in fact showed a significant decrease in accuracy in four of the datasets and no significant change in the others — see Tab. X . Further, efficiency significantly decreased for all datasets. Despite the negative results, it was clear from phase 2 that some form of RRTL was still needed.

As described in Sec. III-E, a new version of RRTL was introduced – RRTL with soft class encouragement (SCRE). Comparing the proposed SCRE with RRTL (Tab. XI), a significant increase in accuracy may be observed in most datasets and an insignificant increase in the other datasets. Mixed differences in efficiency are observed and thus the SCRE variant, despite a high standard deviation, is chosen for further experiments.

*4) **Phase 4 - Comparison with State of the Art Methods**:* Given the results of the previous phases, AISIGA with a) AG splitting b) HE + UB RRs and c) RRTL w SCRE is applied in the comparison with other state of the art algorithms. These include MAIM (base algorithm), Unbounded, AISLSF and Ellipsoidal-AIS. Tab. XII clearly highlights the performance strength of the Ellipsoidal-AIS algorithm. Although the original Unbounded algorithm improved on its predecessor MAIM, it also performed significantly better than AISIGA, which added the advantage of islands. One potential reason may lie in the tailored mutation operator applied in Unbounded compared to the simpler operator applied AISIGA. Specifically, the addition of different RRs in the AISIGA increased the complexity of the chromosome structure when compared to a single RR in MAIM. Therefore the simple mutation operator from MAIM could be insufficient and AISIGA might require a more tuned mutation.

TABLE XII
PHASE 4 - ACCURACY COMPARISON AGAINST THE SOTA

Data	**AISIGA**	**Unbounded**	**MAIM**	**AISLSF**	**Ellipsoidal-AIS**
Wine	93.7(4.5)	98.5(2.7)	96.7(0.3)	97.8(5.8)	**99.8(2.6)**
Diabetes	65.1(0.3)	76.2(3.4)	75.7(0.6)	74.2(9.0)	**80.4(2.4)**
Ionosphere	75.2(4.0)	93.6(3.1)	67.1(2.1)	94.4(4.9)	**97.1(1.6)**
Sonar	67.7(6.5)	83.4(10.5)	67.5(8.5)	**88.1(11.8)**	N/A
Iris	94.5(4.4)	96.7(6.5)	96.5(0.6)	95.7(3.8)	**99.8(0.12)**
Glass	55.3(5.8)	71.1(13.2)	64.0(2.1)	**75.4(16.1)**	N/A

Given the results in Tab. III, it was hypothesized that there would be some accuracy improvement over MAIM, given the improved RRs and addition of SCRE. As shown in Tab. XI the AISIGA accuracy results reflect the improvements in the RR selection and the application of SCRE, particularly notable is the Ionosphere dataset where the improvement is reflected as an improvement compared to MAIM. Although accuracy results are otherwise inconclusive between AISIGA and MAIM, it is interesting to note that the published data provided in the MAIM column is higher than those generated through the AISIGA simulator, so further investigation is needed. Further, it is interesting to note that all algorithms show lower stability on Sonar and Glass, where AISIGA shows more stable results.

Despite the clear efficiency benefits of AISIGA compared to the base algorithm MAIM, it is currently not possible to compare efficiency across the different methods due to lack of data and different evaluation platforms. However, it should be noted that Ellipsoidal AIS, the algorithm that performed best, has a time per fold of **119.8 minutes** [24], compared to an average of 6.8 seconds for the proposed AISIGA model.

V. CONCLUSION

A goal of the work herein was to exploit the island model for AIS introduced in [3] by splitting the AGs across the islands. The results highlight that there is a significant efficiency advantage in splitting the AGs. A further goal was to explore the newer unbounded RRs as well as HE RRs. The study provides a deeper understanding of the newer unbounded RRs, not least the need for the proposed SCRE to handle both CRL and RRTL locking.

Future work will involve improvements aimed at increasing accuracy. Mutation improvements should enable further exploitation of the RRs, particularly the unbounded RRs. Further,

currently only a simple migration strategy is implemented and other migration strategies will be investigated. Finally, investigating the performance on different real world datasets will improve the generalizability of the results.

REFERENCES

[1] Y. Zhu, W. Li, and T. Li, "A hybrid artificial immune optimization for high-dimensional feature selection," *Knowledge-Based Systems*, vol. 260, p. 110111, 2023.

[2] J.-q. Li, Z.-m. Liu, C. Li, and Z.-x. Zheng, "Improved artificial immune system algorithm for type-2 fuzzy flexible job shop scheduling problem," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 11, pp. 3234–3248, 2021.

[3] E. Baug and A. Norstein, "Maim: A novel island based evolutionary classification algorithm," Master's thesis, NTNU, 2019. [Online]. Available: https://hdl.handle.net/11250/2625830

[4] J. Molina, L. F. Valdez, and J. M. Gutiérrez, "An artificial immune system algorithm for classification tasks. an electronic nose case study," *Engineering Applications of Artificial Intelligence*, vol. 133, p. 108457, 2024.

[5] S. J. A. Visser, "Aisiga," https://github.com/Storm-Visser/AISIGA, 2026, accessed: 2026-04-15.

[6] J. Marshall, R. Warrington, W. Watson, and H. Kim, "An introduction to immunology and immunopathology," *Allergy, Asthma & Clinical Immunology*, vol. 14, p. 49, 09 2018.

[7] L. N. de Castro and F. J. V. Zuben., "Learning and optimization using the clonal selection principle." *IEEE Transactions on Evolutionary Computation, 6(3):239-251*, June 2002.

[8] A. W. J. Timmis and L. Boggess, "Artificial immune recognition system (airs): An immune-inspired supervised learning algorithm." *Genetic Programming and Evolvable Machines, 5(3):291-317*, September 2004.

[9] A. B. Watkins and L. C. Boggess, "A resource limited artificial immune classifier," in *Proceedings of the 2002 Congress on Evolutionary Computation*. IEEE, May 2002, pp. 926–931.

[10] P. Karpov, G. Squillero, and A. Tonda, "Valis: an evolutionary classification algorithm," *Genetic Programming and Evolvable Machines*, vol. 19, no. 3, p. 453–471, Sep. 2018.

[11] A. C. Ø. Luiggi-Gørrissen and T. J. Woldseth, "Beyond traditional boundaries; exploiting open dimensions for enhancing fake news detection with artificial immune systems," Master's thesis, NTNU, 2023. [Online]. Available: https://hdl.handle.net/11250/3106114

[12] R. K. Das, S. Dash, R. K. Mishra, and A. Panigrahy, "E-clonalg: A classifier based on clonal selection algorithm," *Transactions on Engineering and Computing Sciences*, vol. 4, no. 6, p. 88, Jan. 2017.

[13] H.-S. Huang, "Distributed genetic algorithm for optimization of wind farm annual profits," *2007 International Conference on Intelligent Systems Applications to Power Systems*, pp. 1–6, 2007.

[14] A. L. Corcoran and R. L. Wainwright, "A parallel island model genetic algorithm for the multiprocessor scheduling problem," in *Proceedings of the 1994 ACM Symposium on Applied Computing*, ser. SAC '94. New York, NY, USA: Association for Computing Machinery, 1994, pp. 483–487.

[15] I. C. Jurado and J. Vanschoren, "Multi-fidelity optimization method with asynchronous generalized island model for automl," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, ser. GECCO '22. New York, NY, USA: Association for Computing Machinery, 2022, pp. 220–223.

[16] J. C. Gómez-López, M. Rodríguez-Álvarez, D. Castillo-Secilla, and J. González, "Tuning multi-objective multi-population evolutionary models for high-dimensional problems: The case of the migration process," *Neurocomputing*, vol. 648, p. 130631, 2025.

[17] H. Homayounfar, S. Areibi, and F. Wang, "An advanced island based ga for optimization problems," *Dynamics of Continuous, Discrete and Impulsive Systems Series B: Applications and Algorithms*, pp. 46–51, 12 2003.

[18] A. Kulawik and F. Krużel, "Evoavx: Island-based genetic algorithm library with avx-512 and mpi," *IEEE Access*, vol. 14, pp. 6944–6953, 2026.

[19] A. Gozali and S. Fujimura, *Localization Strategy for Island Model Genetic Algorithm to Preserve Population Diversity*. Springer International Publishing, 05 2018, pp. 149–161.

[20] Q. Meng, J. Wu, J. Ellis, and P. J. Kennedy, "Dynamic island model based on spectral clustering in genetic algorithm," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 1724–1731.

[21] J. Timmis, T. Knight, L. N. de Castro, and E. Hart, *An Overview of Artificial Immune Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 51–91.

[22] E. Hart, "Not all balls are round: An investigation of alternative recognition-region shapes," *Lecture Notes in Computer Science*, vol. 3627, pp. 29–42, 08 2005.

[23] H. Yang, T. Li, X. Hu, F. Wang, and Y. Zou, "A survey of artificial immune system based intrusion detection," *TheScientificWorldJournal*, vol. 2014, p. 156790, 03 2014.

[24] S. Ozsen and C. Yucelbas, "On the evolution of ellipsoidal recognition regions in artificial immune systems," *Applied Soft Computing*, vol. 31, pp. 210–222, 2015.

[25] G. Dudek, "An artificial immune system for classification with local feature selection," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 6, pp. 847–860, 2012.

[26] E. Cantú-Paz, "Migration policies, selection pressure, and parallel evolutionary algorithms," *Journal of Heuristics*, vol. 7, no. 4, pp. 311–334, Jul. 2001.

[27] S. J. A. Visser, "Aisiga: Exploring a hybrid approach of island genetic algorithms and artificial immune systems utilizing unbounded recognition regions," Master's thesis, NTNU, 2025. [Online]. Available: https://hdl.handle.net/11250/5337934