Politecnico di Torino

Optimization methods for engineering problems

Machine Learning through Artificial Immune Systems (AIS)

Barba Vincenzo, *Dipartimento Energia "Galileo Ferraris"*
Lorenti Gianmarco, *Dipartimento Energia "Galileo Ferraris"*
Moraglio Francesco, *Dipartimento Energia "Galileo Ferraris"*
Tolosano Luisa, *Dipartimento Energia "Galileo Ferraris"*

07.06.2022

Outline

- Artificial Immune Systems
- Three paradigms of learning
  - Unsupervised
  - Supervised
  - Reinforcement Learning
- aiNet for data analysis
  - Overview & algorithm
  - Clustering of points in a 2D space
- Dudek's AIS
  - Overview & algorithm
  - Forecasting of electricity load profiles
- Kato and Kobayashi's IS
- Conclusions

Artificial Immune Systems

"Adaptive system[s], inspired by theoretical immunology and observed immune function, principles and models, which are applied to problem solving." de Castro and Timmis

The main motivation for building immune-inspired solutions to engineering problems arises from the following characteristics of IS:

- **self-organization** of huge numbers of immune cells;
- **distributed operation** throughout the body;
- **pattern recognition** and anomaly detection (**self/nonself discrimination**);
- **optimization** and **memory** to improve and remember immune responses.

Characteristics of the **immune system** as a **defence system**:

- **specialized cells;**
- subdivided into:
  - **innate immunity**: generic defence acting on general classes of pathogens (does not adapt in the lifetime)
  - **adaptive immunity**: ability to adapt and react to unseen pathogens based upon exposure to them (**learning mechanism)**
- **recognition mechanism**: protein molecules on the immune cell surface act as biochemical receptors.

Four main classes of AIS algorithm, inspired by different theories of the IS:

- **Clonal Selection** (affinity maturation)
- **Immune Network** (self-organization of the IS as a regulated network)
- **Negative Selection** (based on the concept of self–nonself discrimination choice)
- **Danger Theory** (danger-theory-inspired systems based on the recognition of danger zone where the intruder acts)

Three paradigms of learning

UNSUPERVISED LEARNING (UL) :

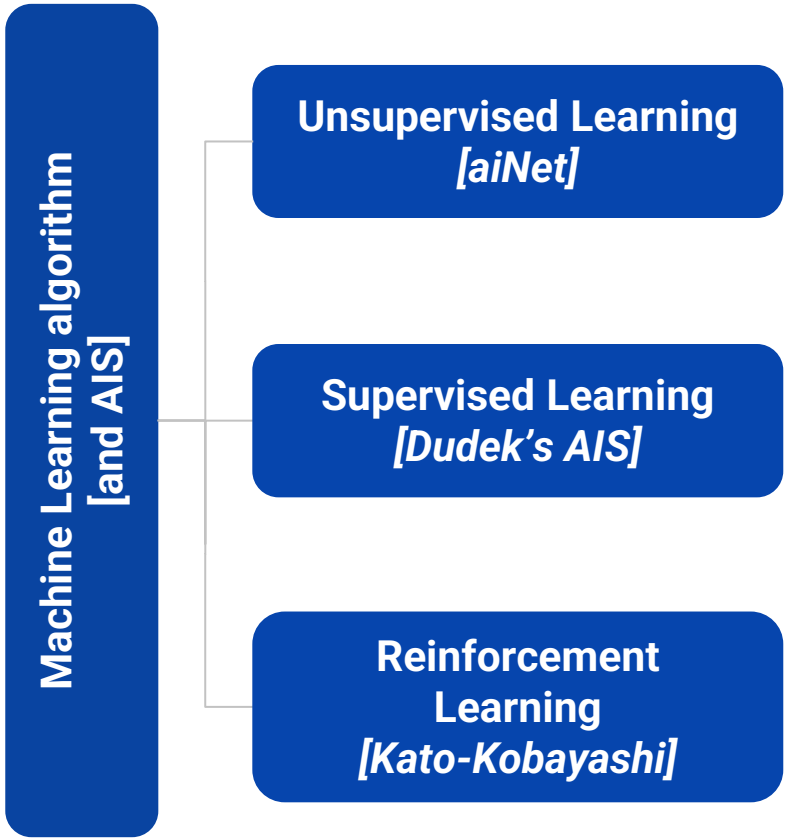
- **learning patterns from untagged data** (e.g. clustering, pattern recognition)
- **self-organization** to capture **patterns** as probability densities or a combination of neural feature preferences.
- two popular methods are **k-means** and **Self-Organizing Maps**.

SUPERVISED LEARNING (SL) :

- task of learning a **mapping function** between input and output
- training samples of input-output pairs are used (**instructive feedback**).
- an SL algorithm uses the training data to **optimize** its mapping function, which can be used for mapping **new** samples (unlabelled data)

REINFORCEMENT LEARNING (RL):

- an **agent** interacts with an **environment** through a sequential process: observe state, take action, get reward .
- the agent does not receive as feedback which action was best but how the action taken was good (**evaluative feedback**)
- the goal of the agent is maximising the **cumulative reward** over time



Richard S. Sutton, Incomplete Ideas. [Online]. Available at: http://www.incompleteideas.net/

aiNet for data analysis

"The aiNet is an edge-weighted graph, not necessarily fully connected, composed of a set of nodes, called antibodies, and sets of node pairs called edges with an assigned number called weight […]."

- the learning algorithm aims at **building a memory** set of antibodies that recognize and represent the **structural organization** of the input **unlabelled** data (i.e. AGs).
- memory set is a **network** composed of AB cells (nodes) interconnected by links (edges) with associated connection strengths, depending on the Ab-Ab affinity (weights)
- this network is the **internal image** of input patterns (AGs).
- **affinity** between ABs and ABs or AGs is inversely proportional to their distance in the **shape space** (L-dimensional space)
- the algorithm makes use of two concepts from AIS theory:
  1. **clonal selection** (and affinity maturation) to locally improve each antibody (**exploitation**)
  2. **network suppression** to eliminate memory cells that are too close to each other (**exploration**)
- the AIS algorithm **does not** perform clustering directly, but it uses **hierarchical clustering** techniques on the aiNet graph.

aiNet algorithm

1. load set of **antigens** $Ag_j \in S_{Ag} \subset \mathbb{R}^{n_{Ag} \times L}$
2. randomly initialize set of **antibodies** $Ab_i \in S_{Ab} \subset \mathbb{R}^{n_{Ab} \times L}$ and **empty memory** set M
3. for $t = 1, \dots, t_{max}$ do:

   3.1 for $Ag_j \in S_{Ag}$ do:

   3.1.1 compute $f_{i,j}$ **affinity** with Ag_j for each $Ab_i \in S_{Ab}$

$$D_{i,j} = \|Ab_i - Ag_j\|$$

$$f_{i,j} = 1/D_{i,j}$$

   3.1.2 **select** n_{best} antibodies with highest affinity to Ag_j

   3.1.3 generate $n_{c,i}$ **clones** $Cl_{k,i}$ for each best antibody (clonal pool)

$$n_{c,i} = n_{Ab} \left(\frac{f_{i,j}}{f_{max,j}} \right)$$

   3.1.4 **mutate** clones with rate inversely proportional to parent's affinity

$$Cl_{k,i} = Ab_i + \alpha_i \, rand(\quad)(Ag_j - Ab_i)$$

$$\alpha_i = \beta \, e^{-f_{i,j}}$$

   3.1.5 put percentage of best clones (highest affinity) into **clonal memory**

   3.1.6 perform ***apoptosis*** (remove clones distant from antigen)

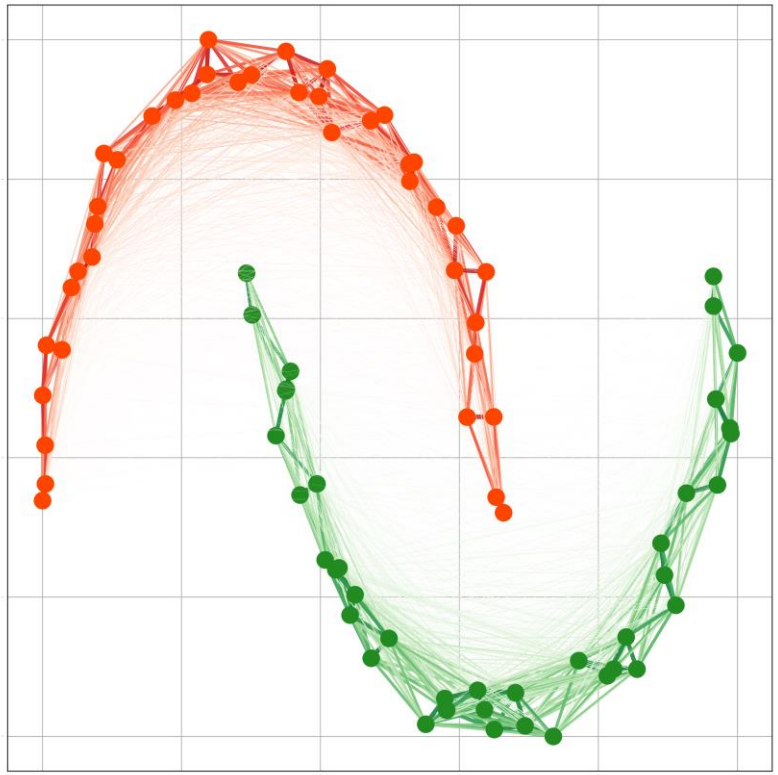
   3.1.7 perform **clonal suppression** (remove clones close to each other)

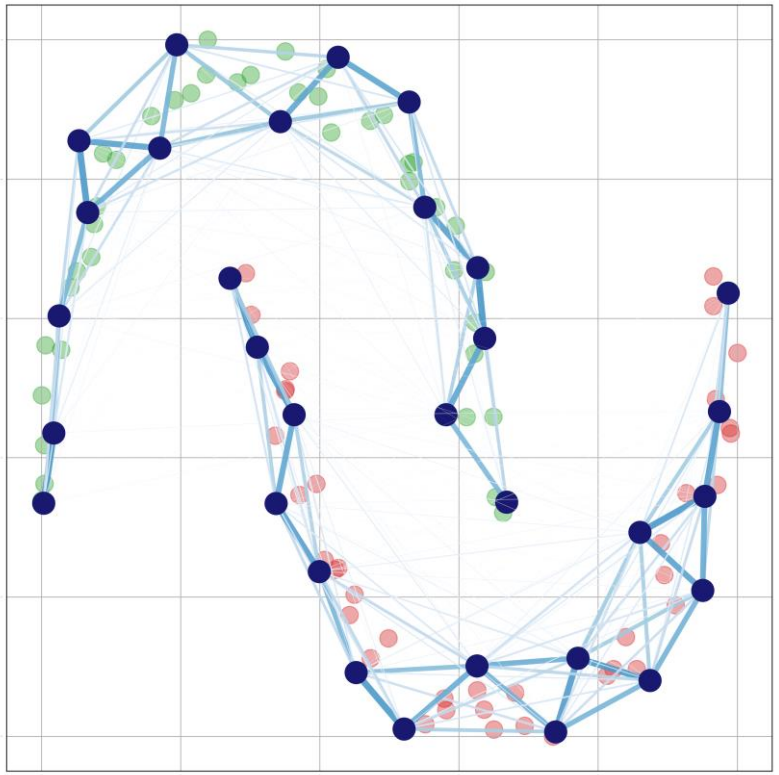
   3.1.8 put surviving clones in **memory** set M

   3.2 perform ***network suppression*** (remove antibodies in M close to each other)

   3.3 **update** S_{Ab} with M *and* n_{new} *randomly generated elements*
4. get final memory set M (**nodes**) and calculate **weights** of Ab-Ab connections (affinity)

Results on moons dataset





- data divided into **2 non-convex** clusters (moon-shaped)
- clustering performed through **hierarchical clustering** (graph analysis)
- results of clustering on the **whole** dataset (left)
- results of clustering after using aiNet to **reduce** number of samples (right)
- the algorithm is still able to find the two clusters while working on a smaller (**representative**) set

Dudek's AIS for regression (SL)

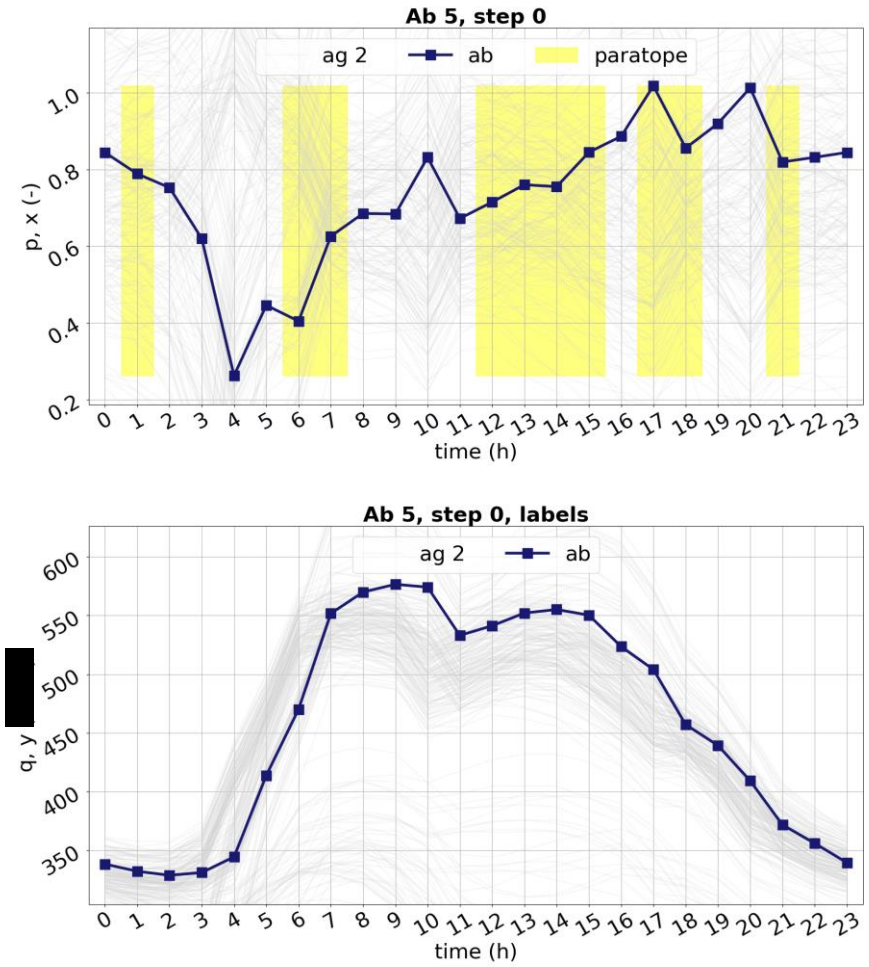
The Dudek's model is based on AIS for **forecasting time series** (e.g. short-term electrical load). The main feature is the **embedded** property of **local selection**: each antibody learns its optimal subset of features (a paratope) to improve its recognition and prediction abilities.

- AGs are input-output pairs that represent the **training samples**:
    1. vector of **features**, **x**, i.e. input variables, (e.g. day-ahed load profile)
    2. label, **y**, containing the related **ouput** variables (e.g. time-series of the load of the following day)

- ABs are expected to **cover** regions of the input space and to provide an output, hence they are composed of:
    1. pattern that **matches** the antigens features (**p**)
    2. label, **q**, containing the **predicted** output values
    3. paratope, **Ω**, i.e. a subset of the active features (**feature selection**)
    4. radius of the **recognition region**, which is an hyper-ball in the paratope sub-space
    5. power, i.e. **number** of antigens with the antibody is able to recognize

- **AB mutation** acts on paratopes.
- System goes on alert whenever an AG is not recognized (nonself), thus allowing further analysis.

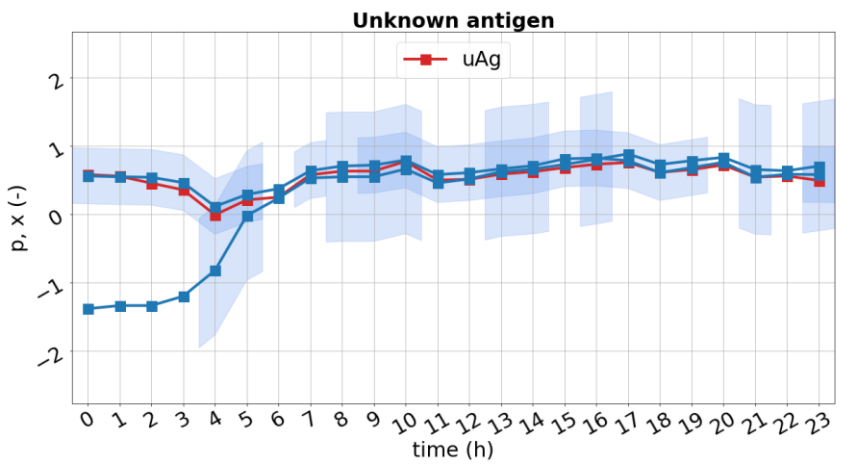
Dudek's algorithm

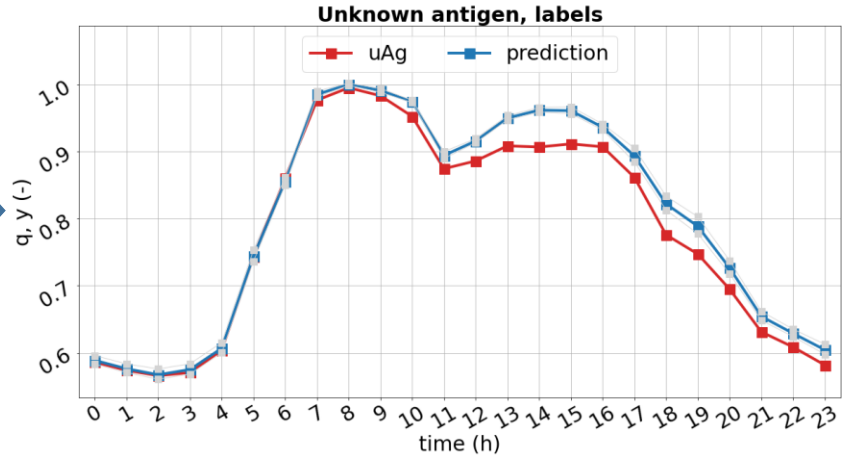


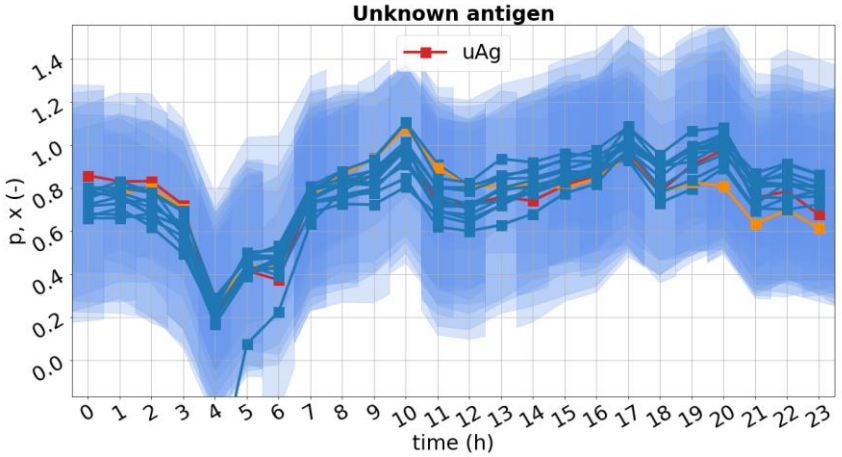
1. load **antigens** population $S_{Ag} = \{Ag_j = (\boldsymbol{x}_j, \boldsymbol{y}_j), \boldsymbol{x}_j \in \mathbb{R}^{L_x}, \boldsymbol{y}_j \in \mathbb{R}^{L_y}\}_{j=1,..,N}$
2. initialize set of **antibodies** $S_{Ag} = \{Ab_i = (\boldsymbol{p}_i, \boldsymbol{q}_i), \boldsymbol{p}_i = \boldsymbol{x}_j, \boldsymbol{q}_i = \boldsymbol{y}_j, \boldsymbol{\Omega}_i, r_i, P_i\}_{i=1,..,N}$
3. for $k = 1, \dots, N$ **do**:
   - 3.1 **select** current antibody $Ab = Ab_k$ (parent)
   - 3.2 while not stop condition **do**:
     - 3.2.1 for $l = 1, \dots, Z$ **do**:
       - 3.2.1.1 create **clone** $Cl_l = Ab$
       - 3.2.1.2 perform hypermutation on **paratope** $\boldsymbol{\Omega}_l = hyp(\boldsymbol{\Omega}_l)$
       - 3.2.1.2 compute **cross-reactivity** threshold $\mathrm{r}_l = cr(\mathrm{r}_l)$
         - **divide** antigens into class1 and class2 (error on labels)
         - identify **closest** antigen of class2, B, ($d_{l,B}$*)
         - select **farthest** antigen of class1, A, ($d_{l,A}$*) s.t. $d_{l,A} \leq d_{l,B}$
         - **evaluate** $r_l = d_{l,A} + C\left(d_{l,B} - d_{lA}\right), \mathrm{C} \in [0,1] \subset \mathbb{R}$

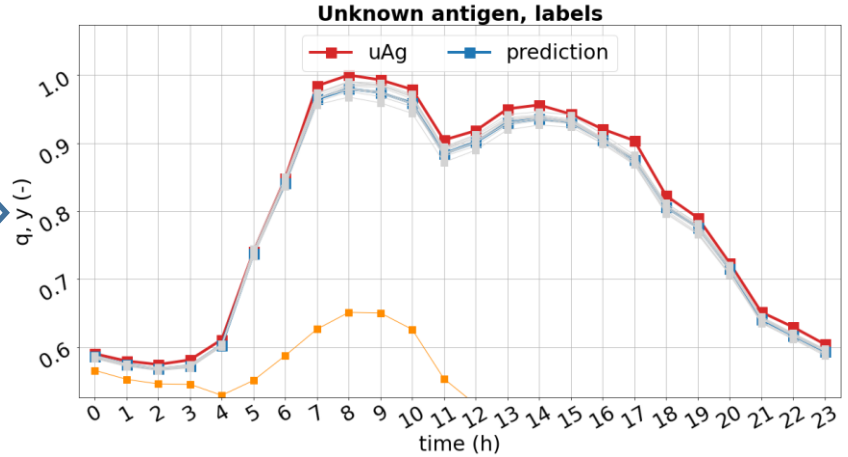
         * $d_{l,j} = \left(\sum_{t \in \Omega_i} \left|p_{l,t} - x_{j,t}\right|^e\right)^{\frac{1}{e}}, \; e = 1 \; or \; 2$
       - 3.2.1.2 compute **affinity** of antigens in class1 (linear function of $d_{l,j}$)
       - 3.2.1.3 compute **power** (number of antigens with affinity larger than 0)
     - 3.2.2 **rank** clones based on: 1. Power, 2. Paratope size, 3. Random choice
     - 3.2.3 select winner to **replace** the parent $Ab_k \leftarrow winner\left(\{Cl_l\}_{l=1,..Z}\right)$
     - 3.2.4 update **label** based on labels of antigens (weighted on the affinity)

Results on electric load forecasting







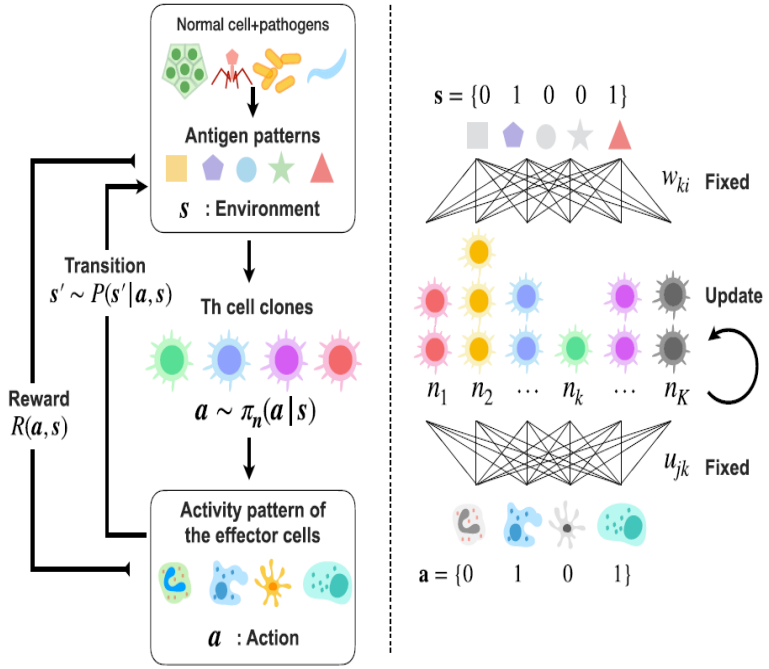


- algorithm trained on 1-years EGEA data of day-next day pairs
- train/test ratio equal to 0.8
- 4.3% MAPE on recognized Ags (38%).
- the figures show how the recognition of **new antigens** works (left plots)
- trained antibodies that **recognise** the new antigen provide their **response** (label, i.e. prediction)
- a **global** response is obtained using weights (affinity of each antibody)
- the **predicted** value is compared with the **actual** output from the test set.

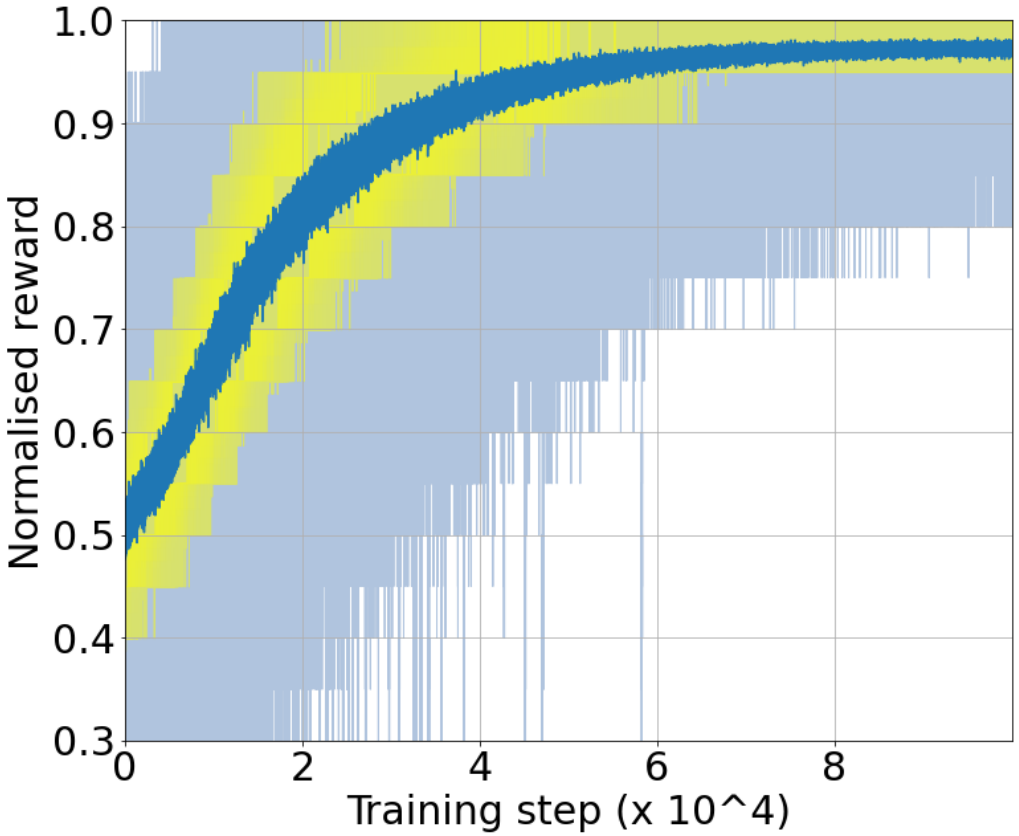
Kato-Kobayashi's model of IS

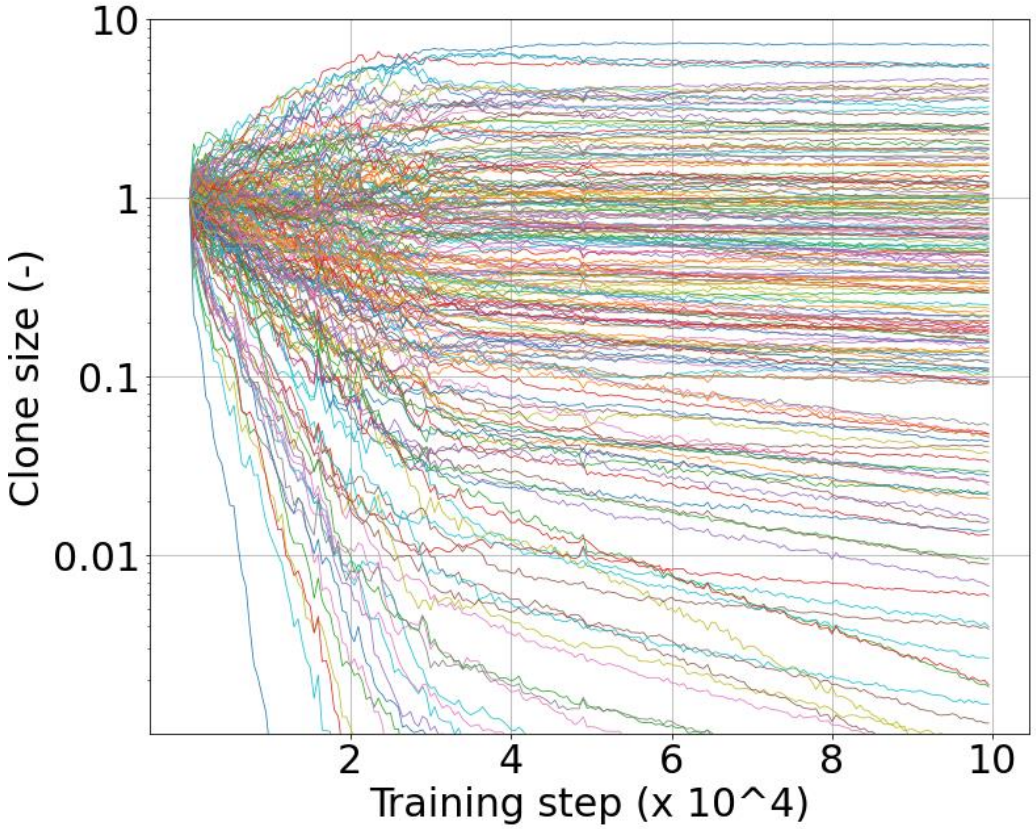
The vast majority of AIS operate at the **individual** level: a clonal selection loop evolves Abs so that offspring tend to posses higher recognition capability. But what if focus on the **population** level?

- Kato and Kobayashi attempt to model **population dynamics** of AIS using RL, thus highlighting several parallels between the two fields.
- based on this assumption, they aim at modelling the **learning dynamics** of the IS which is exposed and has to respond to a set of AG patterns (binary vectors).
- the system's response works on **two** levels:
    1. Different types of **Th cells** recognize the AGs (states) and produce cytokines.
    2. **Effector Cells** understand cytokine production and consequently induce immune response activity (action).
- the system learns how to bias the activity of Th-cells only by adjusting the **distribution** of the sizes of each type: this is realized with the help of **SARSA** algorit
- The RL model induces optimal neural behavior in the cell population: overall net activity provides an estimate of the **Q-function**.
- The body (environment) responds to the IS (agent) with a **reward** and a **transition** depends probabilistically on the action induced by the IS



Results of training





Conclusion

- As stated by Timmis in several works, the application of AIS often failed (or proved ineffective) because of unproper treatment and misunderstanding.
- The immune metaphor still has a lot of potential to express in the domain of **Computational Intelligence** and, w.l.o.g., in optimization settings that can be different from traditional multiobjective problems.
- Statistical Machine Learning seems to us the enabling formal framework to properly analyze, treat and exploit the features of immune systems.
- We believe the most promising paradigms for immune-ispired learning are SL and RL. Particular attention should be given to the investigation of Transfer and Incremental Learning methods.

Politecnico di Torino

Machine Learning through Artificial Immune Systems

Thanks for your attention !

Barba Vincenzo, *Dipartimento Energia "Galileo Ferraris"*
Lorenti Gianmarco, *Dipartimento Energia "Galileo Ferraris"*
Moraglio Francesco, *Dipartimento Energia "Galileo Ferraris"*
Tolosano Luisa, *Dipartimento Energia "Galileo Ferraris"*

07.06.2022