CHAPTER 3

Cellular Frustration Model

The algorithms that solve the Stable Marriage Problem (SMP) create arrangements where all agents are paired in stable bonds. Stable bonds are created when paired agents cannot be destabilized by interacting with other agents. Therefore, solving the Stable Marriage Problem amounts to finding such stable arrangements. [31] The inspiration for the CF model was in part taken from this model, but it operates differently.

With CF algorithms the goal of the dynamic is the opposite. Instead, agents should be designed to engage in a perpetual unstable dynamic, therefore never reaching configurations where all agents are in stable pairings. This dynamic depends on the information presented by one set of agents and contained in a sample of a data set. The principle is that if agents have been prepared to engage in a very unstable dynamic when samples from a normal class are presented, then when samples from an abnormal class are displayed the population should engage in a less frustrated dynamic.

Based on this principle, these algorithms give good results in anomaly detection tasks when compared to currently popular anomaly detection algorithms. More interestingly, since these algorithms derive from models appropriate to describe cellular interactions in the real immune system, this work can gain relevance for building a deeper understanding of how the immune system works [32].

In this chapter, the concepts and terminology that define the CF model are presented. These build upon those presented in [7] and [33].

3.1 Terminology And Definitions

Definition 1 (Agent). *An agent $a_i \in A \; \forall i \in \{0, 1, 2, \ldots, N_A\} \colon N_A = |A|$, is defined as a tuple $a_i = \{s_i, l_i, K_i, m_i\}$, where:*

- *s_i is the signal shown by the agent;*
- *l_i is a sequence of signals called the global preference list of the signals shown by the agents of the opposite type;*
- *K_i is the set of agents with which an agent a_i can interact with;*

- m_i *is the state of an agent* a_i *and contains the index* i *of the agent with which it is matched, or the value* -1 *if it is unmatched.*

Definition 2 (Agent types). *The set of agents* A *is made up of two subsets of agents, where in one exist presenters* P*, and in the other, detectors* D*, such that* $A = P \cup D$ *and* $P \cap D = \emptyset$*. The total number of presenters and detectors is given by* $N_P = |P|$ *and* $N_D = |D|$*, respectively. Hence, the total number of agents is given by* $N_A = N_P + N_D$*.*

Definition 3 (Subtypes). *In this model there can be subtypes for each type of agent. The subtypes of presenters are defined by the ordering of their list of signals* l_i*, where the first signal* s_i *of that list determines its subtype. On the other hand, the subtypes of detectors are defined by the signal* s_i *that they show. A pair of matched agents of the same subtype is called* M*, whereas if their subtypes are different it is called* $\overline{M}$*.*

Definition 4 (Signals). *The set of signals shown by presenters is given by* $X_P = \{s_i \colon s_i \in X \subset \mathbb{R} \wedge a_i \in P\}$*, whereas the set of signals shown by detectors is given by* $X_D = \{s_i \colon s_i \in X \subset \{1, 2\} \wedge a_i \in D\}$*. While presenters can show many signals, detectors only show one signal, which is equal to their subtype. To clarify when one or the other is being referenced, the notation* s_i^P *will be used when mentioning the presenters' set of signals and* s_i^D *when mentioning the detectors' set of signals.*

Definition 5 (Rankings). *The preference lists* l_i*, are ordered according to the preferences of their agents'* a_i*, towards the signals contained in them, with each signal* s_i *having a ranking* $r_{l_i}(s_i)$ *associated with it. Each ranking is simply the index of a signal in* l_i*. A lower index rank is preferred over a higher index one.*

Definition 6 (Detectors' Domains). *Each detector has a normal domain,* $\mathcal{I}$*, and an abnormal domain,* $\mathcal{O}$*, for each feature's distribution of a cluster of samples. A randomly generated threshold probability,* ν_i*, sets how many samples are included in* $\mathcal{I}$*, as* $N_S = 100 - 100\nu_i\%$*, with* $\mathcal{O}$ *having the remaining.*

In statistics the normal domain would be the acceptance region of a distribution, while the abnormal domain would be the rejection region.

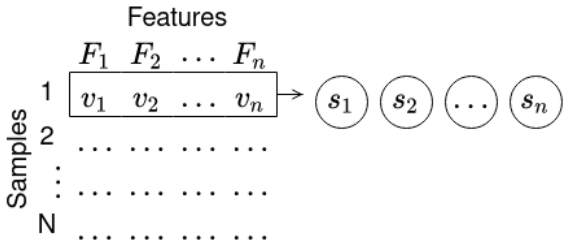
Definition 7 (Detectors' Critical Values). *Each* $\mathcal{I}$ *has symmetric critical values* $z_{\nu_i/2}$ *calculated from the normalized features',* F_f*, distributions of values* $v_f \forall S_u$*, of samples,* S*, within it, that give it lower and upper bounds. The lower bound is given by* $z_{\nu_i/2} = \min\{v_f \forall S_u\}$*, while the upper bound is given by* $z_{1-\nu_i/2} = 1 - z_{\nu_i/2}$*.*

See Figure 3.3 and Figure 3.4 to understand how Definition 6 and Definition 7 relate to the way detectors see signals s_i as either being inside or outside their normal domains.

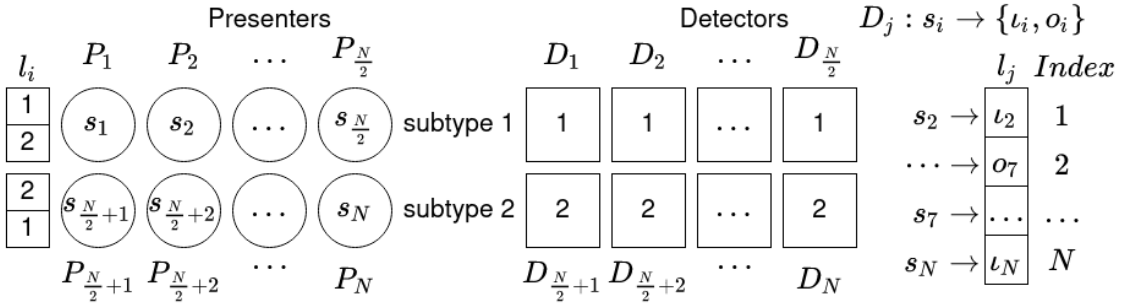
Definition 8 (Signals Mapping). *Detectors establish a mapping,* M*, of signals,* s_i^P*, as* $M : s_i \to \{\iota_i, o_i\} \forall a_i \in P$*, where signals* $\iota_i \in \mathcal{I} \subset X_P \wedge o_i \in \mathcal{O} \subset X_P : \mathcal{I} \cup \mathcal{O} = X_P \wedge \mathcal{I} \cap \mathcal{O} = \emptyset$*. Therefore, detectors see the continuous values shown by the signals* s_i^P *as binary signals.*

See Figure 3.1, Figure 3.3 and Figure 3.4 for a visual representation of how signals are mapped.

Definition 9 (Interactions Between Types Of Agents). *Only interactions between agents of different types are allowed, such that* $K_i \subseteq D \; \forall \; a_i \in P$ *and* $K_i \subseteq P \; \forall \; a_i \in D$. *Furthermore, there can only be pairs of agents, e.g., one agent bound to another.*

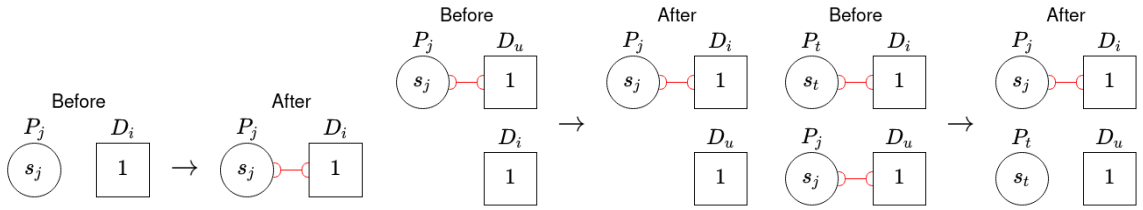


(a)



(b)

Figure 3.1: CF model's mapping of information from the samples in a data set onto presenters and ultimately detectors' preference lists. (a) Shows how each sample's information is mapped onto a set of presenters signals. (b) Shows the preference lists of both presenters and detectors and how the signals they show are mapped onto those lists. Note that presenters show a wide range of signals while detectors only show the signals 1 and 2, which are ranked in the presenters' preference lists according to their subtype, meaning presenters will always prefer detectors that have the same subtype as them. The signals shown by presenters are mapped onto a binary set of signals that represent the signal either being inside the normal domain of a detector or outside it.



(a) **(b)** **(c)**

Figure 3.2: The three fundamental ways agents interact when deciding who to pair with. (a) When both agents are unpaired. (b) When all agents except one are paired. (c) When all agents are paired.

Definition 10 (Decision Rules). *To update the states* m_j, m_t, *of agents* $a_j, a_t \in P$, *and* m_i, m_u, *of agents* $a_i, a_u \in D$, *when these interact with each other, it is necessary to follow the*

update rules R:

1. *if* $m_i = -1 \wedge m_j = -1 \Rightarrow m_i = j, m_j = i$.
2. *if* $m_i = t \wedge m_j = -1 \wedge r_{l_i}(s_j) < r_{l_i}(s_t) \Rightarrow m_i = j, m_j = i, m_t = -1$.
3. *if* $m_i = -1 \wedge m_j = u \wedge r_{l_j}(s_i) < r_{l_j}(s_u) \Rightarrow m_i = j, m_j = i, m_u = -1$.
4. *if* $m_i = -1 \wedge m_j = u \wedge r_{l_j}(s_i) = r_{l_j}(s_u) \wedge s_j \in \mathcal{I}_i \wedge s_j \in \mathcal{O}_u \Rightarrow m_i = j, m_j = i, m_u = -1$.
5. *if* $m_i = t \wedge m_j = u \wedge r_{l_i}(s_j) < r_{l_i}(s_t) \wedge r_{l_j}(s_i) < r_{l_j}(s_u) \Rightarrow m_i = j, m_j = i, m_t = -1, m_u = -1$.
6. *if* $m_i = t \wedge m_j = u \wedge r_{l_i}(s_j) < r_{l_i}(s_t) \wedge r_{l_j}(s_i) = r_{l_j}(s_u) \wedge s_j \in \mathcal{I}_i \wedge s_t \in \mathcal{O}_i \wedge s_j \in \mathcal{O}_u \Rightarrow m_i = j, m_j = i, m_t = -1, m_u = -1$.

It becomes apparent that the agents follow a greedy strategy, since each agent swaps pair whenever they have the opportunity to pair with a preferred agent.

It is important to note the distinction between these rules and the ones considered in the original CF model used as reference in this work. Originally, only rules 1, 2, 3, and 5, existed, with no rule ever mentioning the states of the signals as either being inside or outside a detector's normal domain.

Lemma 1. *Only the R rules, 1, 5, and 6, change the number of unmatched agents.*

Proof. From Definition 10 it can be taken that when all the rules have had their conditions fulfilled, only rule 1 lowers by two the number of unmatched agents and only rules 5, and 6, equally increase that same number, whereas the remaining rules do not change the number of unmatched agents. □

Definition 11 (Stable Matching)**.** *A matching between two agents, a_i and a_j, is said to be stable, when the successive use of rules, R, on every other agent does not change the states, m_i and m_j, of those two agents.*

Definition 12 (Stochastic Iteration)**.** *During a stochastic iteration and given a sequence in which all N_A agents are included, each one of these agents will be sequentially randomly chosen to interact with another agent of the opposite type, which itself was chosen randomly. Next, the states, m_i, of these agents will be updated according to rules, R.*

Definition 13 (Population Configuration)**.** *A population configuration, C, is given by the set of matched agents along with the set of unmatched agents: $C = \{a_i, a_j \colon m_i = j \wedge m_j = i \colon i, j = 1, \ldots, N_A\} \cup \{a_i \colon m_i = -1 \colon i = 1, \ldots, N_A\}$.*

Definition 14 (Stable Configuration)**.** *A configuration, C, is said to be stable when all agents that are within it are in a stable matching.*

Despite being possible to achieve a stable configuration, it is important to note that the probability of a population of agents reaching such a configuration is tiny.

Lemma 2. *The probability of a population, with equal number of presenters and detectors, staying in an unstable configuration by the end of an iteration is* $\frac{N_A}{2} - 1$ *greater than ending up in a stable configuration.*

Proof. In the worst case scenario, consider a population that is just one iteration away from ending up in a stable configuration. This can be achieved by unmatching two agents bound to each other in a population already in a stable configuration. Furthermore, assume that the next iteration involves one of these unmatched agents.

The probability of that agent choosing the only other unmatched agent and therefore returning to a stable configuration is $\frac{1}{\frac{N_A}{2}} = \frac{2}{N_A}$. Therefore, the probability of the population remaining unstable is $1 - \frac{2}{N_A}$. Since $\frac{1-\frac{2}{N_A}}{\frac{2}{N_A}} = \frac{N_A}{2} - 1$ this means the probability of the population remaining unstable is $\frac{N_A}{2} - 1$ greater than ending up in a stable configuration, with this probability increasing the more agents that are involved. □

Definition 15 (Matching Lifetime)**.** *The lifetime of a matching between two agents is determined by the number of iterations that elapsed between the iteration when they matched and the iteration when they unmatched. This is represented by* $\tau_i \in \mathbb{N}$.

3.2 Procedure And Mechanisms

The CF model has several steps and mechanisms that allow it to train detectors and consequently perform detections on data sets. The most important mechanisms will be explained in some detail in order to give a deeper and clearer understanding of the inner workings of the model.

3.2.1 Feature Space Normalization

It is important to normalize feature values in data sets in order to facilitate the mapping of the signals as explained previously. The min-max normalization described in Equation 2.1 was used in this model and the values were mapped to the range $[0, 1] \in \mathbb{R}$.

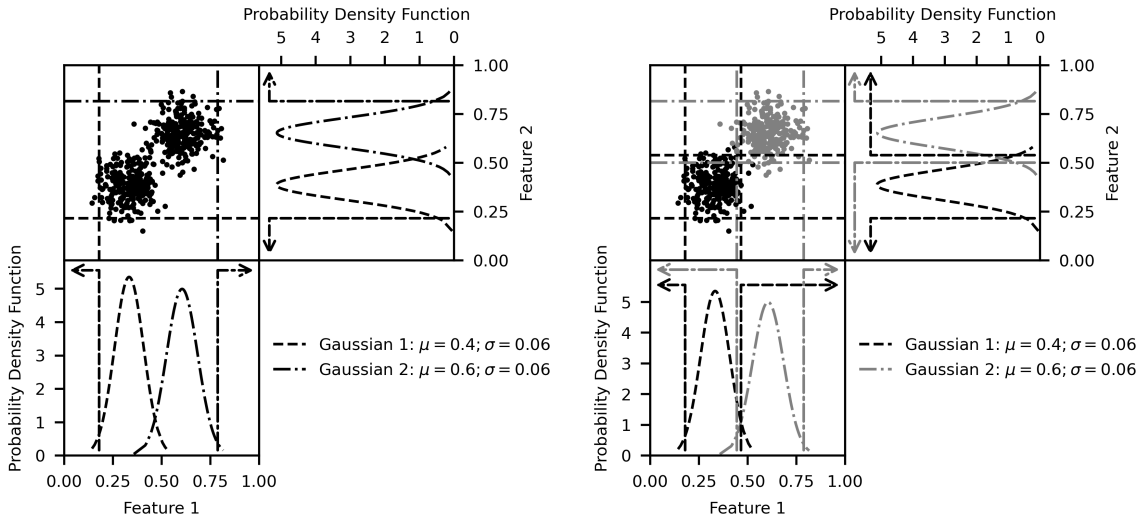
3.2.2 Feature Space Partitioning

One major problem with CF is that detectors require an accurate partition of the feature space, otherwise there will be many false positives or false negatives, hence no detection at all. One way to solve this is by partitioning the feature space into clusters. This way detectors see less feature space as being normal. Note that since detectors take the distribution of values within each feature and divide it into normal and abnormal domains, it becomes crucial to ensure these domains are properly defined.

Figure 3.3 shows the difference between a clustered and non-clustered data set and how this impacts a detector's ability to detect anomalies. In this example K-means clustering with $K = 2$ was used.

After this partitioning, detectors are then able to correctly map the continuous feature values to the binary signals shown by presenters during training, as either being inside or

outside their normal domains. Figure 3.4 describes this in terms of the cumulative distribution function $F_i(z_{\nu_i/2})$, estimated from the data available for training, that is considered when looking at the distribution of a feature's values.



(a) **(b)**

Figure 3.3: Training samples distributions projections (a) without and (b) with clustering. In (a) there is only one cluster, therefore all detectors base their view of the feature space on the left tail of Gaussian 1 and the right tail of Gaussian 2. This has the consequence of making the detectors see a lot of the abnormal feature space as normal, which is the case in [7]. In (b) detectors are assigned to each cluster, therefore they are able to partition the feature space better. This way they are able to better distinguish normal regions of space from abnormal ones. In tests discussed later on, detectors were distributed evenly across all clusters considered during training.

Note how detectors in this work have two tails, which means they should be more robust to false negatives when abnormal feature values show on either side of the feature space distribution, but this might also mean that more normal values will be considered abnormal during testing, hence more false positives. In [7] detectors only have one tail, hence one abnormal domain with which to detect abnormal samples. This means that even if a very obvious abnormal feature value appears in a sample inside the normal domain, it will be considered as a normal value, leading to a false negative. It becomes apparent that the approach in this work is more aggressive than the one in [7] regarding the mapping of feature values into normal and abnormal signals.

During testing, it was noticed that detectors specialized to their assigned cluster (i.e., normal domains based solely on the features' distributions of samples from their cluster), had difficulty dealing with samples from others clusters, because to them anything outside their cluster was abnormal, therefore many false positives would ensue. This led to the development of the shuffling procedure, which is quite simply randomly swapping detectors' pairs of left and right critical values of each feature with other detectors. This led to more robust detections, since now when detectors saw samples from different clusters than the one they were assigned to originally, they would see those as only partly abnormal, which means they would see some features as normal, leading to less false positives.

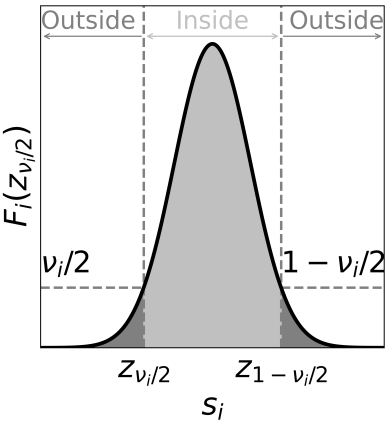


Figure 3.4: Mapping of normalized feature values s_i into binary signals ι_i and o_i. In order for detectors to map signals s_i onto signals inside or outside their normal domain, they must calculate $F_i(z_{\nu_i/2})$ for each feature. When s_i lies in the left, or right, tail of the distribution, $F_i(z_{\nu_i/2}) \leq \nu_i/2$ and $F_i(z_{1-\nu_i/2}) \geq 1 - \nu_i/2$, respectively, and it will be mapped onto a signal o_i, otherwise it will be mapped onto a signal ι_i. $\nu_i \sim U(0, \nu_{max})$, where typically $\nu_{max} \leq 0.2$.

3.2.3 Cellular Frustration Dynamic

There are three main stages when using the CF model, and these are training, calibration, and monitoring. Although they have different purposes and functions, the main loop is the same. This loop is described in Algorithm 3.1. Note that the total number of iterations the algorithm executes is n_{max} and the period between changing samples is T, which typically was $T = 100$.

Algorithm 3.1 Cellular frustration dynamic.

```
Initialize detectors' global preference lists with signals ι_i and o_i randomly ranked
{τ_i} ← 0
for all n ∈ [1, n_max] do
    if n mod T is 0 then
        Change sample shown by presenters
    for all a_j ∈ P ∪ a_i ∈ D do
        Pick a random presenter a_j
        Pick a random detector a_i
        decision(A, a_i, a_j) - apply decision rules in Definition 10
    for all a_i ∈ A do
        if a_i ∈ A is paired then
            τ_i ← τ_i + 1
```

3.2.4 Sample Rotation

Presenters show many samples from the data set throughout the CF dynamic, therefore, there has to be some method of selecting samples from it. Some different methods are: sequentially in the same order as they are stored in the data set; if clustering was applied to the data set, sequentially but on a per-cluster basis, selecting all samples from each cluster before moving on to the next cluster; completely random; randomly selecting a cluster and then randomly selecting a sample from that cluster. The second method was chosen.

3.2.5 Agent Dissociation

To prevent two agents from being paired for too long, possibly even during the entirety of the dynamic, when an agent is selected to take its turn in an iteration, there is a chance that it will become unmatched. This does not significantly alter the results since the probability of this dissociation happening is fairly low.

3.2.6 Detectors' Preference Lists

Detectors have two preference lists, one global and one local. The global list has all the possible binary signals each presenter can show, as defined in Definition 1. The local list only has the binary signals currently shown by presenters given the selected sample, as seen from each detector's perspective regarding its normal and abnormal domains, as seen in Figure 3.5 with reference to Figure 3.6. This list is used for computational reasons, to decrease access times to signals.

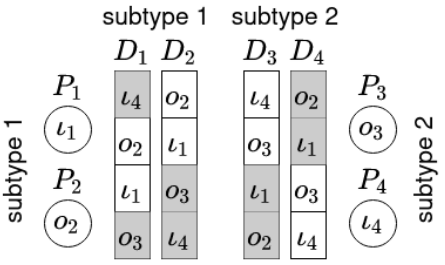


Figure 3.5: Detectors' local preference lists for a sample.

The global preference list of every detector is initially randomized, such that the binary signals corresponding to each presenter are scattered throughout the list without any ordering. Later on there will be some ordering of these signals when education is introduced. These unordered lists can be seen as per the example in Figure 3.6.

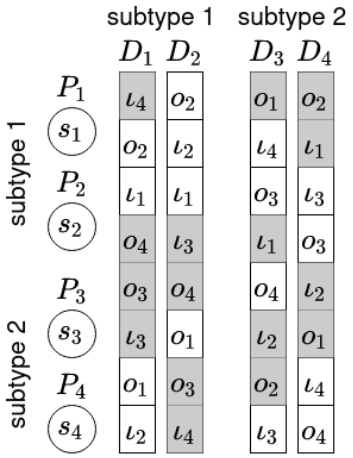


Figure 3.6: Detectors' unordered global preference lists l_i, before training. In this case $N_P = N_D = 4 \implies N_A = 8$, which means there are 4 signals shown in any given moment, while the total number of possible binary signals is 8, which will have to be represented in detectors' preference lists. Signals from presenters of the opposite subtype to each detector are colored in gray.

3.2.7 Training

During training, when agents are paired for too long, this indicates that they both have each other high on their preference lists, therefore detector agents must be trained to avoid this attachment to presenters with subtype equal to theirs that show signals ι_i. The goal is to train detectors to prefer signals o_i, so that when an abnormal sample is shown, detectors will have high affinity with presenters' signals and become very attached to them, leading to long matching lifetimes. Since presenters of the same subtype as detectors will generate the longest matching lifetimes, after training, these presenters will have their signals lowest in detectors' preference lists. This will result in a frustrated dynamic.

Training a batch of detectors is only done every few thousand iterations E_n, in order to allow statistically relevant detectors' matching lifetimes to appear. The condition check that sends a detector to training compares a detector's current τ_i with the threshold for training $\tau_{threshold}$, which is updated to a new lower value based on the highest τ_i amongst all detectors, but only if no detectors were trained after the last E_n iterations, which typically was $E_n = 1500$.

Algorithm 3.2 Detector education.

```
τ_max ← 0
trained ← false
for all a_i ∈ D do
    if τ_i > τ_max then
        τ_max ← τ_i
    if τ_i > τ_threshold then
        a_j is paired with a_i
        trained ← true
        newRank ← random integer greater than rank_{l_i}(s_j)
        In l_i swap signal at rank rank_{l_i}(s_j) with signal at rank newRank
        m_i, m_j ← −1
        τ_i, τ_j ← 0
if trained is false then
    τ_threshold ← τ_max
```

Algorithm 3.3 Training.

```
Initialize detectors' global preference lists with signals ι_i and o_i randomly ranked
{τ_i} ← 0
τ_threshold ← n_max
for all n ∈ [1, n_max] do
    if n mod T is 0 then
        Change sample shown by presenters
    for all a_j ∈ P ∪ a_i ∈ D do
        Pick a random presenter a_j
        Pick a random detector a_i
        decision(A, a_i, a_j) - apply decision rules in Definition 10
    for all a_i ∈ A do
        if a_i ∈ A is paired then
            τ_i ← τ_i + 1
    if n mod E_n is zero then
        education(A, τ_threshold)
```

Algorithm 3.2 describes how the detectors' preference lists are manipulated in order to produce an educated population of detectors capable of recognizing abnormal samples when presented. Algorithm 3.3 is essentially the same as Algorithm 3.1 with the added function that educates detectors.

The education mechanism is visually described in Figure 3.7, and the final educated lists for an example case are shown in Figure 3.8.

3.2.8 Calibration

Before detection can be achieved, detectors must be calibrated with the normal samples used for testing in a two-step procedure. First an activation tau, τ_{act}, is defined as follows. When running the CF dynamic, a sorted vector of taus, $c(\tau_i)$, will count the number of times any detector leaves a pairing with a lifetime τ_i. When the dynamic ends, a cumulative sum is performed on this vector in order to get the number of times each $\tau \geq \tau_i$ occurred. Then these new values in $c(\tau \geq \tau_i)$ are averaged as follows $c_{avg}(\tau \geq \tau_i) = c(\tau \geq \tau_i)/N_S/N_D$, where N_S is the total of all calibration samples. Finally, $\tau_{act} = \max\{c_{avg}(\tau \geq \tau_i) < 1\}$. This is visually shown in Figure 3.9.

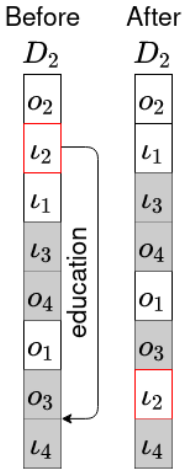


Figure 3.7: Detector being educated with an education operation applied to its global preference list l_i. This operation consists in swapping the signal outlined in red with a signal lower ranked in the l_i. This is triggered by a long matching lifetime $\tau_i > \tau_{threshold}$, which is undesirable when detectors are interacting with samples they perceive as normal, therefore they must learn to dislike these samples. This stable matching happens because D_2 ranks highly the signal shown by P_2, and since they are both of the same subtype, P_2 also ranks D_2 highly in its preference list. On top of this, D_2 also sees the signal s_2 as being inside its normal domain, which according to the decision rules further promotes this stable matching.

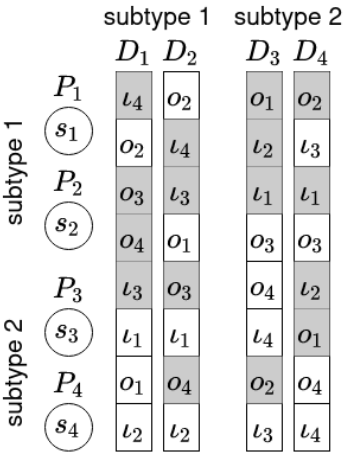


Figure 3.8: Trained detectors and their ordered global preference lists l_i with reference to the lists in Figure 3.6. Some signals were correctly educated to lower ranks while some were not. All lists now have mostly gray colored signals at the top, which means detectors learned to prefer signals from presenters of the opposite subtype, which leads to lower matching lifetimes when presenting normal samples, hence cellular frustration is achieved. In general all lists are trained well. D_2's l_i has one signal on top of the list shown by a presenter of the same subtype, because it never appeared during training. In D_4's l_i, an incorrect education occurred, where a previously correctly ranked signal ι_1 now has a lower rank, and signal ι_3 climbed the list, which is bad, because it leads to long pairings with normal samples. This can happen when the population's configuration is very stable at some point and triggers a long matching. Although very problematic this sort of event is rare.

The idea is that the highest $\tau \geq \tau_i$, where on average detectors have a low $c(\tau \geq \tau_i)$ when shown normal test samples, should be a τ where response is minimum for normal samples but still above 0. Therefore, for abnormal samples it should be easy for even weak responses to trigger detections. It is important to select an activation tau where on average detectors still have $c(\tau \geq \tau_i) > 0$, because if $c(\tau \geq \tau_i) = 0$, when a normal sample is shown during testing it easily triggers a detection, leading to false positives.

The CF dynamic is run for w_{max} iterations for each sample, where $w_{max} = 5000$ in this case. Each sample needs to be evaluated for a few thousand iterations to avoid the influence of statistical fluctuations when registering matching lifetimes.

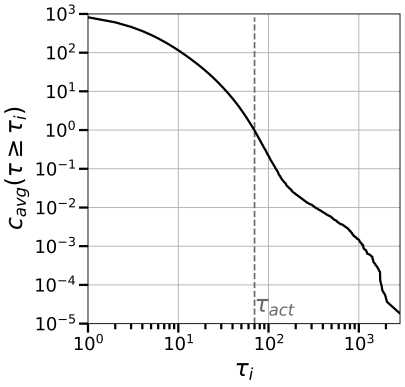


Figure 3.9: Calibration of an activation tau τ_{act}.

Finally, an activation threshold $n_i(\tau_{act}) \forall a_i \in D$ is defined as follows. The same CF dynamic is run again, and a sorted vector $c_{i,j}(\tau_{act}) \forall j$, where j corresponds to a normal sample used for calibration, will register the number of times a detector a_i establishes a matching lifetime $\tau_i \geq \tau_{act}$, for each sample j. Then each detector's activation threshold is calculated as $n_i(\tau_{act}) = c_{i,x}(\tau_{act})$, where $x = N_S \times f$, with $f \in [0, 1]$. In this case $f = 0.05$, which means the 5% largest number of pairing lifetimes greater than τ_{act} were considered.

3.2.9 Detection

The last stage when applying this model is the monitoring stage, where both normal and abnormal samples from the test set are monitored in order to extract the responses from detectors towards each sample. From these responses a ROC curve is plotted to see the performance of the model when trying to distinguish abnormal samples from normal ones.

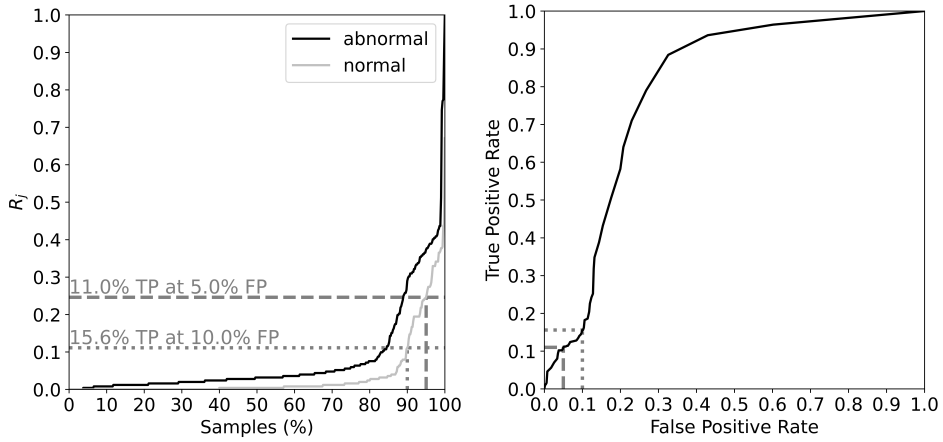


Figure 3.10: Translation of detectors' responses to a typical ROC curve. (a) Shows the normalized responses obtained for the normal and abnormal samples, with two threshold examples for 90% and 95% of the normal samples, represented by the dotted and dashed lines, respectively. All the responses towards normal and abnormal samples above the dotted or dashed lines are considered detections, be it true positives or false positives. (b) Shows the example thresholds translated into the TPR for a given FPR seen in a typical ROC curve.

There should be a frustrated dynamic when a normal sample is presented, whereas when an abnormal sample is presented more stable pairings should occur, leading to long pairing lifetimes, which in turn produce a response from detectors towards these abnormal samples much higher than towards normal samples. Therefore, there should be a clear difference between these responses, hence result in a good detection by the cellular frustrated system (CFS).

In this stage, the same CF dynamic considered in the calibration stage is also run here, for the same w_{max} iterations per sample. Finally, the total response towards a sample j by the collection of detectors, is based on the number of times each detector established a long

pairing lifetime relative to its activation threshold, and it is calculated as follows

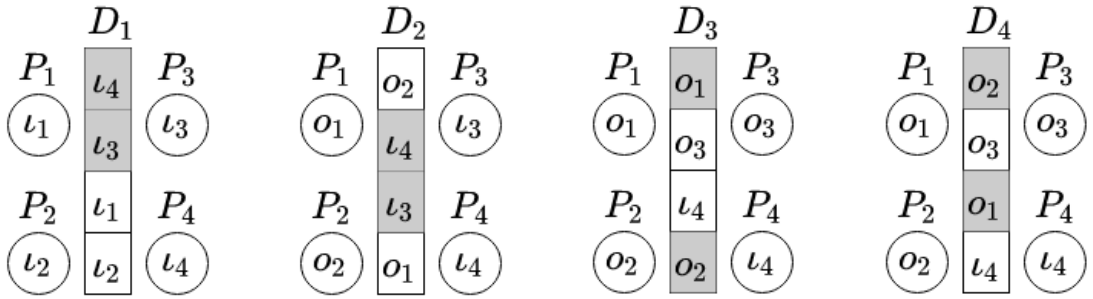
$$R_j = \sum_{i}^{N_D} (c_{i,j}(\tau_{act}) - n_i(\tau_{act}))H(c_{i,j}(\tau_{act}) - n_i(\tau_{act})), \tag{3.1}$$

where H is the Heaviside step function. See Figure 3.10 to better understand how the ROC curve is plotted based on detectors' responses.

The Heaviside step function guarantees a result has a lower bound of zero if the calculation it takes as argument results in a negative value, and it is given by

$$H(x) = \begin{cases} 1 & \text{x > 0} \\ 0 & \text{x < 0.[34]} \end{cases} \tag{3.2}$$

CFSs have the ability to detect three different abnormal patterns in trained detectors' local preference lists, which allow them to make detections when an abnormal sample is shown by presenters. These are the presence of abnormal signals o_i that were rarely or never shown during training, the absence of too many normal signals ι_i compared to how common they were during training, and the absence of combinations of signals ι_i frequently shown together during training. These are more clearly explained in Figure 3.11.



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

Figure 3.11: Mechanisms of local preference lists that allow detectors to make detections, with reference to Figure 3.8. These four examples show the local preference lists of detectors when a certain combination of signals is shown by presenters, and how the order of these signals in preference lists leads to detections. Since each presenter can only show one of its two possible binary signals at any given time, there is no purpose in representing the remaining signals that are not being evaluated at that time, therefore the local preference list representation is more appropriate. (a) Shows how a properly educated list prevents erroneously detections when during monitoring a normal sample is presented, and detectors should not be making long pairings. Since the signals at the top of the list belong to presenters with the opposite subtype to D_1, frustration is achieved. (b) During training the signal o_2 never appeared because it is an abnormal signal, and no sample contained any feature value that could be mapped to it, so it remained in its original randomly generated rank in D_2's global preference list. Therefore, when a sample appears in the monitoring stage with some feature value that can be mapped to o_2, it appears at the top, which provokes a response from D_2, leading to a detection. (c) When the combination of two normal signals at the top of D_3's list become absent, the abnormal signal o_3 is able to climb the list and induce a stable matching between D_3 and P_3. Since both have the same subtype, P_3 becomes very attached to D_3, leading to a detection. This sort of absent combinations can happen when certain features are strongly correlated. (d) When many sequential signals ι_i in D_4's list become absent, the signals o_i are able to climb the list provoking a strong response and triggering a detection. It is important to note that in all these situations, a strong response only occurs when a collection of detectors share the same view of a sample. This is especially important for the detection mechanism that relies on absent combination of signals ι_i, where only a few might disappear in each l_i for an abnormal sample hard to detect.