Dynamic Learning of Heart Sounds with Changing Noise:
An AIS-based Multi-agent Model Using Systemic Computation

Yiqi Deng
University College London
Gower Street, London
United Kingdom
+44(0)7721871959
y.deng.11@ucl.ac.uk

Peter J Bentley
University College London
Gower Street, London
United Kingdom
+44(0)7734102352
p.bentley@cs.ucl.ac.uk

ABSTRACT

Agent-Based Models are used to model dynamic systems such as stock markets, societies, and complex biological systems that are difficult to model analytically using partial differential equations. Many agent-based modeling software are designed for serial von-Neumann computer architectures. That limits the speed and scalability of these systems. Systemic computation (SC) is designed to be a model of natural behavior and, at the same time, a non Von-Neumann architecture with its characteristics similar to multi-agent system. Here we propose a novel method based on an Artificial Immune System (AIS) and implemented on a systemic computer, which is designed to adapt itself over continuous arrival of data to cope with changing patterns of noise without requirement for feedback, as a result of its own experience. Experiments with heartbeat data collected from a clinical trial in hospitals using a digital stethoscope shows the algorithm performs up to 3.60% better in the precision rate of murmur and 3.96% better in the recall rate of murmur than other standard anomaly detector approaches such as Multiple Kernel Anomaly Detection (MKAD).

Keywords

Artificial Immune System; Systemic Computer

1. INTRODUCTION

Real world machine learning, where data is gathered continuously may in theory be classifiable into distinct and unchanging categories but in practice the classification becomes non-trivial because the nature of the background noise continuously changes. One example of such a problem is the classification of heart sounds. Although there may in theory be a limited number of specific categories of heart sound, (e.g. Normal, Extra Heart Sound, Murmur, Etc.), in practice new sounds are gathered within such varied environments that the changing background noise can obscure or corrupt the features being used for learning. For this application, a classifier trained on a static training set may be less effective than a method that can continuously learn and adapt.

GECCO '14, July 12 - 16 2014, Vancouver, BC, Canada

http://dx.doi.org/10.1145/2598394.2605442

This problem is extremely important. According to the World Health Organization, cardiovascular diseases (CVDs) are the number one cause of death globally: more people die annually from CVDs than from any other cause. An estimated 17.1 million people died from CVDs in 2004, representing 29% of all global deaths. Of these deaths, an estimated 7.2 million were due to coronary heart disease[1].

There is also a real need to automate the classification of heart sounds. This is because forming a diagnosis based on sounds heard through either a conventional acoustic or an electronic stethoscope is itself a very special skill, one that can take years to acquire. Because this skill is also very difficult to teach in a structured way, the majority of internal medicine and cardiology teaching programs offer little or no such instruction. Despite its obvious utility, primary care physicians are documented to have poor auscultatory skills [2]. Any method which can help to detect signs of heart disease could therefore have a significant impact on world health.

The aim of this work is to produce methods to do exactly that. Specifically, we are interested in creating the first level of screening of cardiac pathologies both in a Hospital environment by a doctor (using a digital stethoscope) and at home by the patient (using a mobile device). We propose a novel approach to tackling the problem of varied background noise. Instead of supervised machine learning, we propose the use of online learning methods such as anomaly detection. More specifically we propose a novel method based on an AIS and implemented on a systemic computer. The AIS-SC algorithm is designed to adapt itself over the course of a lifetime to cope with changing patterns of noise without requirement for feedback, as a result of its own experience.

The next section provides background to the work. Section 3 details the design of AIS-SC algorithm. Section 4 and 5 compares and presents the performance of AIS-SC model and benchmark model on the classification of heart sound. Finally section 6 concludes our work.

2. Background

2.1 Heartbeat Sounds Classification

Classification of heart sounds is of particular interest to machine learning researchers as it involves classification of audio sample data, where distinguishing between classes of interest is non-trivial. Data is gathered in real-world situations and frequently contains background noise of every conceivable type. The differences between heart sounds corresponding to different heart symptoms can also be extremely subtle and challenging to separate. Success in classifying this form of data requires extremely robust classifiers. Despite its medical significance, to

date this is a relatively unexplored application for machine learning.

So far, heart sound segmentation algorithms reported in literature may be generally classified into two types: (i) using electrocardiogram (ECG) as a reference signal and (ii) without ECG. ECG is a transthoracic (across the thorax or chest) interpretation of the electrical activity of the heart over a period of time, as detected by electrodes attached to the surface of the skin and recorded by a device external to the body [3].

In the first class of approaches, QRS complexes (a name for the combination of three of the graphical deflections seen on a typical ECG) and T-waves (represents the repolarization (or recovery) of the ventricles in ECG) are always detected in order to locate the S1 (the first heart sound) and S2 segments (the second heart sound) [4]. Calvalho applied the variance fractal dimension to adaptively identify the boundaries of sound lobes and used QRS synchronization to detect S1 components while for S2 components a non-supervised classification approach is applied, based on temporal features of the lobes [5]. Kampouraki used support vector machines (SVMs) to classify ECG recordings [6]. Karraz extracted the QRS complex and T-waves from the signal as features and applied them into a Neural Network Classifier based on a Bayesian framework [7].

In the second class of approaches, many signal processing techniques are utilized. For example, Leung characterized digitally recorded pathological and non-pathological phonocardiograms (PCGs) by a time- frequency (TF) method known as the trimmed mean spectrogram (TMS). Features were extracted from the TMS containing the distribution of the systolic and diastolic signatures in the TF domain. Together with the acoustic intensities in systole and diastole, these features were used as inputs to the probabilistic neural networks (PNNs) for classification [8]. A similar TF method was adopted by Yoshida who estimated instantaneous frequency of the phonocardiogram from averaged Wigner-Ville distribution [9]. Groch designed a heart sound gating device which utilized dynamically varying timing windows to anticipate the occurrence of S1 and S2, providing two trigger points through the cardiac cycle for synchronizing medical images [3]. Strunic extracted signals on certain band to reduce anomalies and then set an amplitude threshold to pick out the spikes and realize the segmentation [10]. Haghighi chose an autoregressive (AR) model to estimate the power spectral density (PSD) of the signal as well as the energy in certain frequency bands for consecutive overlapping frames to realize the automatic segmentation [11]. Liang chose Chebyshev type I low-pass filter combined with Shannon energy to attenuate noise and make the findings of low intensity sounds, namely heart beats, easier [12] Developed from Liang's algorithm, Spencer integrated all the segmented heart cycles into one average heart cycle and used it to train the Artificial Neural Network (ANN) to classify heartbeat into Normal, Systolic Murmur caused by Mitral Regurgitation (MR), Systolic Murmur caused by Aortic Stenosis (AS) and Diastole Murmur caused by Aortic Regurgitation (AR) [10]. The way ECG signal is collected conserves fidelity and compresses background noise. Hence segmentation results with the reference of ECG are relatively better than those without it. However, ECG is not a cost-effective choice for preliminary screening of heart problems. Furthermore, with low-quality ECG signals, T-waves are not always clearly visible; in such cases, S2 sounds may be classified by an unsupervised classifier.

For the second approach, most data processed are either simulated or super clean Phonocardiogram (PCG), which is not really the case in daily collection of heartbeat. For instance, according to Spencer's result, when processing simulated heart sounds, the accuracy and sensitivity of ANN could be as high as 76±6.1% and 89.7±5.9% respectively. The accuracy drops to 48.7±12.7% when he used data collected by an electronic stethoscope with a duration of about 5 seconds. In an attempt to apply this system to our real-world datasets for this work the system was unable to differentiate between heartbeats and background noise in most cases. Unfortunately, real life data is always of varying durations and with excessive background noise. Under such circumstances, the differentiation among different subtle heart symptoms can be extremely challenging. To cater to demands from such data, an algorithm is required that may independent of ambient or body sounds.

2.2 Systemic Computation

Systemic computation (SC) is designed to be a model of natural behavior and, at the same time, a model of computation with its characteristics similar to biological systems and processes. As biologically inspired model, cellular automata have proven themselves to be a valuable approach to emergent, distributed computation [14]. Generalizations such as constrained generating procedures and collision-based computing provide new ways to design and analyses emergent computational phenomena [15]. Bio-inspired grammars and algorithms introduced notions of homeostasis (for example in artificial immune systems), fault-tolerance (as seen in embryonic hardware) and parallel stochastic learning, (for example in swarm intelligence and genetic algorithms) [16, 17].

In SC, everything is regarded as a system, or an agent. This implies the notion of the inherent hierarchy in nature and enables SC analysis in different levels of abstraction. Also, SC is designed to operate using any system, meaning that, provided that the interaction pattern is the same, systems of different levels of abstraction can perform the same calculation. Systems can never be destroyed, reflecting the fundamental principle of conservation of energy (first law of thermodynamics). As a result, systemic computations imply metabolism and ecology, since new systems need to be transformed and unwanted computation remnants need to be removed, meaning that the "waste" of one program will have to be recycled as "food" for another.

In SC, each system comprises three elements: two schemata that define the possible systems that may interact in the context of the current system, and a kernel which defines how the two interacting systems will be transformed.

The interaction of two systems can be described by the systems themselves and a third "contextual" system which denotes how/if the interacting systems are transformed after their interaction. The "shape" of a contextual system affects the result of the interaction between systems in its context. This encompasses the general concept that a resultant transformation of two interacting systems is dependent on the context in which that interaction takes place. A different context will produce a different transformation. Since everything in systemic computation is a system, context must be defined by a system.

Systemic computation also exploits the concept of scope. In all interacting systems in the natural world, interactions have a limited range or scope, beyond which two systems can no longer interact (for example, binding forces of atoms, chemical gradients of proteins, physical distance between physically interacting individuals). In cellular automata this is defined by a fixed number of neighbors for each cell. Here, the idea is made more

flexible and realistic by enabling the scope of interactions to be defined and altered by another system. Interactions between two systems may result in one system being placed within the scope of another (akin to the pino membrane computing operation), or being removed from the scope of another (akin to the exo membrane computing operation). So just as two systems interact according to (in the context of) a third system, so their ability to interact is defined by the scope they are all in (defined by a fourth system). Scope is designed to be infinitely recursive so systems may contain systems containing systems and so on. Scopes may overlap or have fuzzy boundaries; any systems can be wholly or partially contained within the scopes of any other systems. Scope also makes this form of computation tractable in simulation by reducing the number of interactions possible between systems to those in the same scope.

3. Algorithm

Le Martelot provided a simple implementation of "artificial tissue" for AIS using systemic computation [17], which serves as the foundation of our work. He created an artificial organism, a program with metabolism that "eats" data, expels waste, clusters cells based on the nature of its food and emits danger signals suitable for an artificial immune system.

3.1 Artificial Immune System

The immune system is a complex of cells, molecules and organs aiming at limiting damage to the host organism by pathogens, which elicit an immune response and thus are called antigens. One type of response is the secretion of antibody molecules by B cells. Antibodies are Y-shaped receptor molecules bound on the surface of a B cell with the primary role of recognizing and binding, through a complementary match, with an antigen. The strength and specificity of the Antigen-Antibody interaction is measured by the affinity (complementarity level) of their match [18].

When stimulated, the B cell proliferates and secretes its receptor molecules as free antibodies. Antibodies thus can either be free molecules or receptors attached to cells. Secretion requires that B cells become activated, undergo proliferation (cloning) and then finally matured into memory cells which are used for antigen recognition. A clone is a cell, or a set of cells, which is the progeny of the same cell. This basic process of pattern recognition and selection is known as clonal selection [19] and is similar to natural selection, except that it occurs on a rapid time scale on the order of days and weeks, within our bodies.

3.2 AIS in Systemic Computation

In Le Martelot's model, the organism should be able to eat food from its environment, converts each data item into a new cell, and attempts to bind that cell to itself in order to grow organs, with cells made from similar data items binding to each other. Thus, a cell unable to bind to any group of cells reveals itself to be significantly different from them. If this abnormal cell dies unbound, it can therefore be spotted as a potential anomaly. To regulate this growth and introduce the notion of time, a decay process simulates the aging of cells. When cells die, a split process splits them from the adhesion surfaces they are bound to. In the model dead cells can thus be recycled into new food. The whole ecosystem could be visualized in Figure 1 below.

We based on our model on Le Martelot's ecosystem to model the process of antigen recognition. While his model mainly focused on clustering data through the growth of adhesion surface and left the AIS mechanism unexploited, we intend to implement the full function of AIS with antigens being "eaten" continuously into the organism and inspired the secretion of antibodies which serves as the classifier to recognize later-come antigens. More specifically, data from the same category is regarded as sharing a same/similar feature which in the scenario of AIS would correspond to antigen. Those data with same/similar antigen would then be recognized by memory antibodies trained previously through clonal selection.

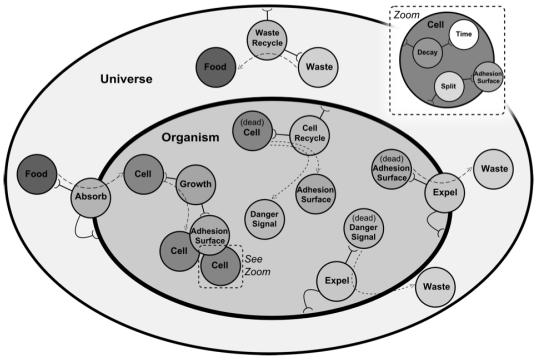

Figure 1 Systemic organization of Le Martelot's organism.

3.3 Systemic Analysis

When programming with SC it is of great importance to conduct a systemic analysis in order to identify and interpret appropriate systems and their organization. The first stage is to identify the low-level systems (i.e. determine the level of abstraction to be used). In most artificial immune systems, the level of abstraction is the cell: few approaches require modeling of the internal organelles or genome of cells, and few require modeling of populations of organisms. Here we intend to model the consumption of "food" (data items), the secretion of antibodies and the recognition of antigens. Thus an abstraction at the cellular level is appropriate, with systems being used to explicitly model each element. Below is an overview of the whole system.

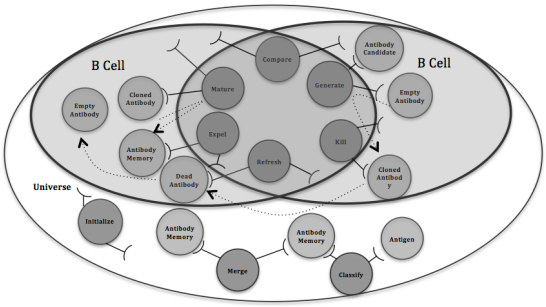

Figure 2 Systemic Organization of our model

The whole system is organized in the scope of Universe which is an abstraction of the immune system. Inside it, training data could be eaten into B cell to generate memory antibody which would be later secreted and used to recognize data with similar features (antigens). Based on Le Martelot's idea, we associate data systems with the candidate antibodies is a straightforward choice, as they both carry the information for recognition. However, in order to enable each system to represent a complete data item this work introduces a new extension to the systemic computation paradigm – the use of pointers. In this work each system may

behave as a pointer to a fixed and unique region of memory where a larger or more complex data structure can be held..

As systems are set as antigens or randomly initialized into antibody candidates after the beginning of the program, the SC model should initially include non-initialized data systems. Moreover, the organism should be able to stimulate an adaptive immune response where a spectrum of clones of antibody candidates is produced with different affinity. Antibody memories, those with the highest affinity, would be expelled to the outside environment to recognize antigens (classify data). Therefore antibody candidates and antibody memories would be seen as different states of the same systems. Moreover, antigen system asks for an on-off mode to denote its status in first phase and second phase of maturation. Thus, five distinct types of data systems, identified using bit 15 to bit 17 in the function of each data system, are required to represent all possible solution: non-initialized, antibody candidate, antigen not ready, antigen ready and antibody memory (see in figure below).

```
#label waste                %b00000000000000000????????000000
#label antigen              %b00000000000000100????????000000
#label clonecandidate       %b00000000000000011????????000000
#label bodymemory           %b00000000000000111????????000000
```

Figure 3 Representation of different data systems in SC

After identifying the elementary systems, we move to the identification of contextual systems which are in charge of system interaction and transformation.

An initialize context is implemented in order to initialize the B-cell with training data. The production of antibody memories could be divided into 2 maturation phases according to the literature above. However, our attempt on organizing the scope based on different phases the separate design suffers from low efficiency due to the adoption of too many long chains and the way it store the data centrally go against the distributed design of systemic computing. Therefore an integrated B-cell system scopes are implemented by using an on-off mode to denote different phases.

In the first phase of maturation, the Compare system calculates the distance between B-cell and antibody candidate and compares it with the distance stored in B-cell last round of comparison. It is evident that Compare system acts as context systems, defining the measurement of the distance between 2 data systems and updating the distance stored in B-cell with the smaller one for further use. The counter implemented in Compare system keeps track of continuous winning time for the current stored distance. When it reaches the counter threshold, the B-cell would be turn to on-mode to be handover to Mature context.

The second phase of maturation follows the similar thought process. The only difference is that the Kill context holds the winner and keeps comparing it with others with the help of chain mechanism. It is worth noting that within each Compare/Kill context a counter is stored in the core to keep a record of consecutive winning time which will be reset if encountering a better match. The data systems would either be discarded for Refresh context to recycle or selected as the final winner if it wins more than 5 times consecutively.

Moving into the next pair of interactions, Refresh system updates the loser antibody candidate with new randomized value to introduce new solutions to the system. Thus Refresh system acts a context, defining the refresh of failed data systems in the maturation.

Looking at the Generate reaction, it acts as a context to proliferate 5 mutated copies of the winner antibody candidates resulted from comparison process with the help of chain.

The mature context is designed to have two functions. In the first phase of maturation, it interacts with successful selected antibody candidate and turns the B-cell to on-mode to proliferate. In the second round, it is in charge of changing the state of antibody winner to antibody memory.

Finally, if antigens in the environment fall into the category of the antibody memory, the expelled antibody memory will classify them by absorbing them into its own scope and update the antibody memory data with the averaged data. Borrowed from the idea of aiNet [18], the expelled antibody memories could be further merge if their distance is below the threshold which indicates they might come from the same category.

Pseudo-code for the classify context clustering method and merge context. d, m are given thresholds for classify and merge context respectively. The distance function calculates the Euclidian distance of two vectors.

```
if distance(Antigen data, Antibody Memory data) ≤ d
   Absorb Antibody into the scope of Antibody Memory
   Antibody Memory data ← Average(Antigen data, Antibody
   Memory data)
end if
if distance(Antibody Memory1 data, AntibodyMemory2 data) ≤ d
   Integrate all the antigens in both antibody memories into the
   scope of antibody memory 1 and set 2 as waste
   Antibody Memory 1 data ← Average(Antibody Memory 1 data,
   Antibody Memory 2 data)
end if
```

After identifying the systems involved in the SC integrated data clustering model above, the last part for the systemic analysis is with regards to the required scope of interaction. This model separates systems according to which B-cell they relate to. Therefore, an exact representation of the model would require that once an antibody candidate C, modeled as a separate system, is bound to an antigen during selection, this C could not be able to be re-bound to another antigen. Furthermore, if this C would need interact again with other antigens, it would need to unbind the specific antigen that had bound before. This functionality, representing the physical location in a real biological system could be implemented using scopes. However, it would add considerable complexity to the SC model if each antigen own itself a set of all the contexts. Hence, all the antigens are designed to share a same set of contexts, which, on the other hand, increases greatly the utility rate of the contexts.

3.4 Model Implementation

A description of the SC contexts functionality is given in Table 1. To further explain it, the code written in SC language is provided below. SC language is intuitively very close to the SC model and is created together with a compiler translating source code into byte-code for the virtual machine. The aim of the SC language is thus to aid the programmer when defining systems, declaring instances of them and setting scopes between them. Defining a system involves defining its kernel and its two schemata. When a system acts as a context, the two schemata are used as the two templates of the systems to interact with, and the kernel encodes

the context behavior. In order to make the code more readable, string labels can be defined and then used in the program instead of their value [16].

Table 1. Summary of the Process-oriented AIS SC model functions

Function Names	Description
Initialize	Initializes B-cells with training data.
Compare	Compares the distance between the antigen itself and the interacted antibody candidates and stores better the index and distance in the right schema of the context. The context also keeps a counter of consecutive comparison time.
Mature	Transform the antigen to the state of second round of selection and mark the final selected antibody candidate as antibody memory.
Generate	Duplicates the winner antibody candidate with a mutation on each dimension proportional to its distance to the antigen it compares to.
Kill	Compares the distance between the two mutated duplicates antibody candidates and their corresponding antigens, mark the winner and record its winning time in the counter. If the counter exceeds counter threshold, transforms the winner to antibody memory.
Expel	Pulls the generated antibody memory out of the scope of the second selection to the root scope.
Merge	Calculates the distance between the two interacted antibody memories and if it is below the sigmoid threshold, pulls all the classified test data systems into one antibody memory and update the value of that memory with the average of the two memories. Meanwhile, transforms the other antibody memory into waste for recycle.
Refresh	Updates the loser antibody candidate with new randomized value to introduce new solutions to the system.
Classify	Decides whether the interacted test data systems belongs to this category. If so, tags them.

```
// define some useful labels
#label zero                 %b0000000000000000
#label dontcare             %b????????????????
#label maxsamnum            %b1111111111111111
#label hunger               %b1111110000000000
#label waste                %b0000000000000000?????????000000
#label food                 %b0000000000000001?????????000000
#label data                 %b0000000000000010?????????000000
#label finiantigen          %b0000000000000101?????????000000
#label antigen              %b000000000000010?????????000000
#label antigenok            %b0000000000000100?????????000000
#label generalantigen       %b000000000000010?????????000000
#label bodycandidate        %b0000000000000100?????????000000
#label deadbodycandidate    %b000000000000001000?????????000000
#label bodyready            %b0000000000000110?????????000000
#label clonecandidate       %b000000000000011?????????000000
#label bodymemory           %b0000000000000111?????????000000
#label generalantibody      %b000000000000001???????????000000
#label generalbodyready     %b000000000000011???????????000000
#label organ                %b0000000000000000100000000000000
#label inidata              %b0000000000000010100000000000000
#label inibodycandidate     %b0000000000000010010000000000000
#label iniwaste             %b0000000000000000000000000000000
#label inifood              %b0000000000000000100000000000000
#label inicell              %b0000000000000000101000000000000
#label inideadcell          %b0000000000000000100000000000000
#label iniadhesion          %b0000000000000011100000000000000
#label iniorgan             %b0000000000000000010000000000000
#label zero2                %b0000000000000000000000000000000
#label decayfunc            %b1110010000000001111111111111111
```

Figure 4 Representation of contexts in SC code.

The SC Integrated AIS model has been implemented applying the suggested development methodology in the previous section. In order to show the function of each context, the source code of the model is given in Figure 5. Its direct mapping to Figure 2 is evident as, after the functions and some useful labels are defined, the data systems, contexts and scopes are defined exactly as represented graphically.

```
// and the program begins here:
main (%d0 zero2 maxsamnum)
[0:199]antibody (zero inibodycandidate zero)
[0:199]datasys (zero inidata zero)
[0:39]wastesys (zero iniwaste zero)
INITIALIZEsys ([zero waste zero] INITIALIZE(0,0) [dontcare zero2 maxsamnum])
expelsys        ([dontcare generalantigen dontcare] EXPEL(0,0) [dontcare generalbodyready dontcare])
[0:2]comparesys      ([dontcare antigen dontcare] COMPARE1(0,0)[dontcare bodycandidate dontcare])
[0:1]killsys           ([dontcare finiantigen dontcare] KILL(0,0)[dontcare clonecandidate dontcare])
generatesys     ([dontcare bodyready dontcare] GENERATE(0,0) [dontcare generalantibody dontcare])
maturesys     ([dontcare bodyready dontcare] MATURE(0,0) [dontcare generalantigen dontcare])
refreshsys     ([dontcare deadbodycandidate dontcare] REFRESH(0,0) [dontcare dontcare dontcare])
ClASSIFYsys   ([dontcare bodymemory dontcare] ClASSIFY(0,0) [dontcare data dontcare])

#chain [0:3]comparesys
{
   (?B maturesys $L)+(B generatesys ?C)+(B generatesys ?D)+(B generatesys ?E)
}
#chain [0:1]killsys
{
   ($R maturesys $L)
   (?A maturesys $L)
}

// set up the scopes
#scope main
{
    main
    [0:39]wastesys
    [0:199]datasys
    INITIALIZEsys
    [0:199]antibody
    refreshsys
    ClASSIFYsys
}

#scope [0:39]wastesys
{
  wastesys
  [0:3]comparesys
  [0:1]killsys
  expelsys
  refreshsys
}

#systemic end
```

Figure 5 SC code of AIS-SC Model

4. Heart Data

4.1 Data description

Our work uses data provided for the first PASCAL Challenge on Classifying Heart Sounds. The dataset comprises data collected from a clinical trial in hospitals using the digital stethoscope DigiScope [20]. In the dataset, sounds were recorded at a frequency of 4000Hz. In this work we consider 2 groups within the set: Normal and Murmur.

The audio files are all of varying lengths, between 1 second and 30 seconds (some have been clipped to reduce excessive noise and provide the salient fragment of the sound). Most information in heart sounds is contained in the low frequency components, with noise in the higher frequencies. It is common to apply a low-pass filter at 195 Hz. Fast Fourier transforms are also likely to provide useful information about volume and frequency over time. More domain-specific knowledge about the difference between the categories of sounds is provided below.

4.1.1 Normal Category

In the Normal category there are normal, healthy heart sounds. These may contain noise in the final second of the recording as the device is removed from the body. They may contain a variety of background noises (e.g., traffic, radios, doors slamming, children crying). They may also contain occasional random noise corresponding to breathing, or brushing the microphone against clothing or skin. A normal heart sound has a clear “lub dub, lub dub” pattern, with the time from “lub” to “dub” shorter than the time from “dub” to the next “lub” (when the heart rate is less than 140 beats per minute). Note the temporal description of “lub” and “dub” locations over time in the following illustration:

lub.....dub.....lub.....dub.....

In medicine we call the lub sound "S1" and the dub sound "S2". Most normal adult heart rates at rest will be between about 60 and 100 beats (‘lub dub’s) per minute. However, note that since the data may have been collected from children or adults in calm or

excited states, the heart rates in the data may vary from 40 to 140 beats or higher per minute.

4.1.2 Murmur Category

Heart murmurs sound as though there is a "whooshing, roaring, rumbling, or turbulent fluid" noise (Bentley P. 2011) in one of two temporal locations: (1) between "lub" and "dub", or (2) between "dub" and "lub". They can be a symptom of many heart disorders, some serious. There will still be a "lub" and a "dub". One of the things that confuses non-medically trained people is that murmurs happen between lub and dub or between dub and lub; not on lub and not on dub. Below, we illustrate with an asterisk* at the locations a murmur may appear:

*lub...****...dub.....lub...****...dub*

or

*lub.....dub...******...lub.....dub...******...*

4.2 Data Pre-processing

No machine learning approach can cope with uncleaned data. For this work we attempt to clean the data as much as feasible so that traditional machine learning methods may be used in addition to our proposed method, for comparison purposes. It should be noted that although significant cleaning is performed, the levels of noise are sufficiently high that the extracted features (described in the following section) are still significantly affected.

At first, the original signal was down sampled by a factor of 10. Since the heart sound feature with the highest frequency is murmur which is up to 600Hz [12], the new sampling frequency 4410Hz is still more than seven times higher. Thus no useful features of heart sounds are missed. After that, we reduce noise within the sound clip by wavelet coefficient soft thresholding using global positive threshold. Then a fifth-level discrete wavelet decomposition of the signal was done to obtain the coefficients of all the components of the decomposition. Using these coefficients, the details and approximation in desired level were obtained by reconstruction. The details and approximations vary depending on the wavelet families and orders used in the decomposition and reconstruction. Order six Daubechies filters were used in our work [12]. After reconstructing the heart sound, the envelope of the reconstructed signal is calculated to further emphasize heart beat information while reduce trifle disturbance. Because Shannon Entropy accentuates the medium intensity signal and attenuates low intensity value much more than high intensity signal, which shortens the difference of the envelope intensity between the low and high intensity sounds and makes the finding of low intensity sounds easier, we calculate the average Shannon energy in continuous 0.02-second segments with 0.01-segment overlapping (see Equation below):

$$E = -x^2 \cdot \log x^2$$

$$E = -1/n \cdot \sum_{i=1}^{n} -s^2(i) \cdot \log s^2(i)$$

For the convenience of parameter setting, the normalized average Shannon energy is computed as follows,

$$E_{norm}(t) = \frac{E(t) - M(E(t))}{Std(E(t))}$$

where $M(E(t))$ is the mean value of $E(t)$ and $Std(E(t))$ is the standard deviation. Then we find the heart spike based on the normalized Shannon Energy. Figure 6 shows the original signal and its average Shannon energy.

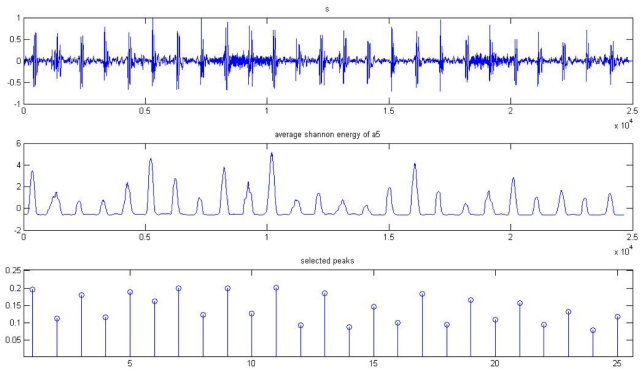

Figure 6 Signal s, a5's average Shannon Energy and selected heart spikes.

4.3 Feature Extraction

Before classification, raw audio of heart beat needs to be processed due to its large dimension and redundancy. By exploiting the results from previous sections, we obtain altogether 8 indicative parameters representing the features of each heart sound clip.

1. N: heart rate describes the number of heartbeat per minute. Heart rate not in the range of 40 to 140 bpm indicates potential abnormality.

2. Rs1 & Rs2: the ratio of the standard deviation of S1 (δ_{s1}) and S2 (δ_{s2}) over total standard deviation (δ_{total}) respectively. Here we assume S1 corresponds to the shorter interval which we tag it as the systolic period and S2 corresponds to the longer one which is diastolic period. Rs1 and Rs2 describes the duration stability of systolic and diastolic period of heartbeat. Extra or skipped heart sound would result in large Rs1 and Rs2.

3. Rm1 & Rm2: the ration of the mean of S1 and S2 over the total mean.

4. Std1 and Std2: the standard deviation of sound duration and interval duration. They mainly evaluate the stability of sound and non-sound intervals. As mentioned in the previous work that heart murmurs usually exhibit high frequency content which is more complex compared to S1 and S2 sounds, hence in order to detect murmurs more features should be extracted. We first calculate the Shannon energy of the detail d4 achieved from the wavelet decomposition from the previous step to further attenuate the noise. Then a binary threshold is constructed and applied to Shannon energy curve to calculate Std1 and Std2 respectively. In this work, the threshold is experimentally chosen as 1/10 of the maximum Shannon energy of d4. The binary threshold is,

$$S_{th} = \begin{cases} 1 & E_{d4} \geq 0.1\max(E_{d4}) \\ 0 & E_{d4} < 0.1\max(E_{d4}) \end{cases}$$

5. Prop: sound duration proportion in the whole sound clip length. Prop is in charge of judging the likelihood of heartbeats with a regular pattern being murmur heartbeat. Large Prop means that heart keeps making sound in the most of the time during the systole period, which is an obvious evidence for murmur heartbeat.

5. Experiments

In order to test AIS-SC model's performance in classifying data against changing background noise, we select 20 testees with normal heartbeat and 20 testees with murmur heartbeat. For each of them, four clips of heartbeat sound are recorded in different

time and location, which makes the total of samples 160. We reserve one clip of each person for training and use the rest 120

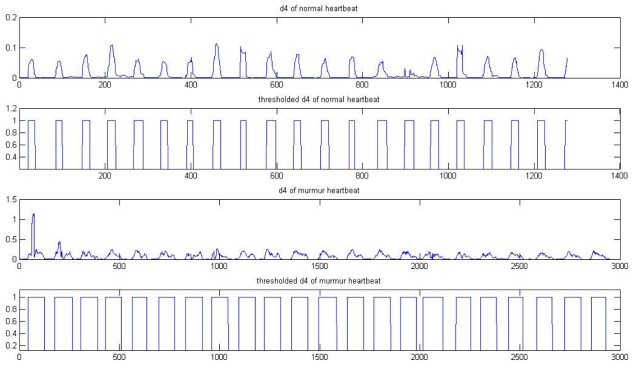

Figure 7 **1) Wavelet-decomposition of normal heartbeat at 4th depth level, 2)Thresholded Shannon energy of d4 of normal heartbeat, 3) Wavelet-decomposition of murmur heartbeat at 4th depth level, 4) Thresholded Shannon energy of d4 of murmur heartbeat.**

samples as test data. In other words, we initialize the B-cells in our model with the 40 training data clips and set rest of them as antigens who are designed to be classified by antibody memories generated by B-cells after maturation. As a comparison, we also feed the heartbeat data into Multiple Kernel Anomaly Detection (MKAD) which is designed for anomaly detection over a set of files [21]. Since theoretically speaking, there is no ending point of systemic computation unless it reaches the limit of the set interaction times, we here set the number of interactions as 3000000 as it is the lowest number to guarantee the complete recognition of all the test data. Then we calculate the average result of 20 rounds experiments on SC and MKAD, respectively. The result is as following:

Table 2 Comparison of Classification Result from AIS-SC and MKAD

	AIS-SC	MKAD
Precision of Normal	78.67%	81.29%
Precision of Murmur	65.72%	63.12%
Recall Rate of Normal	77.25%	82.37%
Recall Rate of Murmur	66.97%	63.01%

As can be seen from Table 2, the AIS-SC model's performance in recognizing normal heartbeat is quite close to that of MKAD while it outperforms MKAD in the precision of Murmur. This advantage becomes more obvious in the recall rate.

To have a better idea on how the AIS-SC model adapts itself along the time, we divide the 120 training data into 5 data chunk and present them one at each time to the model while keep the training data unchanged. Again we run the test 20 times to get the average results. Below depicts how the AIS-SC model improves itself overtime:

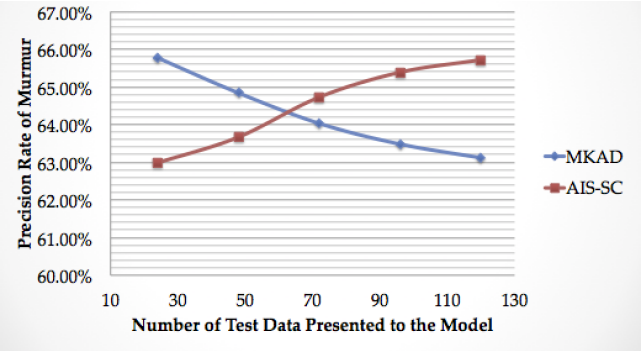

Figure 8 Precision Rate Comparison Against Number of Presented Test Data between AIS-SC and MKAD

Similar pattern could also be observed from the recall rate of murmur.

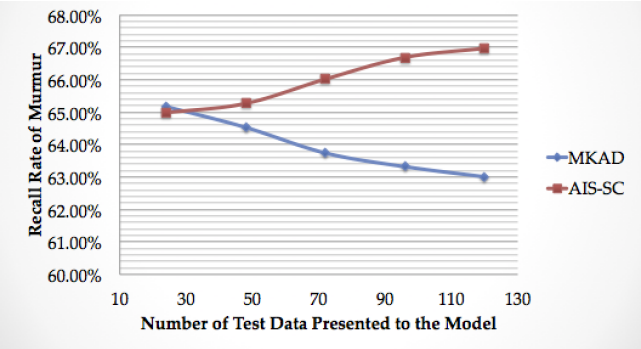

Figure 9 Recall Rate of Murmur Comparison Against Number of Presented Test Data Presented to the Model

From the figure above, It could be inferred that with more data coming in, the model will better adapt itself to the heartbeat sound as well as the background noise. Considering the model is used to help doctor detect potential heartbeat issues, it is more important for the model to tell the illness than confirming the wellness. Hence, it could be reasonable concluded that AIS-SC could reach the similar classification level with current classification model such as MKAD and shows a potential to improve with the increase of data.

6. CONCLUSIONS

In this paper, we examine the possibility to realize an agent-based model on a non von Von-Neumann computing architecture. We propose a novel method based on an AIS and implemented on a systemic computer, which is designed to adapt itself over continuous arrival of data to cope with changing patterns of noise without requirement for feedback, as a result of its own experience. Experiments on the heartbeat data classification shows the algorithm performs up to 3.60% better in the precision rate of murmur and 3.96% better in the recall rate of murmur than other standard anomaly detector approaches such as MKAD. To improve the model, future work will explore more on the increase of system efficiency and test on its ability to cope with more than two classes.

7. REFERENCES

[1] *Global status report on non-communicable diseases 2010.* Geneva, World Health Organization, 2011.

[2] Choi, Samjin, and Zhongwei Jiang. Comparison of envelope extraction algorithms for cardiac sound signal segmentation. *Expert Systems with Applications 34.2* (2008): 1056-1069.

[3] Groch, Mark W., James R. Domnanovich, and William D. Erwin. A new heart-sounds gating device for medical imaging. *Biomedical Engineering, IEEE Transactions on* 39.3 (1992): 307-310.

[4] Pan, J., & Tompkins, W. J. 1985. A real-time QRS detection algorithm. *Biomedical Engineering, IEEE Transactions on* 3 (1985): 230-236.

[5] Carvalho, P., et al. Low complexity algorithm for heart sound segmentation using the variance fractal dimension. *Intelligent Signal Processing, 2005 IEEE International Workshop on*. IEEE, 2005.

[6] Kampouraki, Argyro, George Manis, and Christophoros Nikou. Heartbeat time series classification with support vector machines. *Information Technology in Biomedicine, IEEE Transactions on* 13.4 (2009): 512-518.

[7] Karraz, G., and G. Magenes. Automatic classification of heartbeats using neural network classifier based on a Bayesian framework. *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE*. IEEE, 2006.

[8] Leung, T. S., et al. Classification of heart sounds using time-frequency method and artificial neural networks. *Engineering in Medicine and Biology Society, 2000. Proceedings of the 22nd Annual International Conference of the IEEE*. Vol. 2. IEEE, 2000.

[9] Yoshida, H., H. Shino, and K. Yana. Instantaneous frequency analysis of systolic murmur for phonocardiogram. *Engineering in Medicine and Biology Society, 1997. Proceedings of the 19th Annual International Conference of the IEEE*. Vol. 4. IEEE, 1997.

[10] Strunic, Spencer L., et al. Detection and classification of cardiac murmurs using segmentation techniques and artificial neural networks. *Computational Intelligence and Data Mining, 2007. CIDM 2007. IEEE Symposium on*. IEEE, 2007.

[11] Haghighi-Mood, A., and J. N. Torry. Coherence analysis of multichannel heart sound recording. *Computers in Cardiology, 1996*. IEEE, 1996.

[12] Liang, H., S. Lukkarinen, and I. Hartimo. Heart sound segmentation algorithm based on heart sound envelogram. *Computers in Cardiology 1997*. IEEE, 1997.

[13] Von Neumann, John. *Theory of self-reproducing automata*. (1966).

[14] Wolfram, Stephen. *A new kind of science*. Vol. 5. Champaign: Wolfram media, 2002.

[15] Adamatzky, Andrew. Collision-based computing in biopolymers and their automata models. *International Journal of Modern Physics C* 11.07 (2000): 1321-1346.

[16] Bentley, Peter J. Systemic computation: A model of interacting systems with natural characteristics. *The International Journal of Parallel, Emergent and Distributed Systems* 22.2 (2007): 103-121.

[17] Le Martelot, Erwan, Peter J. Bentley, and R. Beau Lotto. Eating data is good for your immune system: an artificial metabolism for data clustering using systemic computation. *Artificial Immune Systems*. Springer Berlin Heidelberg, 2008. 412-423.

[18] De Castro, L. N., & Von Zuben, F. J. (2001). aiNet: an artificial immune network for data analysis. Data mining: a heuristic approach, 1, 231-259.

[19] De Castro, Leandro Nunes, and Fernando J. Von Zuben. The clonal selection algorithm with engineering applications. *Proceedings of GECCO*. Vol. 2000. 2000.

[20] Bentley, P. and Nordehn, G. and Coimbra, M. and Mannor, S.(2011). The Pascal Classifying Heart Sounds Challenge. http://www.peterjbentley.com/heartchallenge/index.html

[21] Das, S., Matthews, B. L., Srivastava, A. N., & Oza, N. C. (2010, July). Multiple kernel learning for heterogeneous anomaly detection: algorithm and aviation safety case study. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 47-56). ACM.