

EMPLOYEE'S HEART RATE USING NON LINEAR CELLULAR AUTOMATA MODEL

N. GURURAJAN, A. SURESH POOBATHY AND S. SENTHIL KUMAR

ABSTRACT: Non-linear Cellular Automata model is very much use in practice. It has wonderful application for medical diagnosis and the intensity of the disease. Here in this paper, we discuss about Heart Rate Variability (HRV) of employee's through non-linear Cellular Automata model. Cellular Automata model gives us striking results for further studies.

AMS Classification: 00A69, 70K.

Keywords: Cellular Automata (CA), Heart rate variability (HRV), Time domain method and Frequency domain method.

1. INTRODUCTION

Cellular Automata (CA) have starting points far back in the sciences. They were introduced by *John Von Neumann*, in the 1940's and described by *Arthur Burks*, in 1970. During the 1970's and 1980's Cellular Automata had a strong revival through the work of Stephen Wolfram, who published an interesting survey. Today CA has become a very important modeling and simulation tool in science and technology, from physics, chemistry and biology, to computational fluid dynamics in airplane and ship design, to philosophy and sociology [5].

CA is a branch of Automata, which is a branch of Computer Science. A Cellular Automaton is an array of identically programmed automata or "cells" which interact with one another [7]. It is a dynamical system in which cells are generated according to some law. This generation is based on initial conditions. So starting with initial condition, applying certain law, cells are being generated. The arrays usually form either a 1-dimensional string of cells, a 2-dimensional solid.

Dynamics of CA is entirely discrete. It is ROBOT, which gives specific responses to specific inputs. The space of the system, which consists of cells of one, two or more dimensions, may be finite or infinite. In each cell, the system can assume a discrete number of state values, say ' k ' values. The configuration of the entire system at any time is defined by the set of state values, $\{S_i\}$ in all cells $\{i\}$.

For example, S_i may have the possible values,

$$s_i = 0, 1, 2, 3, \dots, k - 1 \quad (\text{State space, } s)$$

and

$$i = 0, 1, 2 \dots \quad (\text{Over the entire space finite or infinite})$$

We can say CA is perfect feedback machines. More precisely, they are mathematical finite state machines, which change the state of their cells step by step. Each cell has one out of ' k ' possible states. Sometimes we speak of a k -state Cellular Automaton [7].

The automaton can be 1-dimensional where its cells are simply linked up like a chain or 2-dimensional where cells are arranged in an array covering the plane. Sometimes we like to draw the succeeding steps of 1-dimensional CA one below the other and call the steps 'layers'. When running the machine it grows layer by layer. To run a Cellular Automaton we need two entities of information: (i) an initial state of its cells (i.e., an initial layer). (ii) a set of rules or laws [5].

These rules describe how the state of a cell in a new layer (in the next step) is determined from the states of a group of cells from the preceding layer. The rules should not depend on the position of the group within the layer.

1.1 Automata Rules

There are several ways a rule may determine the state of a cell in the succeeding layers.

In Fig. 1(a), the state of a new cell is determined by the states of 2 cells.

In Fig. 1(b), by the state of 3 cells.

In Fig. 1(c) and 1(d) the states of 5 cells determine the state of a new cell, but note that the position of the new cell with respect to the group is different in 1(c) and 1(d).

1.2 Automaton Rules

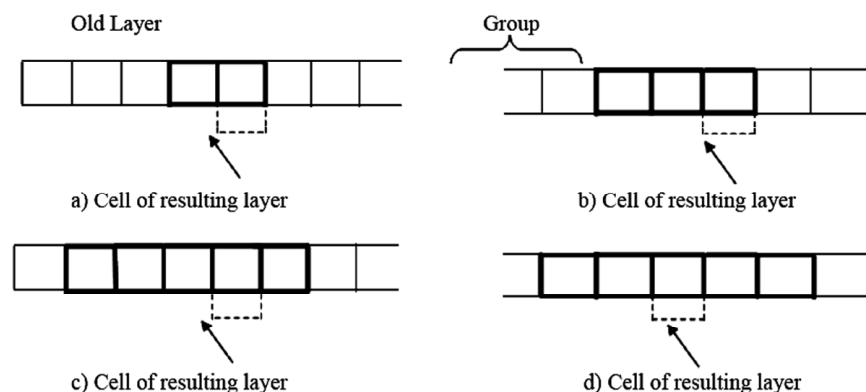


Figure 1: Automaton Rules

1.3 Example

Consider the infinite one-dimensional CA,

$$S_i = 0, 1 \quad (i = 0, +1, +2, \dots)$$

Here $k = 2$

We define the dynamics by

$$S_i(t+1) = [S_{i-1}(t) + S_{i+1}(t)] \pmod{2}$$

With these conditions we generate CA as:

```
00000001011000000 (t = 0) initial value
00000010011100000 (t = 1)
00000101110110000 (t = 2)
00001001010111000 (t = 3)
00010110000101100 (t = 4)
00100111001001110 (t = 5)
```

The simplest CA is the one-dimensional binary formed by a line of cells (the *lattice*) where each cell can assume states 0 or 1. These results can be represented in a more compact form by replacing the state $s = 1$ with a black mark and making no mark when $s = 0$. We get a figure through which one can get the behaviour of any dynamical system. For different initial configuration CA can be generated. We get different figures. A simple underlying mechanism is sufficient to support a whole hierarchy of structures, phenomena and properties [7].

Cellular Automata can be generated to the solution of any dynamical system. It constitutes a large class of models that produce evolving patterns of cells.

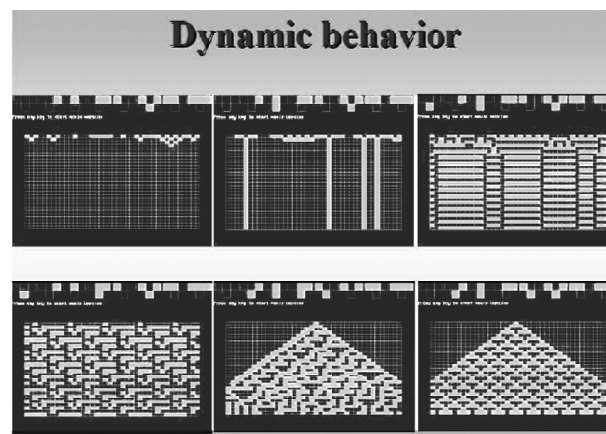


Figure 2: Dynamic Behaviors of Cellular Automata

2. HEART RATE VARIABILITY

Heart rate variability (HRV) is a physiological phenomenon where the time interval between the heart beat varies. It is measured by the variation in the beat-to-beat intervals. Other term RRvariability, where R is the point corresponding to the peak of QRS complex of the ECG waves.

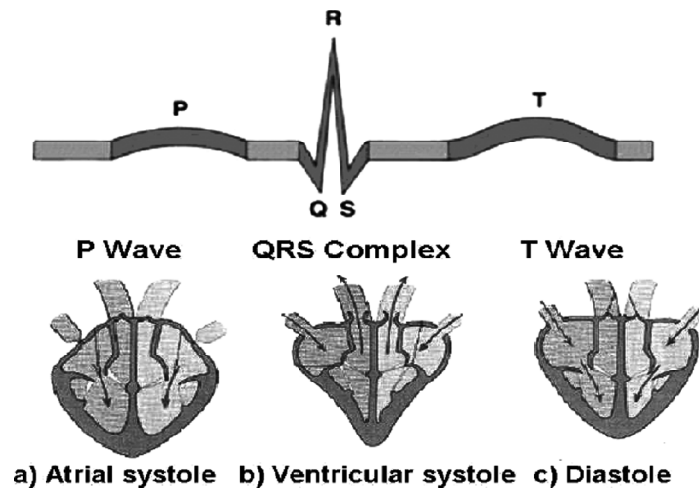


Figure 3: On an ECG Curve (Top), The Main Events in the Cardiac Cycle (Bottom) Can be Identified

The RR interval signals are non-linear because they result from complex interactions of hemodynamic, electrophysiological, and humoral variables, as well as autonomic and central nervous regulations. Very few studies have been done regarding non-linear analysis, so there are no standards. However, this is a promising area of research to discover tools that will help us better understand the complexity of the human system [2].

3. MEASUREMENT USING CONVENTIONAL METHODS

3.1 Time Domain Analysis

The time domain methods include standard statistical methods such as calculating standard deviations (SDNN [ms]) or graphing density distributions of successive RR-intervals. The so called HRV triangular index (HRV-TI) is a standard measure that helps to discriminate normal from reduced heart rate variability. It can be obtained by dividing the total number of RR-intervals by the number of RR-intervals with the value that has been counted to be most frequent in the total RRinterval series [4]. From this it follows that the smaller the value of the HRV index is (provided the data sets have equal total number of RR-intervals), the more RR-intervals have the same value,

meaning that there is less and less dispersion and variability of the RR-intervals. A HRV index with a value smaller than 20 has been established to be pathological [4].

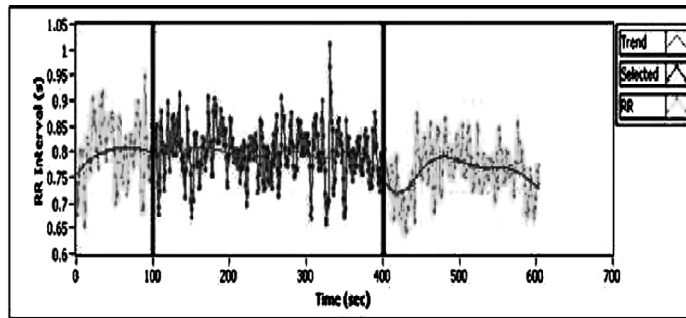


Figure 4: Sample RR Interval Signals Using the Time Domain Analysis Method

By means of the time domain methods, it can be worked out how much variability there is at all, but for a deeper insight into the dynamics underlying the beat-to-beat RR variations, more advanced techniques have to be applied [6] [1].

3.2 Frequency Domain Analysis

The frequency domain analysis is used to identify the underlying rhythms of heart rate time series.

They represent different oscillating physiological processes. Applying the Fast Fourier Transform (FFT) to the heart rate variability signal, the signal is decomposed in its constituent harmonic frequency components. A power spectral density graph shows how the total power, usually indicated as power spectral density measured in ms^2 , is distributed as a function of the various frequencies measured in Hertz (Hz). The power corresponds here to the variance of the signal at a certain frequency [4] [3].

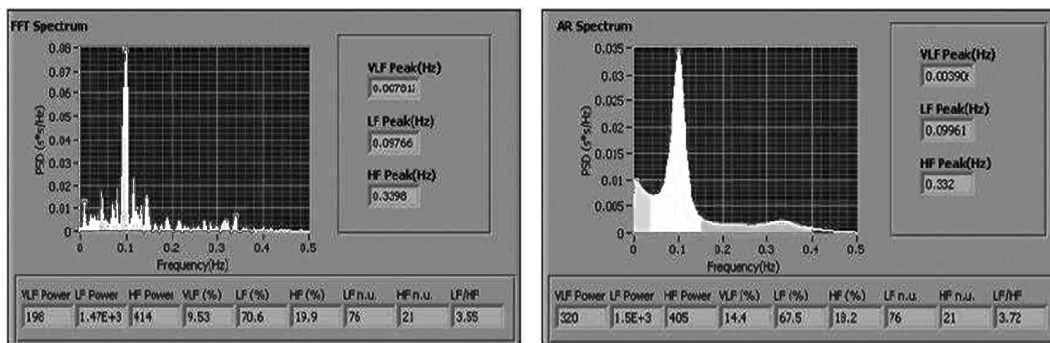


Figure 5: HRV Analysis Using the Frequency Domain Analysis Method

In a short-term heart rate variability assessment of about 5 min, the power spectrum is conventionally divided into three main frequency ranges: Very low frequency (VLF), low frequency (LF), and high frequency (HF). Long period rhythms are contained in the very low-frequency range, between 0 and 0.03 Hz. They account for the long-term regulation mechanisms probably related to thermoregulation or to the hormonal system. In the low-frequency (LF) range, between 0.03 and 0.15 Hz, there is a rhythm whose physiological interpretation is still controversial. Both sympathetic and parasympathetic contributions can be involved in this activity. However, an increase in the LF power is generally accepted as a marker of sympathetic activation. As explained above, the frequency in a range between 0.18 and 0.4 Hz can be related to the respiratory cycle. This highfrequency (HF) component in the power spectrum marks activity of the vagus nerve (major nerve of the parasympathetic nervous system) [1] [3].

In this present paper, we study about the employee's heart rate using one nonlinear Cellular automata model.

4. PURPOSE OF THE PRESENT STUDY

1. The main purpose of the present research is to assess the clinical applicability of new dynamical analysis methods derived from nonlinear dynamics of heart rate behaviour.
2. The specific aims of the individual sub-studies are:
 - i. to compare dynamical measures of heart rate behaviour between normal respiratory patients and unhealthy controls.
 - ii. to compare dynamical measures of heart rate behaviour between prehypertension patients and healthy controls.
 - iii. to compare dynamical measures of heart rate behaviour between hypertension.

5. METHODOLOGY

5.1 Selection of the Employees

In the present study and effort is made to select employees' heart rate from JIPMER Hospital, Pondicherry, during the year 2010-2011. Employees were selected by ECG recording. There are ranged between 45-55 years. There are no drop out in the study and all the employees co-operated willingly during experimentation and testing periods. ECG readings are recorded. For the recorded reading of heart rates, we convert it into binary digit and after that cellular automaton is drawn using JAVA applet program.

5.2 Normal1 Respiratory Patients and Unhealthy Controls

```

import java.applet.*;
import java.awt.*;
public class normal1 extends Applet
{
    public void paint(Graphics g)
    {
        int r=100,c=20,r1=800,c1=100;
        int tb=0,tw=0,tb1=0,tw1=0;
        int a[][]={
            {0,1,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1},
            {0,1,1,0,1,0,0,0,1,1,0,1,0,1,0,0,1,1,1,1},
            {0,1,1,0,1,0,1,1,1,0,1,0,0,1,0,1,1,1,1,0},
            {0,1,1,1,0,0,1,0,0,1,0,0,1,1,0,1,1,1,0,1},
            {0,1,1,1,0,0,1,0,0,0,0,0,1,1,0,0,0,1,0,0},
            {0,1,1,0,1,0,1,1,0,1,0,0,0,0,1,1,1,0,0,1},
            {0,1,1,0,1,0,1,0,1,0,1,0,0,1,1,1,1,1,0,1},
            {0,1,1,0,1,1,1,0,0,1,0,1,0,1,1,0,0,0,0,0},
            {0,1,1,0,1,1,1,1,0,1,0,1,1,1,0,0,0,0,1,0},
            {0,1,1,0,0,1,1,1,1,1,1,0,1,1,1,1,1,0,0,1},
            {0,1,1,0,0,1,0,0,0,0,0,1,1,0,0,0,1,0,0,1},
            {0,1,1,0,0,1,0,1,1,0,0,0,0,0,0,1,0,0,0,0},
            {0,1,1,0,0,1,1,0,0,0,0,0,0,1,0,0,0,0,0,1},
            {0,1,1,0,0,1,1,0,1,1,1,0,1,0,0,1,0,1,1,1},
            {0,1,1,0,0,1,0,0,0,0,1,1,0,0,0,1,0,0,1,0},
            {0,1,1,0,0,1,0,1,1,0,1,0,0,0,0,1,1,1,0,0},
            {0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0},
            {0,1,1,0,1,0,0,0,0,1,1,1,0,0,1,0,1,0,1,1},
            {0,1,1,0,1,0,0,0,1,1,0,1,0,1,0,0,1,1,1,1},
            {0,1,1,0,1,0,0,0,0,0,1,1,0,0,0,1,0,0,1,0},
            {0,1,1,0,0,1,1,0,0,0,0,0,1,0,0,0,0,0,1},
            {0,1,1,0,0,1,1,0,1,1,1,0,1,0,0,1,0,1,1,1},
            {0,1,1,1,0,0,0,1,0,1,0,0,0,1,1,1,1,0,1,0},
            {0,1,1,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1},
            {0,1,1,1,1,0,0,0,0,0,1,1,0,0,0,1,0,0,1,0},
        }
    }
}

```

```

{0,1,1,1,0,0,0,0,1,0,0,0,0,0,1,1,0,0,0,1},
{0,1,1,0,1,1,1,0,0,0,0,1,0,1,0,0,0,1,1,1},
{0,1,1,1,0,1,1,1,0,0,0,0,1,0,1,0,0,0,1,1},
{0,1,1,1,1,1,0,1,0,1,1,1,0,0,0,0,1,0,1,0},
{0,1,1,1,1,0,0,1,1,1,0,1,1,0,1,1,0,0,1,0},
{0,1,1,1,0,0,1,1,0,1,0,1,0,0,1,1,1,1,1,1},
{0,1,1,1,0,1,0,1,0,1,0,0,0,1,1,1,0,1,0,1,1},
{0,1,1,1,1,0,0,1,0,1,0,1,1,0,0,0,0,0,0,1},
{0,1,1,1,1,0,0,0,0,1,1,1,0,0,1,0,1,0,1,1},
{0,1,1,1,0,0,1,0,1,1,0,1,0,0,0,0,1,1,1,0},
{0,1,1,1,0,0,0,1,1,0,0,0,1,0,0,1,0,0,1,1},
{0,1,1,1,0,1,0,0,1,0,1,1,1,1,0,0,0,1,1,0},
{0,1,1,1,0,0,0,1,0,1,0,0,0,1,1,1,0,1,0},
{0,1,1,0,1,1,1,1,0,1,0,1,1,1,0,0,0,1,0},
{0,1,1,1,0,0,1,0,1,0,0,0,1,1,1,0,1,0,1},
{0,1,1,1,0,0,0,1,0,1,1,0,1,0,0,0,0,1,1,1},
{0,1,1,0,1,1,0,0,0,1,1,0,1,0,1,0,0,1,1,1},
{0,1,1,0,1,0,1,0,0,1,0,1,1,1,1,0,0,0,1,1},
{0,1,1,1,0,0,1,0,1,1,1,1,0,0,0,1,1,0,1,0},
{0,1,1,1,1,0,0,1,0,1,0,1,1,0,0,0,0,0,0,1},
{0,1,1,1,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0},
{0,1,1,1,0,0,1,1,0,1,1,1,0,1,0,0,1,0,1,1},
{0,1,1,0,1,0,1,0,0,0,0,1,1,1,0,0,1,0,1,0},
{0,1,1,0,1,0,1,1,1,1,1,0,0,1,1,1,0,1,1,0},
{0,1,1,1,0,0,1,1,0,0,0,1,0,0,1,0,0,1,1,0}
};
for(int i=0;i<=49;i++)
{
    for(int j=0;j<=19;j++)
    {
        if(a[i][j]==0)
        {
            g.drawRect(r,c,20,10);
            tw++;
        }
    }
}

```



```
        else
        {
            g.fillRect(r,c,20,10);
            tb++;
        }
        r=r+20;
    }
    c=c+10;
    r=100;
}
g.drawString("Total number of black box="+tb,600,50);
g.drawString("Total number of white box="+tw,600,80);
}
}
```

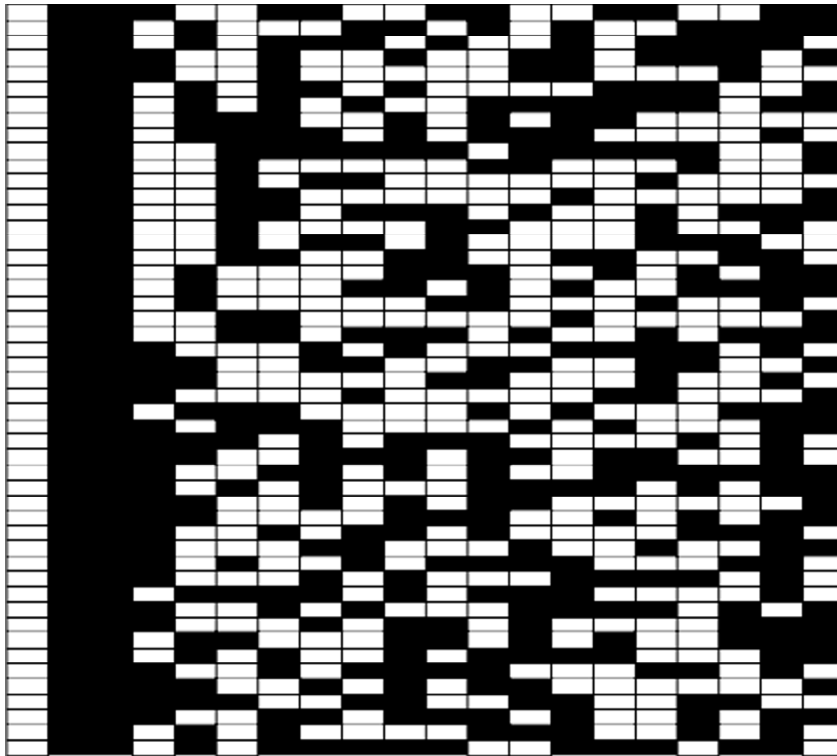


Figure 6: Cellular Automata for Normal1 Respiratory Patients and Unhealthy Controls

Number of white boxes = 486

Number of black boxes = 514

Ratio = no. of white boxes/no. of black boxes = 486/514
= 0.9455

5.3 Normal2 Respiratory Patients and Unhealthy Controls

```
import java.applet.*;
import java.awt.*;
public class normal2 extends Applet
{
    public void paint(Graphics g)
    {
        int r=100,c=20,r1=800,c1=100;
        int tb=0,tw=0,tb1=0,tw1=0;
        int a[][]={
            {0,1,1,1,1,1,0,0,0,0,1,0,1,0,0,0,1,1,1,1},
            {0,1,1,1,0,0,1,0,1,1,1,1,0,0,0,1,1,0,1,0},
            {0,1,1,1,0,1,0,0,0,0,1,1,1,0,0,1,0,1,0,1},
            {0,1,1,1,1,0,1,0,0,1,0,1,1,1,1,0,0,0,1,1},
            {0,1,1,1,0,0,1,1,1,1,0,1,0,1,1,1,0,0,0,0},
            {0,1,1,1,0,0,0,1,1,0,0,0,1,0,0,1,0,0,1,1},
            {0,1,1,1,0,0,1,0,1,1,0,1,0,0,0,0,1,1,1,0},
            {0,1,1,1,0,1,1,0,1,1,1,0,1,0,0,1,0,1,0,1},
            {0,1,1,1,1,1,1,0,0,1,0,1,0,1,1,0,0,0,0,0},
            {0,1,1,1,1,0,1,1,0,0,1,0,0,0,1,0,1,1,0,1},
            {0,1,1,1,0,0,0,0,1,0,0,0,0,0,1,1,0,0,0,1},
            {0,1,1,0,1,0,0,1,0,0,1,1,0,1,1,1,0,1,0,0},
            {0,1,1,0,1,1,0,0,1,1,1,0,1,1,0,1,1,0,0,1},
            {0,1,1,1,0,0,1,1,1,1,0,1,0,1,1,1,0,0,0,0},
            {0,1,1,0,1,0,1,1,1,1,0,0,0,1,1,0,1,0,1,0},
            {0,1,1,0,0,0,1,1,0,1,0,1,0,0,1,1,1,1,1,1},
            {0,1,1,0,1,0,1,1,0,1,1,0,0,1,0,0,0,1,0,1},
            {0,1,1,0,1,0,1,0,0,1,0,1,1,1,1,0,0,0,1,1},
            {0,1,1,1,0,0,0,1,1,1,1,0,1,0,1,1,1,0,0,0},
            {0,1,1,1,0,1,1,0,0,0,1,0,0,1,0,0,1,1,0,1},
        }
```

```

{0,1,1,1,1,0,0,0,1,0,1,1,0,1,0,0,0,0,1,1},
{0,1,1,1,0,1,0,0,0,0,0,1,1,0,0,0,1,0,0,1},
{0,1,1,1,0,1,1,0,0,1,0,0,0,1,0,1,1,0,1,0},
{0,1,1,1,0,0,0,0,0,1,0,0,0,0,0,1,1,0,0,0},
{0,1,1,1,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0},
{0,1,1,1,0,1,0,0,1,0,0,1,1,0,1,1,1,0,1,0},
{0,1,1,0,1,1,1,0,1,1,0,1,1,0,0,1,0,0,0,1},
{0,1,1,1,0,0,0,1,1,0,1,0,1,0,0,1,1,1,1,1},
{0,1,1,1,0,1,0,1,0,0,0,1,1,1,1,0,1,0,1,1},
{0,1,1,1,0,1,1,0,1,0,0,0,0,1,1,1,0,0,1,0},
{0,1,1,1,0,0,0,0,1,1,1,0,0,1,0,1,0,1,1,0},
{0,1,1,1,0,1,1,1,0,1,1,0,1,1,0,0,1,0,0,1},
{0,1,1,1,0,1,1,1,0,1,1,0,1,1,0,0,1,0,0,0},
{0,1,1,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0},
{0,1,1,1,0,1,1,0,0,0,0,0,0,1,0,0,0,0,0,1},
{0,1,1,1,0,1,1,1,0,0,1,0,1,0,1,1,0,0,0,0},
{0,1,1,1,0,1,1,1,0,0,0,0,1,0,1,0,0,0,1,1},
{0,1,1,1,0,1,1,1,0,1,0,0,1,0,1,1,1,1,0,0},
{0,1,1,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1},
{0,1,1,1,1,1,0,0,1,0,1,0,1,1,0,0,0,0,0,0},
{0,1,1,1,0,0,0,1,1,1,1,0,1,0,1,1,1,0,0,0},
{0,1,1,1,0,0,1,1,0,1,1,1,0,1,0,0,1,0,1,1},
{0,1,1,1,0,1,1,1,1,1,1,0,1,1,1,1,1,0,0,1},
{0,1,1,1,0,1,0,0,0,0,0,1,1,0,0,0,1,0,0,1},
{0,1,1,0,1,1,1,0,1,0,1,1,1,0,0,0,0,1,0,1},
{0,1,1,1,0,1,1,0,0,1,0,0,0,1,0,1,1,0,1,0},
{0,1,1,1,0,1,1,1,1,1,0,0,1,1,1,0,1,1,0,1},
{0,1,1,1,0,0,1,1,0,1,0,1,0,0,1,1,1,1,1,1},
{0,1,1,1,0,0,0,0,0,0,1,0,0,0,0,0,1,1,0,0},
{0,1,1,1,1,0,0,0,0,1,0,1,0,0,0,1,1,1,1,0}
};
for(int i=0;i<=49;i++)
{
    for(int j=0;j<=19;j++)
    {
        if(a[i][j]==0)

```

```

        {
            g.drawRect(r,c,20,10);
            tw++;
        }
    else
    {
        g.fillRect(r,c,20,10);
        tb++;
    }
    r=r+20;
}
c=c+10;
r=100;
}
g.drawString("Total number of black box="+tb,600,50);
g.drawString("Total number of white box="+tw,600,80);
}
}

```

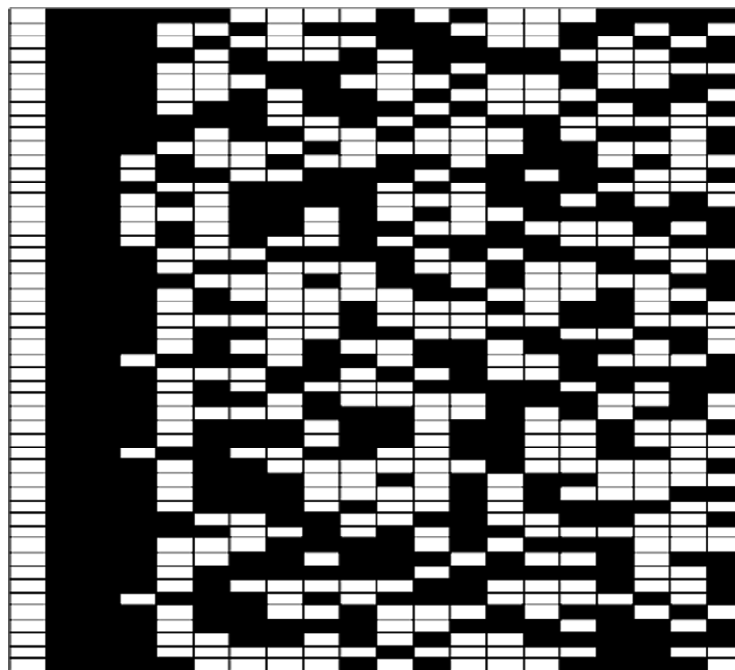


Figure 7: Cellular Automata for Normal2 Respiratory Patients and Unhealthy Controls

Number of white boxes=475

Number of black boxes=525

Ratio = no. of white boxes/no. of black boxes = 475/525
= 0.9047

5.4 Prehypertension1 Patients

```
import java.applet.*;
import java.awt.*;
public class prehyper1 extends Applet
{
    public void paint(Graphics g)
    {
        int r=100,c=20,r1=800,c1=100;
        int tb=0,tw=0,tb1=0,tw1=0;
        int a[][]={
            {0,1,1,0,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1},
            {0,1,1,0,1,0,1,0,0,0,1,1,1,1,0,1,0,1,1,1},
            {0,1,1,0,1,0,1,1,0,1,0,0,0,0,1,1,1,0,0,1},
            {0,1,1,0,1,0,0,0,1,1,0,1,0,1,0,0,1,1,1,1},
            {0,1,1,0,0,1,1,0,1,0,1,0,0,1,1,1,1,1,1,0},
            {0,1,1,0,0,1,1,1,1,1,0,1,1,1,1,1,0,0,1},
            {0,1,1,0,1,0,1,0,0,0,1,1,1,1,0,1,0,1,1,1},
            {0,1,1,0,1,1,0,1,1,1,1,1,0,0,1,1,1,0,1,1},
            {0,1,1,0,1,1,1,1,0,1,0,1,1,1,0,0,0,0,1,0},
            {0,1,1,0,1,0,0,1,1,0,1,1,1,0,1,0,0,1,0,1},
            {0,1,1,0,1,0,0,1,0,1,0,1,1,0,0,0,0,0,0,1},
            {0,1,1,0,1,1,0,0,1,0,1,0,1,1,0,0,0,0,0,0},
            {0,1,1,0,1,1,0,1,0,0,1,0,1,1,1,1,0,0,0,1},
            {0,1,1,0,1,1,0,1,0,0,1,0,1,1,1,1,0,0,0,1},
            {0,1,1,0,0,1,1,0,1,1,1,0,1,0,0,1,0,1,1,1},
            {0,1,1,0,0,1,1,1,1,0,1,0,1,1,1,0,0,0,0,1},
            {0,1,1,0,1,0,0,1,0,0,0,1,0,1,1,0,1,0,0,0},
            {0,1,1,0,1,0,1,0,0,1,1,1,1,1,0,1,1,1,1,1},
            {0,1,1,0,1,1,0,0,0,1,0,0,1,0,0,1,1,0,1,1},
            {0,1,1,0,0,1,0,1,1,1,0,0,0,1,0,1,0,0,0,0},
            {0,1,1,0,1,0,1,0,1,1,1,1,0,0,0,1,0,1,0,1},
            {0,1,1,0,0,1,0,1,1,1,0,0,0,1,0,0,0,0,0,0},
            {0,1,1,0,1,0,0,1,0,0,0,1,0,1,1,0,1,0,0,0},
            {0,1,1,0,1,0,1,0,0,1,1,1,1,1,0,1,1,1,1,1},
            {0,1,1,0,1,1,0,0,0,1,0,0,1,0,0,1,1,0,1,1},
            {0,1,1,0,0,1,0,1,1,1,1,0,0,0,1,0,1,0,1,1},
        }
```

```

{0,1,1,0,0,1,1,0,0,0,1,0,0,1,0,0,1,1,0,1},
{0,1,1,0,1,0,0,0,0,0,1,1,0,0,0,1,0,0,1,0},
{0,1,1,0,1,0,0,0,0,0,1,1,0,0,0,1,0,0,1,0},
{0,1,1,0,1,0,0,0,1,1,1,1,0,1,0,1,1,1,0,0},
{0,1,1,0,0,1,0,0,1,0,0,1,1,0,1,1,1,0,1,0},
{0,1,1,0,0,1,0,0,1,0,1,1,1,1,0,0,0,1,1,0},
{0,1,1,0,1,0,0,0,1,1,0,1,0,1,0,0,1,1,1,1},
{0,1,1,0,1,0,1,1,0,0,0,0,0,0,1,0,0,0,0,0},
{0,1,1,0,1,1,0,0,0,1,1,0,1,0,1,0,0,1,1,1},
{0,1,1,0,0,1,0,1,1,1,1,0,0,0,1,1,0,1,0,1},
{0,1,1,0,0,1,0,1,0,1,1,0,0,0,0,0,1,0,0},
{0,1,1,0,1,0,0,0,0,0,1,1,0,0,0,1,0,0,1,0},
{0,1,1,0,1,0,0,1,1,0,1,1,1,0,1,0,0,1,0,1},
{0,1,1,0,1,0,1,1,1,0,0,0,0,1,0,1,0,0,0,1},
{0,1,1,0,1,0,1,1,0,0,1,0,0,0,1,0,1,1,0,1},
{0,1,1,0,0,1,0,1,0,1,1,0,0,0,0,0,1,0,0},
{0,1,1,0,0,1,0,1,1,0,1,0,0,0,0,1,1,1,0,0},
{0,1,1,0,1,0,1,0,0,0,0,1,1,1,0,0,1,0,1,0},
{0,1,1,0,1,0,1,1,1,1,0,0,0,1,1,0,1,0,1,0},
{0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
{0,1,1,0,1,0,0,0,1,0,1,1,0,1,0,0,0,0,1,1},
{0,1,1,0,1,0,0,0,1,1,1,1,0,1,0,1,1,1,0,0},
{0,1,1,0,1,1,0,1,1,1,1,1,0,0,1,1,1,0,1,1},
{0,1,1,0,1,1,1,1,0,0,0,1,1,0,1,0,1,0,0,1},
{0,1,1,0,1,1,1,0,1,1,1,1,1,0,0,1,1,1,0,1},
{0,1,1,0,1,0,0,1,1,1,0,1,1,0,1,1,0,0,1,0},
{0,1,1,0,1,0,0,0,1,0,0,1,0,0,1,1,0,1,1,1},
{0,1,1,0,1,0,0,1,1,1,1,1,1,0,1,1,1,1,0},
{0,1,1,0,1,0,1,0,1,1,0,0,0,0,0,1,0,0,0},
{0,1,1,0,1,0,1,1,1,1,0,0,0,1,1,0,1,0,1,0}

```

```

};
for(int i=0;i<=49;i++)
{
    for(int j=0;j<=19;j++)
    {
        if(a[i][j]==0)

```

```
        {
            g.drawRect(r,c,20,10);
            tw++;
        }
        else
        {
            g.fillRect(r,c,20,10);
            tb++;
        }
        r=r+20;
    }
    c=c+10;
    r=100;
}
g.drawString("Total number of black box="+tb,600,50);
g.drawString("Total number of white box="+tw,600,80);
}
```

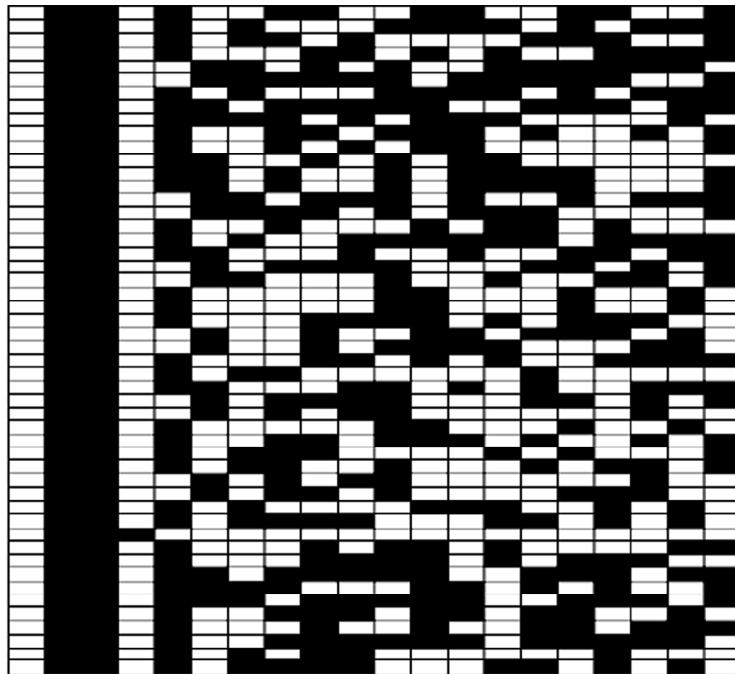


Figure 8: Cellular Automata for Prehypertension1 Patients and Healthy Controls

Number of white boxes=491

Number of black boxes=509

Ratio = no. of white boxes/no. of black boxes = $491/509$
= 0.9646

5.5 Prehypertension2

```
import java.applet.*;
import java.awt.*;
public class prehyper2 extends Applet
{
    public void paint(Graphics g)
    {
        int r=100,c=20,r1=800,c1=100;
        int tb=0,tw=0,tb1=0,tw1=0;
        int a[][]={
            {0,1,1,0,0,1,1,1,1,1,1,0,1,1,1,1,1,0,0,1},
            {0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
            {0,1,1,0,0,0,0,1,1,0,0,0,1,0,0,1,0,0,1,1},
            {0,1,1,0,0,1,0,0,0,0,0,0,1,1,0,0,0,1,0,0,1},
            {0,1,1,0,0,1,1,1,0,1,1,0,1,1,0,1,1,0,0,1,0,0,0},
            {0,1,1,0,1,0,1,0,1,0,1,1,0,0,0,0,0,0,1,0,0,0},
            {0,1,1,0,0,0,1,1,0,1,0,1,0,0,1,0,0,1,1,1,1,1},
            {0,1,1,0,0,1,0,0,0,1,0,1,1,0,1,0,1,0,0,0,0,1},
            {0,1,1,0,1,0,0,1,1,1,1,1,0,1,1,1,1,1,0,0},
            {0,1,1,0,1,1,1,0,0,1,0,1,0,1,1,0,0,0,0,0},
            {0,1,1,0,1,1,0,1,1,1,0,1,0,0,1,0,0,1,0,1,1,1},
            {0,1,1,0,0,1,1,1,1,1,0,1,1,1,0,1,1,1,1,0,0,1},
            {0,1,1,0,0,1,1,0,1,1,1,0,1,0,0,1,0,0,1,1,1},
            {0,1,1,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,0,0},
            {0,1,1,0,1,1,1,1,1,0,0,1,1,1,0,1,1,0,1,1,0,1,1},
            {0,1,1,0,1,1,1,1,1,0,0,1,1,1,0,1,1,0,1,1,0,1,1},
            {0,1,1,0,0,1,0,0,1,1,0,1,1,1,0,1,0,0,1,0,0,1,0},
            {0,1,1,0,0,1,0,0,0,1,0,1,1,0,1,1,0,1,0,0,0,0,1},
            {0,1,1,0,0,1,0,1,1,0,1,0,0,0,0,0,1,1,1,0,0},
            {0,1,1,0,1,0,0,0,0,1,1,1,0,0,1,0,1,0,1,0,1,1},
        }
```



```

{0,1,1,1,0,0,0,0,0,1,0,0,0,0,0,1,1,0,0},
{0,1,1,0,0,1,0,1,0,1,1,0,0,0,0,0,1,0,0},
{0,1,1,0,0,1,1,0,1,1,1,0,1,0,0,1,0,1,1},
{0,1,1,0,1,1,0,1,0,0,0,0,1,1,1,0,0,1,0,1},
{0,1,1,1,0,0,1,1,1,0,1,1,0,1,1,0,0,1,0,0},
{0,1,1,0,1,0,1,1,1,0,0,0,0,1,0,1,0,0,0,1},
{0,1,1,0,1,0,0,0,0,1,1,1,0,0,1,0,1,0,1,1},
{0,1,1,0,1,0,1,0,0,1,1,1,1,1,1,0,1,1,1,1},
{0,1,1,1,0,0,0,0,1,0,1,0,0,0,1,1,1,1,0,1},
{0,1,1,0,1,1,0,0,1,0,0,0,1,0,1,1,0,1,0,0},
{0,1,1,0,0,1,0,1,1,1,1,0,0,0,1,1,0,1,0,1},
{0,1,1,0,0,0,1,0,0,0,0,0,1,1,0,0,0,1,0,0},
{0,1,1,0,0,0,1,1,1,0,0,1,0,1,0,1,1,0,0,0},
{0,1,1,0,0,1,0,1,1,1,1,0,0,0,1,1,0,1,0,1},
{0,1,1,0,0,1,1,0,1,0,1,0,0,1,1,1,1,1,0},
{0,1,1,0,0,0,1,0,1,1,0,1,0,0,0,0,1,1,1,0},
{0,1,1,0,0,1,1,0,1,0,1,0,0,1,1,1,1,1,0},
{0,1,1,0,1,0,0,1,1,0,1,1,1,0,1,0,0,1,0,1},
{0,1,1,0,1,1,1,0,1,0,0,1,0,1,1,1,1,0,0,0},
{0,1,1,0,1,0,1,1,0,1,0,0,0,0,1,1,1,0,0,1},
{0,1,1,0,0,0,1,1,1,1,0,1,0,1,1,1,0,0,0,0},
{0,1,1,0,1,0,1,1,0,1,0,0,0,0,1,1,1,0,0,1},
{0,1,1,0,1,1,0,0,0,0,0,0,1,0,0,0,0,0,1,1},
{0,1,1,0,1,0,0,1,0,1,1,1,1,0,0,0,1,1,0,1},
{0,1,1,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1},
{0,1,1,0,1,0,1,0,0,1,1,1,1,1,0,1,1,1,1},
{0,1,1,0,1,1,1,1,0,1,0,1,1,1,0,0,0,0,1,0},
{0,1,1,1,0,0,0,1,1,0,1,0,1,0,0,1,1,1,1,1},
{0,1,1,0,1,0,0,1,0,0,1,1,0,1,1,1,0,1,0,0},
{0,1,1,0,1,0,0,1,1,1,1,1,0,1,1,1,1,0}
};
for(int i=0;i<=49;i++)
{
    for(int j=0;j<=19;j++)
    {
        if(a[i][j]==0)

```

```

        {
            g.drawRect(r,c,20,10);
            tw++;
        }
        else
        {
            g.fillRect(r,c,20,10);
            tb++;
        }
        r=r+20;
    }
    c=c+10;
    r=100;
}
g.drawString("Total number of black box="+tb,600,50);
g.drawString("Total number of white box="+tw,600,80);
}
}

```

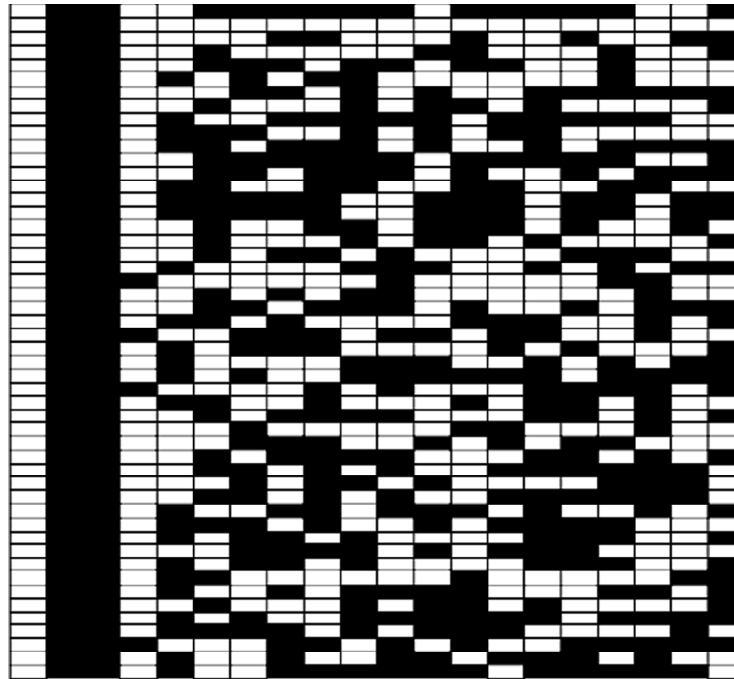


Figure 9: Cellular Automata for Prehypertension2 Patients and Healthy Controls

Number of white boxes=489

Number of black boxes=511

Ratio = no. of white boxes/no. of black boxes = 489/511
 = 0.9569

5.6 Hypertension1

```
import java.applet.*;
import java.awt.*;
public class hyper1 extends Applet
{
    public void paint(Graphics g)
    {
        int r=100,c=20,r1=800,c1=100;
        int tb=0,tw=0,tb1=0,tw1=0;
        int a[][]={
            {0,1,0,1,1,1,1,1,0,0,1,1,1,0,1,1,0,1,1,0},
            {0,1,1,0,0,0,0,0,0,1,0,0,0,0,0,1,1,0,0,0},
            {0,1,1,0,0,0,0,1,0,1,0,0,0,1,1,1,1,0,1,0},
            {0,1,1,0,0,0,0,0,1,0,0,0,0,0,1,1,0,0,0,1},
            {0,1,0,1,1,1,1,0,1,0,1,1,1,0,0,0,0,1,0,1},
            {0,1,0,1,1,1,1,1,0,1,1,1,1,1,0,0,1,1,1,0},
            {0,1,0,1,1,1,1,0,1,1,1,1,1,0,0,1,1,1,0,1},
            {0,1,1,0,0,0,1,0,1,1,0,1,0,0,0,0,1,1,1,0},
            {0,1,1,0,0,0,1,0,1,0,0,0,1,1,1,1,0,1,0,1},
            {0,1,1,0,0,0,0,1,0,0,0,0,0,1,1,0,0,0,1,0},
            {0,1,1,0,0,0,0,1,1,0,0,0,1,0,0,1,0,0,1,1},
            {0,1,1,0,0,0,0,1,1,0,0,0,1,0,0,1,0,0,1,1},
            {0,1,1,0,0,0,1,1,0,0,0,1,0,0,1,0,0,1,1,0},
            {0,1,1,0,0,0,0,0,1,1,0,0,0,1,0,0,1,0,0,1},
            {0,1,1,0,0,0,0,1,0,0,0,0,0,1,1,0,0,0,1,0},
            {0,1,1,0,0,0,0,0,1,0,0,0,0,0,1,1,0,0,0,0},
            {0,1,1,0,0,0,0,0,1,0,0,0,0,0,1,1,0,0,0,1},
            {0,1,1,0,0,0,0,0,1,1,0,0,0,1,0,0,1,0,0,1},
            {0,1,1,0,0,0,0,0,1,1,0,0,0,1,0,0,1,0,0,1},
            {0,1,0,1,1,1,1,1,0,0,1,1,1,0,1,1,0,1,1,0},
        }
```

```

    {0,1,1,0,0,0,0,0,0,1,0,0,0,0,0,1,1,0,0,0},
    {0,1,1,0,0,0,1,0,0,1,0,0,1,1,0,1,1,1,0,1},
    {0,1,1,0,0,0,1,0,0,1,0,0,1,1,0,1,1,1,0,1},
    {0,1,1,0,0,0,0,1,1,1,0,0,1,0,1,0,1,1,0,0},
    {0,1,1,0,0,0,0,0,1,1,0,0,0,1,0,0,1,0,0,1},
    {0,1,1,0,0,0,0,1,1,0,0,0,1,0,0,1,0,0,1,1},
    {0,1,1,0,0,0,1,0,1,0,0,0,1,1,1,0,1,0,1},
    {0,1,1,0,0,0,0,0,1,1,0,0,0,1,0,0,1,0,0,1},
    {0,1,0,1,1,1,1,0,1,0,1,1,1,0,0,0,0,1,0,1},
    {0,1,0,1,1,1,1,1,0,1,1,1,1,0,0,1,1,1},
    {0,1,0,1,1,1,1,0,1,1,1,1,1,0,0,1,1,1,0,1},
    {0,1,0,1,1,1,0,1,1,1,1,1,0,0,1,1,1,0,1,1},
    {0,1,0,1,1,1,1,1,0,1,1,1,1,1,0,0,1,1,1},
    {0,1,0,1,1,1,0,1,1,1,1,1,1,0,0,1,1,1,0,1},
    {0,1,0,1,1,1,1,1,0,1,1,1,1,1,0,0,1,1,1},
    {0,1,1,0,0,0,0,0,1,0,0,0,0,0,1,1,0,0,0,1},
    {0,1,0,1,1,1,1,1,0,1,1,1,1,1,0,0,1,1,1,0},
    {0,1,1,0,0,0,0,1,0,1,0,0,0,1,1,1,1,0,1,0},
    {0,1,1,0,0,0,1,0,0,0,0,0,1,1,0,0,0,1,0,0},
    {0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
    {0,1,1,0,0,0,0,1,1,0,0,0,1,0,0,1,0,0,1,1},
    {0,1,1,0,0,0,0,0,0,1,0,0,0,0,0,1,1,0,0,0},
    {0,1,0,1,1,1,1,1,0,1,1,1,1,1,0,0,1,1,1,0},
    {0,1,1,0,0,0,0,1,0,1,0,0,0,1,1,1,1,0,1,0},
    {0,1,1,0,0,0,0,0,1,1,0,0,0,1,0,0,1,0,0,1},
    {0,1,1,0,0,0,0,0,0,1,0,0,0,0,0,1,1,0,0,0},
    {0,1,1,0,0,0,0,0,1,1,0,0,0,1,0,0,1,0,0,1},
    {0,1,1,0,0,0,0,0,1,1,0,0,0,1,0,0,1,0,0,1},
    {0,1,0,1,1,1,1,1,0,1,1,1,1,1,0,0,1,1,1,0},
    {0,1,1,0,0,0,0,1,0,1,0,0,0,1,1,1,1,0,1,0},
    {0,1,1,0,0,0,1,0,0,0,0,0,1,1,0,0,0,1,0,0},
    {0,1,1,0,0,0,0,1,0,1,0,0,0,1,1,1,1,0,1,0},
    };
for(int i=0;i<=49;i++)
{
    for(int j=0;j<=19;j++)
    {
        if(a[i][j]==0)

```

```
        {  
            g.drawRect(r,c,20,10);  
            tw++;  
        }  
        else  
        {  
            g.fillRect(r,c,20,10);  
            tb++;  
        }  
        r=r+20;  
    }  
    c=c+10;  
    r=100;  
}  
g.drawString("Total number of black box="+tb,600,50);  
g.drawString("Total number of white box="+tw,600,80);  
}
```

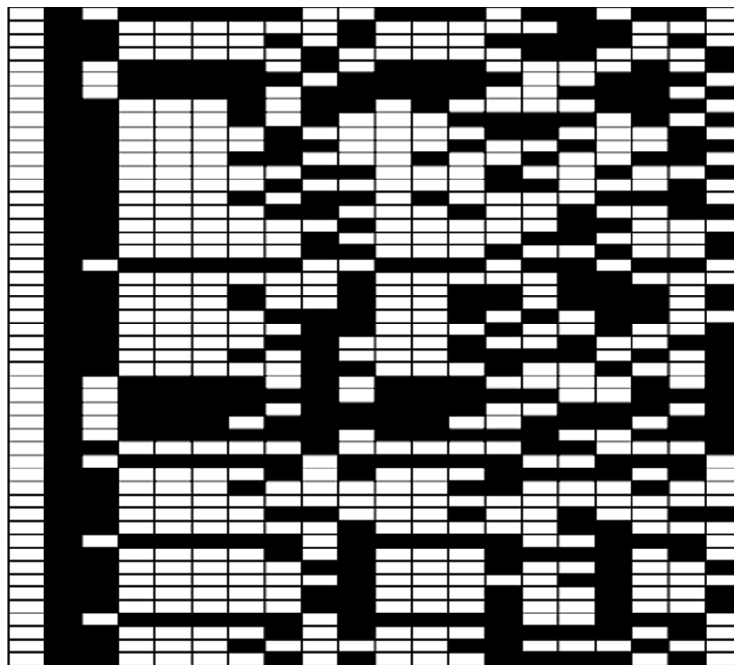


Figure 10: Cellular Automata for Hypertension1

Number of white boxes=554

Number of black boxes=446

Ratio = no. of white boxes/no. of black boxes = 554/446
= 1.24215

5.7 Hypertension2

```
import java.applet.*;
```

```
import java.awt.*;
```

```
public class hyper2 extends Applet
```

```
{
```

```
    public void paint(Graphics g)
```

```
    {
```

```
        int r=100,c=20,r1=800,c1=100;
```

```
        int tb=0,tw=0,tb1=0,tw1=0;
```

```
        int a[][]={
```

```
            {0,1,0,1,0,1,0,1,0,0,1,1,1,1,1,0,1,1,1},
```

```
            {0,1,0,1,0,1,0,1,0,0,1,1,1,1,1,0,1,1,1},
```

```
            {0,1,0,1,1,0,0,1,0,0,0,1,0,1,1,0,1,0,0},
```

```
            {0,1,0,1,1,1,1,1,0,0,1,1,1,0,1,1,0,1,0},
```

```
            {0,1,0,1,1,1,1,1,1,0,1,1,1,1,1,0,0,1,1},
```

```
            {0,1,0,1,1,1,0,1,0,0,1,0,1,1,1,1,0,0,0},
```

```
            {0,1,0,1,1,0,1,0,1,1,1,0,0,0,0,1,0,1,0},
```

```
            {0,1,0,1,1,0,0,0,0,1,0,1,0,0,0,1,1,1,0},
```

```
            {0,1,0,1,0,1,1,0,0,0,0,0,0,1,0,0,0,0},
```

```
            {0,1,0,1,0,1,0,1,1,1,0,0,0,0,1,0,1,0,0},
```

```
            {0,1,0,1,0,1,1,0,0,1,0,0,0,1,0,1,1,0,1},
```

```
            {0,1,0,1,0,1,0,0,0,1,1,1,0,1,0,1,1,1,0},
```

```
            {0,1,0,1,0,1,0,0,1,0,1,1,1,1,0,0,0,1,1},
```

```
            {0,1,0,1,0,1,0,1,1,1,0,0,0,0,1,0,1,0,0},
```

```
            {0,1,0,1,0,1,1,0,0,0,0,0,0,1,0,0,0,0},
```

```
            {0,1,0,1,0,1,1,1,1,0,0,1,1,1,0,1,1,0,1},
```

```
            {0,1,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0},
```

```
            {0,1,0,1,0,1,1,1,1,0,0,1,1,1,0,1,1,0,1},
```

```
            {0,1,0,1,0,1,0,0,0,0,1,1,1,0,0,1,0,1,0},
```

```
            {0,1,0,1,0,0,1,1,0,1,1,1,0,1,0,0,1,0,1},
```

```
            {0,1,0,1,0,0,1,0,1,1,1,1,0,0,0,1,1,0,1},
```

```
            {0,1,0,1,0,0,1,1,1,1,1,0,1,1,1,1,0,0},
```

```
            {0,1,0,1,0,1,1,0,1,1,0,0,1,0,0,0,1,0,1},
```

```
            {0,1,0,1,1,0,0,0,0,1,0,1,0,0,0,1,1,1,0},
```

```
            {0,1,0,1,1,0,1,0,0,0,0,1,1,1,0,0,1,0,1},
```

```
            {0,1,0,1,1,0,0,0,0,1,0,1,0,0,0,1,1,1,0},
```

```
            {0,1,0,1,1,0,0,0,0,1,0,1,0,0,0,1,1,1,0},
```

```
            {0,1,0,1,0,1,1,1,1,0,0,1,1,1,0,1,1,0,1},
```

```
            {0,1,0,1,0,1,1,0,0,1,0,0,0,1,0,1,1,0,1},
```

```
            {0,1,0,1,0,1,1,0,0,1,0,0,0,1,0,1,1,0,1},
```

```

        {0,1,0,1,0,1,0,0,0,1,1,1,1,0,1,0,1,1,1,0},
        {0,1,0,1,0,1,0,1,0,0,1,1,1,1,1,0,1,1,1},
        {0,1,0,1,0,1,0,0,0,1,1,1,1,0,1,0,1,1,1,0},
        {0,1,0,1,0,1,0,0,0,0,1,1,1,0,0,1,0,1,0,1},
        {0,1,0,1,0,0,1,0,0,0,1,0,1,1,0,1,0,0,0,0},
        {0,1,0,0,1,1,1,1,1,0,0,1,1,1,0,1,1,0,1,1},
        {0,1,0,1,0,0,1,0,0,0,1,0,1,1,0,1,0,0,0,0},
        {0,1,0,1,0,0,1,1,0,1,1,1,0,1,0,1,0,1,1,1},
        {0,1,0,1,0,0,1,1,0,1,1,1,0,1,0,0,1,0,1,1},
        {0,1,0,1,0,1,1,1,0,1,1,0,1,1,0,0,1,0,0,0},
        {0,1,0,1,0,1,0,1,0,0,1,1,1,1,1,0,1,1,1},
        {0,1,0,1,0,1,1,1,0,0,0,1,0,1,0,0,0,1,1},
        {0,1,0,1,0,1,1,1,0,0,0,1,0,1,0,0,0,1,1},
        {0,1,0,1,0,1,1,1,0,1,0,0,1,0,1,1,1,0,0},
        {0,1,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1},
        {0,1,0,1,1,0,0,1,0,1,0,1,1,0,0,0,0,0,0,1},
        {0,1,0,1,1,0,1,0,0,0,0,1,1,1,0,0,1,0,1,0},
        {0,1,0,1,1,0,1,0,1,1,1,0,0,0,0,1,0,1,0,0},
        {0,1,0,1,1,0,1,0,0,1,0,1,1,1,1,0,0,0,1,1},
        {0,1,0,1,1,0,0,1,0,1,0,1,1,0,0,0,0,0,0,1},
        {0,1,0,1,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1},
        {0,1,0,1,1,0,0,1,0,1,0,1,1,0,0,0,0,0,0,1},
        {0,1,0,1,1,1,0,1,0,1,1,1,0,0,0,0,1,0,1,0}
    };
    for(int i=0;i<=49;i++)
    {
        for(int j=0;j<=19;j++)
        {
            if(a[i][j]==0)
            {
                g.drawRect(r,c,20,10);
                tw++;
            }
            else
            {
                g.fillRect(r,c,20,10);
                tb++;
            }
            r=r+20;
        }
        c=c+10;
        r=100;
    }
    g.drawString("Total number of black box="+tb,600,50);
    g.drawString("Total number of white box="+tw,600,80);
}
}

```

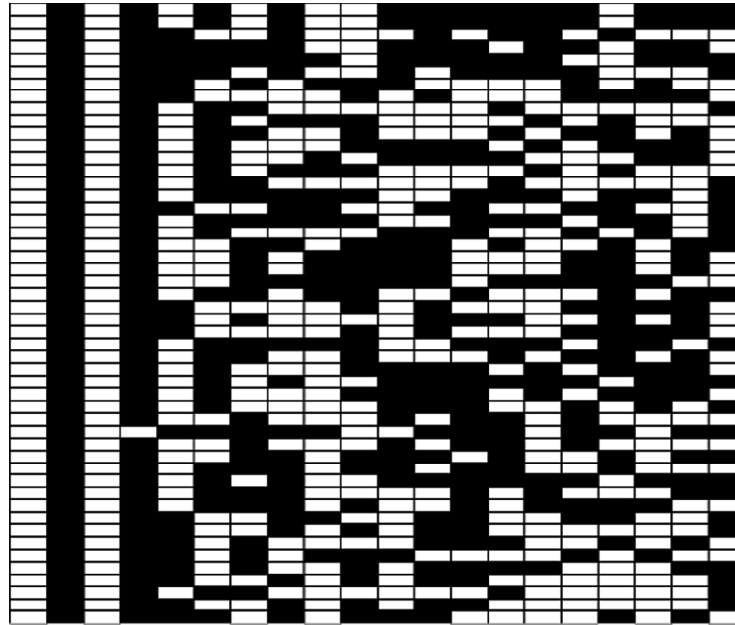


Figure 11: Cellular Automata for Hypertension2

Number of white boxes = 499

Number of black boxes = 501

Ratio = no. of white boxes/no. of black boxes = 499/501

= 0.99600

= 1(approx.)

7. CONCLUSION

It is observed from the Cellular Automata designed above that the ratio between white boxes and black boxes goes on increasing from normal to hypertension patients. Secondly by comparing normal, pre-hypertension and hypertension subjects through Cellular Automata, we observe that it is also increasing. Generally using Cellular Automata we can study effect of Heart Rate Variability through CA by considering variety of cases as mention above. In fact the ratio gives us the exact measure of HRV through CA. The above experiment is done by considering all varieties for so many diseases. Hence CA is helpful in indentifying the intensity of the diseases for a human being.

One can also study brain function and its network through nonlinear CA model.

ACKNOWLEDGEMENT

I would like to thank Miss. C.Manimegalai, Dept. of Computer science, K.M.C.P.G.S., Pondicherry, for her useful suggestion on JAVA applet program.

REFERENCES

- [1] Cerutti S., Bianchi A. M., and Mainardi L. T., Spectral Analysis of the Heart Rate Variability Signal, In: Malik M., Camm A. J. (1995): Heart Rate Variability, Futura Publishing Company Inc., New York, (1995), 64.
- [2] Majercak I., The Use of Heart Rate Variability in Cardiology, In: Bratisl Lek Listy 2002, **103**(10), (2002), 368.
- [3] Malliani A., Association of Heart Rate Variability Components with Physiological Regulatory Mechanisms, In: Malik M., Camm A. J., (1995): Heart Rate Variability, Future Publishing Company Inc., New York, (1995), 147.
- [4] Task Force of the European Society of Cardiology and the North American Society of Pacing and Electrophysiology (1996): Heart Rate Variability – Standards of Measurement, Physiological Interpretation, and Clinical Use, In: *European Heart Journal*, **17**, (1995), 354-355.
- [5] Tommaso Toffoli, “Cellular Automata as an Alternative to Differential Equations”, (1984).
- [6] URL: <http://hrvconsultants.com/documents/HeartRateVariabilityforClinicians2.ppt>, (7th October 2007).
- [7] Wolfram S., “Theory and Application of Cellular Automata”, (1986).

N. Gururajan

Retd. Prof. of Mathematics,
E-mail: vgngpondy@yahoo.com

A. Suresh Poobathy

Lecturer in Mathematics,
Pondicherry University Community College,
Lawspet, Pudhucherry-8
E-mail: spsureshpoobathy22@gmail.com

S. Senthil Kumar

Senior Resident, JIPMER, Pudhucherry-6



This document was created with the Win2PDF "print to PDF" printer available at <http://www.win2pdf.com>

This version of Win2PDF 10 is for evaluation and non-commercial use only.

This page will not be added after purchasing Win2PDF.

<http://www.win2pdf.com/purchase/>