Cloud Publications
International Journal of Advanced Packaging Technology
2015, Volume 3, Issue 1, pp 158-168, Article ID Tech-485
ISSN 2349-6665, doi 10.23953/cloud.ijapt.20

Research Article | Open Access

Insulation Effectiveness of Rice Hull

Siripong Malasri[1], Poomtawan Tiapradit[2], Stephen Russell[1], Possawat Poonpurmsiri[2], Chonnavee Tarkarnviroj[2], Ali Pourhashemi[1], Robert Moats[1] and Bradley Hudson[1]

[1]Healthcare Packaging Consortium, Christian Brothers University, 650 East Parkway South, Memphis, TN, USA
[2]Department of Packaging and Materials Technology, Kasetsart University, 50 Ngam Wong Wan Road, Lad Yao, Chatuchak, Bangkok 10900, Thailand

Correspondence should be addressed to Siripong Malasri, pong@cbu.edu

Publication Date: 28 September 2015

DOI: https://doi.org/10.23953/cloud.ijapt.20

Abstract A previous study showed loose rice hull had reasonable shock absorption even though it was not as effective as bubble wrap and anti-vibration pad. Another study showed rice hull's shock absorption was better than larger grain size crumb rubber, but worse than finer grain size crumb rubber and coconut fiber. A comparative study showed that rice hull was more effective as an insulation material than coconut fiber sheet but less effective than crumb rubber. In this study, a new insulated container was envisioned. Its wall consisted of two layers of single-wall corrugated boards with rice hull filled in the gap between them. Temperature data was collected for three thickness of rice hull, i.e., 0.5", 1.0", and 1.5". For each thickness, nine combinations of three outside temperatures (90°F, 120°F, and 150°F) and three starting interior temperatures (35°F, 45°F, and 55°F) were used. A neural network was trained to recognize the patterns of the interior temperature changes over time. The trained network can then be used for any combinations of exterior and starting interior temperatures. It assists packaging professionals in determining a proper thickness of rich hull needed for distribution. Rice hull insulated containers are suitable for a short distribution route.
Keywords *Rice Hull; Insulated Container; Heat Transfer; Artificial Neural Network; Sustainability*

1. Introduction

Loose rice hull reduced impact acceleration significantly but was less effective than bubble wrap, anti-vibration pad, coconut fiber and fine grain crumb rubber [1, 2]. However, rice hull and coconut fiber are agricultural waste products, which are environmental friendly. A comparative study [3] was performed to see the insulation effectiveness of rice hull, coconut fiber, and crumb rubber. They are potential insulation materials of future environmentally friendly insulated containers. Rice hull is more effective than coconut fiber sheet but less effective than crumb rubber.

In this study rice hull was put in a gap between two corrugated boards with three different gap widths. A total of nine combinations of exterior and interior temperatures were included in the study. Interior temperature versus time data was recorded. A neural network was trained to recognize the interior temperature changes with time under different exterior and interior temperature settings and different gap widths. The purpose of this study is to develop a neural network that can predict the time required to bring the interior temperature to a specified level for a given set of exterior and interior temperature combinations and rice hull thickness.

2. Materials and Methods

An insulated container was built from two single-wall corrugated boxes; 7"x7"x7" for the outer box and 4"x4"x4" for the inner box as shown in Figure 1. The inner box was sealed with two layers of 0.5-inch insulating sheathing on five sides (3 sides, bottom, and top). One side was spaced from the outer box which created a gap to be filled with rice hull. Three different gap widths were used in the study; 0.5", 1.0", and 1.5".

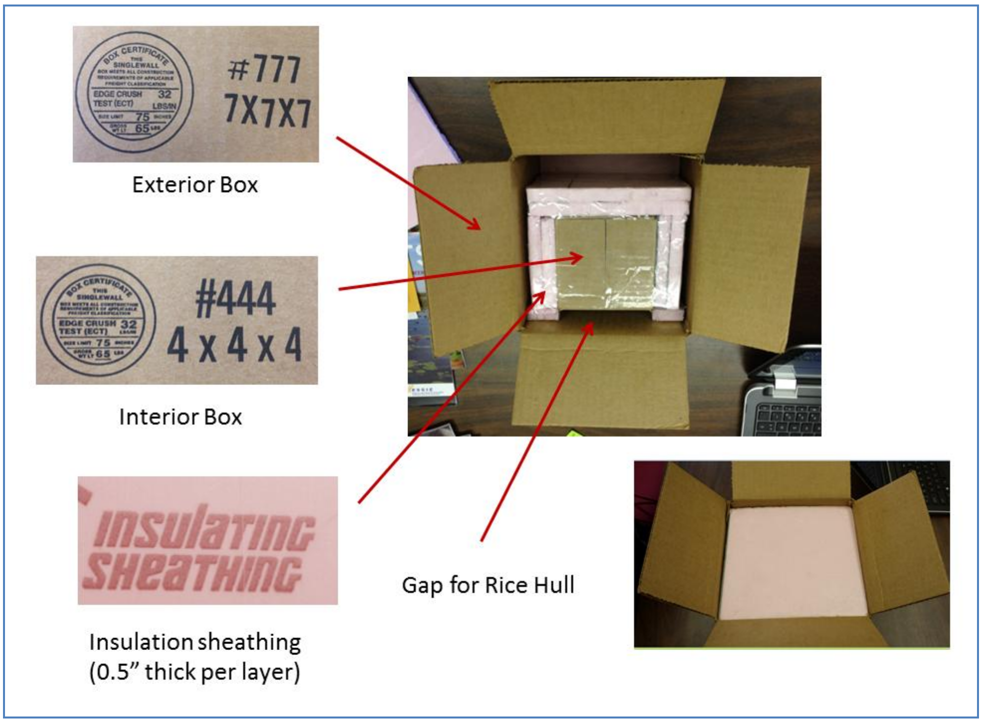
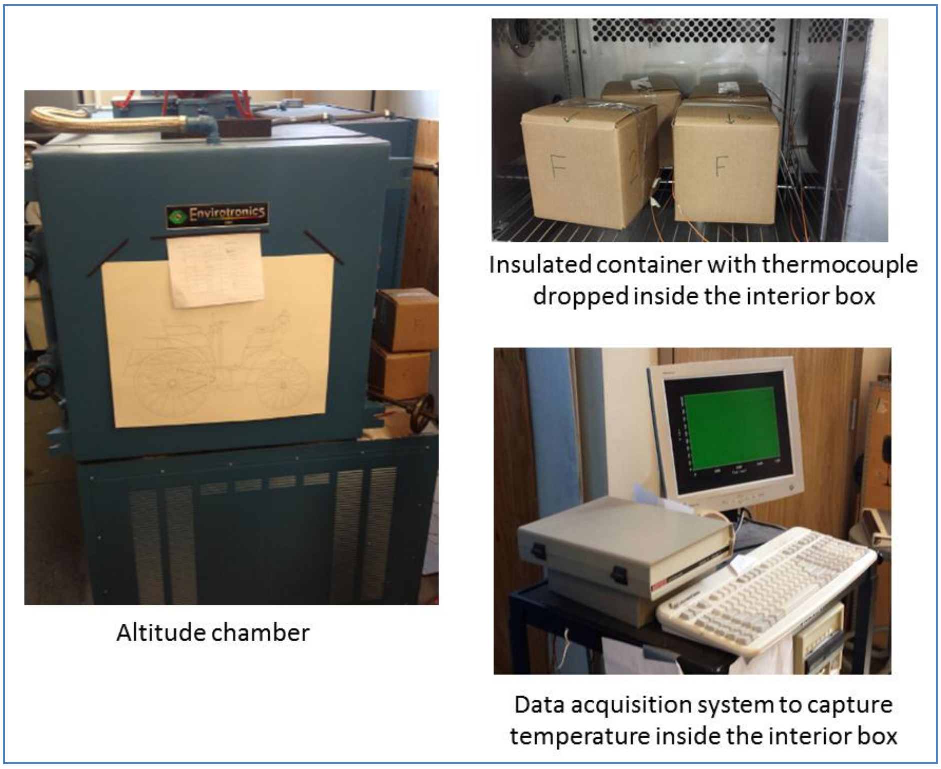


Figure 1: *Insulated Container with Sealed Inner Box for One-Direction Heat Flow*

A thermocouple was inserted inside the inner box. The box was sealed and placed in an altitude chamber. The thermocouple was connected to a data acquisition system as shown in Figure 2. It should be noted that the four insulated containers in the chamber shown in Figure 2 were taken from the previous comparative study [3] on insulation effectiveness of rice hull, coconut fiber, crumb rubber, and air.



Altitude chamber

Insulated container with thermocouple dropped inside the interior box

Data acquisition system to capture temperature inside the interior box

Figure 2: *Insulated Container, Altitude Chamber, Thermocouple, and Data Acquisition System*

The insulated container was first cooled down to a specified low temperature to simulate a cool container. Then it was raised to a specified high temperature to simulate a hot outside environment such as in a warehouse or inside a truck. A total of nine combinations as shown in Table 1 were performed for each gap thickness, i.e., 0.5”, 1.0”, or 1.5”.

Table 1: *Interior and Exterior Temperature Combinations for Each Gap Thickness*

		Interior Temperature		
		35°F	45°F	55°F
Exterior Temperature	90°F	35°F - 90°F	45°F - 90°F	55°F - 90°F
	120°F	35°F - 120°F	45°F - 120°F	55°F - 120°F
	150°F	35°F - 150°F	45°F - 150°F	55°F - 150°F

NeuroShell2 [4] was used to train a neural network to recognize temperature changes with time under each temperature combination and gap thickness.

3. Results and Discussion

Collected temperature data were plotted in Figures 3 – 5 for different thicknesses and temperature combinations.

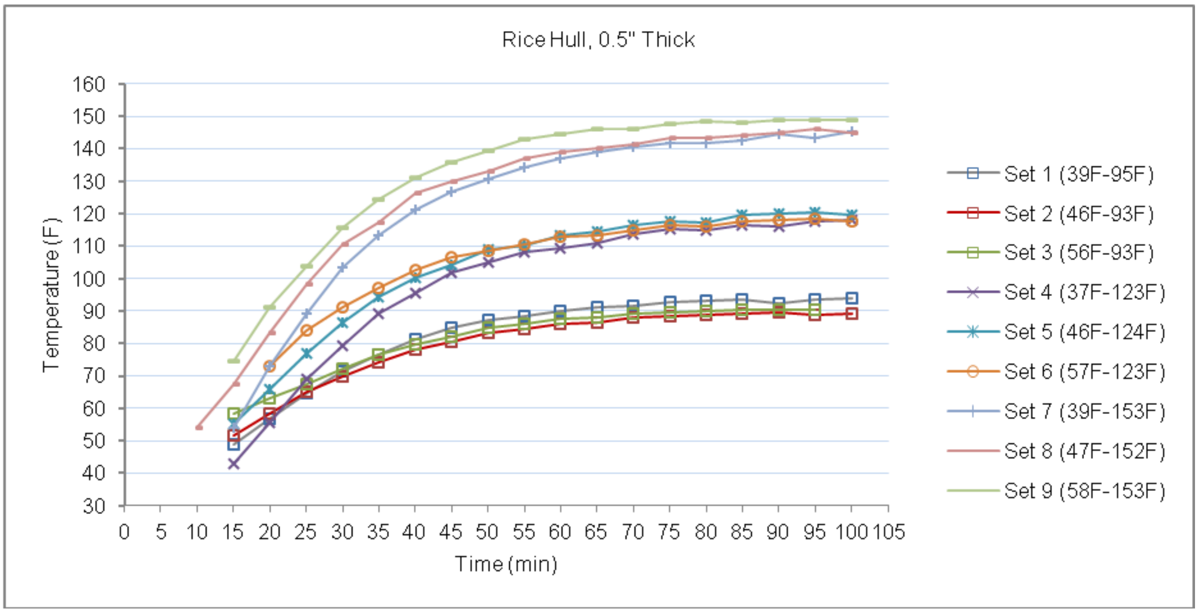


Figure 3: *Temperature Data for 0.5" Thick Rick Hull*

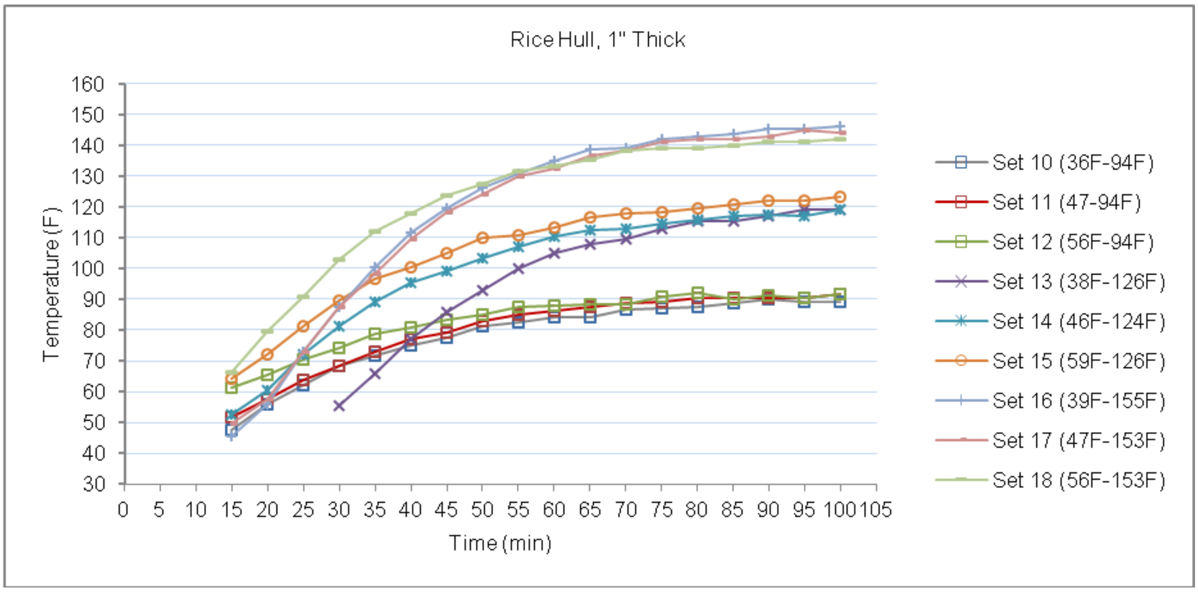


Figure 4: *Temperature Data for 1.0" Thick Rick Hull*

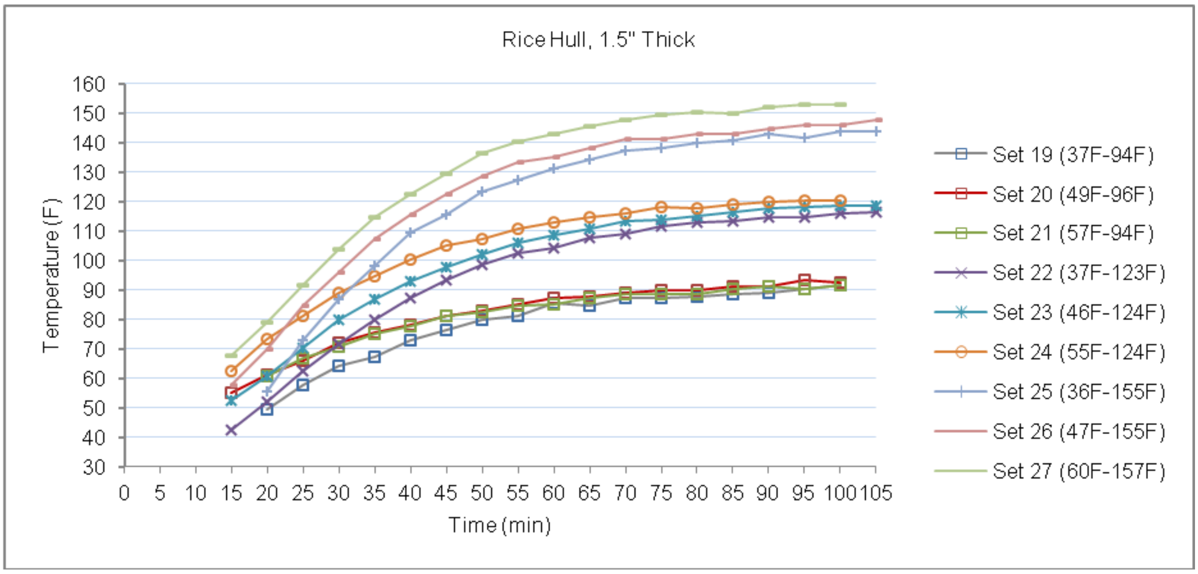


Figure 5: *Temperature Data for 1.5" Thick Rick Hull*

Collected data were tabulated for the neural network training and validation. Table 2 shows sample data obtained from the Set 2 temperature combination, which is one of the 27 sets. Inputs to the neural network are thickness of rice hull (Thick), exterior temperature (Chamber), starting temperature inside the inner box (Inner), the current temperature (Temp), and the time at current temperature (Time). The "T" and "V" marks indicate that the record is used for "Training" or "Validating."

Table 2: *Neural Network Data from Set 2 Temperature Combination*

	Input	Input	Input	Input	Output	
	(in)	(°F)	(°F)	(°F)	(min)	
Set	Thick	Chamber	Inner	Temp	Time	Mark
2	0.5	93	46	51.5	15	T
2	0.5	93	46	58.4	20	T
2	0.5	93	46	65.1	25	T
2	0.5	93	46	69.7	30	T
2	0.5	93	46	74.2	35	T
2	0.5	93	46	77.9	40	T
2	0.5	93	46	80.5	45	T
2	0.5	93	46	83.1	50	T
2	0.5	93	46	84.5	55	T
2	0.5	93	46	85.9	60	T
2	0.5	93	46	86.3	65	T
2	0.5	93	46	87.9	70	T
2	0.5	93	46	88.4	75	T
2	0.5	93	46	88.8	80	T
2	0.5	93	46	89.3	85	T
2	0.5	93	46	89.7	90	T
2	0.5	93	46	88.9	95	T
2	0.5	93	46	89.2	100	T
2	0.5	93	46	71.8	32	V
2	0.5	93	46	84.3	53	V
2	0.5	93	46	89.2	84	V

The network performance was evaluated and summarized in Table 3. “Seen Cases” are those used in network training, while “Unseen Cases” are those used in validating the network. “All Cases” are the sum of seen and unseen cases. In all categories, approximately 97% of each case gave an output within 10% error.

Table 3: *Neural Network Performance*

Category	Number Of Cases	<=5% Error	>5% to 10% Error	>10% to 20% Error	>20% Error
All Cases	568	460 (80.99%)	92 (16.20%)	14 (2.46%)	2 (0.35%)
Seen Cases	487	387 (79.47%)	86 (17.66%)	14 (2.87%)	0 (0%)
Unseen Cases	81	73 (90.12%)	6 (7.41%)	0 (0%)	2 (2.47%)

Once validated, a network source code was obtained from NeuroShell 2 as shown in Appendix. A spreadsheet was developed as a stand-alone application so the user does not need to have the NeuroShell 2 software. Figure 6 shows the input and output of the rice hull spreadsheet. The thickness used was 0.75 inch which is not part of the study. However, the neural network interpolated the result as 23.4 minutes, which is between 16.5 minutes for 0.5” thickness and 24.2 minutes for 1.0” thickness.

	A	B
1	input	
2	Insulation Thickness (in)	0.75
3	Expected Chamber Temperature (°F)	150
4	Content Starting Temperature (°F)	40
5	Content Critical Temperature (°F)	60
6		
7		
8	Output	
9	Maximum Travel Time (min)	23.4
10		

Figure 6: *Rice Hull Spreadsheet*

Figure 7 shows a comparison of different thicknesses of rice hull. As expected when the thickness was increased from 0.5” to 1.0”, it took a longer time for the heat to penetrate into the inner box. However, when the thickness was increased from 1.0” to 1.5”, the times were comparable at higher temperatures. The validity of 1.5” experiment is questionable. Thus, the use of this neural network should be limited to 1.0” thickness or less.

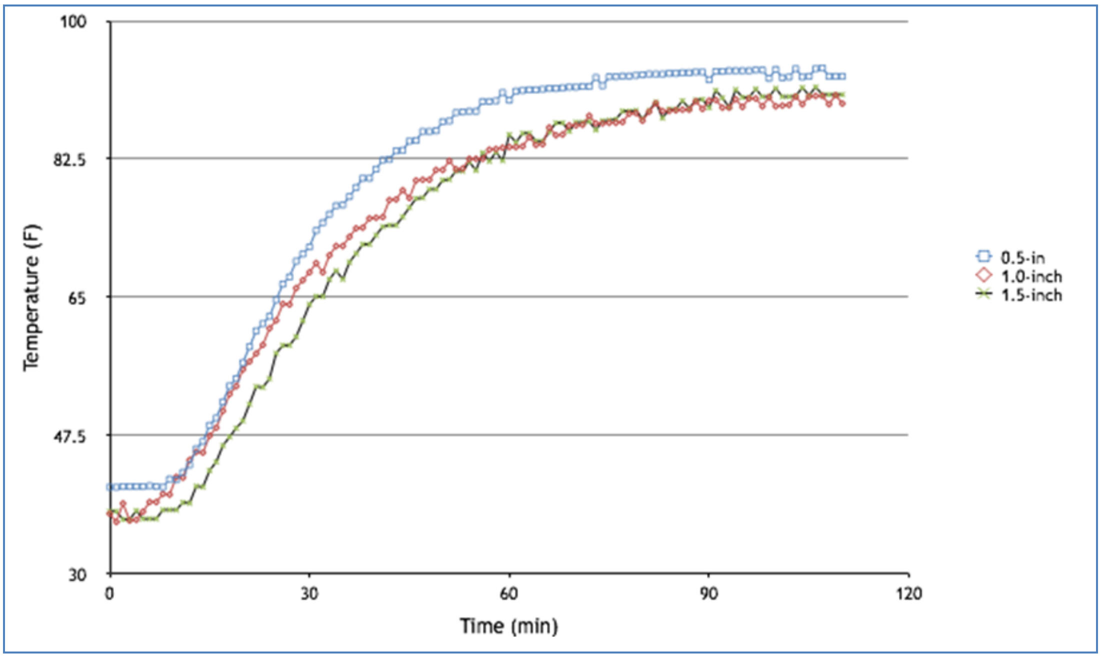


Figure 7: *Comparison of Different Rice Hull Thickness at a Comparable Temperature Range*

4. Conclusion

As seen from Figures 3 – 5, the interior temperature rises to exterior temperature in less than 1.5 hours. Depending on the threshold of the interior temperature, i.e., the temperature limit that will not affect the content, the time duration is even less. Thus, rice hull insulated containers are not suitable for a longer distribution route. It is suitable for distribution within a city, such as a distribution of medical related items.

The neural network performance is excellent. To determine an appropriate rice hull thickness, a thickness is entered into the spreadsheet along with the exterior, starting interior, and the critical or threshold temperatures. The network then gives an output of time before that the interior temperature rises to the critical temperature. It is possible to manufacture insulated containers with different thicknesses for different ranges of temperature combinations.

References

[1] Malasri, S., Stevens, R., Othmani, A., Harvey, M., Griffith, I., Guerrero, D., Johnson, M., Kist, M., Nguyen, C., Polania, S., Qureshi, A. and Sanchez-Luna, Y. *Rice Hulls as a Cushioning Materials*. International Journal of Advanced Packaging Technology. 2014. 2 (1) 112-118.

[2] Alnashwan, W., Aloumi, B., Malasri, S., Kist, M., Othmani, A., Fotso, R., Johnson, M., Polania, S. and Sanchez-Luna, Y. *Shock Absorption of Crumb Rubber and Coconut Fiber.* International Journal of Advanced Packaging Technology. 2014. 2(1) 119-128.

[3] Russell, S., Malasri, S., Poonpurmsiri, P., Tarkarnviroj, C., Tiapradit, P., Pourhashemi, A., Moats, R. and Hudson, B. *Insulation Effectiveness Comparison of Rice Hull, Coconut Fiber, and Crumb Rubber.* Proceedings of the PACKCON 2015 Conference. 2015. 4-7.

[4] Ward Systems Group, Inc. *NeuroShell 2.* http://www.wardsystems.com/neuroshell2.asp (As of September 7, 2015)

Appendix

Generic source code generated by NeuroShell 2:

```
netsum
feature2(25)
Note - the following are names of inputs and outputs:
Note - inp(1) is Thick
Note - inp(2) is Chamber
Note - inp(3) is Inner
Note - inp(4) is Temp
Note - outp(1) is Time
if (inp(1)<0.5) then inp(1) = 0.5
if (inp(1)>1.5) then inp(1) = 1.5
inp(1) = (inp(1) - 0.5)
if (inp(2)<93) then inp(2) = 93
if (inp(2)>157) then inp(2) = 157
inp(2) = (inp(2) - 93) /64
if (inp(3)<36) then inp(3) = 36
if (inp(3)>60) then inp(3) = 60
inp(3) = (inp(3) - 36) /24
if (inp(4)<42.5) then inp(4) = 42.5
if (inp(4)>153) then inp(4) = 153
inp(4) = (inp(4) - 42.5) /110.5
netsum = -77.42249
netsum = netsum + inp(1) * 1.829195
netsum = netsum + inp(2) * 51.18296
netsum = netsum + inp(3) * 2.897887
netsum = netsum + inp(4) * 25.01999
feature2(1) = 1 / (1 + exp(-netsum))
netsum = -34.30855
netsum = netsum + inp(1) * 3.381074
netsum = netsum + inp(2) * 0.6520041
netsum = netsum + inp(3) * 5.827303
netsum = netsum + inp(4) * 28.46277
feature2(2) = 1 / (1 + exp(-netsum))
netsum = 11.80751
netsum = netsum + inp(1) * -0.7572004
netsum = netsum + inp(2) * 16.44867
netsum = netsum + inp(3) * 1.609694
netsum = netsum + inp(4) * -30.24881
feature2(3) = 1 / (1 + exp(-netsum))
netsum = -19.74744
netsum = netsum + inp(1) * 11.08275
netsum = netsum + inp(2) * -11.2922
netsum = netsum + inp(3) * -11.15322
netsum = netsum + inp(4) * 18.53922
feature2(4) = 1 / (1 + exp(-netsum))
netsum = -22.27897
netsum = netsum + inp(1) * 7.165064
netsum = netsum + inp(2) * -16.6314
netsum = netsum + inp(3) * -5.059668
netsum = netsum + inp(4) * 32.52828
feature2(5) = 1 / (1 + exp(-netsum))
netsum = 0.5769625
netsum = netsum + inp(1) * 3.890337
netsum = netsum + inp(2) * -3.766037
netsum = netsum + inp(3) * 1.540869
netsum = netsum + inp(4) * -21.33095
feature2(6) = 1 / (1 + exp(-netsum))
netsum = -2.477142
netsum = netsum + inp(1) * -0.6397823
netsum = netsum + inp(2) * 0.6325201
netsum = netsum + inp(3) * 0.3864034
```

```
netsum = netsum + inp(4) * -1.387421
feature2(7) = 1 / (1 + exp(-netsum))
netsum = 2.504263E-02
netsum = netsum + inp(1) * 4.917343
netsum = netsum + inp(2) * -5.351194
netsum = netsum + inp(3) * 1.058412
netsum = netsum + inp(4) * -3.155575
feature2(8) = 1 / (1 + exp(-netsum))
netsum = -6.079099
netsum = netsum + inp(1) * -18.85942
netsum = netsum + inp(2) * -29.57689
netsum = netsum + inp(3) * -22.49723
netsum = netsum + inp(4) * 23.33121
feature2(9) = 1 / (1 + exp(-netsum))
netsum = -2.30052
netsum = netsum + inp(1) * -0.7811611
netsum = netsum + inp(2) * 0.8631783
netsum = netsum + inp(3) * 0.5986145
netsum = netsum + inp(4) * -1.770426
feature2(10) = 1 / (1 + exp(-netsum))
netsum = -1.613661
netsum = netsum + inp(1) * 1.117724
netsum = netsum + inp(2) * 1.136369
netsum = netsum + inp(3) * -1.109539
netsum = netsum + inp(4) * -0.7703084
feature2(11) = 1 / (1 + exp(-netsum))
netsum = -35.36966
netsum = netsum + inp(1) * 0.5096668
netsum = netsum + inp(2) * -37.69466
netsum = netsum + inp(3) * -4.542458
netsum = netsum + inp(4) * 78.2354
feature2(12) = 1 / (1 + exp(-netsum))
netsum = -17.01482
netsum = netsum + inp(1) * -3.632286
netsum = netsum + inp(2) * -16.90736
netsum = netsum + inp(3) * 7.179433
netsum = netsum + inp(4) * 23.45917
feature2(13) = 1 / (1 + exp(-netsum))
netsum = -2.351869
netsum = netsum + inp(1) * -0.7066271
netsum = netsum + inp(2) * 0.7919896
netsum = netsum + inp(3) * 0.516634
netsum = netsum + inp(4) * -1.666512
feature2(14) = 1 / (1 + exp(-netsum))
netsum = 0.4725413
netsum = netsum + inp(1) * -6.267524
netsum = netsum + inp(2) * -14.388
netsum = netsum + inp(3) * -21.67318
netsum = netsum + inp(4) * -3.762873
feature2(15) = 1 / (1 + exp(-netsum))
netsum = -9.839572
netsum = netsum + inp(1) * 5.391343
netsum = netsum + inp(2) * 9.922738
netsum = netsum + inp(3) * -7.104128
netsum = netsum + inp(4) * -0.6108747
feature2(16) = 1 / (1 + exp(-netsum))
netsum = -32.93026
netsum = netsum + inp(1) * 1.181343
netsum = netsum + inp(2) * -47.83091
netsum = netsum + inp(3) * -1.249812
netsum = netsum + inp(4) * 78.79494
feature2(17) = 1 / (1 + exp(-netsum))
netsum = -2.00805
netsum = netsum + inp(1) * -1.036443
netsum = netsum + inp(2) * 0.7159287
```

```
netsum = netsum + inp(3) * 0.7049346
netsum = netsum + inp(4) * -1.656927
feature2(18) = 1 / (1 + exp(-netsum))
netsum = -3.143473
netsum = netsum + inp(1) * -9.047296
netsum = netsum + inp(2) * 1.569484
netsum = netsum + inp(3) * -5.540841
netsum = netsum + inp(4) * -1.248824
feature2(19) = 1 / (1 + exp(-netsum))
netsum = -7.864933
netsum = netsum + inp(1) * -9.619619
netsum = netsum + inp(2) * 2.728712
netsum = netsum + inp(3) * 9.075184
netsum = netsum + inp(4) * -5.947243
feature2(20) = 1 / (1 + exp(-netsum))
netsum = -26.06628
netsum = netsum + inp(1) * -5.060097
netsum = netsum + inp(2) * -32.91682
netsum = netsum + inp(3) * 14.04668
netsum = netsum + inp(4) * 34.43976
feature2(21) = 1 / (1 + exp(-netsum))
netsum = 3.527127
netsum = netsum + inp(1) * -1.052538
netsum = netsum + inp(2) * 5.13706
netsum = netsum + inp(3) * 3.000223E-02
netsum = netsum + inp(4) * -9.525806
feature2(22) = 1 / (1 + exp(-netsum))
netsum = 3.17608
netsum = netsum + inp(1) * -2.799266
netsum = netsum + inp(2) * -4.537488
netsum = netsum + inp(3) * 5.004307
netsum = netsum + inp(4) * 1.735753
feature2(23) = 1 / (1 + exp(-netsum))
netsum = -1.901363
netsum = netsum + inp(1) * -1.14969
netsum = netsum + inp(2) * 0.4750226
netsum = netsum + inp(3) * 0.7146776
netsum = netsum + inp(4) * -1.402273
feature2(24) = 1 / (1 + exp(-netsum))
netsum = -37.89122
netsum = netsum + inp(1) * -17.27869
netsum = netsum + inp(2) * -1.996307
netsum = netsum + inp(3) * 10.17964
netsum = netsum + inp(4) * 32.27686
feature2(25) = 1 / (1 + exp(-netsum))
netsum = 3.233013
netsum = netsum + feature2(1) * -6.513914
netsum = netsum + feature2(2) * 5.922116
netsum = netsum + feature2(3) * -0.8871983
netsum = netsum + feature2(4) * -4.821428
netsum = netsum + feature2(5) * 3.682183
netsum = netsum + feature2(6) * -0.4269779
netsum = netsum + feature2(7) * -2.162769
netsum = netsum + feature2(8) * -0.6949077
netsum = netsum + feature2(9) * -1.166742
netsum = netsum + feature2(10) * -2.15929
netsum = netsum + feature2(11) * -1.563802
netsum = netsum + feature2(12) * 0.7780318
netsum = netsum + feature2(13) * 11.13845
netsum = netsum + feature2(14) * -2.118476
netsum = netsum + feature2(15) * -3.005923
netsum = netsum + feature2(16) * -0.7979156
netsum = netsum + feature2(17) * 0.9646804
netsum = netsum + feature2(18) * -2.262065
netsum = netsum + feature2(19) * -2.731027
```

```
netsum = netsum + feature2(20) * 0.479648
netsum = netsum + feature2(21) * -4.598935
netsum = netsum + feature2(22) * -1.077305
netsum = netsum + feature2(23) * -1.168869
netsum = netsum + feature2(24) * -2.387797
netsum = netsum + feature2(25) * -2.281764
outp(1) = 1 / (1 + exp(-netsum))
outp(1) = 105 *  (outp(1) - .1) / .8  + 10
```