

Back Propagation Neural Network for Breath Analysis to Detect Diabetes



Arul Pon Daniel D¹ and Michael Anjello JR^{2*}

¹Assistant Professor, Department of Computer Applications, Loyola College of Arts and Science, Mettala, Tamil Nadu, Namakkal District, India

²Associate Director & Science Communicator, Anantharaman Centre for Policy Research in Science & Technology Innovations for Villagers Dignity Empowerment, Don Bosco College (Co-Ed), Yelagiri Hills, Tirupattur, Tamil Nadu, India, e-mail: jothirajan@dbcylagiri.edu.in

Submission: March 26, 2024; **Published:** April 10, 2024

***Corresponding author:** Associate Director & Science Communicator, Anantharaman Centre for Policy Research in Science & Technology Innovations for Villagers Dignity Empowerment, Don Bosco College (Co-Ed), Yelagiri Hills, Tirupattur, Tamil Nadu, India, Email: jothirajan@dbcylagiri.edu.in

Abstract

Diabetes is a group of metabolic diseases which when left untreated may cause many complications. The prevailing methods used for analysing diabetes are invasive, painful and inconvenient. It is difficult to take blood samples from persons suffering from haemoglobin problems and those who are obese. Hence, this work proposes a novel idea of identifying the diabetes level through breath analysis which is a non-invasive method. This breath analyser has three phases which comprises of data acquisition, processing and classification. The two main components of the breath analyser are the sensing system and the processing unit. The sensing system can be a VOC gas sensing element (TGS822), which responds to acetone concentration in the breath out with overlapped sensitivity, producing an output which is processed, classified, and displayed. The proposed system was calibrated with chemical component and tested with exhale breathe, and classified with the accuracy of 88%. In this paper, construction of the intelligent low-cost breath analyser to diagnosis the diabetes was tested and reported.

Keywords: Diabetes; TGS822; Neural Network; Hyper Glycemia; Diagnosis

Introduction

Researchers found that certain volatile organic components in the exhaled breath are sensitive to diabetes. Hence sensors could be used to detect these components in the breath. The study is at a very early stage of undiagnosed diabetes and it will be needed to determine the strength of the link between breath and diabetes detection. Strengthening the methods for early diagnosis of diabetes as well as improved treatments will have a significant impact on cutting down death rates. Despite ongoing investigations into screening technology, research shows that the diabetes death rates have not decreased. Majority of diabetes patients have progressed to advanced stages; at the time they go for diagnosis. Diabetes is one of the leading causes of cancer deaths across the world, both in terms of standardized mortality rates and absolute numbers of people dying [1].

Electronic nose applications have been developed for a wide range of health care sectors including diagnostics, immunology, pathology, physiology, biochemical testing and disease diagnoses. Medical doctors have utilized the sense of smell to facilitate determinations of the physical state and general health of

their patients for centuries. The discovery and recognition of particular volatile organic compounds, released from various diseased human body parts or fluids derived from these tissues have been found to be associated with specific human diseases through the use of specialized modern analytical instruments. The identification of unique biomarkers associated with particular diseases has become an extremely effective and powerful tool for the early detection of diabetes in human body. For example, the analysis of patients' breath odours has had along, history of application for the detection of various human diseases, not only respiratory diseases.

Existing Methods

Diabetes mellitus is characterized by recurrent or persistent hyper glycemia, and is diagnosed by demonstrating any one of the following:

- i. Fasting plasma glucose (FPG): 8 or 12 or 14 hours after eating.
- ii. Oral glucose tolerance test (OGTT): continuous testing.

iii. Random glucose test [2-5].

Fasting Plasma Glucose Test

The FPG test is used to detect diabetes and pre-diabetes. The FPG test measures blood glucose in a person who has fasted for at least 8 hours and is most reliable when given in the morning. People with a fasting glucose level of 100 to 125 mg/dl have impaired fasting glucose (IFG), or pre-diabetes. A level of 126mg/dL or above, confirmed by repeating the test on another day, means a person has diabetes [6].

Oral Glucose Tolerance Test

When used to test for diabetes or pre-diabetes, the OGTT measures blood glucose after a person fasts for at least 8 hours and 2 hours after the person drinks a liquid containing 75 g of glucose dissolved in water. If the 2-hour blood glucose level is between 140 and 199 mg/dL, the person has a type of pre-diabetes called impaired glucose tolerance (IGT). If confirmed by a second test, a 2-hour glucose level of 200 mg/dL or above means a person has diabetes [7-10].

Random Glucose Test

Random glucose test is a blood sugar test taken from a non-fasting subject. The reference values for a "normal" random glucose test in an average adult between 140 -200 mg/dL is considered pre-diabetes, and >200mg/dL are considered diabetes [11].

Proposed Method

The main motivation for electronic noses is the development of qualitative, low-cost, real-time, and non-invasive methods to perform reliable, objective, and reproducible measures of volatile compounds and odours. To overcome the difficulties faced by the existing clinical diagnostic methods, diagnosis of diabetes using breath analysis is proposed. Breath analysis is done using an electronic nose. Breath from the humans is sampled and the Volatile Organic Compounds (VOC) in the sampled breath is analysed to identify the existing component pertaining to diabetes. For example, the patient diagnosed with diabetes will have acetone in the exhaled breath. The proposed work consists of data collection, filtration, selection, classification and testing processes.

Experimental

Three screen-printed commercially available metal-oxide semiconductor gas sensors array are used to construct the proposed electronic nose, manufactured and commercialized by Figaro Inc., for our experiments. The resulting array, populated by sensor devices tagged by the manufacturer as TGS813, TGS822, TGS2620 are placed into a test chamber, where the odorants of interest, in liquid form, are to be injected for trials. The obtained sensor element is mounted onto a stainless-steel substrate with

head of anaesthesia mask and then connected by lead wires to the pins of the sensor package. To generate the required dataset, the said test chamber is connected to a Delta Sigma Pi Analog Digital Converter (ADC), which provides versatility for conveying the chemical compounds of interest at the desired concentrations to the sensing chamber. Then, the analytes under analysis (i.e., ethanol, iso-butane and acetone) are added to this background in random order. The response of the gas sensor array was measured when the operating temperature of sensors which, according to the deterministic one-to-one look-up table provided by the manufacturer [Figaro USA, Inc.], is attained via a built-in heater that is driven by an external DC voltage source set at 5V. The sensor response is read-out in the form of the resistance across the active layer of each sensor; hence each measurement produces a 3-channel time series sequence. The data acquisition board collects the data from the gas sensors and controls the analog voltage signal to every sensor heater. This voltage is utilising a made-in-home, Python IDE environment program running on a Raspberry Pi Platform. The experimental setup is shown in Figure 1[12,13].

Profiling Breath Analyser

The profiling obtained is used for determining the sense resistance. Once the sense resistance values are obtained in Ohms this data is used for finding the right value of load resistance for maximum range ie. it is for this value of load resistance upon which the gas sensor output voltage span would be maximum. The gas concentration level change from 200 ppm to 10,000 ppm and the voltage span is not more than 700mV in this experiment. This poor sensitivity at high gas concentrations is due to the value of load resistance chosen wrongly. It is possible to derive sense resistance values at different gas concentration levels and use them to calculate the right value of load resistance for which the sensitivity would be high. It is observed that a span of 2.85 V for gas concentration changes from 200 ppm to 10,000 ppm.

The Y axis represents the Log (R_s/R_o) where R_s represents the sense resistance and R_o represents the sense resistance at 1000 ppm. X-axis represents the concentration levels in logarithmic terms. From the Sense resistance verses Gas concentration graph, it is observed that there is a linear relation between the voltage output of the sensor module and log of PPM values of gas concentration. X-axis represents the voltage out of the sensor module and Y-axis represents the log of gas concentration. Since there exists an on linear relation between the gas concentration level and voltage output of sensor module the prospect of using simple linear devices to linearize the correlation is miserable. Curve fitting is used to derive a statistical correlation between the sensor output voltage and gas concentration level. If we assume that there are no other factors that affect the sensor response then the above equation could be used to determine the gas concentration level between 200 ppm - 10000 ppm.

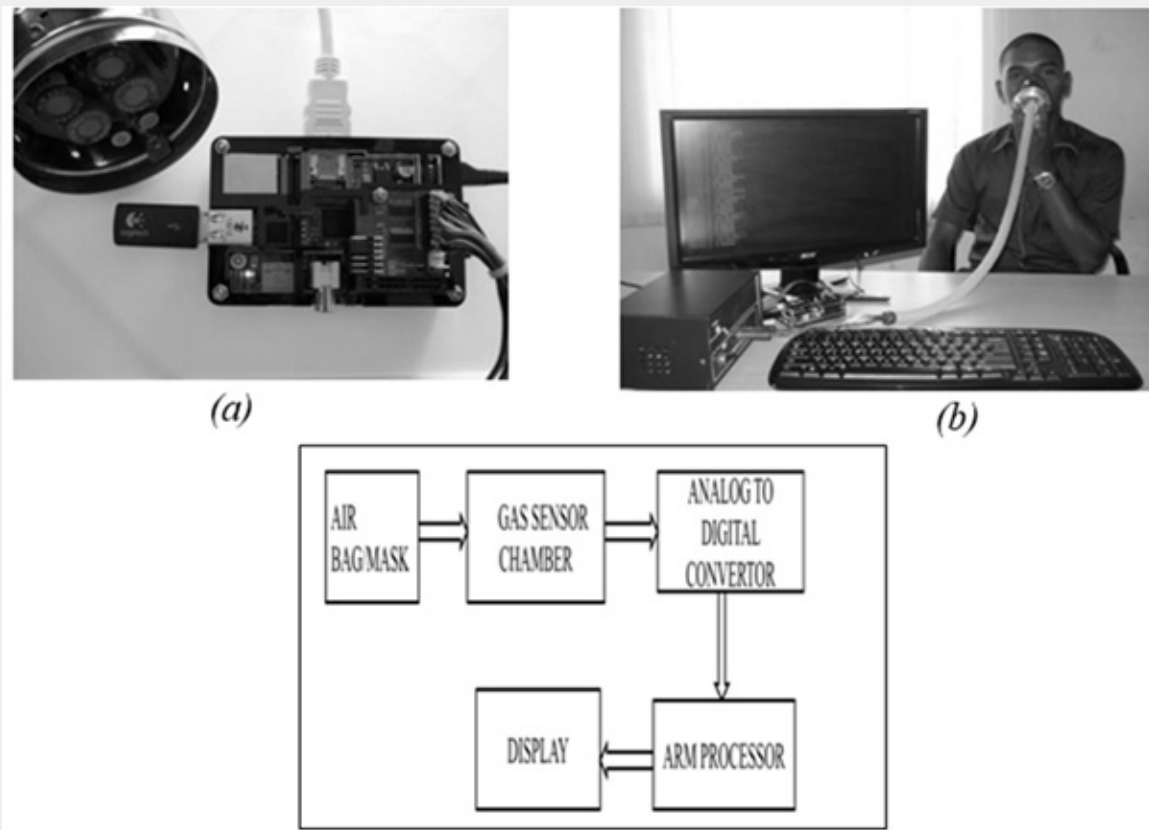


Figure 1: Electronic-Nose System using Raspberry Pi.

The relation derived after fitting a curve is as follows:

Item	Rated Values
Circuit Voltage -	5.0 ± 0.2V
Sensor Resistance -	Gas Concentration of the Sensor in Table 2
Sensor Response -	Output Volt of the Sensor

Collection of a Breath samples

Samples could be collected even from elderly or bed ridden patients without causing discomfort. The collection period was 2.0 min at 0.5 l/min, and two samples were collected; one of the

samples is at Fast, and another one is after Fast. Healthy volunteers were recruited from the Loyola College of Arts & Science, Mettala. All healthy subject samples are collected between 11.00 am - 01.30 pm as reference (Figure 2).

Results and Discussion

There are different kinds of pattern recognition methods available in the literature. In this study, Back-propagation Neural Network method is adopted and discussed in the subsequent section. The proposed electronic nose system has been designed using Python. In this experiment, before the classifying, the features are normalized between -1 to +1. Here the breath samples are tested by the classification algorithms of Naive Bayes and BPN results are achieves the accuracy of 54.7 % and 88.18 % out of that BPN gives the best accuracy (Figure 3 & Table 1). The acetone concentration from the breath out, which is a measure of the diabetic level of the person is measured using TGS 822. The output from the sensor is processed, classified, and displayed in (Tables 2 & 3) [14].

Table 1: Composition of the subject database.

Type of Subject	Number	Male/Female	Age
Healthy	16	07-Sep	23-31
Breast Cancer	6	0/6	38-55

Table 2: Acetone Level of Volunteers.

subject's chart no	fasting plasma sugar mg/100ml3	Acetone mg/100ml
182945	nil	0.39
153910	Nil	0.08
338106	140	0.17
462431	173	0.11
444006	207	0.1
99289	250	0.5
306109	294	0.25
63198	320	0.19
206944	384	0.26
290495	500	5.7

Table 3: Classification accuracy (IN%) by BPN.

Classifier	Accuracy
Naïve Bayes	64.7
BPN	88.18

```

1  513.000000
2  820.000000
3  1181.000000
4  515.000000
5  822.000000
6  1185.000000
7  515.000000
8  822.000000
9  1185.000000
10 514.000000
11 830.000000
12 1207.000000
13 519.000000
14 839.000000
15 1239.000000
16 526.000000
17 850.000000
18 1271.000000
19 530.000000
20 860.000000
21 1315.000000
    
```

Figure 2: Sample collecting using Delta-Sigma Pi.

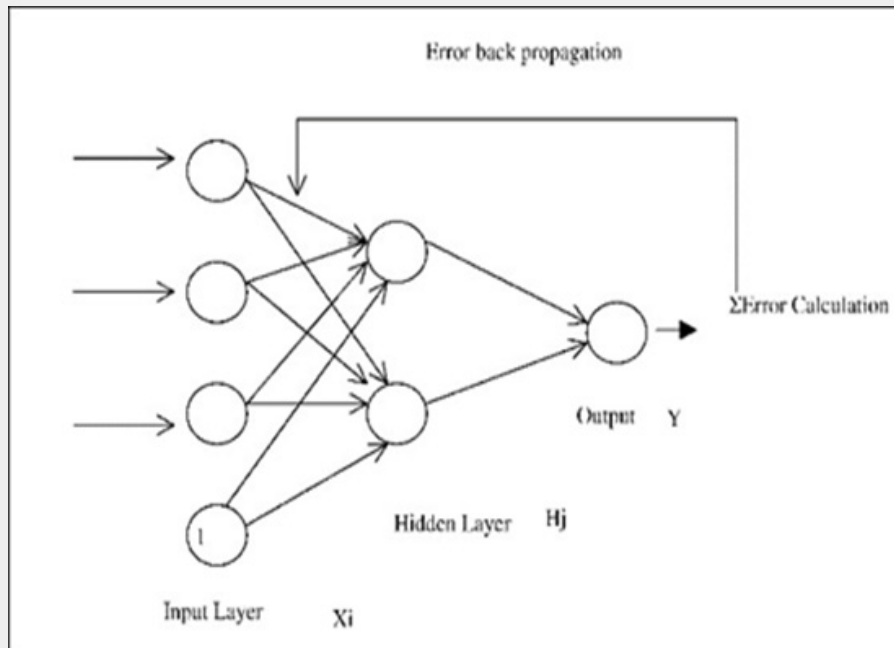


Figure 3: Back-propagation Neural Network.

Conclusion

Many research and development feasibility studies have demonstrated the effectiveness of electronic-nose technologies for detection-type applications in many diverse areas of healthcare and biomedical industries. Electronic noses have proven to be very competent and effective in discriminating between VOCs and other cellular biochemical constituents, showing great potential for improving and speeding up detections for a countless of applications. Most of this feasibility work has been done with expensive laboratory-grade instruments designed to allow maximum discriminations and sample sensitivity for rigorous scientific testing. In this paper, a low-cost artificial intelligence breath analyser is constructed by portable metal oxide based chemical sensory module using Raspberry Pi with Delta-Sigma Pi (version 2.0) ADC and coded with BPN an electronic nose system has been designed to detect the diabetes from the human breath. The human breath has been collected using VOC gas sensor and calibrated with different chemical components. The signals obtained from gas sensor have been converted in to digital signal using ADC. The signals are processed using ARM processor which is programmed with python. The proposed system is tested with chemical components and human breath and BPN classifier achieves the average accuracy of 88.18 % [15,16].

Appendix 1

Programming of ADC

```

import sys
sys.path.append('/home/pi/Python/quick2wire-python-

```

```

api')fromsmbus import SMBus
import readc_address1 = 0x68adc_address2=0x69
# create byte array and fill with initial values to define
sizeadcreading= bytearray()
adcreading.append(0x00)adcreading.append(0x00)
adcreading.append(0x00)adcreading.append(0x00)
varDivisor = 1 # from pdf sheet on adc addresses and
config#detect i2C port number andassign to i2c_bus
for line in open('/proc/cpuinfo').readlines():m=re.
match('(.*?)\s*:\s*(.*)',line)
if m:
(name,value)=(m.group(1),m.group(2))ifname=="Revision":
ifvalue[-4:]in ('0002','0003'):
i2c_bus = 0else:
i2c_bus = 1break
bus=SMBus(i2c_bus)
defchangechannel(address,adcConfig):
tmp= bus.write_byte(address, adcConfig)
defgetadcreading(address,adcConfig):
adcreading = bus.read_i2c_block_data(address,adcConfig)
h=adcreading[0]
l = adcreading[1]s=adcreading[2]

```

```

# wait for new data while (s & 128):
    adcreading = bus.read_i2c_block_data(address, adcConfig)
h = adcreading[0]

l = adcreading[1]
s = adcreading[2]

# shift bits to product result t = (h << 8) | l

# check if positive or negative number and invert if needed

if (h > 128):
    t = ~(0x020000 - t)

return t / varDivisor if = open("subject.txt", "w+") while True:

changechannel(adc_address1, 0x98)

print ("Channel 1: %02f" % getadcreading(adc_
address1, 0x98)) f.write("Sensor1: %02f\n" % getadcreading(adc_
address1, 0x8))

changechannel(adc_address1, 0xB8)

print ("Channel 2: %02f" % getadcreading(adc_
address1, 0xB8)) f.write("Sensor2: %02f\n" % getadcreading(adc_
address1, 0xB8)) changechannel(adc_address1, 0xD8)

print ("Channel 3: %02f" % getadcreading(adc_
address1, 0xD8)) f.write("Sensor3: %02f\n" % getadcreading(adc_
address1, 0xD8)) changechannel(adc_address1, 0xF8)

print ("Channel 4: %02f" % getadcreading(adc_
address1, 0xF8)) f.write("Sensor4: %02f\n" % getadcreading(adc_
address1, 0xF8)) changechannel(adc_address2, 0x98)

print ("Channel 5: %02f" % getadcreading(adc_
address2, 0x98)) f.write("Sensor5: %02f\n" % getadcreading(adc_
address2, 0x98)) changechannel(adc_address2, 0xB8)

print ("Channel 6: %02f" % getadcreading(adc_
address2, 0xB8)) f.write("Sensor6: %02f\n" % getadcreading(adc_
address2, 0xB8)) #changechannel(adc_address2, 0xD8)

# print ("Channel 7: %02f" % getadcreading(adc_
address2, 0xD8)) #changechannel(adc_address2, 0xF8)

# print ("Channel 8: %02f" % getadcreading(adc_
address2, 0xF8)) f.close()

```

Appendix 2

1.1. Programming of ARM Processor

```

# Back-Propagation Neural Networks

import math
import random
import string

random.seed(0)

# calculate a random number where: a <= rand < b defrand(a,
b):

```

```

return (b-a) * random.random() + a

# Make a matrix (we could use NumPy to speed this up)
def makeMatrix(I, J, fill=0.0):

    m = []

    for i in range(I): m.append([fill]*J)

    return m

# our sigmoid function, tanh is a little nicer than the standard
1/(1+e^-x) def sigmoid(x):

    return math.tanh(x)

# derivative of our sigmoid function, in terms of the output
(i.e. y) def dsigmoid(y):

    return 1.0 - y**2

class NN:

    def __init__(self, ni, nh, no):

        # number of input, hidden, and output nodes
        self.ni = ni + 1 # +1 for bias node
        self.nh = nh
        self.no = no

        # activations for nodes
        self.ai = [1.0]*self.ni
        self.ah = [1.0]*self.nh
        self.ao = [1.0]*self.no

        # create weights

```

References

- Vashist SK (2012) Non-invasive glucose monitoring technology in diabetes management: A review *Anal Chin Acta* 750: 16-27.
- Risby and TH, Solga S (2006) Current status of clinical breath analysis. *Appl Phys B, Lasers Opt* 85(2/3): 421-426.
- Mintch TDC, Blake DR, Galassetti PR (2012) The clinical potential of exhaled breath analysis for diabetes mellitus. *Diabetes Res Clin Pract* 97(2): 195-205.
- Hierlemann, R Gutierrez-osuna (2008) Higher-order chemical sensing. *Chem Rev* 108(2): 563-613.
- Shirasu M, Touhara K (2011) the scent of disease: volatile organic compounds of the human body related to disease and disorder. *J Biochem* 150(3): 257-66.
- Adams FE (1994) *Hippocratic Writings: Aphorism 4,5*. Web Atomics, New York, USA.
- Pauling LR, Obinson AB, Teranishi R, Cary P (1971) Quantitative analysis of urine vapor and breath by gas-liquid partition chromatography. *Proc Natl Acad Sci USA* 68(10): 2374-2376.
- Koc-Rauchenwald H, Unterkofler K, Teschl S, King J (2011) Mathematical modelling for breath gas analysis. *Forschungs für u mder Österreich is chen Fachhochs chulen*.
- Dubowski KM (1974) Breath Analysis as a Technique in Clinical Chemistry. *Clin Chem* 20(8): 966-972.
- Manolis A (1983) The diagnostic potential of breath analysis. *Clin Chem* 29(1): 5-15.

11. Francesco FD, Fuoco R, Trivella MG, Ceccarini A (2005) Breath Analysis: trends in techniques and clinical applications. *Microchemical J* 79(1): 405-410.
12. Risby TH, Solga SF (2006) Current status of clinical breath analysis. *Appl Phys B* 85: 421- 426.
13. Miekisch W, Schubert JK, Noeldge-Schomburg GF (2004) Diagnostic potential of breath analysis: focus on volatile organic compounds. *Clin Chim Acta* 347(1): 25-39.
14. Mukhopadhyay R (2004) Don't waste your breath. Researchers are developing breath tests for diagnosing diseases, but how well do they work? *Anal Chem* 76(15): 273A-276A.
15. Cao W, Duan Y (2006) Breath Analysis: Potential for clinical Diagnosis and Exposure Assessment. *Clinical Chemistry* 52(5): 800-811.
16. James A Freeman, David M Skapura (1991) *Neural Networks Algorithms, Applications and Programming Techniques* pp. 115-116.



This work is licensed under Creative Commons Attribution 4.0 License
DOI: [10.19080/CRDOJ.2024.17.555962](https://doi.org/10.19080/CRDOJ.2024.17.555962)

**Your next submission with Juniper Publishers
will reach you the below assets**

- Quality Editorial service
- Swift Peer Review
- Reprints availability
- E-prints Service
- Manuscript Podcast for convenient understanding
- Global attainment for your research
- Manuscript accessibility in different formats
(Pdf, E-pub, Full Text, Audio)
- Unceasing customer service

Track the below URL for one-step submission
<https://juniperpublishers.com/online-submission.php>