

Predictions

Show probabilities for

	RNA/TIPO DE GAS	GAS	PPM	RS/R0
39	1	1	4957.59	0.51
40	1	1	5911.18	0.48
41	1	1	6901.04	0.45
42	1	1	8000.22	0.43
43	1	1	9016.78	0.40
44	1	1	9950.33	0.39
45	2	2	0.10	49.09
46	2	2	0.13	50.99
47	2	2	0.20	49.87
48	2	2	0.30	48.77
49	2	2	0.40	47.66
50	2	2	0.50	47.13
51	2	2	0.60	46.61
52	2	2	0.69	46.64
53	2	2	0.79	45.54
54	2	2	0.90	46.14
55	2	2	1.00	45.60
56	2	2	1.97	43.57
57	2	2	2.97	44.21
58	2	2	3.98	42.68
59	2	2	4.99	41.69
60	2	2	5.99	41.73
61	2	2	6.94	41.77
62	2	2	7.94	41.80
63	2	2	8.94	41.31
64	2	2	9.94	40.83

Model	MSE	RMSE	MAE	R2
RNA/TIPO DE GAS	0.000	0.008	0.005	1.000

Restore Original Order

64

Figura 66. Resultados de predicción para el tipo de gas - 2 (Autoría propia, 2020).

En la figura 64, 65 y 66, se observa los datos de predicción que obtuvo la RNA, frente a los datos del dataset original, de esta forma evidenciando que los datos regenerados por la RNA (RNA/TIPO DE GAS), son similares a los datos de salida esperados (GAS).

Capítulo III – Análisis de resultados del prototipo

Concluido el diseño del prototipo de nariz electrónica para la detección, predicción y clasificación de compuestos volátiles en el aire, caso: gas natural. Se procedió a realizar el pertinente análisis de los resultados obtenidos para este prototipo con el fin de verificar su diseño y funcionamiento.

Dicho análisis comenzó con la verificación del funcionamiento de los sensores de gas empleados ante concentraciones de gases conocidas, en otras palabras, en aire limpio ambiente. De igual forma se verifico el funcionamiento de los sensores de gas, exponiendo dichos sensores a la presencia del gas que se pretendía detectar, de manera que los sensores nos entregaban lecturas acordes a concentración de gas presente en el ambiente. Antes se había mencionado que los sensores eran susceptibles a entorpecer su funcionamiento si estos se exponían a grandes variaciones de temperatura y humedad, sin embargo, al realizar las pertinentes pruebas de funcionalidad de los sensores de gas, estos valores de temperatura y humedad fueron constantes (no variaron), y conforme a ello los sensores de gas no sufrieron contratiempos que entorpecieran su óptimo funcionamiento.

Se evidencio que es posible enviar las lecturas obtenidas de los sensores a una red de internet mediante la conexión de un módulo ethernet mini. Al conectar la tarjeta de Arduino a un servidor web, caso de MySQL logrando transmisión de estas lecturas mediante una red de área local interna.

Posteriormente se examinó la implementación de una red neuronal artificial, para la perdición y clasificación de los diferentes tipos de gas sensados por el arreglo de sensores a fin de evaluar la implementación de la red neuronal, la cual se debió a la necesidad de “reconocer” de manera certera la presencia de gases nocivos para la salud, como lo es el gas natural en ambientes residenciales, de esta forma el prototipo de la nariz electrónica es capaz de reconocer entre la presencia de este gas a otros gases diferentes presentes en el aire, y conocer la concentración de gas presente en el aire. La red neuronal evaluada, bajo los algoritmos demostrados para la ejecución de este proyecto, y los resultados obtenidos a prueba y error mediante el entrenamiento de aprendizaje, demuestra aproximadamente un 90% de eficiencia para la predicción y clasificación de los distintos gases detectados.

Conclusiones

Realizar la respectiva investigación y documentación de los diferentes componentes de la nariz electrónica y las redes neuronales artificiales, me permitió comprender el funcionamiento de las diferentes tecnologías aplicadas en este proyecto, y en donde concentrar los mayores esfuerzos para la realización del prototipo de la forma más eficiente posible.

Mediante un arreglo de sensores de gas, es posible sensor diferentes gases volátiles presentes en el ambiente residencial, de manera que se pueda determinar la concentración del gas presente, donde dichas concentraciones se consideran peligrosas para la salud del ser vivo, como lo son las fugas en altas concentraciones de gas natural en ambientes residenciales.

Efectuar un entrenamiento de una red neuronal artificial perceptrón multicapa aplicado para la predicción y clasificación de los gases sensados en el ambiente residencial, bajo los parámetros establecidos mediante prueba y error para la red neuronal confirma que este modelo de red neuronal artificial es óptimo para la implementación de este tipo de aplicación.

Teniendo en cuenta el trabajo realizado, se corrobora que es válido realizar un prototipo de nariz electrónica para predecir y clasificar diferentes gases volátiles presentes en el entorno residencial, como lo es el gas natural, haciendo uso de instrumentos electrónicos de bajo costo y aplicando técnicas de inteligencia artificial.

Recomendaciones a trabajos futuros

Para la ejecución de trabajos futuros se recomienda utilizar otro tipo de sensores de gas diferentes a los sensores utilizados en este proyecto. Con el fin de evaluar el funcionamiento y comportamiento del prototipo empleando otra tecnología.

También se recomienda crear el software del modelo de la red neuronal artificial, en una plataforma compatible con Arduino, bajo los parámetros de diseño establecidos para este modelo red neuronal artificial.

Referencias

- 330ohms. (11 de 07 de 2016). *330ohms*. Obtenido de <https://blog.330ohms.com/2016/07/11/como-funcionan-los-sensores-de-gas/>
- Alcanos. (19 de 11 de 2020). *Alcanos*. Obtenido de <http://alcanosesp.com/que-es-el-gas-natural>
- Alvarado, H. (19 de 11 de 2020). *github*. Obtenido de <https://github.com/HarryAlvarado28/MachineLearning>
- BluRadio. (26 de 01 de 2020). Incendio en vivienda de Suba deja a una mujer gravemente herida. Bogotá D.C., Colombia.
- Botero, J. (24 de 05 de 2004). *Juriscol*. Obtenido de <http://www.suin-juriscol.gov.co/viewDocument.asp?id=4025320>
- Carbajar, C., Jimenez, A., Corona, E., & Noel, R. (2018). Nariz electrónica: Herramienta para detección de gases empleando redes neuronales artificiales. *Tecnología Digital*, 10.
- Circuitarte. (15 de 08 de 2020). *Circuitarte*. Obtenido de <https://www.circuitarte.com/categoria-producto/modulos/sensores/>
- Diaz, A. (2019). *Sistema de sensado mediante Arduino y una matriz de sensores de gases industriales*. Valencia.
- Ecured. (19 de 11 de 2020). *Ecured*. Obtenido de https://www.ecured.cu/Sensor_de_Humedad
- Flores, C., & Li, A. (2011). *Diseño de una nariz electrónica como discriminador de olores utilizando Algoritmos Genéticos Y redes Neuronales Artificiales*. Lima-Perú.
- Hernández Sampieri, R., Fernández Collado, C., & Baptista, M. (2013). *Metodología de la investigación*. México: McGraw Hill.
- Hilera, J., & Victor, M. (1 de 1 de 1995). *ResearchGate*. Obtenido de https://www.researchgate.net/publication/44343683_Rednes_neuronales_artificiales_fundamentos_modelos_y_aplicaciones_Jose_Ramon_Hilera_Gonzalez_Victor_Jose_Martinez_Hernando
- hwsensor. (19 de 11 de 2020). *hwsensor*. Obtenido de <http://www.hwsensor.com>
- Infobae. (18 de 06 de 2018). Intoxicación por inhalación de gas: Cúales son los riesgos para la salud frente a un escape. *Infobae*, pág. 1.
- Media, P. (10 de 04 de 2017). *Heat Wave*. Obtenido de <http://www.heatwave.com.mx/noticias/fuga-de-gas/>
- país, E. (26 de 10 de 2007). *elpais.com*. Obtenido de

- https://elpais.com/elpais/2007/11/26/actualidad/1196068621_850215.html
- Paredes, A., Sun Kou, M., Escobar, G., Doing, E., & Comina, G. (2016). *Implementación y evaluación de una nariz electrónica para la detección de alcoholes lineales*. Lima.
- Plata, G. (28 de 04 de 2008). *icbf*. Obtenido de https://www.icbf.gov.co/cargues/avance/docs/resolucion_mincomercioit_0936_2008.htm
- Ponce, P. (01 de 06 de 2010). *Inteligencia artificial con aplicaciones a la ingeniería*. Obtenido de <https://www.freelibros.me/ingenieria/inteligencia-artificial-con-aplicaciones-a-la-ingenieria-pedro-ponce-cruz>
- Prieto, P. (19 de 11 de 2020). *Médico*. Obtenido de <https://medicoplus.com/neurologia/partes-neurona>
- Quicazán, M., Díaz, A., & Zuluaga, C. (16 de 06 de 2016). *La nariz electrónica, una novedosa herramienta para el control de procesos y calidad en la industria agroalimentaria*. Obtenido de Scielo: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0121-40042011000200012
- Rubio, J., Aguilar, J., Ávila, F., Stein, J., & Mélendez, A. (2016). Sistema sensor para el monitoreo ambiental basado en redes neuronales. *Ingeniería Investigación y Tecnología*, 12.
- Soto Salomón, A. (10 de 02 de 2020). Colombia superará los 10 millones de usuarios de gas natural, según el Ministerio de Minas y Energía. *La Republica*, pág. 1.
- Technical-data-MQgas-sensor. (15 de 10 de 2019). *hwsensor*. Obtenido de <http://www.hwsensor.com>
- TN23todoennoticias. (19 de 02 de 2019). ¡Cuidado con las fugas de gas! En tres días se han registrado una explosión y un conato de incendio. *TN23 todo en noticias*, pág. 1.
- Vanti. (13 de 07 de 2020). *Vanti*. Obtenido de https://regimenjuridico.grupovanti.com/ver_leyes.php?id=116
- Vázquez, N. D. (2017). *Diseño de dispositivo detector de gases en electrónica flexible para equipos de protección individual*. Oviedo.
- Villamizar, R., & Barberena, A. (21 de 03 de 2011). *apolo.creg.gov.co*. Obtenido de <http://apolo.creg.gov.co/Publicac.nsf/Indice01/Resoluci%C3%B3n-1995-CRG95067>
- wiki, e. q. (10 de 10 de 2020). *Orange (Software)*. Obtenido de [https://es.qaz.wiki/wiki/Orange_\(software\)](https://es.qaz.wiki/wiki/Orange_(software))

Anexos

Se anexa el código fuente en Arduino para la ejecución de este proyecto.

```
#include <Ethernet.h>
#include <SPI.h>
#include <DHT.h>
#include <DHT_U.h>

int sensor_th=3; int Temperatura; int Humedad;

float mq2_volt; float mq2_rs; float MQ2RSR0; float MQ2PPM; float mq2_valor;
float mq3_volt; float mq3_rs; float MQ3RSR0; float MQ3PPM; float mq3_valor;
float mq6_volt; float mq6_rs; float MQ6RSR0; float MQ6PPM; float mq6_valor;

DHT dht (sensor_th, DHT22);
// Configuración ethernet
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFF, 0xEE}; // Dirección MAC
byte ip[] = { 192,168,0,50 }; // Dirección IP del Arduino
byte server[] = { 192,168,0,1 }; // Dirección IP del servidor
EthernetClient cliente;//objeto del ethernet

void setup() {
  Serial.begin(9600);
  Ethernet.begin(mac, ip); // Inicializamos el módulo Ethernet mini
  dht.begin(); // inicialización sensor dht
}

void loop() {

  Temperatura= dht.readTemperature();
  Humedad= dht.readHumidity();
  mq2_valor = analogRead(A0); // Lectura pin analógico A0
  mq2_volt = mq2_valor/1024*5.0;
```

```

mq2_rs = (5.0-mq2_volt)/mq2_volt;
mq3_valor = analogRead(A1);    // Lectura pin analogico A1
mq3_volt = mq3_valor/1024*5.0;
mq3_rs = (5.0-mq3_volt)/mq3_volt;
mq6_valor = analogRead(A2);    // Lectura pin analogico A2
mq6_volt = mq6_valor/1024*5.0;
mq6_rs = (5.0-mq6_volt)/mq6_volt;
MQ2RSR0 = mq2_rs/0.12; // Proporción = RS/R0 aire ambiente
                        //(reemplazar R0) resultado del sketch anterior
MQ3RSR0 = mq3_rs/0.05; // Proporción = RS/R0 aire ambiente
                        //(reemplazar R0) resultado del sketch anterior
MQ6RSR0 = mq6_rs/0.15; // Proporción = RS/R0 aire ambiente
                        //(reemplazar R0) resultado del sketch anterior

float x1= 29.612 * MQ2RSR0;
float x2= 40.500 * MQ3RSR0;
float x3= 1011.2 * MQ6RSR0;

MQ2PPM = pow (x1, -2.941);
MQ3PPM = pow (x2, -20.8);
MQ6PPM = pow (x3, -2.385);

Serial.print("MQ2RSR0 = ");
Serial.println(MQ2RSR0);
Serial.print(" ");
Serial.print("MQ2PPM = ");
Serial.print(MQ2PPM);
Serial.print("MQ3RSR0 = ");
Serial.println(MQ3RSR0);
Serial.print(" ");
Serial.print("MQ3PPM = ");
Serial.print(MQ3PPM);

```

```

Serial.print("MQ6RSR0 = ");
Serial.println(MQ6RSR0);
Serial.println(" ");
Serial.print("MQ6PPM = ");
Serial.print(MQ6PPM);
Serial.println(" ");
Serial.print("Temperatura = ");
Serial.print(Temperatura);
Serial.println("°C");
Serial.println(" ");
Serial.print("Humedad = ");
Serial.print(Humedad);
Serial.println("%");
Serial.println(" ");
delay(5000);

// Proceso de envío de muestras al servidor
Serial.println("Envío de dato, conectando...");
if (cliente.connect(server, 80)>0) { // Conexion con el servidor(client.connect(server, 80)>0
cliente.print("GET/ conexion_arduino.php?MQ2RSR0_php="); // Enviamos los datos por GET
cliente.println(MQ2RSR0);
cliente.print("MQ2PPM_php= ");
cliente.println(MQ2PPM);
cliente.print("MQ3RSR0_php= ");
cliente.println(MQ3RSR0);
cliente.print("MQ3PPM_php= ");
cliente.println(MQ3PPM);
cliente.print("MQ6RSR0_php= ");
cliente.println(MQ6RSR0);
cliente.print("MQ6PPM_php= ");
cliente.println(MQ6PPM);

```

```
cliente.print("Temperatura_php= ");
  cliente.println(Temperatura);
  cliente.print("Humedad_php= ");
  cliente.println(Humedad);
  cliente.println(" HTTP/1.1");
  cliente.println("User-Agent: Arduino 1.1");
  cliente.println();
}
else {
  Serial.println("Fallo en la conexion");
  Serial.println(" ");
}
cliente.stop();
cliente.flush();
delay(5000);
}
```