Received: July 3, 2019

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Optimizing Threshold Using Pearson Correlation for Selecting Features of Electronic Nose Signals

Shoffi Izza Sabilla[1]* **Riyanarto Sarno[1]** **Kuwat Triyana[2]**

[1]*Informatics Department, Institut Teknologi Sepuluh Nopember, Surabaya 60111, Indonesia*
[2]*Department of Physics, Universitas Gadjah Mada Sekip Utara, Yogyakarta, 55281, Indonesia*
* Corresponding author's Email: shoffi12@mhs.if.its.ac.id

Abstract: This research studied the use of sensors in an electronic-nose (e-nose) system to select the most efficient features to detect three classes of data signals. The data signals were collected from sensors that were assembled specifically for recognition of meat types. The development of e-noses has been the subject of many researches. However, researchers tend to install multiple sensors without optimizing their utilization, leading to higher costs. This research proposes three significant contributions to optimize the number of sensors to be used in e-noses, i.e. (i) tuning the threshold of the correlation coefficient; (ii) obtaining significant features by determining the optimal threshold of independent features; (iii) optimizing efficiency by considering the significant features and the accuracy of the results. The proposed efficiency method suggested that the number of sensors in the system could be reduced from 10 to 5. The optimal threshold suggested that 12 out of 20 features were correlated to each other. The highest accuracy obtained in this research was 92% for three classes of meats.

Keywords: Efficiency, Meats, Electronic nose, Correlation, Feature.

1. Introduction

Electronic noses (e-noses) to identify various food smells have been widely studied by researchers. The e-nose is an electronic instrument that has the ability to emulate the human nose in detecting smells. This tool can be used in many fields, for example in biomedicine for detecting diabetes [1–3], in the food-industry [4–6], for environmental monitoring [7], in agriculture [8], for detecting mixtures of food ingredients [9, 10], and detecting the methane gas content of chili [11]. An e-nose consists of a non-selective chemical gas sensor array to capture and transform smell into electric signals or sensor responses. E-noses produce electric signals that represent the overlapping aromas that are captured by the sensors with different sensitivities [12]. The sensors create specific patterns from which different smells can be identified. These patterns are not only identified by analyzing them quantitatively [13]; pattern recognition can be used for further analysis.

Once the patterns have been determined, the sensor array can be used to test samples. If a sensor produces the same value as the free-air value it is possible to eliminate it, which can increase the efficiency of the e-nose system. Up until now many researchers still put all types of sensors into the system based on the assumption that using more sensors produces better results. However, this is inefficient, for instance due to higher investment costs, unresponsiveness of some sensors, and more expensive processing. System efficiency is a challenge in any industry [14] and obviously it is commercially advantageous to cut down the production cost as much as possible [15]. Precise analysis at lower cost is a selling point that can attract customers.

This paper proposes an efficiency method for making e-noses more efficient by generating a significance threshold to determine which sensors have no correlation with other sensors. A lower correlation value can reduce the number of sensors. After the threshold value has been obtained, the significant features can be determined using category samples. Once the combination of significant features has been determined, a cost factor is used as a comparative variable to achieve the most efficient and accurate combination of sensors. This method is expected to produce better analysis at lower cost.

This research provides three significant contributions, i.e. (i) tuning the threshold of the correlation coefficient; (ii) obtaining significant features by determining the optimal threshold for independent features; (iii) optimizing efficiency by considering the significant features and the accuracy of the results. This paper is organized into the following sections: Section 2 explains previous works related to the subject. Section 3 explains the details of the proposed method used in the experiment and the three contributions of this research. Section 4 describes the results of the experiment. Section 5 is the evaluation. The final section contains the conclusion of this paper.

2. Related literature

Up until now, putting all types of sensors into an e-nose system is still done by researchers without considering the efficiency of sensor usage. They overlook the usage of features, which is necessary to produce significant results efficiently. They use the raw signal data and classify them immediately with various classifier methods. This can be avoided by first determining which features have a significant impact on differentiating each class.

Different researches utilize e-noses in different ways, for instance to classify peaches with four peach growth classes using 13 MOS sensors [16]. Another research detected the freshness of meat with very high accuracy by using 10 sensors [17]. This approach is very helpful in some cases, but the cost is very high.

In [16], the Pearson correlation was used to determine the relationship between the variables and their classes. The correlation coefficient, or Pearson's correlation, is an analysis technique to measure the strength of the relationship between two or more variables [18]. Variables have a relationship when other variables are affected when one variable is changed. In 1998, Cohen applied effect size to the strength of correlation relationship. Effect size indicates whether the correlation between two variables is strong or weak, as can be seen in Table 1. The closer the value of r (correlation coefficient) to 1, the stronger the correlation, whereas the correlation is weaker when r is closer to 0. If the value of r is positive, the value of x is above average, and the value of y is above the average of y, the same direction is indicated.

Table 1. Effect size of correlation relationship strength

Coefficients	Relationship strength
0.00	No relationship
0.01 – 0.09	Very Weak
0.10 – 0.29	Weak
0.30 – 0.49	Moderate
0.50 – 0.69	Strong
0.70 – 0.89	Very Strong
≥ 0.90	Near Perfect

Whereas if the value of r is negative, the value of x is above average and the value of y is below average.

$$Covariance(cov) = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{n-1} \quad (1)$$

$$PearsonCorrelation(r) = \frac{Cov(x,y)}{S_x S_y} \quad (2)$$

$$StandartDeviation(S_x) = \sqrt{\frac{\sum(x_i-\bar{x})^2}{n-1}} \quad (3)$$

S_x, S_y are the standard deviations of variables x and y, n, y_i are sample size. Individual sample points are indexed with $i. \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ In general, the Pearson correlation is used to calculate the correlation between independent features and dependent features [19].

3. Proposed methods

3.1 Tuning the threshold of the correlation coefficient

The correlation coefficient plays an important role in measuring the correlation between features. Therefore, a correlation coefficient threshold is needed as a reference to understand the purpose of using the features. In this research, a feature is highly correlated with another feature if the relationship strength is close to 1. It does not have a correlation when the effect size (r) is close to 0, which means it can be replaced by another feature. The method proposed in this paper is explained in Fig. 1. We took a different approach by calculating the correlation between dependent features and other dependent features using Pearson's correlation. The resulting correlation value generates a threshold value. The first threshold value is calculated using Eq. (4) threshold. The first threshold uses algorithm of the Eq. (4).

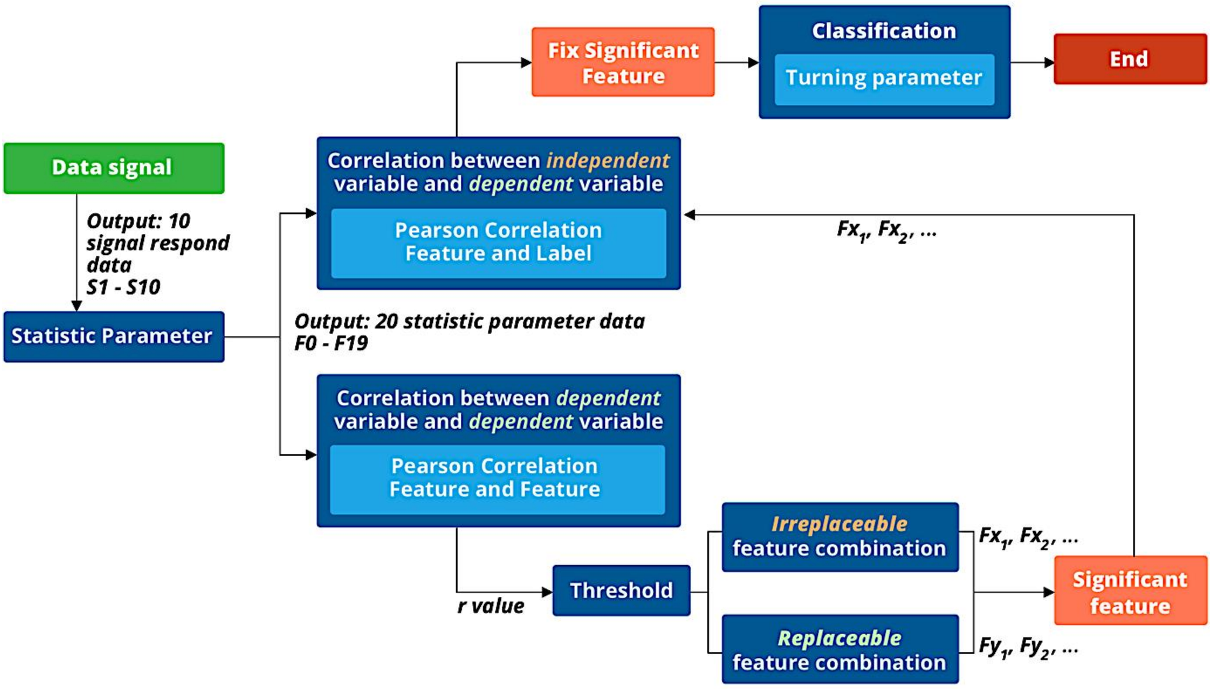

Data signal
Output: 10 signal respond data S1 - S10
Statistic Parameter
Output: 20 statistic parameter data F0 - F19
Correlation between *independent* variable and *dependent* variable
Pearson Correlation Feature and Label
Correlation between *dependent* variable and *dependent* variable
Pearson Correlation Feature and Feature
r value
Threshold
Irreplaceable feature combination
Replaceable feature combination
Fx1, Fx2, ...
Fy1, Fy2, ...
Significant feature
Fix Significant Feature
Classification
Turning parameter
End

Figure. 1 An overview of the proposed method

$$Threshold = 1 - 0.05 \qquad (4)$$

The resulting threshold value is $r \geq 0.95$. The following steps are executed: a) find features that have a value of r more than or equal to 0.95, b) select a number of features, c) check the sensors from the selected features, d) if the number of uncorrelated sensors remains the same as the number of initial sensors, use Eq. (5) to calculate the second threshold ($Threshold(n = 2)$).

$$Threshold(n) = Threshold(n - 1) - 0.05 \quad (5)$$

The steps that follow are the same as for the first threshold, the only difference is that the second threshold value is used. Iteration of the threshold tuning algorithm will be stopped when a strong correlation relationship is reached, i.e. close to 1. A new tuning threshold is calculated with $T(n) = T(n - 1) - 0.05$. $T(n)$ is the new threshold as long as the combination of all features is not found, where $T(n - 1)$is the previous threshold.

3.2 Obtaining significant features by determining the optimal threshold of independent features

All features and sensors are the main input for Pearson's correlation. The coefficient correlation is calculated with Label(Y), i.e. the independent features. The method proposed in this research calculates the coefficient correlation between each dependent feature and the other dependent features. This process aims to filter out features that yield a significant result. After the number of dependent features has been reduced by tuning the threshold, the coefficient correlation with the independent features, Label(Y), is calculated.

3.3 Optimizing efficiency by considering significant features and result accuracy

The main concern in this research was whether the utilization of sensors and features is efficient based on a specific cost allocation. This research used a cost factor to ensure that the expenses are as low as possible and optimal analysis performance is provided. For calculating the error value, this research proposes Eq. (6).

$$Efficiency = (1 - Acc) \times Vc \qquad (6)$$

where Acc, V_c are the accuracy value and the variable cost. The value of $1 - Acc$ is the error value.

4. Result and discussion

4.1 Data acquisition

An e-nose system was built to detect different meats bought at the market using an Arduino Uno microcontroller. The system passed several tests, examining both the hardware and the software components as well as calibrating the system step by step. The overall system configuration can be seen in

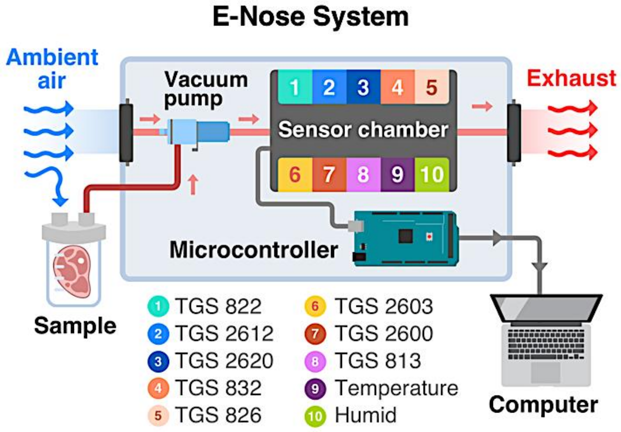


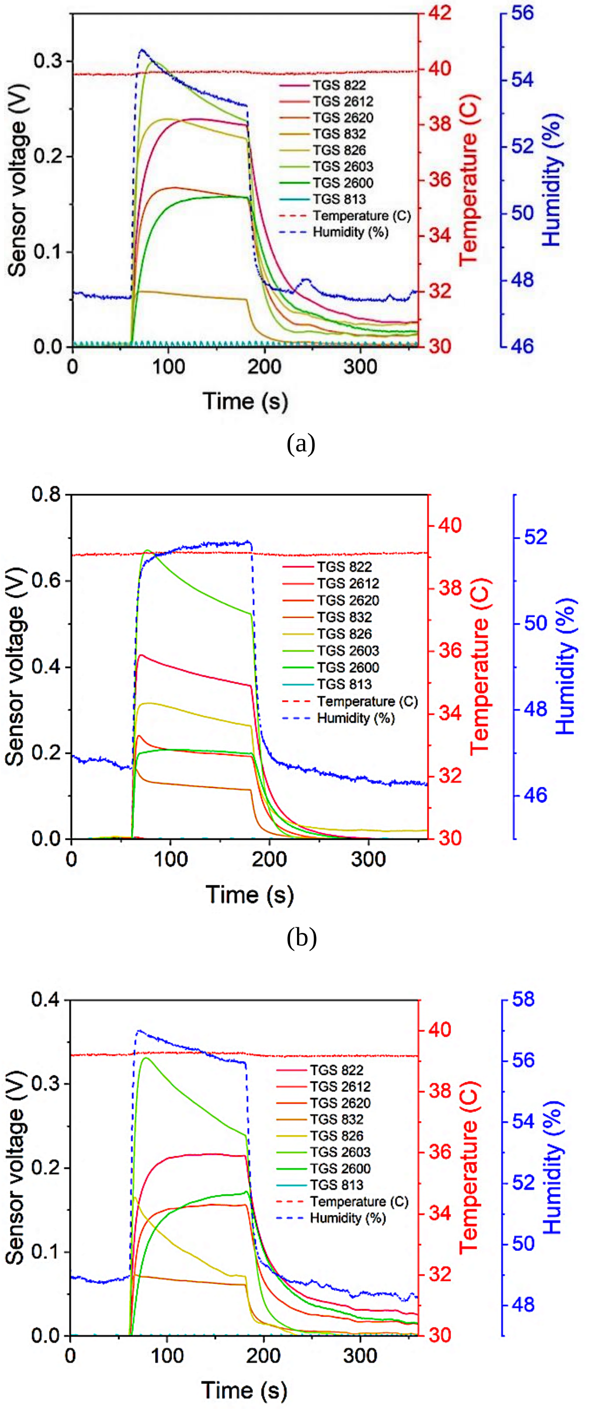
Figure. 2 E-nose system built for this research

Fig. 2 The e-nose system was connected to a computer to operate the e-nose sensor array and process the output of the system. The e-nose had several metal dioxide sensors, assembled in one unit connected to a single microcontroller. This assembly consisted of nine sensors and 10 output signals, i.e. TGS 822, TGS 2612, TGS 2620, TGS 832, TGS 826, TGS 2603, TGS 2600, TGS 813, and a temperature and humidity output sensor (DHT 21). Each sensor can identify specific gases, as shown in Table 2.

The e-nose system executes three processes: flushing, sensing, and cleaning. Each process has a specific execution time, such as 60 seconds for flushing, 120 seconds for sensing, and 180 seconds for cleaning. This research used three kinds of meat, i.e. pork (Class 1), chicken (Class 2), and beef (Class 3). During one month, fresh meat was purchased every day to ensure that the samples were in good condition.

Table 2. Sensors and measured gas contents used in the e-nose system

Sensor	Measured Gas Contents
TGS 822	Ethanol, Acetone, Methane, CO, isobutane, n- Hexane, Benzene
TGS 2612	Methane, Propane, isobutane, Ethanol
TGS 2620	Ethanol, Hydrogen, isobutane, CO, Methane
TGS 832	R22, R12, R134a, Ethanol
TGS 826	Ammonia, Ethanol, isobutane, Hydrogen
TGS 2603	Ethanol, Hydrogen, Methyl mercaptan, Trimethyl amine, H2S
TGS 2600	CO, isobutane, Ethanol, Hydrogen, Methane
TGS 813	Methane, Propane, isobutane, Hydrogen, CO, Ethanol
DHT 21	Temperature, Humidity



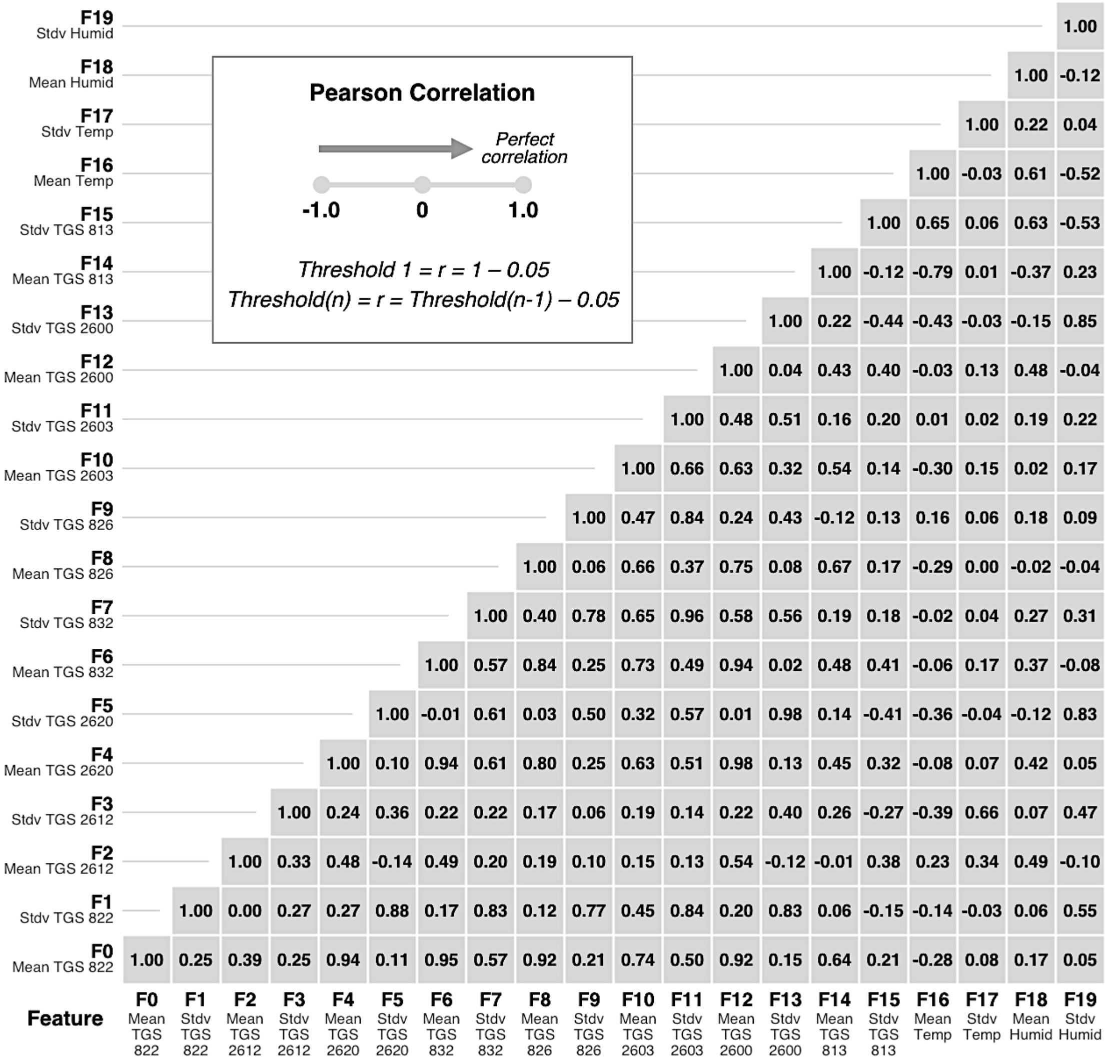
(a)

(b)

(c)

Figure. 3 Signal responses from the ten sensors of e-nose system: (a) raw signal of pork sample, (b) raw signal of chicken sample, and (c) raw signal of beef sample

There were 250 data samples, consisting of 109 chicken samples, 80 pork samples, and 61 beef samples. Each sample was placed in a beaker glass that was connected to a vacuum pump, moving the smell of the meat through the sensor chamber. All samples were tested using the e-nose, with the



Feature	F0 Mean TGS 822	F1 Stdv TGS 822	F2 Mean TGS 2612	F3 Stdv TGS 2612	F4 Mean TGS 2620	F5 Stdv TGS 2620	F6 Mean TGS 832	F7 Stdv TGS 832	F8 Mean TGS 826	F9 Stdv TGS 826	F10 Mean TGS 2603	F11 Stdv TGS 2603	F12 Mean TGS 2600	F13 Stdv TGS 2600	F14 Mean TGS 813	F15 Stdv TGS 813	F16 Mean Temp	F17 Stdv Temp	F18 Mean Humid	F19 Stdv Humid
F19 Stdv Humid																				1.00
F18 Mean Humid																			1.00	-0.12
F17 Stdv Temp																		1.00	0.22	0.04
F16 Mean Temp																	1.00	-0.03	0.61	-0.52
F15 Stdv TGS 813																1.00	0.65	0.06	0.63	-0.53
F14 Mean TGS 813															1.00	-0.12	-0.79	0.01	-0.37	0.23
F13 Stdv TGS 2600														1.00	0.22	-0.44	-0.43	-0.03	-0.15	0.85
F12 Mean TGS 2600													1.00	0.04	0.43	0.40	-0.03	0.13	0.48	-0.04
F11 Stdv TGS 2603												1.00	0.48	0.51	0.16	0.20	0.01	0.02	0.19	0.22
F10 Mean TGS 2603											1.00	0.66	0.63	0.32	0.54	0.14	-0.30	0.15	0.02	0.17
F9 Stdv TGS 826										1.00	0.47	0.84	0.24	0.43	-0.12	0.13	0.16	0.06	0.18	0.09
F8 Mean TGS 826									1.00	0.06	0.66	0.37	0.75	0.08	0.67	0.17	-0.29	0.00	-0.02	-0.04
F7 Stdv TGS 832								1.00	0.40	0.78	0.65	0.96	0.58	0.56	0.19	0.18	-0.02	0.04	0.27	0.31
F6 Mean TGS 832							1.00	0.57	0.84	0.25	0.73	0.49	0.94	0.02	0.48	0.41	-0.06	0.17	0.37	-0.08
F5 Stdv TGS 2620						1.00	-0.01	0.61	0.03	0.50	0.32	0.57	0.01	0.98	0.14	-0.41	-0.36	-0.04	-0.12	0.83
F4 Mean TGS 2620					1.00	0.10	0.94	0.61	0.80	0.25	0.63	0.51	0.98	0.13	0.45	0.32	-0.08	0.07	0.42	0.05
F3 Stdv TGS 2612				1.00	0.24	0.36	0.22	0.22	0.17	0.06	0.19	0.14	0.22	0.40	0.26	-0.27	-0.39	0.66	0.07	0.47
F2 Mean TGS 2612			1.00	0.33	0.48	-0.14	0.49	0.20	0.19	0.10	0.15	0.13	0.54	-0.12	-0.01	0.38	0.23	0.34	0.49	-0.10
F1 Stdv TGS 822		1.00	0.00	0.27	0.27	0.88	0.17	0.83	0.12	0.77	0.45	0.84	0.20	0.83	0.06	-0.15	-0.14	-0.03	0.06	0.55
F0 Mean TGS 822	1.00	0.25	0.39	0.25	0.94	0.11	0.95	0.57	0.92	0.21	0.74	0.50	0.92	0.15	0.64	0.21	-0.28	0.08	0.17	0.05

Figure. 4 Feature of correlation value with the resulted feature using Pearson

sensors being configured every 0.1 seconds. The e-nose captured the signal response, which was sent to the Arduino microcontroller. The microcontroller converted the analogue value into a digital value, resulting in 3600 data rows for each sample. The results of the three class signals were converted to a single diagram, as shown in Fig. 3. After the data signal was stored, the next step was calculating the statistic parameter or feature extraction. The statistic parameter was counted using standard deviation and mean, which expanded the 10 rows of the data signals into 20 rows of data signals. The results of the statistic parameter calculation were analyzed to ascertain the correlation potential between one feature and another. This research used Pearson correlation to measure the correlation potential. The value between the correlation feature and the compared feature must be equal to 1.00.

4.2 Result of optimized threshold for coefficient correlation between dependent features and dependent features

This research used threshold tuning for the proposed correlation coefficient. The first threshold value was $r \geq 0.98$. After the threshold was provided, the correlation value in Fig. 4 was checked, which was $r \geq 0.98$. Because a correlation value of 0.98 indicates a near-perfect correlation feature, it cannot be replaced. This feature can be replaced with other features that have the same correlation potential. The first threshold in Table 3 provided no feature for selection, which means that no combination of features that correlated with each other was found. Next, the threshold was set to $r \geq 0.96$. Similar to the procedure with the first threshold, a feature with a correlation value of $r \geq 0.96$ could not be replaced with another feature.

Table 3. Summary of sequences of the features that cannot be replaced by other features

Features	Symbols	Sensors									
		T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
Mean TGS 822	F0		x	x	x	x	x	x	x	x	x
Stdv TGS 822	F1				x	x	x	x	x	x	x
Mean TGS 2612	F2										x
Stdv TGS 2612	F3								x	x	x
Mean TGS 2620	F4	x	x	x	x	x	x	x	x	x	x
Stdv TGS 2620	F5	x	x	x	x	x	x	x	x	x	x
Mean TGS 832	F6		x	x	x	x	x	x	x	x	x
Stdv TGS 832	F7	x	x	x	x	x	x	x	x	x	x
Mean TGS 826	F8			x	x	x	x	x	x	x	x
Stdv TGS 826	F9					x	x	x	x	x	x
Mean TGS 2603	F10							x	x	x	x
Stdv TGS 2603	F11		x	x	x	x	x	x	x	x	x
Mean TGS 2600	F12	x	x	x	x	x	x	x	x	x	x
Stdv TGS 2600	F13	x	x	x	x	x	x	x	x	x	x
Mean TGS 813	F14						x	x	x	x	x
Stdv TGS 813	F15								x	x	x
Mean Temp	F16						x	x	x	x	x
Stdv Temp	F17								x	x	x
Mean Humid	F18									x	x
Stdv Humid	F19					x	x	x	x	x	x

In Table 4, F4, F5, F7, F12, F13 are non-replaceable features when 3 sensors are used. The second threshold reduced the number of sensors by 7 and reduced the number of features by 15, leaving 3 sensors and 5 features. The threshold needed to be lowered again, reducing it to $r \geq 0.95$. The third threshold produced 5 sensors and 8 features. The combination of features obtained was F0, F4, F5, F6, F7, F11, F12, and F13. As this research was expected to utilize both sensors and features efficiently with high accuracy, another threshold was needed. The fourth threshold was $r \geq 0.90$, producing 6 sensors and 9 features. These features were F0, F4, F5, F6, F7, F8, F11, F12, and F13. The fifth threshold was $r \geq 0.85$, producing 6 sensors and 10 features. The sixth threshold was $r \geq 0.80$, producing 7 sensors and 12 features. The seventh threshold was $r \geq 0.75$, producing 9 sensors and 14 features. Now, the threshold was nearly perfect because the combination of sensors consisted of 9 sensors. One more sensor combination was needed to achieve the target. The eighth threshold was $r \geq 0.70$, producing 10 sensor and 15 features. The final step was obtaining 20 combinations of features. The ninth threshold was $r \geq 0.65$, producing 10 sensors and 18 features. The tenth threshold was $r \geq 0.60$, producing 10 sensors and 19 features. The tenth threshold was the final threshold, where all combinations of sensors and features had been obtained.

This proposed method indicated a number of features that were significantly different from the other features, as shown in In Table 3. These features could not be replaced by other features. Features F4, F5, F7, F12 and F13 must be used with all thresholds. The 'x' symbol in Table 3 presents an initialization of a feature that cannot be replaced due to a high correlation value.

4.3 Classification result of significant features with independent features

After the final combination has been obtained, the next step is to test the correlation features with Label(Y) to ensure that the features are efficient and cannot be reduced any more. The accuracy of the last combination of features was calculated with Label(Y).

This research used several classification methods to provide the parameters to support this process. There are several classification methods available for this process. We refer to the previous researches [12, 20-25] for the selection of the classification method using 2 parameters, which are C and gamma (γ). Both parameters are suitable for the *support vector machine* (SVM) method. The values for C were 0.001, 0.01, 0.1, 1, 5, 10, 100, and the values for gamma (γ) were 0.001, 0.01, 0.1, 1, 5, 10, 100. *K-nearest neighbor* (KNN) uses two parameters, namely number of neighbors and weighting method.

Table 4. Combination result of features and sensors from the threshold

Threshold	Sensor Usage	Feature Usage	Classifier			
			RF	SVM	KNN	MLP
$r \geq 0.98$	0 sensor	0 feature;	0	0	0	0
$r \geq 0.96$	3 sensors: TGS 2600, TGS 2620, TGS 832	5 features: F4, F5, F7, F12, F13	0.688	0.720	0.676	0.564
$r \geq 0.95$	5 sensors: TGS 2600, TGS 2603, TGS 2620, TGS 822, TGS 832	8 features: F0, F4, F5, F6, F7, F8, F11, F12, F13	0.800	0.868	0.808	0.804
$r \geq 0.90$	6 sensors: TGS 2600, TGS 2603, TGS 2620, TGS 822, TGS 826, TGS 832	9 features: F0, F4, F5, F6, F7, F8, F11, F12, F13	0.856	0.884	0.844	0.816
$r \geq 0.85$	6 sensors: TGS 2600, TGS 2603, TGS 2620, TGS 822, TGS 826, TGS 832	10 features: F0, F1, F4, F5, F6, F7, F8, F11, F12, F13	0.868	0.892	0.844	0.832
$r \geq 0.80$	7 sensors: TGS 2600, TGS 2603, TGS 2620, TGS 822, TGS 826, TGS 832, Humid	12 features: F0, F1, F4, F5, F6, F7, F8, F9, F11, F12, F13, F19	0.844	0.896	0.824	0.828
$r \geq 0.75$	9 sensors: TGS 2600, TGS 2603, TGS 2620, TGS 822, TGS 826, TGS 832, Humid, TGS 813, Temp	14 features: F0, F1, F4, F5, F6, F7, F8, F9, F11, F12, F13, F14, F16, F19	0.836	0.920	0.860	0.900
$r \geq 0.70$	9 sensors: TGS 2600, TGS 2603, TGS 2620, TGS 822, TGS 826, TGS 832, Humid, TGS 813, Temp	15 features: F0, F1, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F16, F19	0.852	0.920	0.876	0.888
$r \geq 0.65$	10 sensors: TGS 2600, TGS 2603, TGS 2620, TGS 822, TGS 826, TGS 832, Humid, TGS 813, Temp, TGS 2612	18 features: F0, F1, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16, F17, F19	0.864	0.904	0.884	0.876
$r \geq 0.60$	10 sensors: TGS 2600, TGS 2603, TGS 2620, TGS 822, TGS 826, TGS 832, Humid, TGS 813, Temp, TGS 2612	19 features: F0, F1, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16, F17, F18, F19	0.868	0.904	0.876	0.888
$r \geq 0.50$	10 sensors: TGS 2600, TGS 2603, TGS 2620, TGS 822, TGS 826, TGS 832, Humid, TGS 813, Temp, TGS 2612	20 features: F0, F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16, F17, F18, F19	0.848	0.900	0.868	0.876

The values for number of neighbors were 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, and weighting method was ‘distance’ or ‘uniform’. *Multilayer perceptron* (MLP) uses three parameters, namely activation, number of nodes in the hidden layer, and number of iterations. The activation models given are logistic, tanh, ReLu, and identity. The number of nodes in the hidden layer was 500 and the number of iterations was 10000.

The data were divided into three data types before going into the classification process, which were training data, test data, and validation data. The distribution of the data used cross validation (cv) with the *stratified K-fold* method, and the number of folds given was 7. The program was configured to automatically find the best accuracy value from the classification method and the provided parameters. The classification results can be seen in Table 4. The resulting accuracy value was 0.920 with 9 sensors and 14 features. This is a very satisfying result for three classes. The highest accuracy from the result of the SVM classification method was produced with C = 10 and gamma (γ) = 0.1. Although the result was satisfying, several sensors in the system were still not utilized efficiently.

This study also compared another feature selection method, i.e. *chi square*. The threshold resulted from the proposed method is preferable compared with the results of the *chi square* method using the same classification method and the same treatment parameters. The results of *chi square* accuracy with RF, SVM, KNN, and MLP were 0.83, 0.87, 0.85, 0.85, respectively.

Table 5. Efficiency calculation result

Model	Number of Feature and Sensor	Accuracy	Total Prices	Efficiency
1	0 sensor, 0 feature	0	0	0
2	5 sensors, 3 features	0.720	1115400	312312
3	8 sensors, 5 features	0.868	1797200	237230
4	6 sensors, 9 features	0.884	3382300	392346
5	6 sensors, 10 features	0.892	3382300	365288
6	7 sensors, 12 features	0.896	3482300	362159
7	9 sensors, 14 features	0.920	3802300	304184
8	9 sensors, 15 features	0.920	3802300	304184
9	10 sensors, 18 features	0.904	4233700	406435
10	10 sensors, 19 features	0.904	4233700	406435
11	10 sensors, 20 features	0.900	4233700	406435

4.4 Optimizing efficiency by considering the significant features and the accuracy of the results

To achieve efficiency of sensor usage, Eq. (6) was used to detect the three classes, i.e. beef, chicken, and pork. Table 5 shows that the threshold initialized the model.

As for the sensor prices: Sensor TGS 822, TGS 2612, TGS 2620, TGS 832, TGS 826, TGS 2603, TGS 2600, TGS 813 cost 382,500 (Indonesian Rupiah) IDR, 431,400 IDR, 360,600 IDR, 477,000 IDR, 1,585,100 IDR, 299,300 IDR, 277,800 IDR, 220,000 IDR, and 200,000 IDR respectively [26–30]. The column in Table 5 shows the sum of the sensor prices for each model. The value of efficiency was calculated using Eq. (6); the lowest value is the best one. If the lowest value of efficiency is more than one, then the model with the lowest number of sensors will be chosen. The model with the lowest efficiency value will be chosen, which was model 3. Models 3, 7, 8, 2, 5, 6, 9, 10, and 11 were sorted based on the efficiency value, which can be used as recommendation for further research

5. Evaluation

A confusion matrix is commonly used to test the performance of a classification method. It visualizes a table of comparison between the predicted class and the actual class. In this study, the confusion matrix stores classification information for chicken, pork, and beef produced by SVM. The SVM confusion matrix in Fig. 5 shows 119 data for chicken, out of which 105 actual data were predicted as chicken, beef was predicted in 13 data. For pork, 79 out of 82 data were predicted correctly and for beef, 46 out of 50 data were predicted correctly, while the other 4 data were predicted incorrectly.

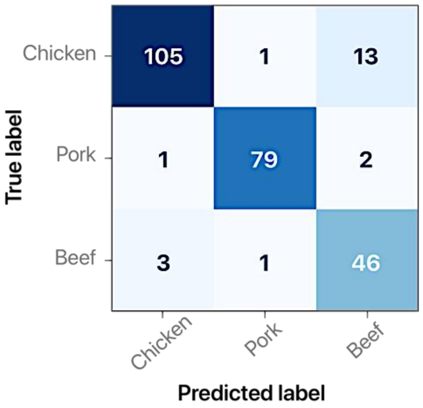


Figure. 5 Result of confusion matrix for classification meat from the market

Hence, we can obtain various indicators, such as accuracy, precision, and recall in order to indicate the performance of the classifier [31], as can be seen in Table 6. For additional analysis, the classification result of SVM is divided into true positive rate (TPR), false negative rate (FNR), true negative rate (TNR), and false positive rate (FPR), as summarized in Table 7.

Some researches did not calculate these values [6, 16, 32, 33], which could actually improve performance measurement. SVM produced an average TPR and TNR of 92% and 96%, as can be seen in Table 7. SVM achieved low FNR and FPR, i.e. 0.08% and 0.04% respectively. These results show that SVM performed well recognizing the classes of meat.

Table 6. Classifier performance result

	Class		
	Chicken	Pork	Beef
Precision	0.96	0.99	0.75
Recall	0.89	0.96	0.92
F1-Score	0.93	0.98	0.83
Accuracy	0.92		

Table 7. Performance measurement of SVM

Class	TP	FN	FN	TN	TPR	TNR
0	105	13	4	128		
1	79	3	1	167	0.92	0.96
2	46	4	15	185		

6. Conclusion

This paper presented several steps of determining thresholds to decide the number of sensors to be used in an e-nose system. An experimental e-nose was used to detect three main meat classes, namely chicken, pork, and beef. The thresholds reduced 10 sensors to 9 sensors and 20 features to 6 features, resulting in an accuracy value of 92%. Because several sensors and features are still present, this research proposed an efficiency method that includes an optimized threshold value in the cost factor. The most efficient sensors with the highest accuracy could be selected confidently. By using 5 sensors and 8 features, a good result was achieved with an accuracy value of 86.6% for the three classes.

Acknowledgments

The authors would like to thank Shidiq, Trisna, Radhea and the Ministry of Research, Technology and Higher Education of Indonesia for supporting this research.

References

[1] Hariyanto, R. Sarno, and D. R. Wijaya, “Detection of diabetes from gas analysis of human breath using e-Nose”, In: *Proc. of the 11th International Conference on Information and Communication Technology and System*, vol. 2018-Janua, pp. 241–246, 2018.

[2] H. M. Saraoglu, A. O. Selvi, M. A. Ebeoglu, and C. Tasaltin, “Electronic nose system based on quartz crystal microbalance sensor for blood glucose and hba1c levels from exhaled breath odor”, *IEEE Sens. J.*, Vol. 13, No. 11, pp. 4229–4235, 2013.

[3] K. Yan and D. Zhang, “Blood glucose prediction by breath analysis system with feature selection and model fusion”, In: *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 2014, pp. 6406–6409, 2014.

[4] R. Rusinek, M. Gancarz, M. Krekora, and A. Nawrocka, “A Novel Method for Generation of a Fingerprint Using Electronic Nose on the Example of Rapeseed Spoilage”, *J. Food Sci.*, Vol. 84, No. 1, pp. 51–58, 2019.

[5] Q. Li, X. Yu, L. Xu, and J.-M. Gao, “Novel method for the producing area identification of Zhongning Goji berries by electronic nose”, *Food Chem.*, Vol. 221, pp. 1113–1119, 2017.

[6] M. Ezhilan, N. Nesakumar, K. J. Babu, C. S. Srinandan, and J. B. B. Rayappan, “Freshness Assessment of Broccoli using Electronic Nose”, *Meas. J. Int. Meas. Confed.*, Vol. 145, pp. 735–743, 2019.

[7] H. Kim, B. Konnanath, P. Sattigeri, J. Wang, A. Mulchandani, N. Myung, M. A. Deshusses, A. Spanias, and B. Bakkaloglu, “Electronic-nose for detecting environmental pollutants: Signal processing and analog front-end design”, *Analog Integr. Circuits Signal Process.*, Vol. 70, No. 1, pp. 15–32, 2012.

[8] S. T. P. Radi, S. Ciptohadijoyo, W. S. Litananda, M. Rivai, and M. H. Purnomo, “Electronic nose based on partition column integrated with gas sensor for fruit identification and classification”, *Comput. Electron. Agric.*, Vol. 121, pp. 429–435, 2016.

[9] J. Ha, S. Kim, J. Lee, S. Lee, H. Lee, Y. Choi, H. Oh, and Y. Yoon, “Identification of Pork Adulteration in Processed Meat Products Using the Developed Mitochondrial DNA-Based Primers”, *Korean J. Food Sci. Anim. Resour.*, Vol. 37, No. 3, pp. 464–468, 2017.

[10] L. Yang, T. Wu, Y. Liu, J. Zou, Y. Huang, V. Babu, and L. Lin, “Rapid Identification of Pork Adulterated in the Beef and Mutton by Infrared Spectroscopy”, *J. Spectrosc.*, vol. 2018, pp. 1–10, 2018.

[11] S. I. Sabilla, R. Sarno, and J. Siswantoro, “Estimating Gas Concentration using Artificial Neural Network for Electronic Nose”, *Procedia Comput. Sci.*, Vol. 124, pp. 181–188, 2017.

[12] D. R. Wijaya, R. Sarno, and E. Zulaika, “Information Quality Ratio as a novel metric for mother wavelet selection”, *Chemom. Intell. Lab. Syst.*, Vol. 160, pp. 59–71, Jan. 2017.

[13] X. Zhu and X. Wu, “Class Noise vs. Attribute Noise: A Quantitative Study”, *Artif. Intell. Rev.*, Vol. 22, No. 3, pp. 177–210, Nov. 2004.

[14] D. Kang and D. H. Lee, “Energy and environment efficiency of industry and its productivity effect”, *J. Clean. Prod.*, Vol. 135, pp. 184–193, 2016.

[15] J. González-Buesa and M. L. Salvador, “An Arduino-based low cost device for the measurement of the respiration rates of fruits and vegetables”, *Comput. Electron. Agric.*, Vol. 162, pp. 14–20, 2019.

[16] H. G. J. Voss, S. L. Stevan, and R. A. Ayub, “Peach growth cycle monitoring using an electronic nose”, *Comput. Electron. Agric.*, Vol. 163, p. 104858, 2019.

[17] D. R. Wijaya, R. Sarno, E. Zulaika, and S. I. Sabila, “ScienceDirect Development of mobile electronic nose for beef quality monitoring”, *Procedia Comput. Sci.*, Vol. 124211, pp. 728–735, 2017.

[18] M. T. Puth, M. Neuhäuser, and G. D. Ruxton, “Effective use of Pearson’s product-moment correlation coefficient”, *Animal Behaviour,* Vol. 93. Academic Press, pp. 183–189, 2014.

[19] H. Zhou, Z. Deng, Y. Xia, and M. Fu, “A new sampling method in particle filter based on Pearson correlation coefficient”, *Neurocomputing*, Vol. 216, pp. 208–215, 2016.

[20] X. Hong, J. Wang, and G. Qi, “Comparison of semi-supervised and supervised approaches for classification of e-nose datasets: Case studies of tomato juices”, *Chemom. Intell. Lab. Syst.*, Vol. 146, pp. 457–463, 2015.

[21] S. Omatu and M. Yano, “E-nose system by using neural networks”, *Neurocomputing*, Vol. 172, pp. 394–398, 2016.

[22] K. Yan, D. Zhang, D. Wu, H. Wei, and G. Lu, “Design of a breath analysis system for diabetes screening and blood glucose level prediction”, *IEEE Trans. Biomed. Eng.*, Vol. 61, No. 11, pp. 2787-2795, 2014.

[23] T. Seesaard, C. Sriphrapradang, T. Kitiyakara, and T. Kerdcharoen, “Self-screening for diabetes by sniffing urine samples based on a hand-held electronic nose”, In: *Proc. of the 9th Biomedical Engineering International Conference*, Vol. 144, pp. 77–85, 2017.

[24] B. S. Kirar and D. K. Agrawal, “Current Research on Glaucoma Detection using Compact Variational Mode Decomposition from Fundus Images”, *International Journal of Intelligent Engineering and Systems*, Vol. 12, No. 3, pp.1-10, 2019.

[25] K. Deepika and N. Sathyanarayana, “Relief-F and Budget Tree Random Forest Based Feature Selection for Student Academic Performance Prediction”, *International Journal of Intelligent Engineering and Systems*, Vol. 12, No. 1, pp.30-39, 2019.

[26] Tashika, “Sensor TGS,” 2014. [Online]. Available: https://www.ebay.com/sch/i.html?_from=R40&_trksid=m570.l1313&_nkw=sensor+TGS&_sacat=0. [Accessed: 26-Jun-2019].

[27] “Buy esp8266 and get free shipping on AliExpress.com.” [Online]. Available: https://www.aliexpress.com/af/sensor-TGS.html?SearchText=sensor+TGS&d=y&initiative_id=SB_20190626011306&origin=n&catId=0&isViewCP=y&jump=afs. [Accessed: 26-Jun-2019].

[28] “sensor tgs - Buy Cheap sensor tgs - From Banggood|Shopping Array.” [Online]. Available: https://sea.banggood.com/search/sensor-tgs.html. [Accessed: 26-Jun-2019].

[29] “Jual Produk Sensor Tgs Murah dan Terlengkap | Bukalapak.” [Online]. Available: https://www.bukalapak.com/products?utf8=✓&source=navbar&from=omnisearch&search_source=omnisearch_organic&search%5Bhashtag%5D=&search%5Bkeywords%5D=sensor+TGS. [Accessed: 26-Jun-2019].

[30] “Jual sensor TGS | Tokopedia.” [Online]. Available: https://www.tokopedia.com/search?st=product&q=sensor tGS. [Accessed: 26-Jun-2019].

[31] Y. Potharaju, M. Kamsali, and C. R. Kesavari, “Classification of Ontological Violence Content Detection through Audio Features and Supervised Learning”, *International Journal of Intelligent Engineering and Systems*, Vol. 12, No. 3, pp.20-30, 2019.

[32] N. Rao Saggurti, J. Sankar Koti Reddy, and D. Naidu Challa, “Suppression of Non-Stationary Noises Through the Generalized Signal Detector”, *International Journal of Intelligent Engineering and Systems*, Vol. 11, No. 1, pp.131-141, 2018.

[33] S. Muzamil Basha, D. Singh Rajput, and V. Vandhan, “Impact of Gradient Ascent and Boosting Algorithm in Classification”, *International Journal of Intelligent Engineering and Systems*, Vol. 11, No. 1, pp.41-49, 2018.