Sensing and Bio-Sensing Research 36 (2022) 100492

Contents lists available at ScienceDirect

Sensing and Bio-Sensing Research

journal homepage: www.elsevier.com/locate/sbsr

Adaptive filter for detection outlier data on electronic nose signal

Doni Putra Purbawa [a], Riyanarto Sarno [a,*], Malikhah [a], M. Syauqi Hanif Ardani [a], Shoffi Izza Sabilla [a], Kelly Rossa Sungkono [a], Chastine Fatichah [a], Dwi Sunaryono [a], Indra Sampe Parimba [b], Arief Bakhtiar [b]

[a] Department of Informatics, Faculty of Intelligent Electrical and Informatics Technology, Institut Teknologi Sepuluh Nopember (ITS) Sukolilo, Surabaya 60111, Indonesia

[b] Department of Pulmonology, Faculty of Medicine, Airlangga University, Surabaya, Indonesia

ARTICLE INFO

Keywords:
Artificial olfaction
Deep neural network
Electronic nose
Outlier detection
Signal processing

ABSTRACT

Outliers are extreme cases that different from other observations. The existence of these outliers can damage data quality. The electronic nose (*E*-nose) is a tool for imitating the human nose work system that is cheap, accurate, and widely used to solve various scientific fields. Moreover, several feasibility studies have been conducted in recent years, such as detecting a mixture of beef and pork authenticity, diabetes patient detection, lung cancer, and respiratory infectious disease (SARS) detection based on the smell of armpit sweat. However, the inevitable challenge is detecting invalid data and preventing it from coming into the dataset. This paper proposes an adaptive filter using a deep neural network (DNN) and self-feature extraction to overcome the invalid data (outlier) presence from *E*-nose signal in case of SARS detection by armpit sweat odor. Our proposed method in the outlier detection task has a promising performance with 90.4% of average balanced accuracy (BA). These results indicate that the proposed DNN and self-feature extraction outperforms conventional methods such as Support Vector Machine (SVM), Naïve Bayes (NB), K-Nearest Neighbour ($k-$NN), and ensemble models like Random Forest (RF), XGBoost, also the Euclidean-z-score combination. The proposed system can be applied for real-time outlier detection on *E*-nose and improve the SARS detection performance.

1. Introduction

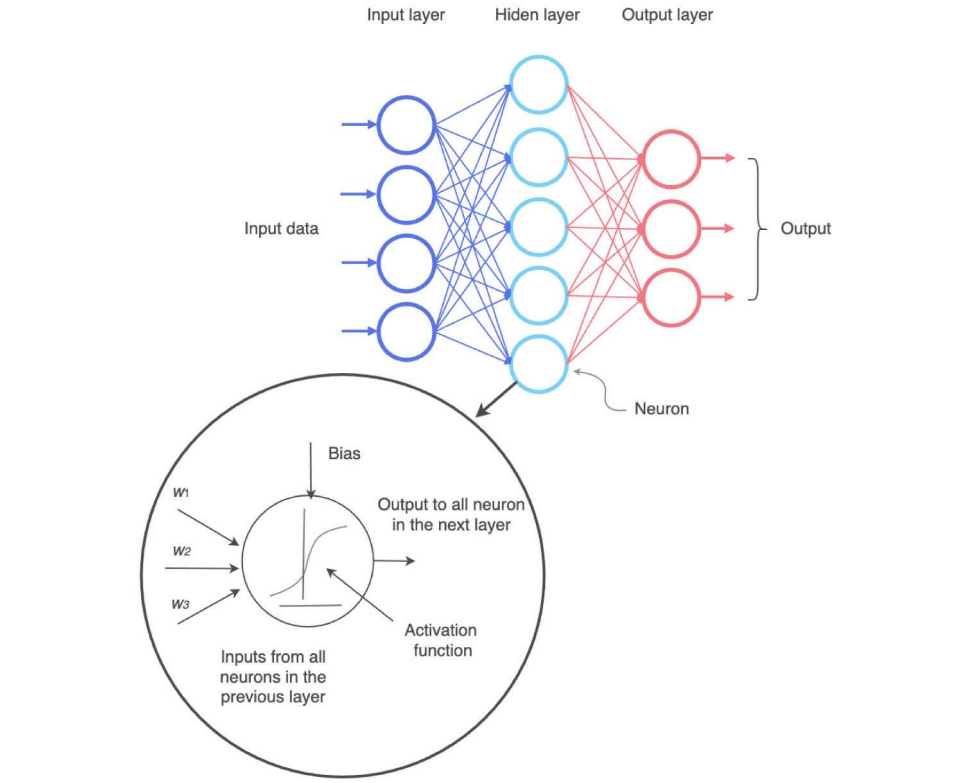
Outliers are data or cases with unique characteristics located far from the normal population. According to [1,2], outliers can damage data quality and interfere with the classification or prediction. Usually, outliers appear in extreme values for either a single variable or a combination. According to [3] there are four causes of outliers (1) data entry errors, (2) missing values in computer programs, (3) not being a member of the sample population, (4) having extreme values and not normally distributed. We must have training data from both normal and outlier data to determine outliers adaptively without static constraints. In *E*-Nose data that has more than two sensors, usually more outliers are found that arise from a combination of variables, so there is a need for a multivariate approach to detect outliers.

Electronic Nose (*E*-nose) is an artificial olfaction tool for imitating the nose work system that is cheap, easy, and accurate in implementing machine learning methods. Inside the *E*-nose component, several sensors are used to capture odors, and each has a sensitivity to a substance. Some researchers use E-nose to detect the authenticity of beef mixed with pork [4,5] and monitor food quality [6]. In addition, the application of machine learning and E-nose is currently being widely studied in the medical field, such as the detection of diabetes patients [7], lung cancer [8,9], knowing the potential for leukemia between men and women [10] and early detection of TB-infected patients [11]. However, *E*-nose has many challenges, such as outlier data; for example, in SARS detection through the smell of armpit sweat, deodorants and perfumes can interfere with the classification or prediction process. As well as detecting TB-infected patients through exhalation, the pungent smell of food, drink, or drugs will interfere with the classification process. In the *E*-nose community, the use of machine learning and even artificial neural networks has been recognized for its performance to process E-nose signals. Several previous studies, namely noise filtering using discrete wavelet transform (DWT) to monitor beef quality [12], then DWT to optimize the classification process of diabetic patients [13] and Naïve Bayes algorithm to distinguish beef from fresh pork [14].

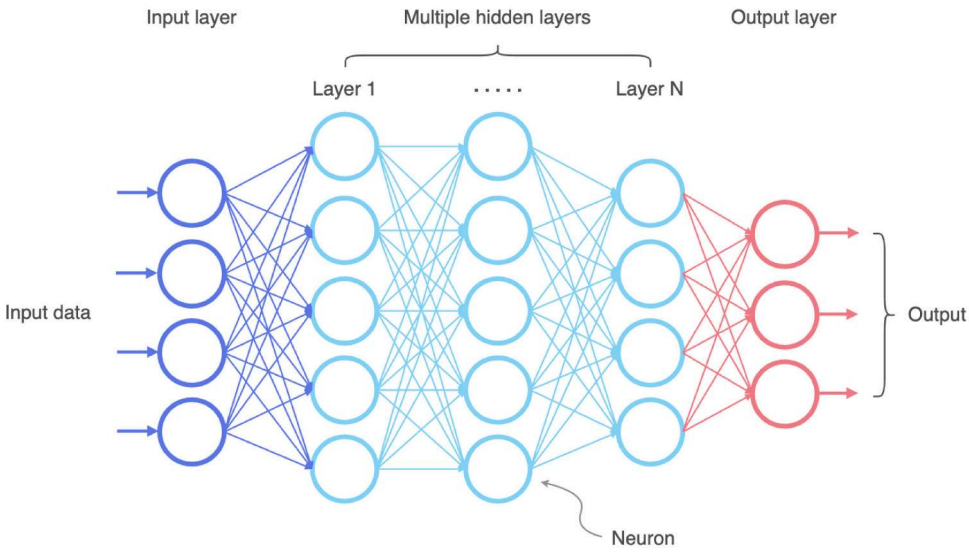
In the data sampling process, the sensor array on the *E*-nose device

* Corresponding author.
E-mail addresses: riyanarto@if.its.ac.id (R. Sarno), kelly@its.ac.id (K.R. Sungkono), chastine@if.its.ac.id (C. Fatichah), arief-b@fk.unair.ac.id (A. Bakhtiar).

https://doi.org/10.1016/j.sbsr.2022.100492
Received 18 December 2021; Received in revised form 18 March 2022; Accepted 24 March 2022
Available online 1 April 2022



(a)



(b)

Fig. 1. The comparison of the (a) Artificial Neural Network; (b) Deep Neural Network.

can produce incorrect values, or even some sensors are not working properly. This anomaly needs to be reported and analyzed in real-time using appropriate feature extraction and efficient algorithms and is the focus of this research. Several researchers have also created an outlier detection framework for E-nose signals. For example, the use of robust principal components analysis (PCA) for E-nose data validation [15], then [16] the proposed combination of data validation and imputation using partial least squares. The inherent feature of the gas sensor also causes some difficulties for smell distinction. Some of the problems are below:

(i) The prevalence of manufactured sensor collections will experience a decrease in accuracy over time (drift).
(ii) Cross-sensitivity of gas sensors is unavoidable in the sensor collections layout. This situation produces data with high correlation.

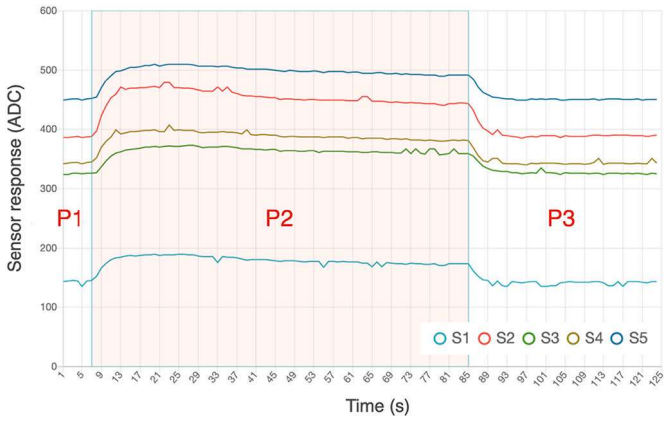


Fig. 2. *E*-nose data sample

(iii) The behavior of the sensor is instantly affected by the surrounding chemical and physical conditions. The sensor's response is highly dependent on the temperature or humidity of the gas being examined.

The multivariate responses of the gas sensor arrays underwent different pre-processing before the axillary sweat odor signal classification. Amin et al. [17], Wilson et al. [18], Pardo et al. [19], Shao et al. [20], and Yan et al. [21] discusses diverse methods of feature extraction for gas sensor data by undervaluing redundancy in the data.

Based on this problem, there are some reasons to complete this analysis as follows:

1. In the previous studies, outlier detection systems were built [22] [23] [24], and it very useful. In disparity, we were motivated to perform real-time outlier detection for hospital patients with SARS during the data collection process. In addition, we also set up the label for all SARS data with two labels, valid or invalid, and the more data that can be obtained, the better the performance of the outlier and SARS detection system will be.
2. Monitoring data quality in the *E*-nose sampling process yields time-series data. Thus, a suitable approach is required to deliver more acceptable performance than traditional techniques to solve classification problems.
3. In addition, the challenge of detecting infectious respiratory diseases using the smell of armpit sweat, namely the use of deodorants, perfumes, and the presence of scents around the subject, is very important.

Based on our knowledge, several studies use DNN for E-nose classification case, and it is widely used to detect outlier or anomaly as discussed in research [25]. However, most of the data used for anomaly detection is image, text, video, tabular data, and only a few use sequence data also none use signal or time series data. According to above motivations and some references, an outlier detection system is appropriate to be created. DNN with extraction features based on analysis of invalid signal characteristics is a prominent combination to validate that the signal data belongs to valid or invalid (outliers).

The structure of this paper is arranged as follows. Section 2 explains about DNN principle and the proposed method. Section 3 concerned the experimental setup, datasets, and results. Eventually, In Section 4, the conclusion of this study is described.

2. Methodology

This section briefly explains the principle of DNN and feature extraction. Then, our proposed system DNN and self-feature extraction details are also discussed.

2.1. The Deep Neural Network (DNN) principle

Recently, researchers have been building deep learning algorithms. The success of their research in various cases was also reported. There are several artificial neural networks, but they always have the same components: neurons, synapses, weights, biases, and activation functions as detailed in Fig. 1 (a). In Fig. 1, the white circles are the neurons that store the weight of data other neurons as an input. Then each weight will be added up using the summing function ($\sum$) and the result will be processed using the activation function. The result of the sum will be compared with a certain threshold, if it exceeds the threshold the activation will be revoked, otherwise, if it is below the threshold the neuron will be activated as a weight for the next neuron. There are three types of layers: input, hidden, and output. Neuron for the input layer mostly using the extracted feature, one feature as one neuron, and each neuron weighting to hidden layer by using some activation functions.

The hidden layer also weighs each neuron with an activations function to get the value on the output layer. There are three activation functions which mostly used in the hidden layer, such as Rectified Linear Activation (ReLU), Logistic (Sigmoid), and Hyperbolic Tangent (Tanh). ReLU returned all negative values as 0, the sigmoid function allows the value within the interval 0 to 1, and the tanh is the same as sigmoid but within -1 and 1. The output layer also needs an activation function, which usually uses linear, sigmoid or softmax for classification or regression cases.

DNN is an artificial neural network-based algorithm that adapts the human brain for decision-making [26]. DNN is composed of a neural network known as neutrons. Based on the name, a deep neural network means an artificial neural network with many hidden layers between the input and output layers. The most striking difference between ANN and DNN lies in the number of hidden layers. The ANN architecture only has one hidden layer as shown in Fig. 1 (a), while the DNN has more than one to n-hidden layers that are interconnected as in Fig. 1 (b).

2.2. Self-feature extraction

The feature extraction process takes the characteristics of data representing important information to be analyzed or used in making a classification model. Time series signal data usually uses statistical parameters to get the features of each signal [27]. In this paper, the feature extraction process is carried out based on observations from the *E*-nose signal data. Adaptive filters are digital filters that can adjust their coefficients to give the best match to the desired signal. When an adaptive filter operates in a changing environment, the filter coefficient can adapt in response to changes in the applied input signal. The signal in the E-nose data has many variations, this is influenced by many factors such as the use of deodorant, data collection errors, and damage to hardware or sensors. This causes noise such as ripple and peak which can interfere with the extraction process. Fast Fourier Transform (FFT) is used in the pre-processing stage which adaptively removes noise or peaks and performs low pass filtering before the feature extraction process. FFT converts the signal in the time domain into the frequency domain using Eq. 1, and returns from the frequency domain to the time domain using Eq. 2.

$$X(f) = \int_{\infty}^{\infty} X(t) e^{-jwt} dt \tag{1}$$

$$X(t) = \frac{1}{N} \int_{\infty}^{\infty} X(f) e^{-jwt} dt \tag{2}$$

where $X(t)$ is the signal in the time domain, e^{-jwt} is the kernel function, $X(f)$ is the frequency domain function, f is the frequency, t is the time and N is the total sample of the signal.

The data generated from the *E*-nose sensor is time-series data

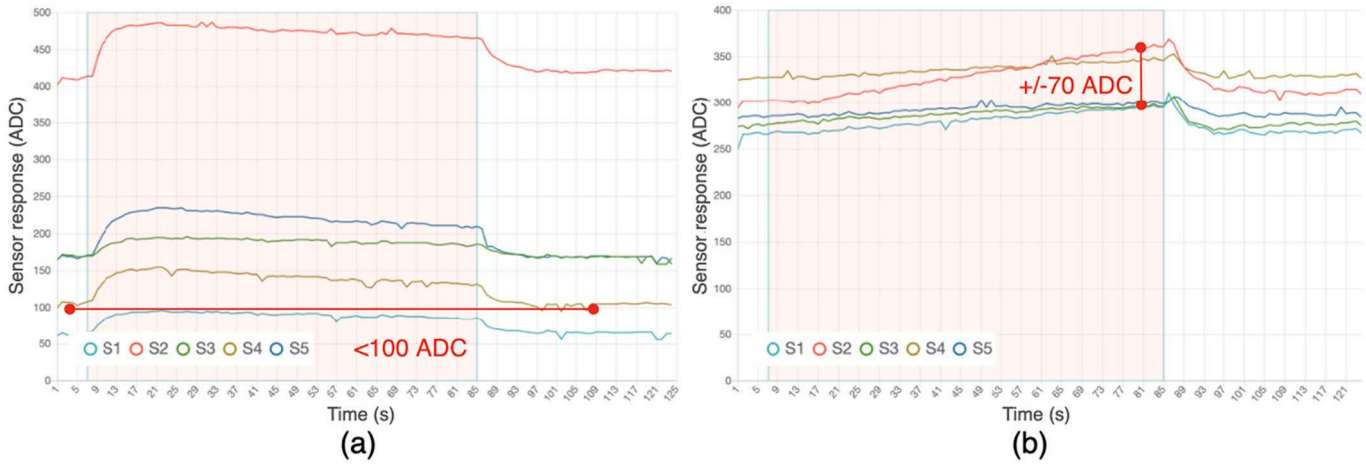


Fig. 3. Example Signals (a) Below 100 ADC, and (b) Extreme Increment.

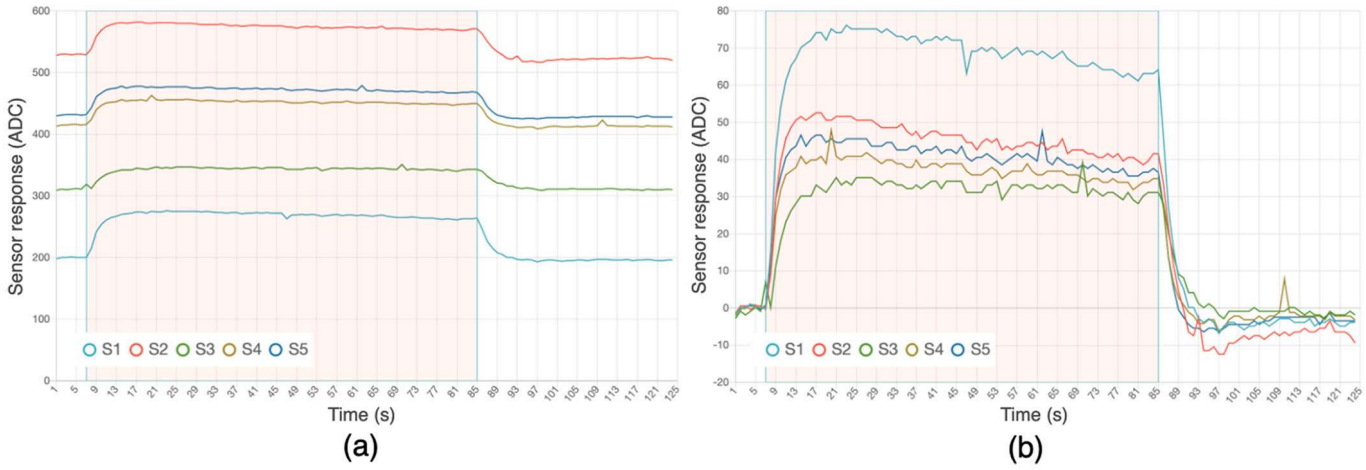


Fig. 4. Example of (a) Original Signal, and (b) After Normalization

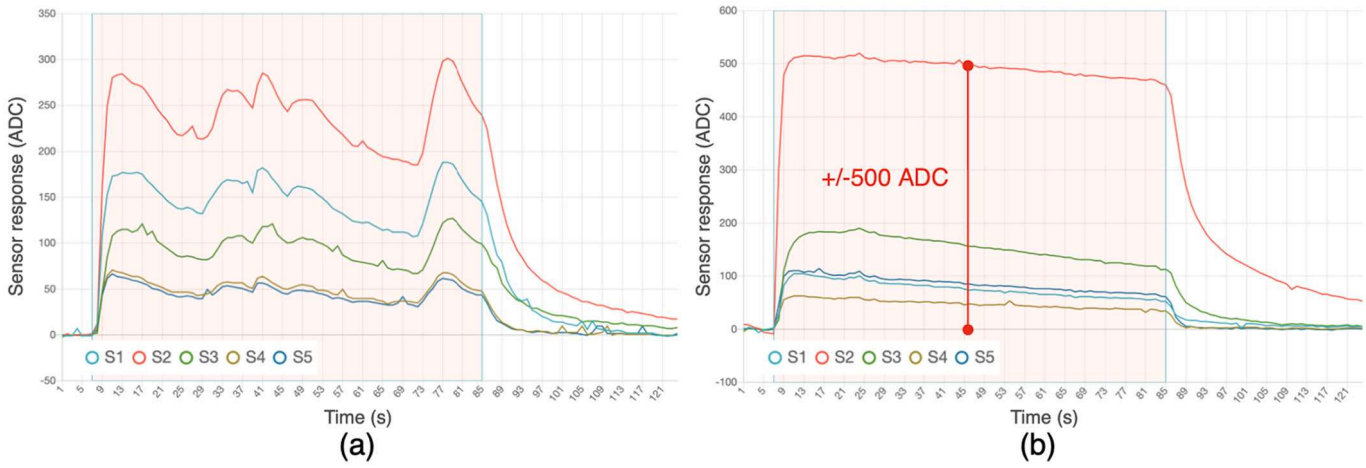


Fig. 5. Sample of (a) Irregular Signal, and (b) Deo Signal After Normalization.

consisting of 1900 lines where each line represents the sensor value per one millisecond. The data collection process consists of three stages, Flushing (P1), Sampling (P2), and Purging (P3), whose ranges are as shown in Fig. 2. The x-axis in Fig. 2 is a measurement of the time required during the patient data collection process in seconds, while the y-axis is the response value of the sensor in ADC units and S1 to S5 are the sensors used in this study.

Based on the experiments and specifications of each sensor, there are five categories of data that are said to be invalid, as follows:

2.2.1. Two-digit signal

The datasheet of each sensor states that when the sensor value is less than 100 ADC or 500 mV (mV), then the sensor has an anomaly or can be said to be invalid, as shown in Fig. 3a. So, in this case, the minimum value feature of each sensor is required.

2.2.2. Extreme increment or decrement in signal

The increment value of each signal is usually at the beginning of entering P2, and in the P2 process, it has a stable value at a certain value. However, if there is a drastic increase or decrease, it is categorized as a sensor anomaly, as in Fig. 3b. The difference between each sensor's maximum and minimum values or amplitude is needed to overcome this problem.

$$X_{diff} = max(X_{P2}) - min(X_{P2}) \tag{3}$$

2.2.3. Sensor value after normalization

Almost the same as the first category, however, due to the variety of *E*-nose devices or more than one device, allowing for the inconsistency of initial values and the increment value in every sensor in all devices. So, this study normalized the signal to equalize the value of the same sensor in all devices using Eq. 4. The difference can be seen in Fig. 4.

$$X_{norm} = X_{P2} - mean\left(\sum X_{P1}\right) \tag{4}$$

where X is the value of each sensor obtained from the *E*-nose sampling process, X_{P2} is the value of the P2 phase as in Fig. 2, where each value is reduced by the average value of all data in P1 or X_{P1}.

2.2.4. Irregular signal in P2

The process of collecting data on the patient's armpit sweat is strongly influenced by various conditions in the field, such as the position of the hose being too compressed or in the sampling process, the patient moves too much. So, the data generated from E-nose is very irregular and cannot be used for making prediction models. This study performs signal filtering using Fast Fourier Transform (FFT) to avoid peak values that are too high, then calculates the slope value and takes the maximum and minimum values from each sensor to overcome this issue. The details of the algorithm are written in Algorithm 1, and the sample of signal is on Fig. 5a.

2.2.5. Deo/Parfum contaminated armpit sweat

The most urgent thing to be investigated further is when the smell of the patient's sweat is mixed with deodorant or perfume. If the data is not filtered, it will interfere with making prediction models for SARS disease. The content of deodorants and perfumes varies, but they usually contain alcohol, ethanol, isobutane and others [28,29]. This study uses five sensors, some of which are sensitive to alcohol and some ingredients in deodorant or perfume, based on their datasheets [30,31]. We calculate the stable sensor value from the slope value to know the average of each sensor response and use their max sensor value to overcome this issue. Before stable value averaged, we divided the slope into six regions to avoid some parts getting peaked and some getting deep valley. The detail is on Algorithm 2 and the sample of deodorant signal is on Fig. 5b.

Algorithm 1 : sensorSlope

Input : Sensor raw signal

Output : Slope max and min value

```
P1 ← load(data_per_patient, "P1")[2:-2] (Avoid start and end tick)
P2 ← load(data_per_patient, "P2")[15:] (Remove the transient P1 to P2)
P1_mean ← mean(P1)

NormalizedP2 ← P2 – P1_mean
slopeOutput ← []
for sensor ← S1 to sensorList do
    P2 ← NormalizedP2[sensor]
    P2_fft ← fft(P2)
    slopes ← []
    for count, tick ← 1 to enumerate(P2_fft) do
        if (count = 0) then
            continue
        else
            slope = (tick - P2_fft[count-1])/(count-(count-1))
            slopes.add(slope)
        end if
    end for
    slopeOutput.add(max(slopes))
    slopeOutput.add(min(slopes))
end for
return slopeOutput
```

Algorithm 2 : deoSensorValue

Input : S1 and S2 raw signal

Output : S1 and S2 stable and max value

```
MAX_SLOPE ← 1
MIN_SLOPE ← -1
DELTA_SLOPE_MAX ← 0.1
region ← 6
slopeValue ← Algorithm1(S1, S2)
stableValue ← []
for sensor ← S1,S2 do
    slope ← sloveValue[sensor]
    slopeDivided ← divideIntoRegion(slope, region)
    stables ← []
    for reg, data ← 1 to enumerate(slopeDivided) do
        (original Index, slopes) ← data
        maxVal ← max(slopes)
        minVal ← min(slopes)
        if maxVal <= MAX_SLOPE and minVal >= MIN_SLOPE then
            for index, plotSlope ← 1 to enumerate(slopes) do
                if (abs(plotSlope) < DELTA_SLOPE_MAX) then
                    stables.add(slope[originalIndex[index]])
                end if
            end for
        end if
    end for
    if (len(stables) = 0) then
        stableValue.add(0)
    else
        stableValue.add(avg(stables))
    end if
end for
stableValue.add(max(S1))
stableValue.add(max(S2))
return stableValue
```

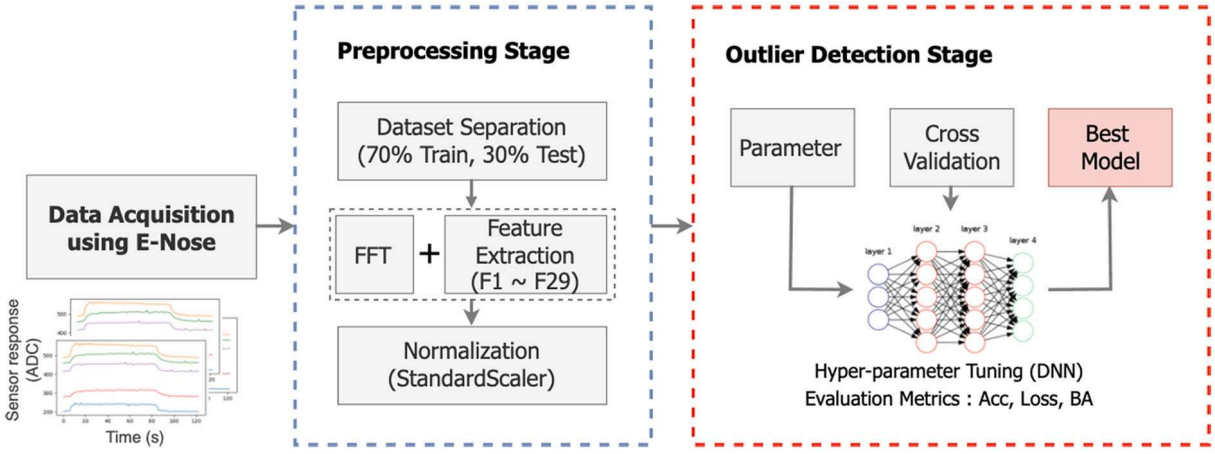


Fig. 6. Proposed outlier detection system.

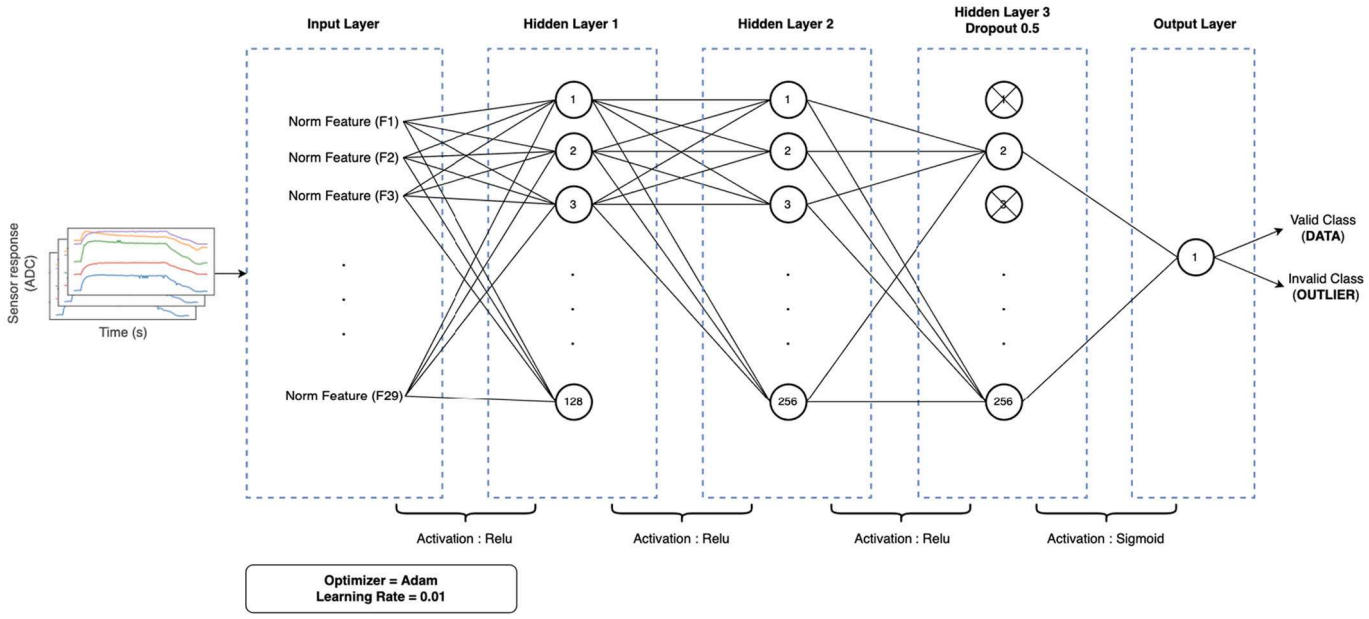


Fig. 7. Proposed DNN Network.

2.3. Proposed system using DNN

In this section, a detailed description of the DNN is discussed. As previously mentioned, the performance of the SARS disease detection model is highly influenced by data outliers from the data collection process. Therefore an outlier detection system as in Fig. 6 is proposed. There are three parts of the system: data sampling, pre-processing, and outlier detection. In the pre-processing stage, the datasets are divided into 70% train data and 30% test data which will be explained in more detail in the next section. After that, the feature extraction process is carried out based on the five categories represented in the previous section. Due to different sensor values and different extraction results, normalization is performed after feature extraction using a standard scaler (SC) as in Eq. 5.

$$x_{scaled} = \frac{(x-u)}{s} \tag{5}$$

where s is the standard deviation of the training samples and u is the average of the training samples. Centring and scaling occur independently of each feature by calculating the applicable statistics in the training sample.

The next process is to develop a DNN model for outlier detection based on the features obtained. We used hyperparameter tuning to determine the number of neurons, the required dropout value, the appropriate optimizer, learning rate value, and model evaluation metrics. Dropout is a process of preventing overfitting and also accelerating the learning process. Dropout focuses on eliminating neurons in the form of hidden and visible layers in the network. The neurons are removed randomly, and each neuron will be assigned a probability that is between 0 and 1 [32].

In deep learning or neural networks, the gradient descent optimization process is carried out by optimizing the derivative function itself [33,34]. The change in gradient descent value can be calculated from Eqs. 6 and 7.

$$z = f(w)\eta \tag{6}$$

$$w = w - \eta^* \frac{dz}{dw} \tag{7}$$

where w is the coordinates, z is the height of the value (the function we will optimize) and η is a measure of the learning rate. The purpose of the training network process is to find the value of w so that the value of z is minimal.

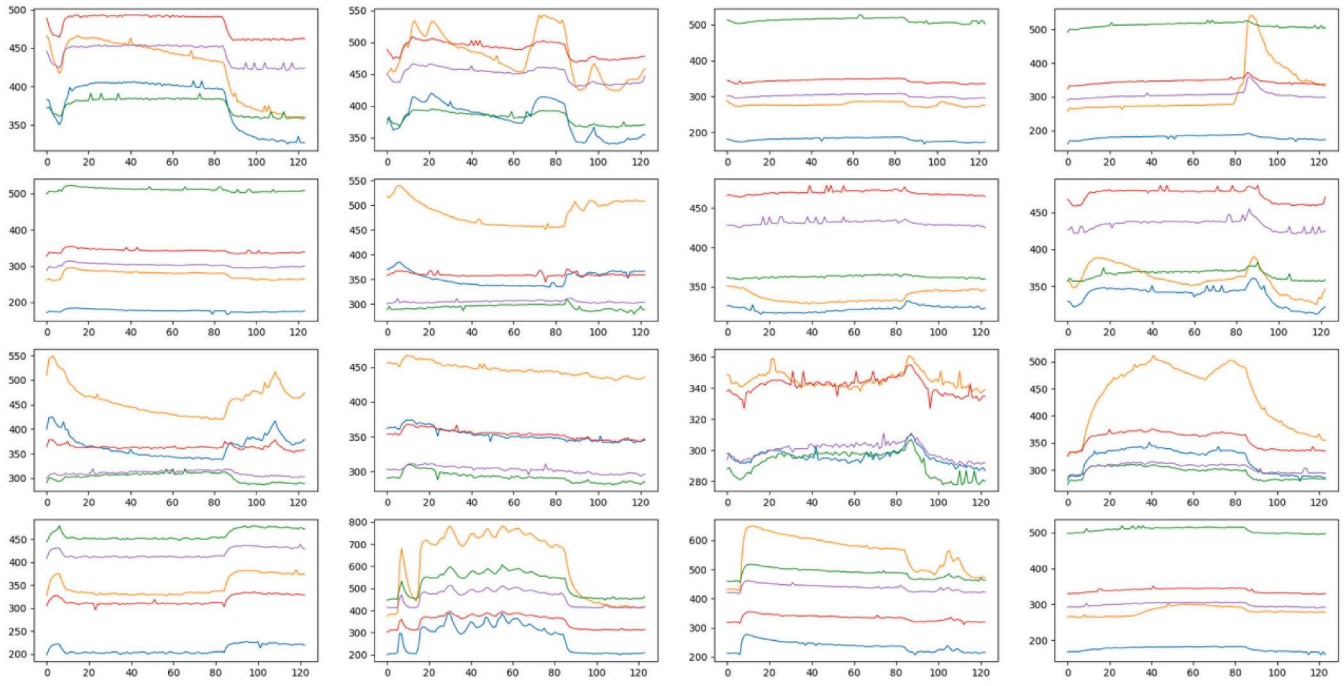
One of the factors that influence the achievement of the optimal solution for gradient descent is the size of the learning rate. A large learning rate will make large changes to variables and vice versa. Based on [34], using a learning rate that is too large will cause large changes, and the optimal solution will not be achieved. However, a learning rate that is too small will also cause the process of reaching the optimal solution to be very slow. Therefore, in this study, we performed hyperparameter tuning to compare the performance of the DNN model. As in Fig. 7, the resulting best model has one input layer with 29 neurons, hidden layer 1 with 128 neurons, hidden layer 2 with 256 neurons and hidden layer 3 with 256 neurons and 0.1 is hidden using dropout. Based on the three optimizers tuning in, Adam performed better than SGD and RMSprop. Then the learning rate with an interval of 0.0001 to 0.1 was found to be 0.01, resulting in the optimal value. Next, the activation function used in the hidden layer is reLu to change the minus value to 0. The sigmoid function in the output layer produces 1 as invalid data (outlier) and 0 as valid data (normal).

Table 1
E-nose gas sensor package.

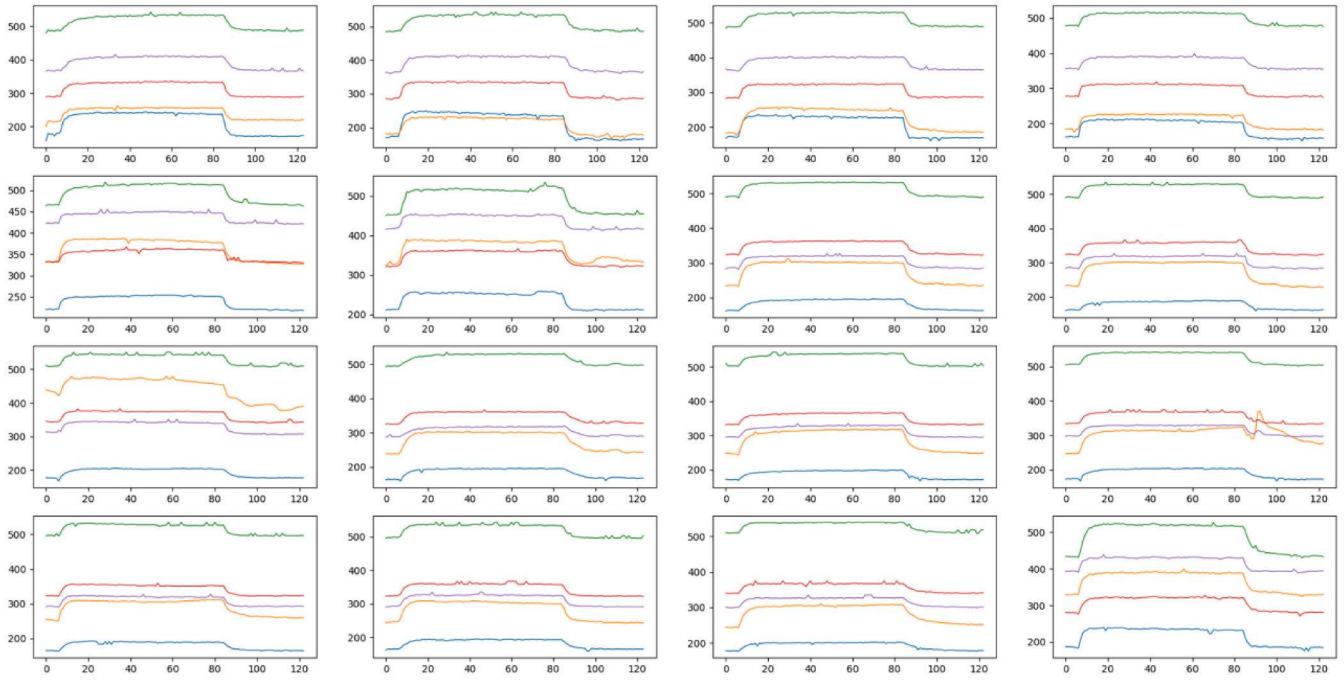
Sensor	Volatile organic compounds
S1	H_2, LPG, CH_4, CO, alcohol, smoke, propane, air
S2	Alcohol, benzenic, CH_4, hexane, LPG, CO, air
S3	LPG, CH_4, H_2, CO, alcohol, smoke, air
S4	Air, CO, C_2H_6O, NH_3
S5	Benzene, n-hexane, CH_4, CO, alcohol, propane, air

3. Experiments

This part explains the experimental setup in detail, including materials, datasets, performance measurements, and comparisons. The research outcomes discussed include classifying valid and invalid signals (outliers) of the proposed model and comparing several machine learning models.



(a)



(b)

Fig. 8. The example of invalid or outlier signals (a) and valid signals (b).

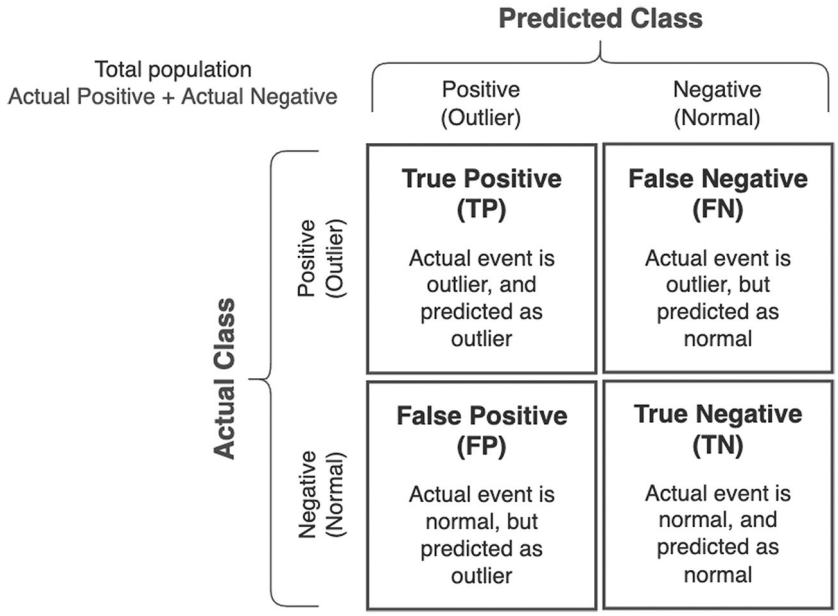


Fig. 9. Confusion matrix framework.

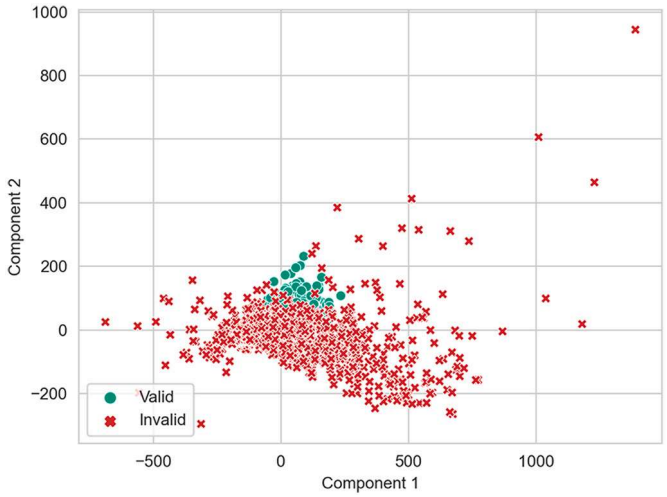


Fig. 10. Data distribution using two principal components.

3.1. Dataset and experimental setup

The electronic nose tools was created by previous research to detect SARS through the smell of armpit sweat. *E*-nose data is used to create an outlier detection model whose best model will be used for real-time data validation. This *E*-nose device consists of a mini-computer, gas sensor array, NFC and Wi-Fi connectivity for data synchronization to the cloud. The metal-oxide-semiconductor (MOS) gas sensor is used in the Raspberry mini-computer. This study used a gas sensor from Zhengzhou Winsen company. Table 1 characterizes the MOS gas sensor utilized in the E-nose device and its sensitivity to volatile organic compounds (VOC). MOS gas sensors are used because of their low cost, good sensitivity, and quick response [35]. It has been successfully utilized in various studies such as diabetic patients detection [13], detection of tuberculosis [11], monitoring of meat quality [12,36], and detection of halal food [14,37].

Furthermore, the raw signal, which is invalid and valid, are shown in Fig. 8. It can be seen that the invalid signal has very different characteristics from the valid signal. Each colour indicates a different response from the gas sensor in the sensor array, represented by resistance (ohms). Then the x-axis indicates the sampling time in seconds.

Table 2
The comparison of feature extraction.

Feature extraction		Training			Testing		
	Total features	Acc	Loss	BA	Acc	Loss	BA
Mean	5	0.865	0.324	0.850	0.836	0.405	0.820
Mean (+ SC)	5	0.873	0.303	0.859	0.857	0.342	0.843
Mean + std. + max + min	20	0.945	0.146	0.942	0.890	0.301	0.887
Mean + std. + max + min (+ SC)	20	0.957	0.111	0.954	0.900	0.264	0.895
Percentile 25 + 50 + 75	15	0.917	0.205	0.915	0.877	0.332	0.876
Percentile 25 + 50 + 75 (+ SC)	15	0.927	0.178	0.923	0.893	0.268	0.889
Kurtosis + skewness	10	0.793	0.439	0.784	0.572	0.848	0.554
Kurtosis + skewness (+ SC)	10	0.848	0.362	0.842	0.567	0.928	0.548
Peak-to-peak	5	0.758	0.489	0.727	0.722	0.578	0.690
Peak-to-peak (+ SC)	5	0.758	0.493	0.723	0.729	0.568	0.695
Proposed	29	0.953	0.126	0.951	0.880	0.445	0.877
Proposed (+ SC)	29	**0.973**	**0.076**	**0.972**	**0.905**	**0.286**	**0.904**

Table 3
The comparison of classification model.

Model	Optimal parameter	Training			Testing		
		Acc	Loss	BA	Acc	Loss	BA
SVM	(kernel = 'rbf', C = 10, gamma = 1)	0.953	–	0.954	0.859	–	0.866
Naïve Bayes	(var_smoothing = 0.005)	0.788	–	0.758	0.807	–	0.776
K-NN	(n_neighbors = 5)	0.909	–	0.898	0.874	–	0.861
Random Forest	(n_estimators = 10, criterion = "gini")	0.959	–	0.959	0.895	–	0.897
XGBoost	(n_estimators = 1000, gamma = 4.2, max_depth = 12)	0.953	–	0.951	**0.905**	–	0.901
ED + *Z*-Score (th ±1)	–	0.659	–	0.590	0.663	–	0.595
ED + Z-Score (th ±2)	–	0.623	–	0.547	0.636	–	0.556
ED + Z-Score (th ±3)	–	0.609	–	0.530	0.615	–	0.531
Proposed DNN	–	**0.973**	**0.076**	**0.972**	**0.905**	**0.286**	**0.904**

Moreover, each gas sensor produces different magnitudes. Hence, proper feature extraction and standard scaler are needed to improve the outlier detection model.

In this experiment, sampling was carried out using 16 different *E*-nose devices. The sampling process runs for 190 s, with the composition of P1 10 s, P2 120 s, and P3 60 s. Total data is 5171 consisting of 3030 valid, and 2141 invalid are used as a dataset. The composition is 70% for training and 30% for testing. During the modelling process 70% of the training data is used to train the model, and the rest is used as validation data (30%). This study also compared the proposed algorithm with some conventional algorithms like SVM, k-NN, Naïve Bayes, then ensemble model like Random Forest, XGBoost and the combination of Euclidean Distance with z-score from the previous study [38].

In addition, experiments were carried out using a personal computer with Intel(R) Core i7 CPU @ 2.70 GHz Quad-Core (4CPUs), 16 GB RAM 2133 MHz, and macOS Catalina Version 10.15.7 operating system. K-NN, SVM, Naïve Bayes and Random Forest were computed using Python 3.9.7 and the scikit-learn library. XGBoost was computed using xgboost library. Meanwhile, the proposed DNN is implemented using the TensorFlow 2.5.0 library. For the performance measurement, and based on these references [39,40] related comparison of performance metric for the imbalanced dataset, balanced accuracy (BA) is used in this research. This is because BA has a smaller value than other classification performance metrics like accuracy, F1-score, Macro Average Recall (MAR), Macro Average Precision (MAP) and it also has a smaller bias. So that BA is reasonably used to determine the best model for the imbalanced dataset in this research.

Balanced accuracy is calculated by using the formula as follows:

$$precision = \frac{TP}{(TP + FP)} \tag{8}$$

$$recall = \frac{TP}{(TP + FN)} \tag{9}$$

$$BA = \frac{precision + recall}{2} \tag{10}$$

Based on the confusion matrix framework as Fig. 9, True Positive (TP) is outlier data that is correctly predicted by the model as an outlier, while True Negative (TN) is normal data that is correctly predicted as

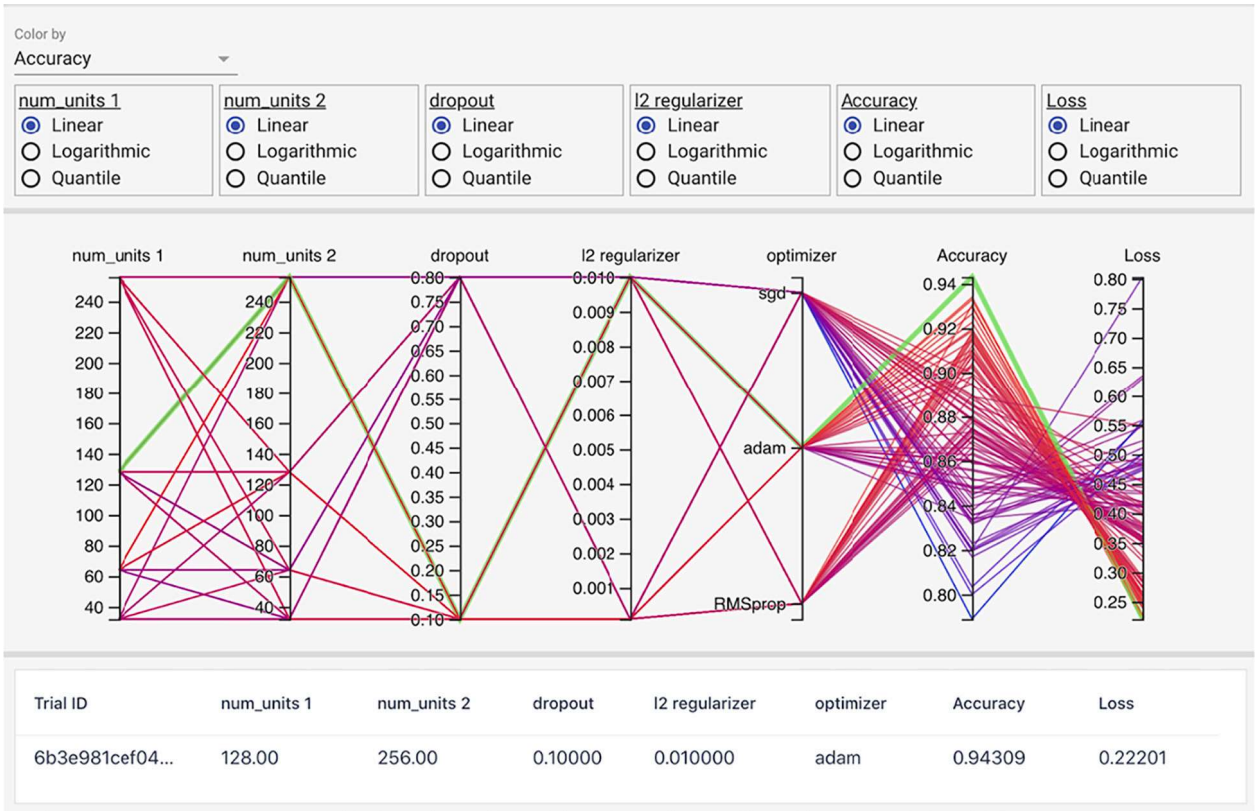


Fig. 11. Visualization of hyper-parameter tuning.

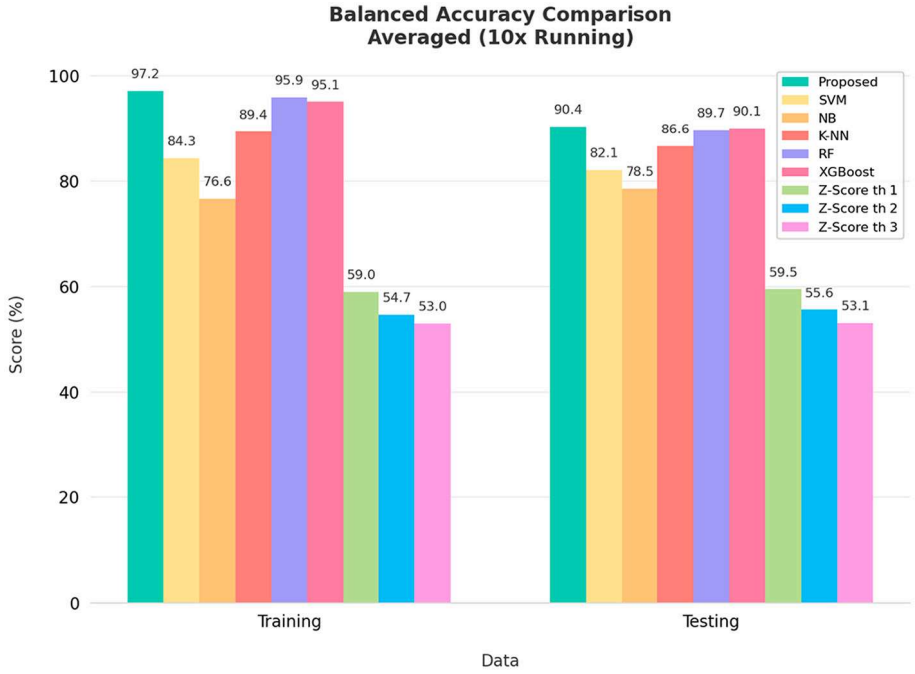


Fig. 12. Proposed DNN and other methods BA comparison.

normal data. Next, in the upper right corner of Fig. 9, it is False Negative (FN) where the original data is an outlier but incorrectly predicted as normal data. Meanwhile, False Positive (FP) is normal data, but wrongly predicted as outlier data. Precision is a comparison of TP data with the number of data that is predicted to be positive, while recall is a comparison of TP data with the total data that is actually positive.

3.2. Results and discussion

In this part, the experiment results, including the self-feature extraction, outlier detection, and comparison with several machine learning methods are discussed.

3.2.1. Outlier detection system

Some conventional machine learning models like K-NN, SVM, Naïve Bayes, also ensemble models such as Random Forest, XGBoost and Euclidean combined with z-score are performed to prove the proposed method performance by comparing them. Various combinations of feature extraction and calculating accuracy using the vanilla DNN model. Fig. 10 shows the valid and invalid data distribution using principal component analysis (PCA). It can be seen that the data looks very overlapping, so the self-feature extraction is proposed. Based on the comparison results in Table 2, the proposed self-feature extraction has a better performance in determining valid and invalid data. The use of a standard scaler is also considered effective to improve performance. In the next section, the best feature combinations used to compare the performance of the proposed method with various classification methods. The results of the model comparison are described in Table 3.

This study uses the Adam optimizer to optimize the stochastic gradient based on the hyperparameter tuning results with the highest accuracy. We used three hidden layers, but the number of neurons, dropout probability, optimizer, learning rate and metrics evaluation model are tuned using hyper-parameters to get the best performance.

Fig. 11 results from hyper-parameter tuning, which shows each combination and the accuracy and loss training model with epoch 50. Also, Fig. 12 is the result of a comparison of classification using several machine learning methods and a z-score with the static threshold.

Based on the results of the hyper-parameter tuning that has been done, the best results are obtained using the XGBoost ensemble learning. The optimal parameters used are 1000 estimators, gamma 4.2, and a maximum depth of 12 levels getting an accuracy of 0.905 and BA of 0.901. Based on the results of this experiment, the proposed model still has a better BA, which is 0.904.

4. Conclusions

E-nose with the proposed DNN and self-feature extraction has successfully distinguished valid and invalid data (outliers). There are a total of 5171 data as datasets that are used to make outlier detection models and compare the feature extraction process. DNN was proposed to overcome the outlier detection from an E-nose signal. Several machine learning algorithms such as k-NN, Naïve Bayes, SVM, and ensemble learning like Random Forest, XGBoost, also z-score statistical formulas are implemented in classification tasks. The comparison results show that the proposed DNN and self-feature extraction get the highest average balanced accuracy, which is more stable than other conventional methods. Based on the experimental results, conventional machine learning model, and ensemble algorithm, also the statistical formula z-score are susceptible to being unable to predict outliers from E-nose devices thoroughly. Hence, the proposed outlier detection system can monitor E-nose data which is used to predict SARS disease through armpit sweat and improve its performance.

In future work, we will retrofit signals detected as invalid or outliers so that the data is not discarded but is still used to develop a prediction model. Moreover, the improvement of the DNN network is needed to deal with imbalanced data.

Declaration of Competing Interest

None.

Acknowledgement

This research was funded by the Indonesian Ministry of Education and Culture under Penelitian Terapan Unggulan Perguruan Tinggi (PTUPT) Program, and Institut Teknologi Sepuluh Nopember (ITS)

under project scheme of the Publication Writing and IPR Incentive Program (PPHKI).

References

[1] H. Pham (Ed.), Springer Handbook of Engineering Statistics. London, Springer, London, 2006.

[2] T. Singhal, A Review of Coronavirus Disease-2019 (COVID-19), Indian J. Pediatr. vol. 87 (4) (2020) 281–286. Springer Apr. 01, https://doi.org/10.1007/s12098-020-03263-6.

[3] I. Ghozali, Aplikasi Analisis Multivariete SPSS 23, 2016.

[4] B. Kuswandi, A.A. Gani, M. Ahmad, Immuno strip test for detection of pork adulteration in cooked meatballs, Food Biosci. 19 (Sep. 2017) 1–6, https://doi.org/10.1016/j.fbio.2017.05.001.

[5] L. Yang, et al., Rapid identification of pork adulterated in the beef and mutton by infrared spectroscopy, J. Spectrosc. 2018 (2018), https://doi.org/10.1155/2018/2413874.

[6] S.I. Sabilla, R. Sarno, J. Siswantoro, Estimating gas concentration using artificial neural network for electronic nose, Procedia Comp. Sci. 124 (Jan. 2017) 181–188, https://doi.org/10.1016/j.procs.2017.12.145.

[7] G. Swapna, R. Vinayakumar, K.P. Soman, Diabetes detection using deep learning algorithms, ICT Express 4 (4) (Dec. 2018) 243–246, https://doi.org/10.1016/j.icte.2018.10.005.

[8] J.A. Cruz, D.S. Wishart, Applications of Machine Learning in Cancer Prediction and Prognosis, 2006.

[9] D.M. Wong, et al., Development of a breath detection method based E-nose system for lung cancer identification, in: *Proceedings of 4th IEEE International Conference on Applied System Innovation* 2018, ICASI, Jun. 2018, pp. 1119–1120, https://doi.org/10.1109/ICASI.2018.8394477.

[10] I.A. Sabilla, D.P. Purbawa, R. Sarno, A. Al Fauzi, D.R. Wijaya, R. Gunawan, Men and women classification at night through the armpit sweat odor using electronic nose, in: 2021 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob), Apr. 2021, pp. 121–127, https://doi.org/10.1109/APWiMob51111.2021.9435273.

[11] A.M.I. Saktiawati, et al., eNose-TB: A trial study protocol of electronic nose for tuberculosis screening in Indonesia, PLoS One vol. 16 (4) (Apr. 2021), https://doi.org/10.1371/journal.pone.0249689 e0249689.

[12] D.R. Wijaya, R. Sarno, E. Zulaika, Noise filtering framework for electronic nose signals: an application for beef quality monitoring, Comput. Electron. Agric. 157 (Feb. 2019) 305–321, https://doi.org/10.1016/J.COMPAG.2019.01.001.

[13] R. Hariyanto, Sarno, D.R. Wijaya, Detection of diabetes from gas analysis of human breath using e-Nose, in: Proceedings of the 11th International Conference on Information and Communication Technology and System vol. 2018, ICTS 2017, Jan. 2018, pp. 241–246, https://doi.org/10.1109/ICTS.2017.8265677. Janua.

[14] D.R. Wijaya, R. Sarno, A.F. Daiva, Electronic nose for classifying beef and pork using Naïve Bayes, in: Proceedings - 2017 International Seminar on Sensor, Instrumentation, Measurement and Metrology: Innovation for the Advancement and Competitiveness of the Nation vol. 2017, ISSIMM 2017, Nov. 2017, pp. 104–108, https://doi.org/10.1109/ISSIMM.2017.8124272. Janua.

[15] M. Mirshahi, V. Partovi Nia, L. Adjengue, An online data validation algorithm for electronic nose, Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 10163 LNCS (2016) 104–120, https://doi.org/10.1007/978-3-319-53375-9_6.

[16] M. Mirshahi, V. Partovi Nia, L. Adjengue, Automatic odor prediction for electronic nose 45, 15, Nov. 2018, pp. 2788–2799, https://doi.org/10.1080/02664763.2018.1441382.

[17] D. L. García-González and R. Aparicio, "Sensors: from biosensors to the electronic nose.".

[18] D. M. Wilson, K. Dunman, T. Roppel, and R. Kalim, Rank extraction in tin-oxide sensor arrays.

[19] M. Pardo, et al., Data preprocessing enhances the classification of different brands of Espresso coffee with an electronic nose, Sensors Actuators B Chem. 69 (3) (Oct. 2000) 397–403, https://doi.org/10.1016/S0925-4005(00)00499-8.

[20] X. Shao, H. Li, N. Wang, Q. Zhang, Comparison of different classification methods for analyzing electronic nose data to characterize sesame oils and blends, Sensors 15 (10) (Oct. 2015) 26726–26742, https://doi.org/10.3390/S151026726.

[21] J. Yan, et al., Electronic nose feature extraction methods: a review, Sensors Vol. 15 (11) (Nov. 2015) 27804–27831, https://doi.org/10.3390/S151127804.

[22] H. Alimohammadi, S. Nancy Chen, Performance evaluation of outlier detection techniques in production timeseries: a systematic review and meta-analysis, Expert Syst. Appl. 191 (Apr. 2022) 116371, https://doi.org/10.1016/J.ESWA.2021.116371.

[23] Y. Yang, C. Fan, L. Chen, H. Xiong, IPMOD: an efficient outlier detection model for high-dimensional medical data streams, Expert Syst. Appl. vol. 191 (Apr. 2022), 116212, https://doi.org/10.1016/J.ESWA.2021.116212.

[24] M. Baldomero-Naranjo, L.I. Martínez-Merino, A.M. Rodríguez-Chía, A robust SVM-based approach with feature selection and outliers detection for classification problems, Expert Syst. Appl. vol. 178 (Sep. 2021), 115017, https://doi.org/10.1016/J.ESWA.2021.115017.

[25] G. Pang, C. Shen, L. Cao, A. Van Den Hengel, Deep Learning for Anomaly Detection, ACM Comput. Surv. vol. 54 (2) (Mar. 2021), https://doi.org/10.1145/3439950.

[26] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science vol. 313 (5786) (Jul. 2006) 504–507, https://doi.org/10.1126/SCIENCE.1127647.

[27] M. Christ, N. Braun, J. Neuffer, A.W. Kempa-Liehr, Time series FeatuRe extraction on basis of scalable hypothesis tests (tsfresh – a Python package), Neurocomputing vol. 307 (Sep. 2018) 72–77, https://doi.org/10.1016/J.NEUCOM.2018.03.067.

[28] M.V. Heisterberg, et al., Deodorants are the leading cause of allergic contact dermatitis to fragrance ingredients, Contact Dermatitis vol. 64 (5) (May 2011) 258–264, https://doi.org/10.1111/J.1600-0536.2011.01889.X.

[29] A. Manayi, S. Saeidnia, Cosmetics and personal care products, Encycl. Toxicol. Third Ed. (Jan. 2014) 1043–1049, https://doi.org/10.1016/B978-0-12-386454-3.00979-9.

[30] MQ-2 Datasheet | HANWEI ELETRONICS. https://datasheetspdf.com/pdf/622943/Hanwei/MQ-2/1 (accessed Dec. 16, 2021).

[31] MQ-3 Datasheet | HANWEI ELETRONICS. https://datasheetspdf.com/datasheet/MQ-3.html (accessed Dec. 16, 2021).

[32] S.I. Sabilla, R. Sarno, K. Triyana, K. Hayashi, Deep learning in a sensor array system based on the distribution of volatile compounds from meat cuts using GC–MS analysis, Sens. Bio-Sensing Res. vol. 29 (Aug. 2020), https://doi.org/10.1016/j.sbsr.2020.100371.

[33] D. Learning, 技術者が知っておきたい Deep Learning の基礎と 組込みでの利用 ～今さら聞いてください Deep Learning ～, Nature vol. 26 (7553) (2016) 436. Accessed: Dec. 16, 2021. [Online]. Available: https://mitpress.mit.edu/books/deep-learning.

[34] F. Häse, L.M. Roch, C. Kreisbeck, A. Aspuru-Guzik, An overview of gradient descent optimization algorithms, undefined vol. 4 (9) (Sep. 2016) 1134–1145, https://doi.org/10.1021/ACSCENTSCI.8B00307.

[35] H.K. Patel, R.H. Austin, J. Barber, The electronic nose: artificial olfaction technology, Biol. Med. Physics, Biomed. Eng. (2014) 247, https://doi.org/10.1007/978-81-322-1548-6. Accessed: Dec. 17, 2021. [Online]. Available:.

[36] D.R. Wijaya, R. Sarno, E. Zulaika, DWTLSTM for electronic nose signal processing in beef quality monitoring, Sensors Actuators B Chem. 326 (Jan. 2021), 128931, https://doi.org/10.1016/J.SNB.2020.128931.

[37] R. Sarno, K. Triyana, S.I. Sabilla, D.R. Wijaya, D. Sunaryono, C. Fatichah, Detecting pork adulteration in beef for halal authentication using an optimized electronic nose system, IEEE Access (2020), https://doi.org/10.1109/ACCESS.2020.3043394.

[38] N.B. Chikodili, M.D. Abdulmalik, O.A. Abisoye, S.A. Bashir, Outlier Detection in Multivariate Time Series Data Using a Fusion of K-Medoid, Standardized Euclidean Distance and Z-Score vol. 1350, Springer International Publishing, 2021.

[39] A. Luque, A. Carrasco, A. Martín, A. De Las Heras, The impact of class imbalance in classification performance metrics based on the binary confusion matrix, Pattern Recogn. vol. 91 (Jul. 2019) 216–231, https://doi.org/10.1016/J.PATCOG.2019.02.023.

[40] E. Mortaz, Imbalance accuracy metric for model selection in multi-class imbalance classification problems, Knowledge-Based Syst. vol. 210 (Dec. 2020), https://doi.org/10.1016/J.KNOSYS.2020.106490.