Sensor Array Optimization for the Electronic Nose via Different Deep Learning Methods (Manuscript)

Xijia Zhang[#], Tao Wang[#], Wangze Ni, Wen Lv, Yongwei Zhang, Min Zeng, Jianhua Yang, Yanjie Su, Nantao Hu, Zhi Yang[*]

Key Laboratory of Thin Film and Microfabrication (Ministry of Education), Department of Micro/Nano Electronics, Institute of Marine Equipment, School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, P. R. China.

[#]These authors contributed equally to this work.

*Corresponding author, E-mail: zhiyang@sjtu.edu.cn

Abstract

A sensor array accounts for the key component of an electronic nose (E-nose). However, the practical applications of E-nose are often inhibited by its size and energy consumption arising from the numbers of sensors. How to achieve a high-performance E-nose with a minimum number of sensors is key and challenging for its practical applications. Herein, different machine learning models have been studied and compared to optimize the performance of the E-nose. The results show that Convolutional neural network (CNN) is the optimal model, which has an accuracy of 0.986 for classification, and an R-square score of 0.979 for concentration prediction, superior to Gated Recurrent Unit (GRU), Long Term Short Memory (LSTM), Multi-Layer Perceptron Neural Network (MLP), and Support Vector Machine (SVM). The performance of the E-nose is slightly changed when the number of sensors participated in pattern recognition decreases to four, where the CNN model can yield an accuracy of 0.905 for classification and an R-square of 0.972 for concentration prediction. To further quantify the loss and gain of array optimization, a *cost-effective metric* is designed to reveal the suitable array size under different scenarios. This work can provide a valuable guidance in the design of portable E-nose devices with a smaller size and optimal performances.

Keywords: Electronic Nose; Array Optimization; Convolutional Neural Network; Recurrent Neural Network; Deep Learning; Pattern Recognition

1. Introduction

An electronic nose (E-nose) is a bionic device that simulates the function of the mammalian olfactory system to distinguish gases. It replaces the olfactory nose receptors with a multi-sensor array and the bionic neural process with a pattern recognition algorithm [1-4]. As a rapid, objective, sensitive, and low-cost approach to identify the types and to predict the concentration of target gases, the electronic nose has been applied in multiple fields, including food and beverages [5, 6], medicine, health care [3, 7], industrial odor detection [8, 9], etc. With advances in materials, sensors, and machine learning [10-12], the technology of the electronic nose is under fast development.

However, the scale of the sensor array holds the electronic nose back from industry and households [2]. Including a large series of sensors for different measurement purposes, the device is not able to fit the demand of certain tasks in space. The material cost and energy consumption of the electronic nose system is also proportional to the number of gas sensors. In addition, redundant sensors may even affect the performance of pattern recognition when implementing certain tasks [13, 14]. As a result, researchers have been working on the optimization of the electronic sensor array in various applications, including but not limited to detection of the stored duration of wheat, evaluation of Chinese pecan quality, and discrimination of apple juices [13-15]. But since their works are devoted to specific scenario settings, a general pattern could not be concluded. This study aims to find a more generalized discipline between model performance and the number of active sensors.

Pattern recognition algorithm is an essential component in the electronic nose system. Machine Learning approaches such as Support Vector Machine [16], K Nearest Neighbor [17], Multi-Layer Perceptron Neural Network [18], Decision Tree [19], and Random Forest [20] have been applied in the electronic nose system and achieved considerable results. Most traditional methods tend to extract features from the data and recognize specific patterns [2, 21]. As a result, the performance of the system is closely related to the features obtained. By contrast, deep learning avoids the process of feature design and selection. The deep neural network is an End-to-End model which can automatically recognize patterns from the input based on a raw dataset [22, 23].

Convolutional neural networks (CNN) can automatically and adaptively learn spatial-like features through backward propagation. Since the introduction of Neocognitron in 1980 [11], the network has yielded considerable outcomes in object classification [24], image analysis [25], and computer vision [26]. In recent years, CNN has been applied to the electronic nose. In 2017, Qi replaces the traditional data processing methods of electronic noses with an integrated CNN network [27]; in 2019, Shi et.al developed a CNN structure that identifies beer olfactory information and achieved an accuracy of 96.67%. CNN has become one of the most competitive models in the E-nose field [27-28].

Recurrent neural network (RNN) is another class of artificial neural network with several variants. Based on David Rumelhart's work [29], the earliest recurrent network executes a back propagation along temporal sequences. However, error signals flowing back through time would either blow up or vanish in the process, leading to an exponential evolution of model error. The Long Term Short Memory (LSTM) model is therefore introduced to solve this issue [12]. In 2014, Cho et.al simplified the structure of LSTM to develop a more intuitive way of signal encoding and decoding [10], which is named the Gated Recurrent Unit (GRU). The structure has then been adapted to form several variants, including but not limited to the original Fully Gated Recurrent Unit [10], the Minimal Gated Unit [30], and the Content Adaptive Recurrent Unit [31]. LSTM has recently been applied to monitor beef quality [32], predict sensor drift [33], etc. while GRU has been used to detect lung cancer [34], to predict gas concentration [35], etc. They are both competitive models that emerge in the field.

Based on a comparison of CNN and RNN models, this study aims to explore the best way of optimizing the electronic nose sensor array. In general, optimization is mainly accomplished through optimizing multi-sensor array and pattern recognition algorithms. Optimization can be achieved either by reselecting necessary sensors or by applying the fittest algorithm. To find the algorithm which has the best performance, CNN, LSTM, and GRU together with other machine learning methods (SVM, KNN, MLP, DT, RF) based models are designed and compared. The models are trained with two tasks: type classification and concentration prediction. A classifier and a regressor are built for each model. To find the general pattern between model performance and the number of working sensors, we compare the significance of the sensors and reduce them one by one. The deep learning models are designed to handle inputs of different array sizes. The relation between the model performances and the number of sensors can be obtained in this work. As the number of sensors decreases, the change in the model performances can be observed. Therefore, this study provides a general guide for sensor array optimization in the electronic nose system.

2. Experimental Session

2.1 Data Processing

The dataset used in this study is collected by our previous work [36]. The sensors are purchased from Weisheng Electronics Co., Ltd, of which the models are MQ-2, MQ-3, MQ-4, MQ-5, MQ-6, MQ-7, MQ-8, and MQ-9, respectively. Details of the sensors are attached in Table. S1 in Supporting Information. To minimize the effect of environmental disturbance, an unmanned automatic gas-sensing system has been developed to carry out all the experiments [36]. In addition, each experiment is repeated for more than ten times. In this study, the dataset contains the gas-sensitivity response data of eight species of gases. The response and recovery curves are shown in Fig. S1 and Fig. S2 in Supporting Information. Classification labels of the gases are set to be: butyl acetate: 0, acetone: 1, ethanol: 2, methanol: 3, ethyl acetate: 4, n-propanol: 5, isopropanol: 6, and ethylene glycol: 7, respectively.

To form a data structure that can be recognized by deep neural network models, raw data is grouped by a certain length. The data is converted into sample groups that contain a partial time series of the length, regardless of the dimension in width. In other words, the length and the width constitute the size of each sample array. A self-designed data loader loads the converted sample arrays with corresponding labels. Such a way of data processing provides both CNN and RNN with suitable input structures. In a convolutional neural network, the sample array is equivalent to a monochrome graph, where individual responding values are viewed as the brightness of pixels. In a recurrent neural network, the sample array is flattened to a multiple temporal sequence. After the conversion of data, both models are estimated with 10-fold cross validation and are repeated five times for every fold.

2.2 Model Design

2.2.1 Output and Loss

Our models deal with two tasks: type classification and concentration prediction of target gases. Typically, a classification model of the deep neural network is designed to generate an output array that is of the same length as classification labels. The output array is then sorted by a SoftMax matrix, which picks up the largest value from the array and claims the corresponding label to be the result. Such a design trains the model to generate a probability value of each label between 0 and 1. The corresponding loss function is Cross Entropy Loss, which measures the overall performance of a classification model that yields arrays of probability values. In this study, CNN, LSTM, and GRU models of classification are structured to generate eight-dimensional outputs and evaluated with Cross Entropy Loss. The criterion of model performance is accuracy.

Predicting the concentration of target gases, however, should not be treated as a classification task. The dataset labels the concentration of samples as well. Label 1 to 10 suggests a gas concentration of 1 to 10 ppm respectively, while label 0 suggests the condition that the sensors are not responding. But in a real-world application, gas concentration may not be an exact value between 1 and 10 ppm. Although more experiments could be done to obtain the response information of middle values like 3.5 or 4.5 ppm, new non-integral values could come to occur. In essence, predicting gas concentration is a regression problem rather than a classification problem.

Consequently, in this study, CNN, LSTM, and GRU models of concentration are structured to yield single-value outputs. Outputs are directly evaluated by Mean Squared Error Loss. This loss function computes the mean value of squared differences between true and predicted values. As a result, the outputs of neural networks are regulated by how much they differ from true values. If a non-integral value of gas concentration is examined, the output would not be enforced to match the discrete labels. Accordingly, the criterion of model performance is R-square, root means square error, and Mean Average Error.

2.2.2 Data Length

The data length of the model determines the volume of each sample array. As has been covered in Section **2.1**, the width of the sample array equals to the number of sensors, while the length of the sample array determines how many sampling points to select each time from the original time series. The selected data forms a new multi-dimensional temporal series that is the input of neural networks. On one hand, to obtain a rapid model response, the data length should be small so that the system could gather enough data for recognition in a short time. On the other hand, to ensure the accuracy of the model response, enough information must be contained in the time series. Otherwise, the model is prone to yield false judgments. In this study, a data length of 8 is adopted. According to our previous study [21], this length yields the best output among multiples of 8. Additionally, such a length makes the CNN model easier to design in eight sensor condition. Since our data is collected every 0.5 second, a data length of 8 implies that models can respond after four seconds of data acquisition.

2.2.3 CNN Model

Typical CNN networks are a combination of convolutional layers, pooling layers, and fully connected layers [11, 22, 37, 38]. The convolutional layers perform linear operations while reserving the original features of the input. These operations are typically followed by a batch normalization operation that could mitigate internal covariate shift. Since a deep neural network requires non-linearity to develop self-learning properties, an activation function follows next. This allows the network to deal with problems that are not linearly separable. In this study, the structure of the convolutional layer, batch normalization, and the ReLU activation function are adopted and repeated.

The basic CNN architecture used in this study is shown in Fig.1(a). The fully connected layer is adaptive to different tasks: the output is either an eight-dimensional array for the classification task, or a single value for the concentration prediction task. This architecture consists of two repetitive blocks. In each block, there is a convolutional layer, a max-pooling layer, a convolutional layer, and a convolutional layer by order. To resolve dimensionality mismatches, zero paddings are added as long as the pooling layer expects an input array of odd width according to this architecture. The sequence of component layers and their corresponding dimensions are shown in Table 1. An example of the padding structure is illustrated in Fig.1(b), where the model gets data from five active sensors. These paddings increase the adaptability of the model so that it could work in different sensor conditions.

An exception occurs when the model gets data from one sensor. In this case, 2D padding is meaningless given that the data has only one dimension. Consequently, we use single dimensional max-pooling in this case. The original architecture is preserved in the length dimension, where max-pooling yields the larger value between adjacent numbers. The width dimension remains single during the process.

Table 1. Structure and dimensionality of the CNN model under n-sensor input.

Component (in order)	Mapping Relation
Convolutional Layer	$8 \bullet n \rightarrow 8 \bullet n$
Pooling layer	$8 \bullet n \rightarrow 4 \bullet [\frac{n+1}{2}]$
Convolutional Layer	$4 \bullet [\frac{n+1}{2}] \rightarrow 4 \bullet [\frac{n+1}{2}]$
Convolutional Layer	$4 \bullet [\frac{n+1}{2}] \rightarrow 4 \bullet [\frac{n+1}{2}]$
Convolutional Layer	$4 \bullet [\frac{n+1}{2}] \rightarrow 4 \bullet [\frac{n+1}{2}]$
Pooling Layer	$4 \bullet [\frac{n+1}{2}] \rightarrow 2 \bullet [\frac{n+3}{4}]$
Convolutional Layer	$2 \bullet [\frac{n+3}{4}] \rightarrow 2 \bullet [\frac{n+3}{4}]$
Convolutional Layer	$2 \bullet [\frac{n+3}{4}] \rightarrow 2 \bullet [\frac{n+3}{4}]$
Fully Connected Layer	$2 \bullet [\frac{n+3}{4}] \rightarrow 8 \ or \ 1$

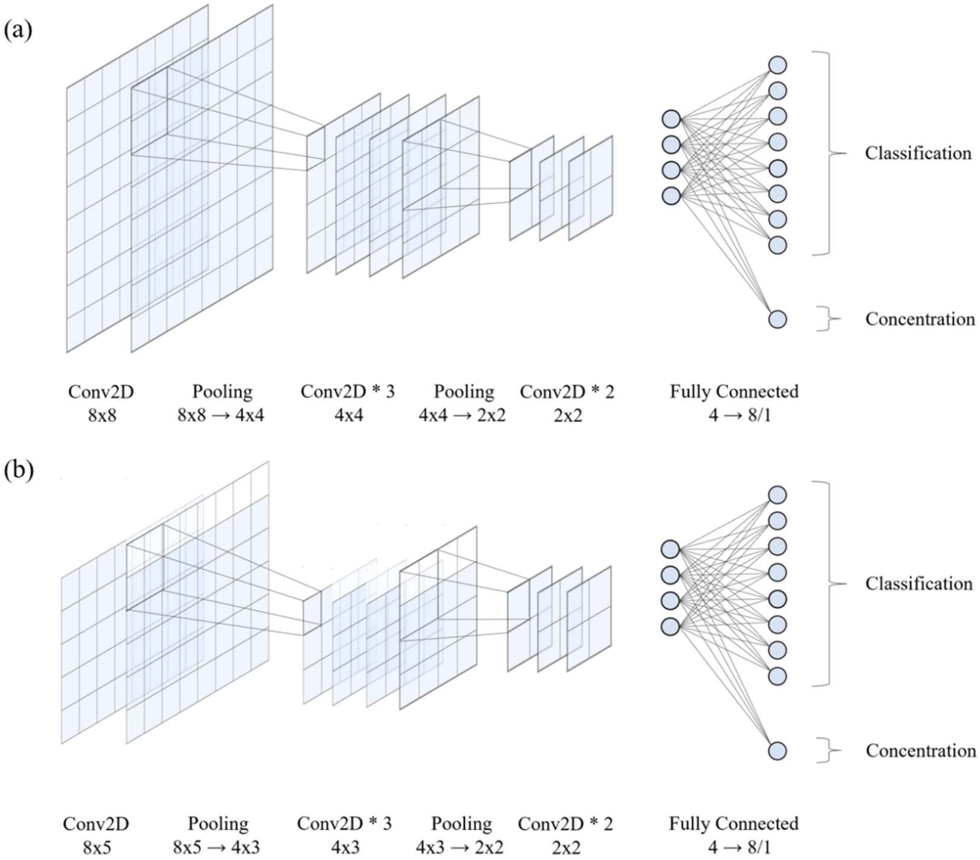


Fig. 1. (a) CNN network structure with eight sensors. (b) CNN network structure with five sensors.

2.2.4 RNN Models

RNN models are typically used for time series prediction and sequential pattern discrimination. They can treat with time series data and sequential data, especially those with repetitive patterns. In this study, the data collected by a sensor is sequential data that oscillates between certain values and shows periodic behavior. The responding curves of the sensors are attached in Supporting Information. Accordingly, the data collected by the sensor array is a multi-dimensional time sequence that can be interpreted with RNN models. This study involves the implementation of two RNN models, LSTM and GRU. The structures of LSTM cell and GRU cell are shown in Fig. 2.

The LSTM unit contains a cell, an input gate, an output gate and a forget gate [12]. This structure is developed to deal with the vanishing gradient problem of traditional RNN [4, 12]. As shown in the formula, there exists a recurrent relation between the input gate i_t, the output gate o_t, the forget gate f_t, the hidden state h_t, and the memory unit c_t.

$$f_t = \sigma_s\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{1}$$

$$i_t = \sigma_s(W_i x_t + U_i h_{t-1} + b_i) \tag{2}$$

$$o_t = \sigma_s(W_o x_t + U_o h_{t-1} + b_o) \tag{3}$$

$$g_t = \phi_h\left(W_g x_t + U_g h_{t-1} + b_g\right) \tag{4}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{5}$$

$$h_t = o_t \odot \sigma_t(c_t) \tag{6}$$

where σ_s denotes the sigmoid function, ϕ_h denotes the tanh function, $\odot$ denotes the element-wise product. W, U, and b are weight matrices and biases that are learned by training. x is the input vector. t is the iteration index of time. Initial conditions are $c_0 = 0$ and $h_0 = 0$.

The Gated Recurrent Unit is a simplification of the LSTM Unit with fewer memory gates. It is simpler to compute and implement [10]. There are several variations of GRU, including but not limited to Fully Gated Unit [10], Minimal Gated Unit [30], and Content-Adaptive Recurrent Unit [31]. In this study, we adopt the Minimal Gated Unit model, which is composed of forget vectors f_t, input vectors x_t, output vectors h_t, and candidate activation vectors $\hat{h}_t$. Their relations are shown in the formula.

$$f_t = \sigma_s\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{7}$$

$$\hat{h}_t = \phi_h(W_h x_t + U_h(f_t \odot h_{t-1}) + b_h) \tag{8}$$

$$h_t = (1 - f_t) \odot h_{t-1} + f_t \odot \hat{h}_t \tag{9}$$

By modifying the dimension of the output vector, we could fit LSTM and GRU models to both classification and regression tasks. In the case of identifying gas classification, we set the dimension to be the same as the length of the label array. Consequently, the models are trained to generate an array of possibility values. In the case of predicting gas concentration, we set the output vectors to be single-dimensional so that the concentration is derived directly.

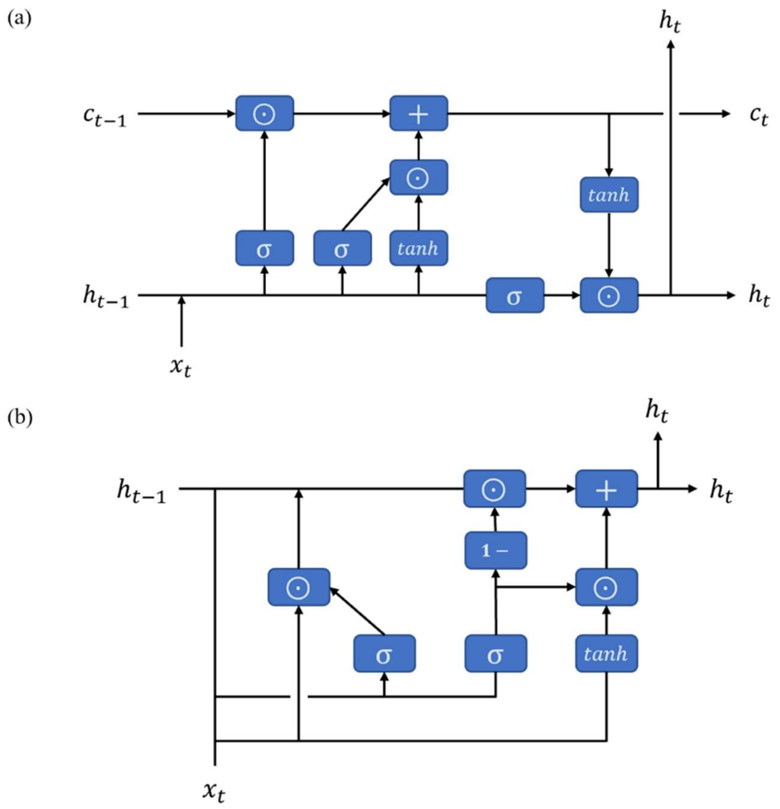


Fig. 2. (a) Structure of LSTM cell. (b) Structure of GRU cell.

2.3 Platform Setup

The measurement of model response time is influenced by computing power. To ensure the homogeneity of training conditions, all experiments are carried out on the same server without multi-processing. Operating system on the server is *Windows 10*. Deep learning models are implemented and trained with *Py-torch* [35, 39, 40]. Traditional machine learning methods used for comparison are implemented and trained with *Sklearn* [41, 42]. Codes are run in a virtual *python* environment inside *Anaconda3*, where the *python* version is 3.9.9, the *torch* version is 1.11.0, and the *Cuda toolkit* version is 11.6.0. Computations of the deep learning networks are accelerated by GPU

GTX 970[40], with *Nvidia* driver version 456.71 and *Cuda* version 11.1.

3. Model Performance

3.1 Performance of CNN Model

CNN training involves a choice of multiple hyperparameters. Some hyperparameters, such as epochs and data volume, tend to affect the sufficiency of the training only. Data volume is the size of data, and a small data volume can cause significant fluctuations in results. In preliminary work, it has been verified that the size of our dataset is sufficient to train a stable model [21]. Epoch number counts the time that the process of passing the complete dataset to the network is repeated. Multiple epochs are required for the model to be more stable. To ensure that the number of epochs is chosen correctly, the models are validated after every epoch of training. Note that gradient descent is not applied to the models during this process to ensure the originality of the validation set. The result from a single training of the CNN classification model is shown in Fig. 3(a), which demonstrates the relation between model performance and the number of epochs. As is shown in Fig. S3-S8 in Supporting Information, after 200-250 epochs of iteration, the validation accuracy no longer increases. Therefore, choosing 300 as the number of epochs is sufficient.

Two major factors influencing the performance of the CNN model are batch size and data length. Data length is the length of sampling points that is included in a single input. It suggests the length of our input matrix. Taking the response speed into account, a fixed data length of 8 is chosen (discussed in 2.1). Therefore, we mainly investigate the impact of various batch sizes on model performances. Batch size refers to the number of samples that is propagated through the neural network at a time. It reflects the frequency of the model being modified. In neural network training, parameter matrices are modified based on the calculation of gradient after every iteration of samples.

Various batch sizes are compared based on these considerations. The CNN classification model is evaluated by accuracy (proportion of correct identifications), while the CNN concentration model is evaluated by R-square. Each experiment is repeated 5 times with a mean result and a sample variance derived. The results of the experiments are shown in Fig. 3(a) and (b), which depict the relationship between model performances and batch size. Although the two models are performing different tasks, similar trends could be observed. In general, the performances of the models increase with batch size. The gain of slope reduces gradually until a nearly stable point. Accordingly, we select batch size = 900 for both tasks.

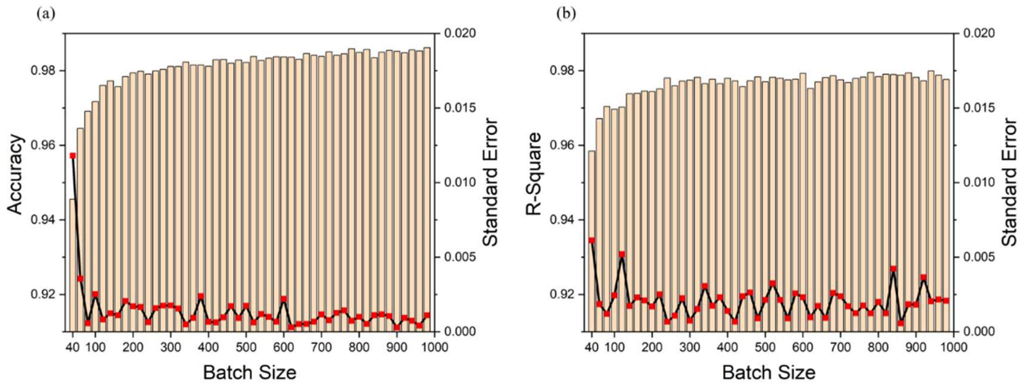


Fig. 3. (a) Accuracy vs. batch size of the CNN classification model. (b) Accuracy vs. batch size of the CNN concentration model.

3.2 Performance of RNN Models

The hyper parameters involved in RNN training are mostly the same as those of CNN. Data volume and epochs ensure the sufficiency of the training while data length and batch size influence the model performances. An epoch of 300 is selected which is sufficient to yield stable results, and it has been demonstrated in the supporting file. The data length is considered to be 8. To investigate the relationship between the performance of RNN model and the batch sizes, experiments are done and the results are visualized in Fig. 4 and Fig. 5.

Another significant hyperparameter, the dimension of the hidden state, applies only to recurrent neural networks. This parameter resembles the length of the hidden state vector h_t. In RNN training, the vector is propagated along the time steps t. A larger dimension of the hidden state suggests more features to be computed. If the features are not enough, necessary information could be omitted. However, expanding the number of dimensions too extensively could also include redundant features. Therefore, the choice of hidden state dimension requires experimental support.

Various batch sizes and hidden state dimensions are compared. The RNN classification models are evaluated by accuracy (proportion of correct identifications), while the RNN concentration models are evaluated by R-square. Mean results from 5 times of repetitive experiments are visualized. Fig. 4 shows the performances of LSTM classification model and concentration model against batch sizes and hidden state dimensions. Fig. 5 shows the performances of the GRU classification model and concentration model against batch sizes and hidden state dimensions. The mean accuracy and R-square from the training are displayed by bars related to the left axe. The variances from repetitive training are displayed by line-dot curves related to the right y axe. Both axes are under the same scale.

Except for the fluctuation, some general trends in the LSTM models could be concluded. Fig. 4(a) and (c) shows that the performance of the LSTM model is negatively related to batch size numbers. By Fig. 4(d), the concentration model performance of LSTM is positively related to the hidden state dimension with decreasing slope. The trend of the classification model performance of LSTM against hidden state dimension shows a maximum point of around 120. According to these experimental results, we select the hyperparameters of LSTM classification model to be batch size = 140, hidden dimension state = 120; the hyperparameters of LSTM concentration model to be batch size batch size = 220, hidden dimension state = 110.

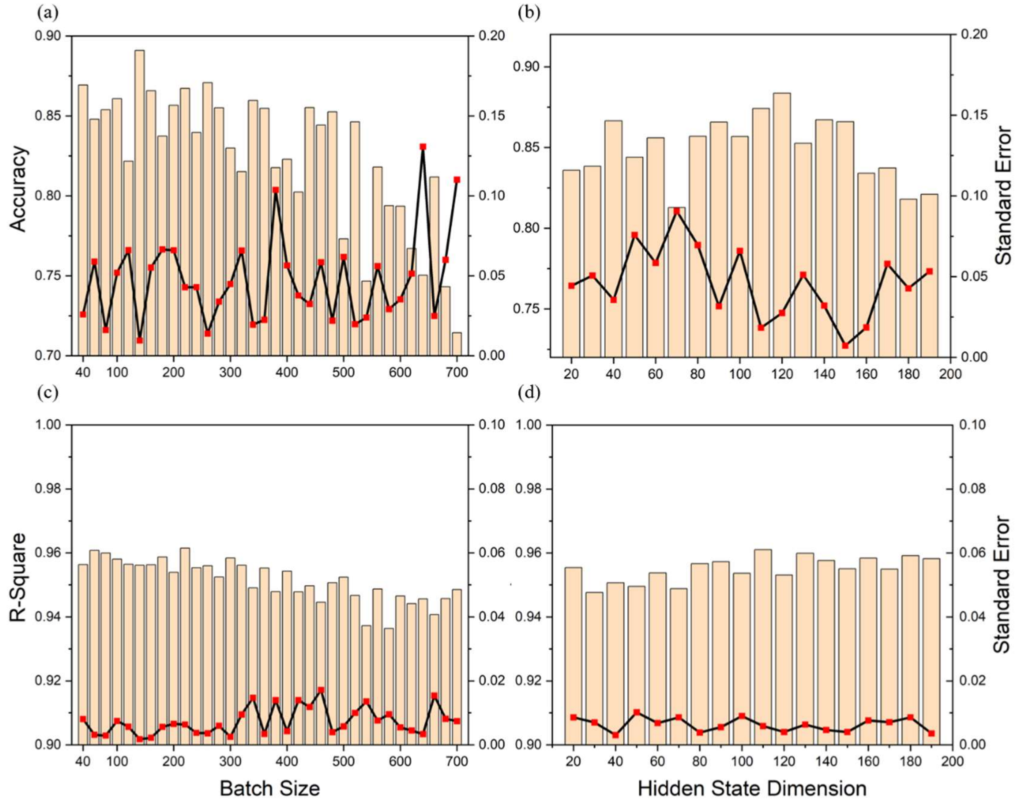


Fig. 4. (a) Accuracy vs. batch size of the LSTM classification model. (b) Accuracy vs. hidden state dimension of the LSTM classification model. (c) R-Square vs. batch size of the LSTM concentration model. (d) R-Square vs. hidden state dimension of the LSTM concentration model.

As to the GRU models, by Fig. 5(a), the performance of the classification model rises sharply and then falls gradually over the increasing batch. A maximum value is observed around batch = 140. As shown in Fig. 5(b), the performance of the concentration model rises gradually, except for the fluctuation. Fig. 5(c) and (d) shows that the performance of the GRU concentration model does not vary significantly along batch or hidden state dimensions, though some local maximums could be derived. According to these experimental results, the hyperparameters of the GRU classification model are selected, including batch size = 140, hidden dimension state = 110; while we set batch size = 460, hidden dimension state = 80 in the hyperparameters of the GRU concentration model.

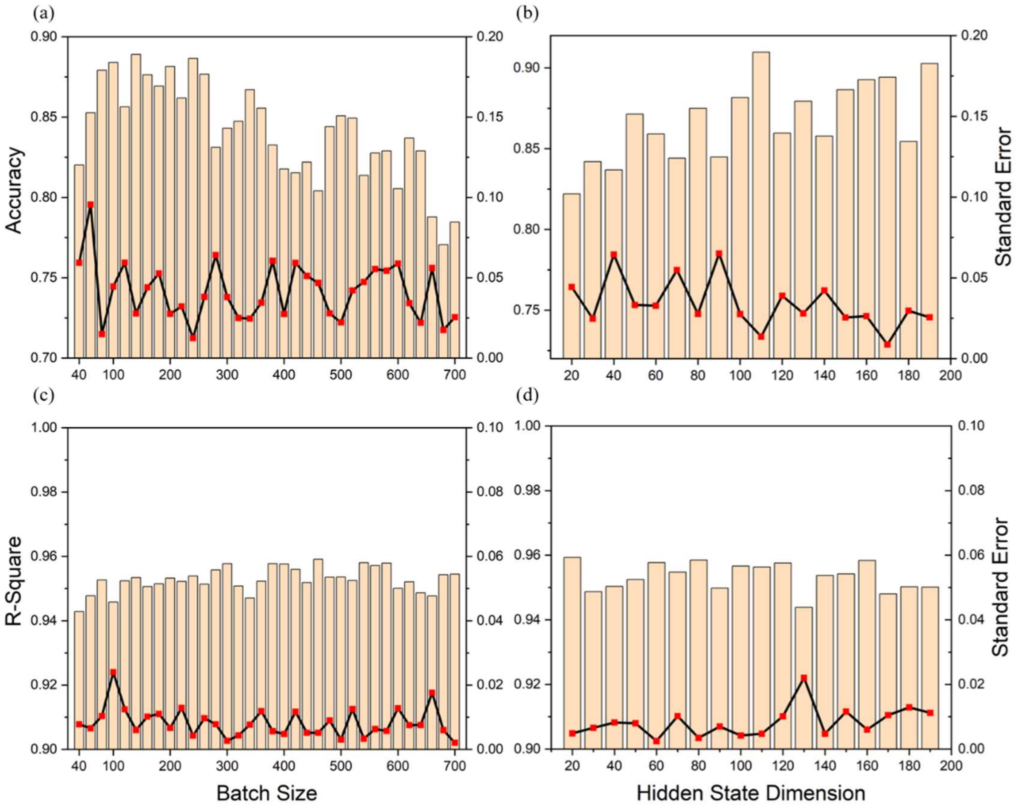


Fig. 5. (a) Accuracy vs. batch size of the GRU classification model. (b) Accuracy vs. hidden state dimension of the GRU classification model. (c) R-Square vs. batch size of the GRU concentration model. (d) R-Square vs. hidden state dimension of the GRU concentration model.

3.3 Performance Comparison

From the results shown above, we discover that the performance of CNN is significantly better than LSTM and GRU under full-sensor condition. Between the RNN networks, GRU performs slightly better than LSTM in the case of classification; slightly worse than LSTM in the case of concentration. To verify the overall performance of deep neural networks, we compare CNN, LSTM and GRU with traditional machine learning methods over the same data set. Four classical models are selected, which are Support Vector Machine (SVM), K Nearest Neighbor (KNN), Multi-Layer Perceptron Neural Network (MLP), and Decision Tree (DT). They are implemented with Scikit-Learn Libraries.

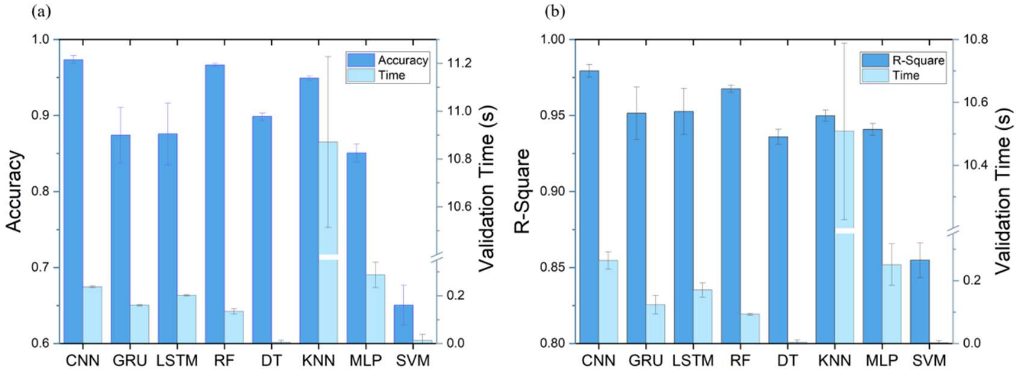


Fig. 6. (a) Accuracy and validation time vs. classification models. (b) R-square and validation time vs. concentration models.

Fig. 6 demonstrates the performances of classification models and concentration models respectively. For the comparison between models, we also take the validation time into account. This is because a model with slightly better outcomes is not necessarily superior if its computation takes too much time. Performance and validation time are both plotted in Fig. 6 as bars. Mean accuracy and R-square are related to the left axe, while validation time is related to the right y axe. In identifying the classification of gases, CNN is the leading model with the highest accuracy / F1-score and a rather low variance, as shown in Table 2. It is followed by KNN, DT, GRU, LSTM, MLP, and SVM, respectively. In predicting the concentration of gases, CNN is also the leading model considering R-square and root mean square error, as shown in Table 3.

Above all, we find that CNN generates the best performance in both tasks, with the best quantitative indicators (accuracy 0.986 for classification; R-square 0.979 for concentration prediction) and fairly short response times (0.393 s for classification; 0.264 s for concentration prediction). In contrast, LSTM and GRU are outperformed by Decision Tree, and also by KNN if the factor of response time is taken out of consideration. In full-sensor conditions, CNN is the most suitable choice for the electronic nose system.

Table 2. Performance of Models in Identifying Classification

Model	Accuracy	Error	F1-Score	Error	Time (s)	Error
CNN	0.986	0.003	0.986	0.003	0.393	0.028
GRU	0.876	0.037	0.877	0.037	0.171	0.006
LSTM	0.846	0.043	0.847	0.043	0.222	0.005
RF	0.969	0.002	0.970	0.002	0.136	0.010
DT	0.905	0.004	0.906	0.004	0.004	0.007
KNN	0.951	0.002	0.951	0.002	10.876	0.302
MLP	0.853	0.012	0.854	0.012	0.332	0.081
SVM	0.652	0.019	0.649	0.019	0.014	0.032

Table 3. Performance of Models in Predicting Concentration

Model	R-Square	Error	RMSE	Error	Time (s)	Error
CNN	0.979	0.004	0.479	0.051	0.264	0.028
GRU	0.952	0.017	0.733	0.114	0.123	0.029
LSTM	0.953	0.015	0.722	0.101	0.170	0.023
RF	0.968	0.002	0.364	0.025	0.093	0.003
DT	0.936	0.005	0.718	0.048	0.004	0.007
KNN	0.950	0.004	0.564	0.039	10.508	0.281
MLP	0.941	0.004	0.665	0.045	0.251	0.066
SVM	0.855	0.012	1.633	0.143	0.003	0.006

4. Array Optimization

4.1 Comparing the Significance of Sensors

We would now investigate the performances of deep neural network in the case of fewer sensors. When optimizing the sensor array, it is necessary to compare the significance of the sensors and leave the more important ones [3, 4, 14, 15]. Instead of analyzing the significance of sensors by traditional machine learning methods, we adopt a more direct approach that is implemented by deep learning. To explore the significance of each sensor, we exclude the corresponding data and train the model with data from the other seven sensors. Intuitively, the more significantly the sensors are removed, the more the model performances will degrade. Accordingly, we can obtain the significant order of the sensors in the sensor array.

The results of exclusive training are shown in Fig. 7. Fig. 7(a) shows the exclusive significance of the total six deep learning models. The classification models are labeled as cl and evaluated by accuracy. The concentration models are labeled as cc and evaluated by R-square. The x axe denotes the series of the sensor that is excluded. Fig. 7(b) shows the exclusive response time of the models, which are measured by seconds. After analyzing the results of each model, it can be concluded that the sensors do not show significant differences in their role during model learning.

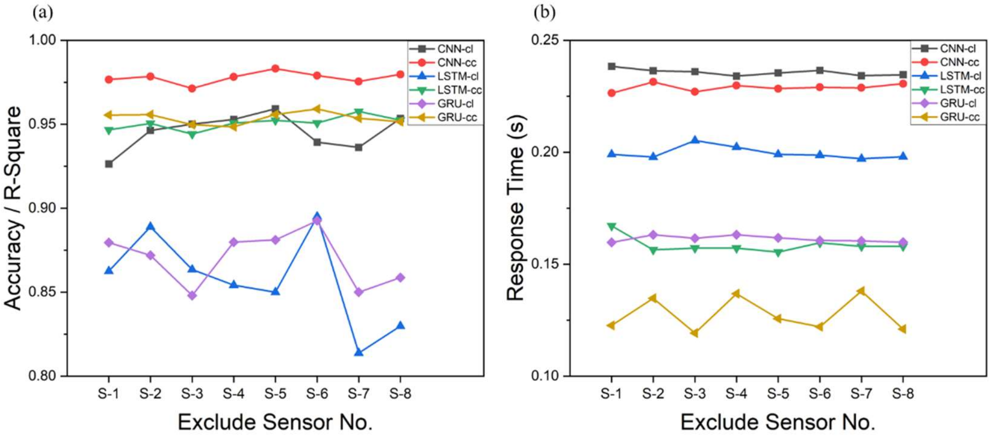


Fig. 7. Comparison of sensor significances: (a) Accuracy / R-Square, and (b) Validation time.

4.2 Sensor Array Optimization

This study reduces the number of active sensors to investigate array optimization of the electronic nose. Accordingly, the dimension of the input time series (or the width of the input data array) changes from 8 to 1 according to different study needs. When a certain width of the array is needed, data from the remaining sensors are discarded. Given that no significant difference between the role of various sensors has been detected, sensors are excluded by random order. Here, to prevent systematical error caused by a fixed order, 5 random permutations have been used in the exclusion of sensors, while the training under each permutation is repeated 5 times. Their average value and standard error are calculated and evaluated. Thereby, we could obtain a relation between model performances and the number of active sensors.

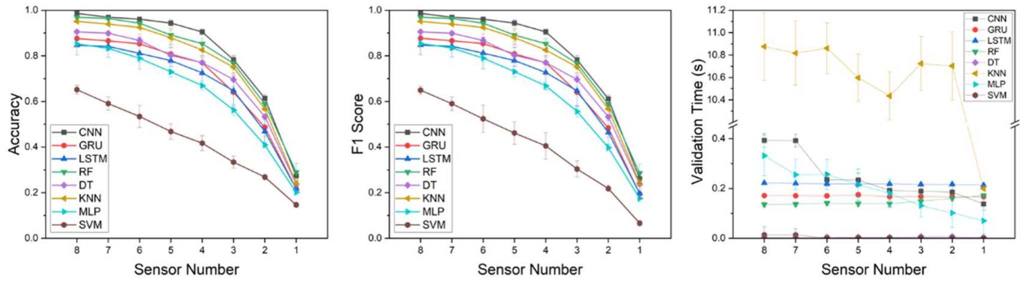


Fig. 8. Array optimization results of classification models: (a) Accuracy and (b) Validation time.

The results of classification models are shown in Fig. 9. Two measures are adopted to quantize model performance, which are accuracy and F1-score. A general trend of performance could be observed in all models alike. Overall, the model performance decreases with the decline of active sensors. The slope of decrease encounters a turning point at the 4-sensor condition. Before the active sensor number is reduced to 4, the performances of most models are under a slow linear decline except the SVM.

Performances between models are not influenced by the change in active sensor numbers. CNN is still the leading model, followed by KNN, DT, GRU, LSTM, MLP, and SVM, respectively. Fig. 9(c) shows the response time of the models with decreasing sensor numbers. The response time of CNN decreases by an even number, while the response time of LSTM and GRU remains almost the same. This is supported by their model design. CNN reduces the dimension of convolution layers by an even number, while the hidden dimension state of RNN remains the same.

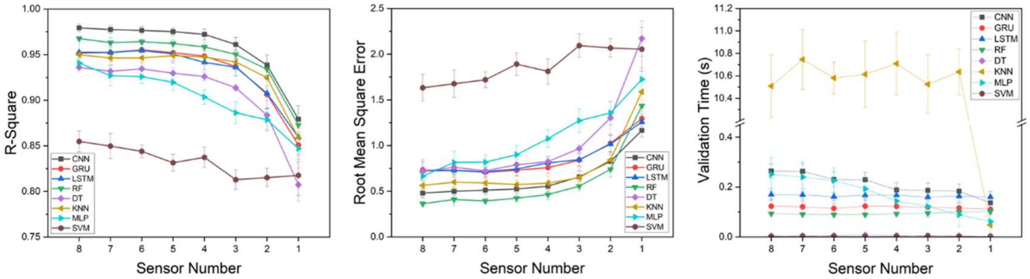


Fig. 9. Array optimization results of concentration models: (a) R-square, (b) root mean square error, (c) mean absolute error, and (d) validation time.

Fig. 9. includes the results of concentration models. Two measures are adopted to quantize model performance, which is R-square and root mean square error. A general trend of performance could be observed in all models alike. The performances of concentration models decrease with the decline of active sensors. Before the active sensor number is reduced to 4, the decrease in model performances is mild. After the number of active sensors is reduced to 4, the model performance decrease of model performance is intense. The sensor number of 4 is still the turning point of the performance decline. As to the comparison between models, their rank in performance does not change either. The response time is also similar to the results of classification, which is mostly influenced by model structure.

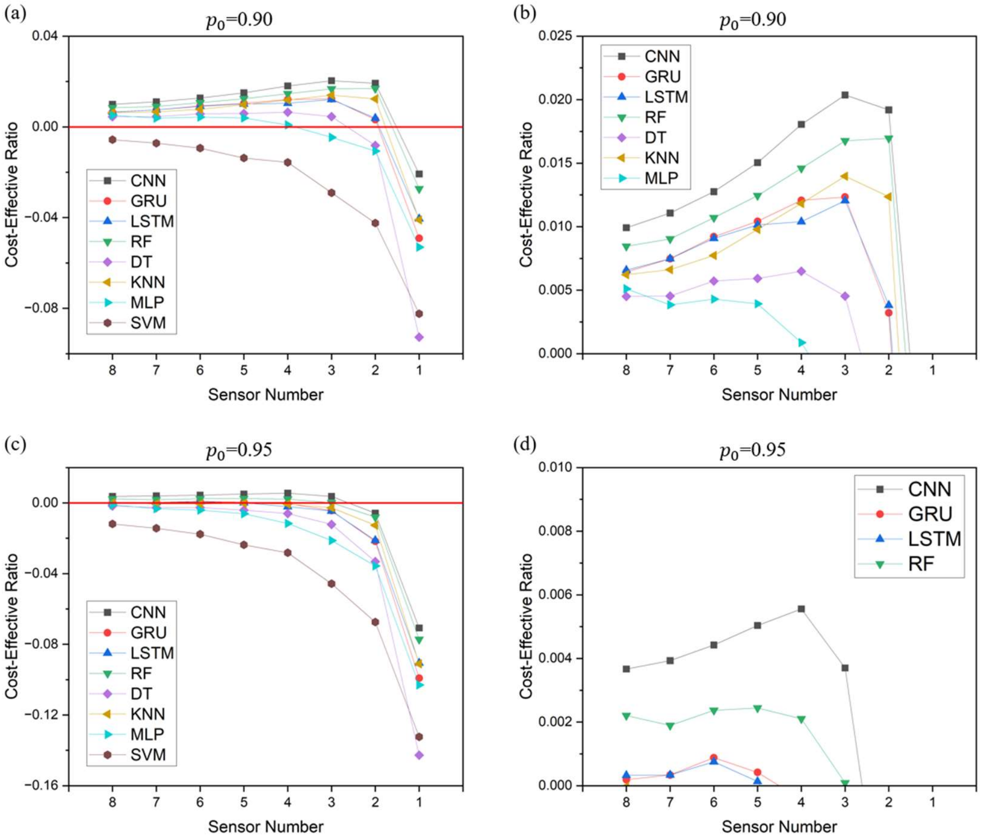
Overall, As the size of the sensor array shrinks, the performances of the models decrease accordingly, and when the number of sensors in each sensor array is reduced to 4, it comes to a turning point Meanwhile, the response time does not necessarily decrease with the number of sensors, and it depends mostly on the model structure. Multiple models do not show different patterns in sensor array optimization. CNN is the leading model in both performance and response time, while LSTM and GRU don't have much advantage over traditional machine learning methods.

4.3 Optimization Criterion for Practical Applications

The relation between model performance and the number of sensors discovered in this study can be used to guide the choice of E-noses in industries and households. Instead of expanding the size of the sensor array as much as possible for better performance, investors prefer a more cost-effective way especially when the intended application scenario is tight in space. When an E-nose is installed in the kitchen for a fire alarm, it needs to operate at the highest precision available. But if an E-nose is installed in a cellphone to detect environmental air quality, space and cost are more important factors to consider. Here we design an optimization criterion as a reference for e-nose producers.

$$k = \frac{p - p_0}{n}$$

In this equation, p refers to some quantized standard reflecting how well the model works, p_0 refers to the lowest acceptable value of that standard, and n refers to the number of sensors. We name this quotient k as the *cost-effectiveness ratio*. The suggested size of the sensor array is yielded by the non-negative maximum of the *cost-effectiveness ratio*, if any. If no such maximum exists, then p_0 needs to be set smaller.



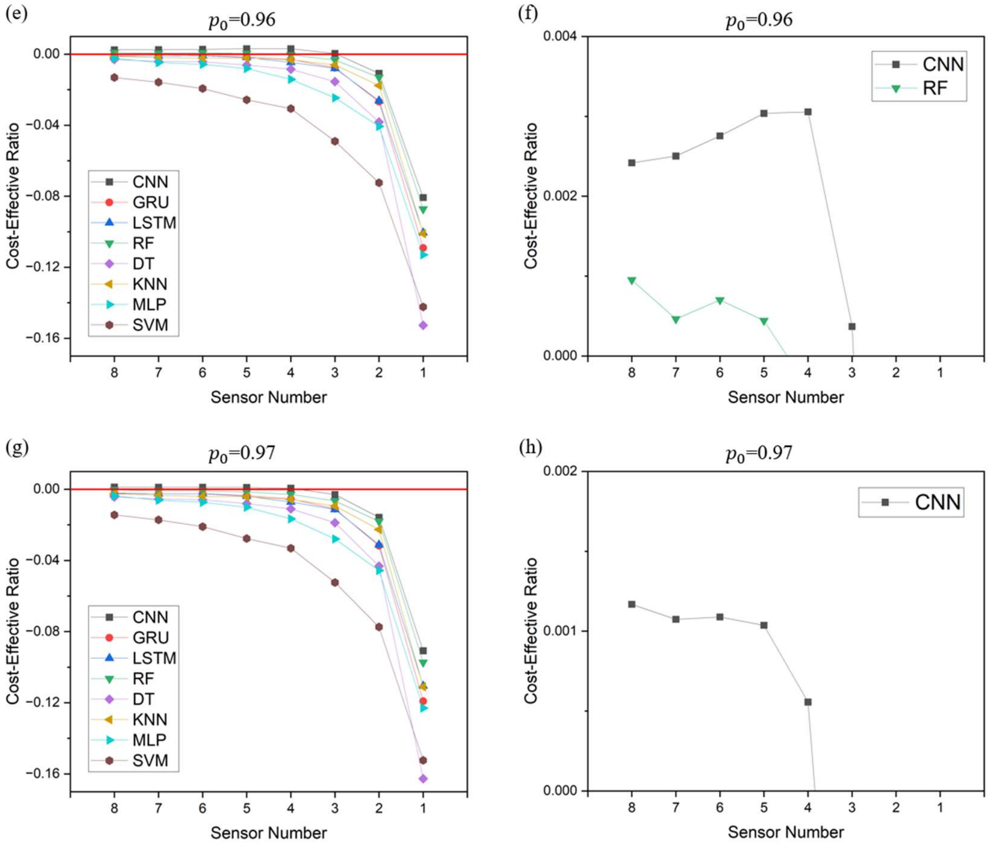


Fig. 10. Relation between the cost-effective ratio and sensor number of the concentration model (standard: R-square): (a)(b) $p_0 = 0.90$, (c)(d) $p_0 = 0.95$, (e)(f) $p_0 = 0.96$, and (g)(h) $p_0 = 0.97$.

For instance, the concentration models in this study with R-square as the performance standard are investigated by *cost-effectiveness ratio*. Fig. 10 shows the relation when p_0, the lowest acceptable R-square, is set to different values. Subfigures on the right show the non-negative parts of the curves on the left, namely, the parts above the red line. The parts below the red line are meaningless since they can't fulfill the minimum performance requirement. As can be seen, Fig. 10(b), (d), (f), and (g) show non-negative maximums at $n = 3, 4, 4, 8$, respectively. This implies that scenarios requiring a minimum R-square of 0.90, 0.95, 0.96, or 0.97 in predicting concentration is suggested to use 3, 4, 4, and 8 sensors. The *cost-effectiveness ratio* is an objective reference for the optimization of e-noses in practical applications.

5. Conclusion

In this study, the performances of three deep learning networks (CNN, LSTM, and GRU) and five traditional machine learning methods (RF, KNN, SVM, DT, and MLP) are compared for the pattern recognition of an E-nose system. CNN occurs to be the leading model for type classification and concentration prediction tasks, while LSTM and GRU are outperformed by traditional machine learning methods. The sensors used in this study have approximately equal contributions to the training results. A general pattern in array optimization can be concluded that model performance decreases by the number of sensors in each sensor array, with a turning point at 4. Before the number of sensors is reduced to 4, model performance declines slowly (classification by 8.22%; regression by 0.73%; from 8 to 4), and the model performance declines sharply (classification by 69.7%; regression by 9.57%; from 4 to 1) while the number of sensors is less than 4. By designing a *cost-effectiveness ratio* to quantize the benefit of adding or declining sensors, this work provides an optimization criterion that reveals the most gainful array size under different scenarios.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61971284 and 62101329), the Oceanic Interdisciplinary Program of Shanghai Jiao Tong University (SL2020ZD203 and SL2020MS031), the Scientific Research Fund of the Second Institute of Oceanography, Ministry of Natural Resources of P. R. China (SL2003), Shanghai Sailing Program (21YF1421400), and Startup Fund for Youngman Research at Shanghai Jiao Tong University. We also acknowledge analysis support from the Instrumental Analysis Center of Shanghai Jiao Tong University and the Center for Advanced Electronic Materials and Devices of Shanghai Jiao Tong University.

Reference

[1] W.S. Al-Dayyeni, S. Al-Yousif, M.M. Taher, A.W. Al-Faouri, N.M. Tahir, M.M. Jaber, et al., A review on electronic nose: coherent taxonomy, classification, motivations, challenges, recommendations and datasets, IEEE Access 9(2021) 88535–88551.

[2] D. Karakaya, O. Ulucan, M. Turkan, Electronic nose and its applications: A survey, Int. J. Autom. Comput. 17(2020) 179–209.

[3] H. Sun, F. Tian, Z. Liang, T. Sun, B. Yu, S.X. Yang, et al., Sensor array optimization of electronic nose for detection of bacteria in wound infection, IEEE Trans. Ind. Electron. 64(2017) 7350–7358.

[4] Y. Zou, J. Lv, Using recurrent neural network to optimize electronic nose system with dimensionality reduction, J. Electron. 9(2020) 2205.

[5] J.W. Gardner, H.V. Shurmer, T.T. Tan, Application of an electronic nose to the discrimination of coffees, Sens. Actuators B Chem. 6(1992) 71–75.

[6] J.A. Ragazzo-Sanchez, P. Chalier, D. Chevalier, M. Calderon-Santoyo, C. Ghommidh, Identification of different alcoholic beverages by electronic nose coupled to GC, Sens. Actuators B Chem. 134(2008) 43–48.

[7] S. Dragonieri, G. Pennazza, P. Carratu, O. Resta, Electronic nose technology in respiratory diseases, Lung 195(2017) 157–165.

[8] J. Burgués, M.D. Esclapez, S. Doñate, S. Marco, RHINOS: A lightweight portable electronic nose for real-time odor quantification in wastewater treatment plants, iScience 24(2021) 103371.

[9] S. Deshmukh, R. Bandyopadhyay, N. Bhattacharyya, R.A. Pandey, A. Jana, Application of electronic nose for industrial odors and gaseous emissions measurement and monitoring–An overview, Talanta 144(2015) 329–340.

[10] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, et al., Learning phrase representations using RNN encoder-decoder for statistical machine translation, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1724–1734.

[11] K. Fukushima, Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, Biol. Cybern. 36(1980) 193–202.

[12] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput, 9(1997) 1735–1780.

[13] H. Wu, T. Yue, Z. Xu, C. Zhang, Sensor array optimization and discrimination of apple juices according to variety by an electronic nose, Anal. Methods. 9(2017) 921–928.

[14] H. Zhang, J. Wang, X. Tian, H. Yu, Y. Yu, Optimization of sensor array and detection of stored duration of wheat by electronic nose, J. Food Eng. 82(2007) 403–408.

[15] K. Xu, J. Wang, Z. Wei, F. Deng, Y. Wang, S. Cheng, An optimization of the MOS electronic nose sensor array for the detection of Chinese pecan quality, J. Food Eng. 203(2017) 25–31.

[16] M. Pardo, G. Sberveglieri, Classification of electronic nose data with support vector machines, Sens. Actuators B Chem. 107(2005) 730–737.

[17] H.R. Estakhroueiyeh, E. Rashedi, Detecting moldy Bread using an E-nose and the KNN classifier, 2015 5th International Conference on Computer and Knowledge Engineering, 2015, pp. 251–255.

[18] L. Zhang, F. Tian, Performance study of multilayer perceptrons in a low-cost electronic nose, IEEE Trans. Instrum. Meas. 63(2014)1670–1679.

[19] J.H. Cho, P.U. Kurup, Decision tree approach for classification and dimensionality reduction of electronic nose data, Sens. Actuators B Chem. 160(2011) 542–548.

[20] A.u. Rehman, A. Bermak, Heuristic random forests (HRF) for drift compensation in electronic nose applications, IEEE Sens. J. 19(2019) 1443–1453.

[21] T. Wang, H. Zhang, Y. Wu, W. Jiang, X. Chen, M. Zeng, et al., Target discrimination, concentration prediction, and status judgment of electronic nose system based on large-scale measurement and multi-task deep learning, Sens. Actuators B Chem. 351(2022) 130915.

[22] Z. Wen, R. Xu, J. Du, A novel convolutional neural networks for emotion recognition based on EEG signal, 2017 International Conference on Security, Pattern Analysis, and Cybernetics, 2017, pp. 672–677.

[23] G. Xu, X. Shen, S. Chen, Y. Zong, C. Zhang, H. Yue, et al., A deep transfer convolutional neural network framework for EEG signal classification, IEEE Access 7(2019) 112767–112776.

[24] H. Gao, B. Cheng, J. Wang, K. Li, J. Zhao, D. Li, Object classification using CNN-based fusion of vision and LIDAR in autonomous vehicle environment, IEEE Trans. Industr. Inform. 14(2018) 4224–4231.

[25] Z. Liang, A. Powell, I. Ersoy, M. Poostchi, K. Silamut, K. Palaniappan, et al., CNN-based image analysis for malaria diagnosis, 2016 IEEE international conference on bioinformatics and biomedicine, 2016, pp. 493–496.

[26] D. Bhatt, C. Patel, H. Talsania, J. Patel, R. Vaghela, S. Pandya, et al., CNN variants for computer vision: history, architecture, application, challenges and future scope, J. Electron. 10(2021) 2470.

[27] P.F. Qi, Q.H. Meng, M. Zeng, A CNN-based simplified data processing method for electronic noses, 2017 ISOCS/IEEE International Symposium on Olfaction and Electronic Nose, 2017, pp. 1–3.

[28] Y. Shi, F. Gong, M. Wang, J. Liu, Y. Wu, H. Men, A deep feature mining method of electronic nose sensor data for identifying beer olfactory information, J. Food. Eng. 263(2019) 437–445.

[29] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Nature 323(1986) 533–536.

[30] G.-B. Zhou, J. Wu, C.-L. Zhang, Z.-H. Zhou, Minimal gated unit for recurrent neural networks, Int. J. Autom. Comput. 13(2016) 226–234.

[31] K.-H. Chan, W. Ke, S.-K. Im, CARU: A content-adaptive recurrent unit for the transition of hidden state in NLP, in: H. Yang, K. Pasupa, A.C.-S. Leung, J.T. Kwok, J.H. Chan, I. King (Eds.), Neural Information Processing, 2020, pp. 693–703.

[32] D.R. Wijaya, R. Sarno, E. Zulaika, DWTLSTM for electronic nose signal processing in beef quality monitoring, Sens. Actuators B Chem. 326(2021) 128931.

[33] Q. Wang, H. Qi, F. Liu, Time series prediction of e-nose sensor drift based on deep recurrent neural network, 2019 Chinese Control Conference, 2019, pp. 3479–3484.

[34] B. Lu, L. Fu, B. Nie, Z. Peng, H. Liu, A novel framework with high diagnostic sensitivity for lung cancer detection by electronic nose, Sensors 19(2019) 5333.

[35] S. Wang, Y. Hu, J. Burgués, S. Marco, S. Liu, Prediction of gas concentration using gated recurrent neural networks, 2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems, 2020, pp. 178–182.

[36] T. Wang, Y. Wu, W. Jiang, Y. Zhang, W. Lv, X. Chen, et al., Unmanned gas-sensing system for large-scale measurement of electronic nose, Proceedings of 2021 International Conference on Autonomous Unmanned Systems, 2021, pp. 629–637.

[37] M. Meng, Y.J. Chua, E. Wouterson, C.P.K. Ong, Ultrasonic signal classification and imaging system for composite materials via deep convolutional neural networks, Neurocomputing. 257(2017) 128–135.

[38] R. San-Segundo, M. Gil-Martín, L.F. D'Haro-Enríquez, J.M. Pardo, Classification of epileptic EEG recordings using signal transforms and convolutional neural networks, Comput. Biol. Med. 109(2019) 148–158.

[39] J.C. Rodriguez Gamboa, A.J. da Silva, I.C. S. Araujo, E.S. Albarracin E, C.M. Duran A, Validation of the rapid detection approach for enhancing the electronic nose systems performance, using different deep learning models and support vector machines, Sens. Actuators B Chem. 327(2021) 128921.

[40] D. Yu, Y. Gu, A Machine Learning Method for the Fine-Grained Classification of Green Tea with Geographical Indication Using a MOS-Based Electronic Nose, Foods 10(2021) 795.

[41] M. Camardo Leggieri, M. Mazzoni, S. Fodil, M. Moschini, T. Bertuzzi, A. Prandini, et al., An electronic nose supported by an artificial neural network for the rapid detection of aflatoxin B1 and fumonisins in maize, Food Control. 123(2021) 107722.

[42] P. Srinivasan, J. Robinson, J. Geevaretnam, J.B.B. Rayappan, Development of electronic nose (Shrimp-Nose) for the determination of perishable quality and shelf-life of cultured Pacific white shrimp (Litopenaeus Vannamei), Sens. Actuators B Chem. 317(2020) 128192.