SIR-3DCNN: A Framework of Multivariate Time Series Classification for Lung Cancer Detection

Ran Liu, *Member, IEEE*, Shidan Wang, Fengchun Tian, *Member, IEEE*, and Lin Yi

***Abstract*— Electronic noses (e-noses) have proven effective in detecting lung cancer (LC) by analyzing volatile organic compounds (VOCs) in breath samples, with multivariate time series classification (MTSC) as the primary task. However, challenges remain in effectively capturing spatiotemporal information for MTSC. To address this, a novel method called SIR-3DCNN for MTSC in LC detection is introduced. A pivotal aspect of SIR-3DCNN is the sensor array optimization (SAO), an algorithm based on linear discriminant analysis (LDA) that can decrease the number of sensors from 22 to 8 while increasing accuracy by 2.35% points in our study. Furthermore, SIR-3DCNN incorporates an advanced technique for representing spatiotemporal information, converting optimized multivariate time series (MTS) into maximum trajectory matrix images (MTMIs) and arranging them to maximize the sum of interframe mutual information (SIFMI). In addition, we have developed C3D-Light, a lightweight yet effective 3-D convolutional neural network (3DCNN) that demonstrates superior performance compared to other models. Comparative analyses with state-of-the-art methodologies reveal that SIR-3DCNN consistently outperforms existing methods, achieving perfect sensitivity (100%), the highest specificity (80%), accuracy (92.94%), AUC (94.85%), precision (90.26%), and F1-score (94.86%) among all compared methods. This advancement holds significant promise for LC detection using electronic noses. The source code is available at https://github.com/cqu-3dteam/sir-3dcnn**

***Index Terms*— 3-D convolutional neural network (3DCNN), electronic nose (e-nose), lung cancer (LC) detection, multivariate time series (MTS), sensor array optimization (SAO).**

I. Introduction

LUNG cancer (LC), which is responsible for 28% of all cancer-related deaths and results in over 1.6 million lives lost globally annually [1], is one of the most prevalent life-threatening cancers. Current medical diagnostic techniques for LC encompass a range of approaches, including radiological imaging, histopathology, and molecular assays. Although these methods have shown efficacy, they still face challenges in meeting the requirements of noninvasiveness, cost, and speed. For instance, blood-based cell-free DNA (cfDNA) analysis [2], a type of molecular assay technique, is effective but is constrained by invasive sampling, expensive equipment, and lengthy processing times. In contrast, electronic nose (e-nose) technology offers a noninvasive, cost-effective, and relatively rapid alternative by analyzing volatile organic compounds (VOCs) in exhaled breath for LC detection [3], [4]. However, e-nose systems still struggle with suboptimal accuracy [5] and computational inefficiency due to inherent limitations in multivariate time series classification (MTSC) methods.

Researchers have endeavored to develop various MTSC methods to enhance the effectiveness and efficiency of e-nose detection. These methods can be divided into two categories: traditional methods and deep learning (DL) methods. Traditional methods encompass TSF [6], HIVE-COTE 2.0 [7], and FreshPRINCE [8]. These methods typically require extensive handcrafted feature engineering, which often relies on personal experience to capture discriminative spatiotemporal features in multivariate time series (MTS) [6], [7], [9], [10], [11].

DL methods [12], [13], [14] predominantly consist of convolutional neural network (CNN)-based approaches, such as ResNet [12], FCN [12], TapNet [13], and others. Alongside CNN-based methods, Transformer-based (e.g., TST [15] and FormerTime [16]) and graph neural networks-based methods (e.g., TodyNet [17]) have recently been integrated into MTSC tasks. These recent techniques have exhibited strong performance in the latest studies [18] and often serve as benchmarks for performance comparison. Another important DL technique for time series classification is the phase-space reconstruction (PSR)-based method. The PSR-based method shows potential for enhancing spatiotemporal information extraction through a detailed nonlinear representation of signal features [19].

Although the aforementioned traditional and DL methods exhibit certain advantages in MTSC tasks, they still face inherent challenges in enhancing the effectiveness and efficiency of MTSC.

1) *Data Redundancy:* MTS samples often include redundant univariate time series (UTS) from cross-sensitive sensors. Existing methods typically necessitate the input of a full MTS sample. Redundancy may arise when all UTS in an MTS are utilized as classifier inputs, which not only diminishes classification accuracy but also increases computational latency (e.g., prolonged training or testing time).

Received 20 November 2024; revised 18 March 2025; accepted 22 March 2025. Date of publication 21 April 2025; date of current version 2 May 2025. This work was supported in part by the Natural Science Foundation Project of Chongqing under Grant CSTB2023NSCQ-MSX0413 and in part by the Open Fund Project of Chongqing Key Laboratory of Translational Research for Cancer Metastasis and Individualized Treatment. The Associate Editor coordinating the review process was Dr. Zhenghua Chen. *(Ran Liu and Shidan Wang are co-first authors.) (Corresponding author: Lin Yi.)*

Ran Liu is with the College of Computer Science, Chongqing University, Chongqing 400044, China (e-mail: ran.liu_cqu@qq.com).

Shidan Wang and Fengchun Tian are with the School of Microelectronics and Communication Engineering, Chongqing University, Chongqing 400044, China (e-mail: shidan_wang@qq.com; fengchuntian@cqu.edu.cn).

Lin Yi is with Chongqing Key Laboratory of Translational Research for Cancer Metastasis and Individualized Treatment, Chongqing University Cancer Hospital, Chongqing 400030, China (e-mail: linyi_cqu@163.com).

Digital Object Identifier 10.1109/TIM.2025.3563000

2) *Ineffective MTS Representation:* The arrangement of UTS within MTS can vary widely, resulting in multiple forms of representation for MTS. Poorly arranged MTS representation can make it difficult for the classifier to recognize discriminative features of the MTS, as discriminative features exist not only within individual UTS but also between multiple UTS [20].
3) *Unsuitable Classifier:* Existing methods predominantly rely on conventional classifiers such as support vector machine (SVM) and 2DCNN. However, these classifiers struggle to effectively process MTS data due to its high dimensionality and inherent spatiotemporal complexities. Such challenges often obscure critical discriminative features, resulting in suboptimal classification accuracy. To address this gap, we need to develop classifiers that are more suitable for MTSC tasks.

Our motivation is to explore an innovative method that effectively and efficiently addresses the aforementioned challenges. To achieve this, we introduce a method called SIR-3DCNN. This method initially preprocesses each MTS to reduce redundancy, and then represents the optimized MTS as image sequences. Finally, we leverage a specially designed 3-D convolutional neural network (3DCNN) to learn spatiotemporal information from the image sequences and perform classification. The main contributions of this study are as follows.

1) We implement a sensor array optimization (SAO) algorithm to minimize the number of sensors for classification, reducing data redundancy. This algorithm evaluates the contribution of each sensor on the training set using linear discriminant analysis (LDA) and subsequently selects the minimal number of sensors required to achieve maximum accuracy at the first attempt on the validation set, thus enhancing classification performance and decreasing time complexity.
2) We propose a spatiotemporal information representation (SIR) method for MTS representation, converting each UTS into a 2-D image via the maximum trajectory matrix image (MTMI) technique [10]. To facilitate the acquisition of task-relevant information and hence improve performance, these images are arranged to maximize the sum of interframe mutual information (SIFMI), achieving optimal spatiotemporal representation.
3) We develop C3D-Light, a lightweight 3DCNN model for classification, which treats the outputs of the SIR method as a video, effectively handling MTS spatiotemporal information. It focuses on the image's top-left area, crucial for capturing initial-stage discriminative features like transient features in e-nose signals, thereby leading to improved classification. C3D-Light exhibits notable computational efficiency while maintaining robust performance.

The remaining sections of this article are organized as follows. Section II outlines the MTMI technique. Section III provides a detailed description of the proposed SIR-3DCNN framework. Section IV presents the experimental results and discussions. Finally, conclusions are drawn in Section V.

II. Related Work

In this section, we briefly describe our previously proposed MTMI technique, which is highly relevant to this study. The MTMI technique is designed to convert an UTS into a high-resolution image [10].

Let $\mathbf{X} = [x_1, x_2, \ldots, x_d]^T$ represent an ordered set of real numbers that constitute a scalar time series. For instance, data gathered by a sensor can form an UTS. The length of $\mathbf{X}$, denoted as d, corresponds to the number of real numbers in the series.

Given positive integers τ and m, if $\mathbf{X}$ is embedded into an $m-$dimensional phase space, we can obtain an $l \times m$ trajectory matrix

$$\mathbf{\Theta} = \begin{bmatrix} x_1 & x_{1+\tau} & \cdots & x_{1+(m-1)\tau} \\ x_2 & x_{2+\tau} & \cdots & x_{2+(m-1)\tau} \\ \vdots & \vdots & \ddots & \vdots \\ x_l & x_{l+\tau} & \cdots & x_{l+(m-1)\tau} \end{bmatrix} \quad (1)$$

where $l = d - (m-1)\tau$. When all elements in matrix $\mathbf{\Theta}$ are linearly mapped to integer intervals [0, 255], it can be regarded as a grayscale image with a resolution of $l \times m$, termed the trajectory matrix image (TMI). Essentially, TMI represents a 1-D time series through a 2-D image, enabling the use of DCNNs for automatic feature extraction.

The resolution of the image $(l \times m = (l - (m-1)\tau) \times m)$ depends on τ and m. Generally, a larger image leads to better classification performance by DCNNs. Therefore, determining the optimal values of τ and m to maximize the image resolution becomes crucial. This issue is resolved by the following theorem [10].

Theorem 1: When $\tau = 1$ and $m = \lfloor (d+1)/2 \rfloor$(or $m = \lceil (d+1)/2 \rceil$), the image $\mathbf{\Theta}$ attains the highest resolution.

We refer to the corresponding $\mathbf{\Theta}$ as the MTMI. Unless specified otherwise, $\mathbf{\Theta}$ denotes an MTMI in the remainder of this article. With the MTMI technique, we can convert each UTS in an MTS into an image, ultimately achieving the conversion of MTS to video.

III. Proposed Method

This study introduces SIR-3DCNN, a method that converts time series into images for MTSC tasks. Fig. 1 provides an overview of the SIR-3DCNN framework, where the input comprises the original MTS, and the output generates the predicted labels. SIR-3DCNN encompasses three distinct stages: SAO, SIR, and Classification. During the SAO stage, we evaluate the contributions of each sensor in every sample using LDA, and then select sensors based on their individual contributions. The data collected by selected sensors are then used for imaging and subsequent classification. The SIR stage consists of two pivotal steps: converting each UTS in the MTS into a 2-D image using the MTMI technique, followed by the arrangement of these 2-D images into a sequence to enhance classification performance. The final stage involves the implementation of a 3DCNN (C3D-Light), specifically engineered to effectively classify the arranged image sequences.

A. Sensor Array Optimization

One of the notable characteristics of the sensor array is cross-sensitivity, which often leads to redundancy in sensor

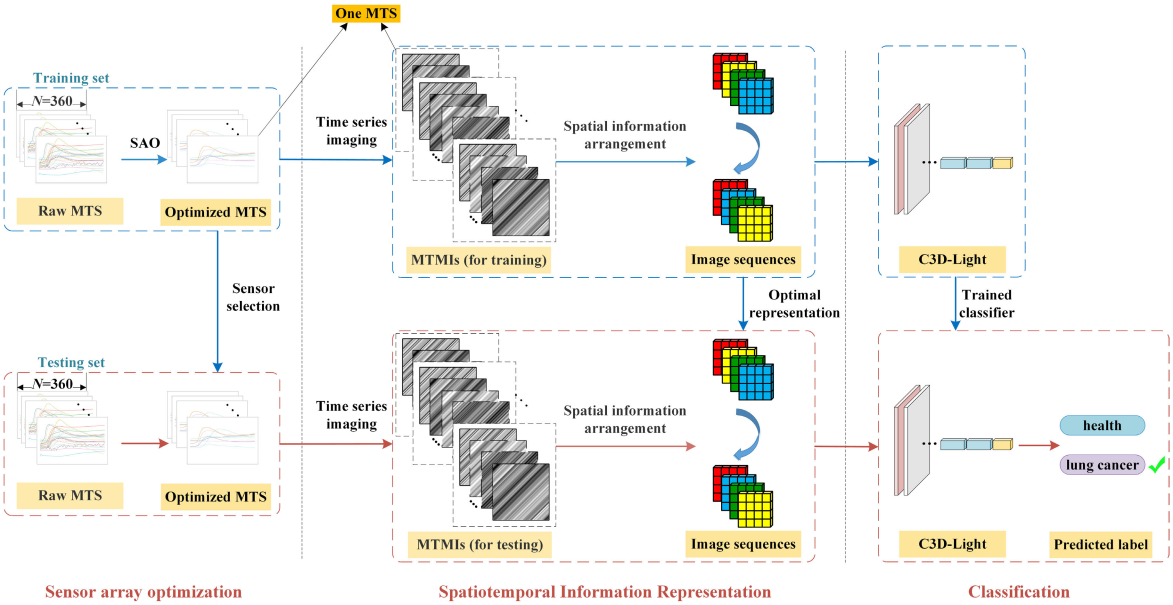


Fig. 1. Overview of the proposed framework SIR-3DCNN.

channels [21]. This redundancy not only fails to contribute positively to MTSC but can also compromise classification accuracy. More importantly, it significantly increases data dimensionality, resulting in higher time complexity for classification. Therefore, employing the SAO algorithm is essential to eliminate redundant channels, thereby reducing time complexity and enhancing classification accuracy.

The LDA algorithm is widely recognized for SAO [21], [22]. It aims to transform high-dimensional data into a lower-dimensional space while retaining as much class-related information as possible [22]. LDA identifies a projection direction that maximizes interclass distances and minimizes intraclass distances. In our LC dataset, with two classes ($c = 2$) and data from D channels ($D = 22$), the projection direction is defined by the linear discriminant function (LDF) as follows [21]:

$$g(\boldsymbol{F}) = \mathbf{W}^T \boldsymbol{F} \tag{2}$$

where $\boldsymbol{F} = [f_1, f_2, \ldots, f_i, \ldots, f_D]^T$, and $f_i = \frac{1}{n}\sum_{l=1}^{n} \max(h(\mathbf{M}_l^{\mathrm{tr}}, i))$ represents the feature of Sensor i ($1 \le i \le D$). The function $h(\mathbf{M}_l^{\mathrm{tr}}, i)$ returns the UTS $\mathbf{X}_i$ in the sample $\mathbf{M}_l^{\mathrm{tr}}$, and the function $\max(\cdot)$ returns the maximum value of all the elements in $\mathbf{X}_i$. $\mathbf{W} = [w_1, w_2, \ldots, w_i, \ldots, w_D]^T$ represents the weight vector, which can be calculated using the within-class scatter matrix and the between-class scatter matrix generated on the training set. The magnitude of w_i in $\mathbf{W}$ indicates the contribution of the ith sensor to LDA. The larger the $|w_i|$, the more significant its contribution to the ith sensor. Therefore, sensors should be selected based on the descending order of $|w_i|$. Subsequently, we select the minimal number of sensors that can achieve maximum accuracy at the first attempt on the validation set. The experiments in Section IV validate that our SAO algorithm effectively reduces the number of sensors required.

B. Spatiotemporal Information Representation

In the SIR stage, each MTS is transformed into a format conducive to classification by a 3DCNN. This stage includes two critical components: time series imaging and spatial information arrangement (SIA).

1) Time Series Imaging: We employ the MTMI technique to convert each UTS $\mathbf{X}_i^{\mathrm{tr}}$ ($1 \le i \le D$) in $\mathbf{M}_l^{\mathrm{tr}}$ into a 2-D image. MTMI employs two primary techniques: first, a novel time-series imaging technique based on PSR that converts trajectory matrices into 2-D images, minimizing information loss; second, it incorporates parameter optimization tailored for time-series imaging, thereby maximizing the TMI resolution and determining the optimal reconstruction parameters. As a result, each MTS is converted into a set of D images (or D MTMIs).

2) Spatial Information Arrangement: In this component, we view the D images as a video consisting of D frames, despite the absence of temporal correlation among them. A key challenge lies in assembling these 2-D images into a sequence, as different arrangements lead to different representations, which significantly affect the classification performance of 3DCNN.

In this study, we propose a method called SIA to arrange the 2-D images of each MTS into a single image sequence. As more mutual information (MI) among adjacent frames facilitates the acquisition of more task-relevant information and leads to better performance [23], our method is designed to maximize the SIFMI in the arranged image sequence. The detailed description of SIA is as follows.

a) Calculation of the NMIM: Consider a set $\boldsymbol{\Theta}_s = \{\theta_1, \theta_2, \ldots, \theta_{H\times W}\}$, representing the MTMI converted from an UTS $\mathbf{X}_s$ of an MTS $\mathbf{M}$, where H and W denote the height and width of the $\boldsymbol{\Theta}_s$, respectively. The normalized MI (NMI) of two converted images ($\boldsymbol{\Theta}_s$ and $\boldsymbol{\Theta}_t$, corresponding to Channel s and Channel t) of an MTS can be formulated as follows [24]:

$$\mathrm{NMI}_{s,t} = \frac{2 \times MI(\boldsymbol{\Theta}_s; \boldsymbol{\Theta}_t)}{H(\boldsymbol{\Theta}_s) + H(\boldsymbol{\Theta}_t)}, s, t \in [1, D] \tag{3}$$

where $MI(\boldsymbol{\Theta}_s; \boldsymbol{\Theta}_t)$ denotes the MI between $\boldsymbol{\Theta}_s$ and $\boldsymbol{\Theta}_t$

$$MI(\boldsymbol{\Theta}_s; \boldsymbol{\Theta}_t) = \sum_{s=1}^{H\times W} \sum_{t=1}^{H\times W} p(\theta_s, \theta_t) \log_2 \frac{p(\theta_s, \theta_t)}{p(\theta_s)p(\theta_t)} \tag{4}$$

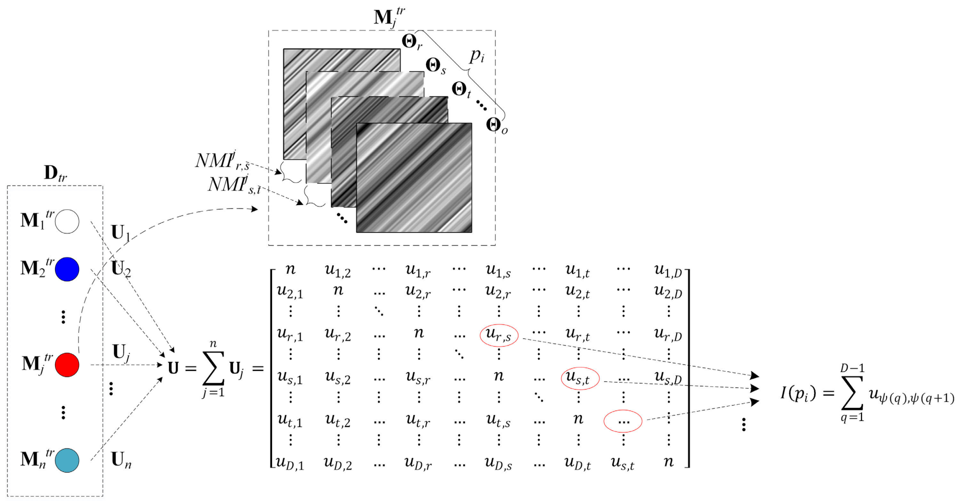


Fig. 2. Illustration of the calculations of the MIMTS and the SIFMI.

where $p(\theta_s, \theta_t)$ represents the joint probability mass function between mass θ_s and θ_t, while $p(\theta_s)$ and $p(\theta_t)$ are their respective marginal probability mass functions. $H(\mathbf{\Theta}_s)$ and $H(\mathbf{\Theta}_t)$ denote the entropies of $\mathbf{\Theta}_s$ and $\mathbf{\Theta}_t$, respectively [24]. The values of NMI range from 0 to 1, where values close to 0 indicate two largely independent assignments, and values close to 1 indicate significant agreement [25].

For any two images within an MTS, an NMI value can be computed. Thus, we can define an NMI matrix (NMIM) $\mathbf{U}_j$ $(1 \le j \le n)$ for each sample $\mathbf{M}_j^{\text{tr}}$, which incorporates all NMI values

$$\mathbf{U}_j = \begin{bmatrix} 1 & \text{NMI}^j_{1,2} & \cdots & \text{NMI}^j_{1,D} \\ \text{NMI}^j_{2,1} & 1 & \cdots & \text{NMI}^j_{2,D} \\ \vdots & \vdots & \ddots & \vdots \\ \text{NMI}^j_{D,1} & \text{NMI}^j_{D,2} & \cdots & 1 \end{bmatrix}. \tag{5}$$

Given that $\text{NMI}^j_{s,t} = \text{NMI}^j_{t,s}$, $\mathbf{U}_j$ is a symmetric matrix with diagonal elements equal to 1. Therefore, we only need to calculate $D(D-1)/2$ elements to obtain $\mathbf{U}_j$.

b) Calculation of the MIMTS: This step involves summing the NMIMs of all samples in the training set ($\mathbf{D}_{\text{tr}}$) to form a square matrix $\mathbf{U}$ of order D, as shown in Fig. 2, where n represents the number of samples, and each element $u_{s,t}$ in $\mathbf{U}$ satisfies

$$u_{s,t} = \sum_{j=1}^{n} \text{NMI}^j_{s,t} \tag{6}$$

where $u_{s,t} \in [0, n]$. This matrix $\mathbf{U}$ is refered to as the MI matrix for training set (MIMTS).

Please note that in the proposed SIA method, the MTMIs within each sample of both training and testing sets are arranged in the same order. Accordingly, the element $u_{s,t}$ represents the sum of NMI values between channels s and t across all samples in $\mathbf{D}_{\text{tr}}$.

c) Calculation of the SIFMI: We assume that the D MTMIs in sample $\mathbf{M}_j^{\text{tr}}$ form a permutation $p_i = (\mathbf{\Theta}_r, \mathbf{\Theta}_s, \mathbf{\Theta}_t, \ldots, \mathbf{\Theta}_o)$, where $1 \le i \le D!$, and the corresponding channels are arranged as $(r, s, t, \ldots, o)$. Here, permutations are represented as ordered n-tuples. The SIFMI for permutation p_i can be defined by

$$I(p_i) = \sum_{q=1}^{D-1} u_{\psi(q),\psi(q+1)},\ u_{\psi(q),\psi(q+1)} \in \mathbf{U} \tag{7}$$

where $\psi(q)$ maps the qth element in p_i to its subscript. For example, $\psi(1) = r$ for the above p_i. Fig. 2 illustrates the step-by-step process for calculating the SIFMI.

d) Calculation of the OPS: Define $S = \{p_1, p_2, \ldots, p_i, \ldots, p_{D!}\}$ as the set of all possible permutations formed by D MTMIs in each sample in $\mathbf{D}_{\text{tr}}$, with $|S| = D!$. Considering the symmetry of matrix $\mathbf{U}_j$, we define the reverse of permutation p_i as p_i^{-1}

$$p_i^{-1} = (\mathbf{\Theta}_o, \ldots, \mathbf{\Theta}_t, \mathbf{\Theta}_s, \mathbf{\Theta}_r). \tag{8}$$

This definition has the notable property that

$$I(p_i) = I\left(p_i^{-1}\right). \tag{9}$$

To calculate SIFMI efficiently, only $D!/2$ permutations need to be considered. By excluding reversed permutations in S, we obtain the refined set $\hat{S}$. We introduce the permutation filtering (PF) algorithm below to exclude reversed permutations in S, operating with a time complexity of $O(D!)$. For each permutation in $\hat{S}$, we calculate its SIFMI, resulting in the set $A = \{I(p_a), I(p_b), I(p_c), \ldots, I(p_i), \ldots, I(p_{D!/2})\}$, where $a, b, c, i \in [1, D!]$. Let the set $B = \{p_i \mid p_i \in \hat{S} \wedge I(p_i) = \max(A)\} \subseteq \hat{S}$, then set B is referred to as the optimal permutation set (OPS), with each element termed as an optimal permutation. The SIFMI for each optimal permutation p_i reaches its maximum value.

e) Arrangement of the MTMIs: MTMIs of all samples are arranged according to any optimal permutation in OPS, creating a novel representation of MTS. Recall that the arrangement order of MTMIs is consistent across all datasets, ensuring uniform representation for all samples. Once arranged, these newly represented samples are input into the classifier to obtain classification results.

Algorithm 1 PF

```
Input: The set of all possible permutations, $S$
Output: The filtered set of permutations $\hat{S}$ with $|\hat{S}| = \frac{D!}{2}$
for each $p_i \in S$ do
    if $p_i^{-1} \in S$ then
        $S \leftarrow S \setminus \{p_i^{-1}\}$
    end if
end for
$\hat{S} \leftarrow S$
return $\hat{S}$
```

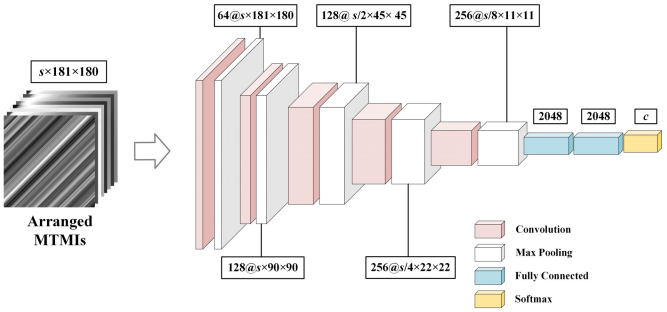


Fig. 3. Structure of C3D-Light.

C. Classification

In this section, we propose C3D-Light, a novel lightweight variant of the well-established C3D network architecture [26] for classification purposes. The C3D network architecture was selected for its ability to preserve spatial and temporal information across multiple frames, thereby enhancing the spatiotemporal relationships in the feature maps [26]. C3D-Light was meticulously developed as a more resource-efficient version of the C3D model. The structure of C3D-Light, which includes five convolutional layers, five max-pooling layers, two fully connected layers, and a softmax output layer, is detailed in Fig. 3. The output shape of each layer is detailed in the figure, where s represents the number of input MTMIs, and c denotes the number of classes. The 3-D convolutional operations use $3 \times 3 \times 3$ kernels with a spatial and temporal stride of 1. The 3-D pooling layers utilize two $2 \times 2 \times 2$ pooling kernels and one $1 \times 2 \times 2$ kernel. Each fully connected layer contains 2048 output units. Through these convolution and pooling operations, C3D-Light efficiently extracts deep features from the input data, streamlining the feature engineering process.

IV. Experiments and Discussions

A. Sensor Array Implementation

The sensor array is a key module for realizing the e-nose LC detection function and must be designed very carefully. In our research, the implementation of the sensor array follows the following three principles.

1) *Sensitivity to Specific VOCs:* The chosen gas sensors must be sensitive to the VOCs found in the exhaled breath of LC patients. According to [27], these VOCs include hydrocarbons, benzene and its derivatives, alcohols, aldehydes, ketones, esters, and other related compounds. The sensors should be capable of detecting these specific VOCs, as well as oxygen and carbon dioxide, due to potential metabolic changes in patients. In addition, the detection range should encompass the concentration levels of these VOCs.
2) *Broad-Spectrum Responsiveness and High Selectivity:* Due to the wide variety of VOCs in the exhaled breath of LC patients, the sensor array requires broad-spectrum responsiveness. Metal oxide semiconductor (MOS) sensors are ideal for this purpose, as they can detect multiple gas components simultaneously and are highly sensitive to VOCs linked to pulmonary conditions like lung carcinoma and chronic obstructive pulmonary disease (COPD) [28]. Therefore, MOS sensors comprise more than half of the sensors listed in Table I. To complement this, electrochemical sensors, known for their high selectivity in detecting specific gases, are also included. Incorporating diverse sensor types enhances diagnostic accuracy, as suggested in [29]. Our array comprises six sensor types: micro-electro-mechanical system (MEMS), MOS, hot-wire, solid electrolyte, electrochemistry, and photoionization. These sensors were selected for their sensitivity to LC VOC biomarkers [27]. Compared to our previous research [29], this study adds three sensors (Nos. 1, 16, and 22 in the sensor Index) to improve detection performance. All sensors are commercial products to ensure robustness and stability.
3) *Optimal Operating Temperatures:* Sensors must operate at their optimal temperatures for maximum efficiency. Our e-nose features two separate sensor chambers: one dedicated to MOS sensors, which require specific operating temperatures, and another for the remaining sensor types [30]. This separation ensures each sensor type functions optimally.

B. Dataset

In our experiments, we used the LC dataset, which was collected by our e-nose, to evaluate the performance of SIR-3DCNN. For detailed information about our e-nose system, please refer to [30] and [31], as well as Fig. 4(a). The dataset consists of 169 exhaled breath samples from 169 individuals, including 107 LC patients confirmed via biopsy and 62 healthy controls. Consequently, each subject contributes a single sample to this dataset. Table II provides details of the population characteristics, illustrating that the LC cohort encompasses a wide demographic range, including age (33–89 years), gender (69M:38F), and lifestyle factors (58% smokers), reflecting the epidemiological patterns of LC prevalence. The dataset includes multiple histopathological subtypes (squamous cell carcinoma, adenocarcinoma, and small cell carcinoma), although the majority of cases are middle- and late-stage, which is consistent with real-world diagnostic challenges where early-stage detection is rare. This stage distribution mirrors real-world diagnostic patterns where early-stage detection remains challenging due to asymptomatic presentation, though it may constrain the model's applicability to early-stage malignancies.

All subjects in this study voluntarily signed the informed content approved by the institutional review board (IRB)

TABLE I

INFORMATION OF GAS SENSORS IN SENSOR ARRAY

Index	Model	Detectable gases	Type	Manufacturer (Country)
1	GM-502B	Alcohols, Aldehydes, Benzenes, Ketones	MEMS	Winsen (China)
2	TGS822	Organic solvent vapors, Methane, Carbon monoxide, Isobutane, N-hexane, Benzene, Alcohol, Acetone	MOS	Figaro (Japan)
3	TGS2600	Air pollutants, Hydrogen, Carbon monoxide, Ethanol, Methane, Isobutane	MOS	Figaro (Japan)
4	TGS2602	Air pollutants, Hydrogen, Ammonia, Ethanol, Benzene, Hydrogen sulfide	MOS	Figaro (Japan)
5	TGS826	Nitrogenous compounds (ammonia, amines, etc.), Alcohols, Hydrocarbons (methane, butane, etc.)	MOS	Figaro (Japan)
6	TGS8669	Acetone, Benzene, Toluene	MOS	Figaro (Japan)
7	MS1100	Formaldehyde, Toluene	MOS	Ogam (Korea)
8	SP3S-AQ2	Methane, Carbon monoxide, Hydrogen	MOS	FIS (Japan)
9	MQ3	Alcohol vapor	MOS	Winsen (China)
10	MQ138	Benzene, Toluene, Methanol, Alcohol, Acetone, Formaldehyde	MOS	Winsen (China)
11	MP801	Benzene, Toluene, Formaldehyde, Alcohol, Smoke, Hydrogen, Acetone, Carbon monoxide, etc.	MOS	Winsen (China)
12	MP901	Toluene, Formaldehyde, Benzene, Alcohol, Acetone, etc.	MOS	Winsen (China)
13	WSP2110	Benzene, Toluene, Aldehydes, Hydrogen, etc.	MOS	Winsen (China)
14	MR516	Combustible gases, Formaldehyde, etc.	Hot-wire	Winsen (China)
15	TGS4161	Carbon dioxide	Solid electrolyte	Figaro (Japan)
16	ME4-H2S	Hydrogen sulfide, Phosphine, Formaldehyde, Carbon monoxide, Hydrogen, Carbon dioxide, Ammonia	Electrochemistry	Winsen (China)
17	4S	Sulfur dioxide	Electrochemistry	City (UK)
18	CH2O/M-10	Formaldehyde	Electrochemistry	Membrapor (Switzerland)
19	4-CH3SH-10	Methanethiol	Electrochemistry	Solidsense (Germany)
20	4OXV	Oxygen	Electrochemistry	City (UK)
21	ME4-C6H6	Benzene, Toluene, Xylene	Electrochemistry	Winsen (China)
22	PID-AH	Ammonia, Nitrogen monoxide, Nitrogen dioxide	Photoionization	Alphasense (UK)

TABLE II

DEMOGRAPHIC AND CLINICAL CHARACTERISTICS OF THE STUDY POPULATION

Attribute	Lung cancer	Health
Number	107	62
Gender (M/F)	69/38	23/39
Age range	33∼89	22∼59
Height range (cm)	140∼182	153∼185
Weight range (kg)	38.5∼89	40∼85
Smoking (Y/N)	62/45	3/59
Drinking (Y/N)	42/65	7/55
Stage (early/middle- and late-)	2/105	N/A

TABLE III

INFORMATION ON THE LC DATASET

Set	Health	Lung Cancer	Total
$\mathbf{D}_{tr}$	50	86	136
$\mathbf{D}_{va}$	6	10	16
$\mathbf{D}_{te}$	6	11	17
Total	62	107	169

before the experiment. The study was conducted in accordance with the Declaration of Helsinki, and the protocol was approved by the IRB of Chongqing University Cancer Hospital (19-012, 2/12/2019).

Our experimental procedure for data collection involves eight stages [29]: helium baseline collecting (HBC, S_1), sample gas adsorption (SGA, S_2), postadsorption (PA, S_3), thermal desorption (TD, S_4), sample injection (SI, S_5), preconcentration subsystem rinse at high temperature (PSRHT, S_6), preconcentration subsystem rinse at low temperature (PSRLT, S_7), and sensor array rinse (SAR, S_8). Each sampling session lasts 66.5 min, generating 3990 data points. Fig. 4(b) shows the sensor response curves of a sample in our LC dataset. Only data from the SI stage (S_5) were used for classification, where helium at 40 mL/min carries desorbed compounds to sensor chambers over 6 min, yielding 360 data points per channel. Each sample, referred to as an MTS, comprises data from 22 channels (equivalent to 22 sensors, forming a sensor array shown in Table I; $D = 22$). Each channel records 360 data points ($\mathbf{X}_i = [x_1, x_2, \ldots, x_d]^T$, with $d = 360$, $1 \leq i \leq D$) during the SI stage. The training ($\mathbf{D}_{\text{tr}}$) and testing ($\mathbf{D}_{\text{te}}$) sets of the LC dataset are detailed in Table III, with the validation set ($\mathbf{D}_{\text{va}}$) constituting approximately 10% of the training set.

C. Experiment Settings

Our experiments were conducted in Python using PyTorch and Scikit-learn libraries on an NVIDIA TITAN Xp GPU with 3840 cores and 12 GB memory. Evaluation metrics included sensitivity (Sens), specificity (Spec), accuracy (Acc), the area under the receiver operating characteristic curve (AUC), precision (Pre), F1-score (F1), training time (Tr, measured in minutes), and testing time (Te, measured in seconds). We also considered parameters (P) and Giga floating-point operations per second [GFLOPS, denoted by F(G)] to illustrate performance enhancements due to methodological design rather than increased model complexity. The results presented in Sections IV-D–IV-G represent the average outcomes of five independent runs, with optimal values highlighted in bold in the tables.

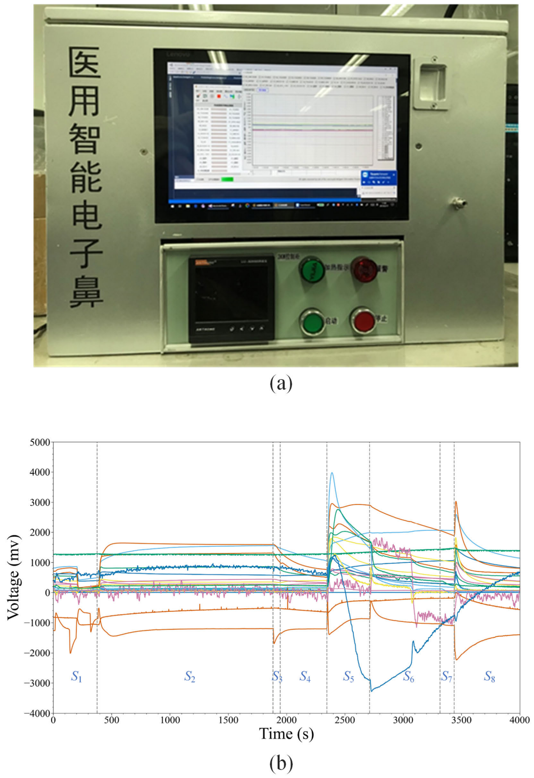


Fig. 4. (a) Proposed e-nose system for LC detection. (b) Sensor response curves of a sample captured by the system in (a). A total of 22 curves are shown. Only the data points from the SI stage (S_5) were utilized for classification.

Please note that in this study, we chose not to employ cross-validation to avoid excessively long training times. While cross-validation is indeed a powerful technique for evaluating model performance and mitigating overfitting, it can be computationally prohibitive, particularly for deep models that demand substantial training resources. Implementing cross-validation would require training multiple instances of these already resource-intensive models, which could make the process extremely time-consuming or even infeasible. In DL, it is common practice to use a fixed validation set instead of cross-validation due to the high computational demands of training complex models like 3DCNNs [10]. Consequently, in recent years, many models have adopted fixed validation sets for evaluation [10], [15], even when working with small datasets, where a fixed validation set is often designated [15]. In addition, a characteristic of our method makes it unsuitable for cross-validation. In our study, we employ an SAO algorithm to reduce redundancy and select the most informative sensors for classification tasks. This involves applying LDA to rank the sensors by their contribution and selecting the minimal number of sensors that can achieve the highest accuracy at the first attempt on the validation set. Cross-validation, however, requires that the validation set changes across different folds. If a fixed validation set is not used, this could lead to varying results for each SAO, resulting in inconsistent input representations for the model. Such variability could cause the model to learn less robust features, potentially diminishing its generalization ability.

TABLE IV
DoC of Sensors Calculated by LDA

Index	DoC	Index	DoC
4	**0.6907**	8	0.1215
22	**0.4653**	3	0.1192
10	**0.3132**	14	0.1184
11	**0.1988**	16	0.1145
15	**0.1672**	13	0.0958
7	**0.1422**	19	0.0887
1	**0.1226**	5	0.0660
21	**0.1219**	2	0.0646
17	0.0634	20	0.0503
9	0.0440	12	0.0224
18	0.0063	6	0.0006

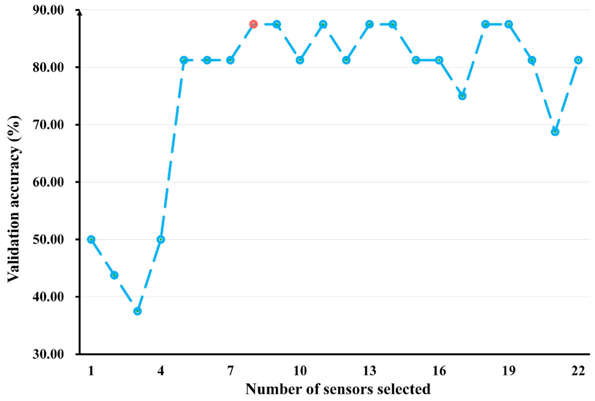


Fig. 5. Validation accuracy on the validation set of our LC dataset when parameter s varies.

D. Evaluation of SAO

We performed the SAO algorithm using both the training and validation sets of the LC dataset. Table IV details the degree of contribution (DoC) of all sensors, calculated by LDA and presented in descending order. Let s be the number of sensors selected. The optimal value of s is determined by the highest validation accuracy. Different from direct or stepwise selection methods [21], [32], our method started with an empty set, adding sensors sequentially in descending DoC order (see Table IV) until all were included. After each addition, the features of the selected sensors were input into a classifier (a support vector machine was used here due to its low time complexity) for classification. We tracked validation accuracy changes with each sensor addition and selected the number s that first achieved the highest accuracy. As shown in Fig. 5, s was set to 8. This indicated that the top 8 sensors in Table IV, highlighted in bold, were selected for further classification.

By employing the SAO algorithm, the number of sensors was reduced, and the selected sensors were arranged in ascending order of their Index. The effectiveness of the SAO algorithm on the LC dataset was evaluated through ablation experiments with C3D-Light. The classifier was configured with the following hyperparameters: the optimizer was stochastic gradient descent (SGD) with a learning rate of 0.0005 and a momentum of 0.9. The batch size was set to 8,

TABLE V

PERFORMANCE COMPARISON OF C3D-LIGHT WITH AND WITHOUT SAO

Case	Accuracy (%)	Tr (min)	Te (s)
w/o SAO	88.24	40.93	1.43
w/ SAO	**90.59**	**17.47**	**1.03**

and the model was trained over 200 epochs. The classification performance with and without the SAO algorithm is compared in Table V. Please note that the reported training or testing time covers the entire training or testing set, rather than just a single sample. In addition, the duration includes the time required for time series imaging. No SIA was employed to streamline the comparison. Table V shows that applying SAO results in a 2.35% point increase in classification accuracy, and reduces training and testing times by 58% and 28%, respectively. This indicates the effective removal of redundant channels and highlights our LDA-based SAO algorithm's ability to significantly enhance the classification performance of MTS.

The choice of LDA for SAO stems from its suitability for supervised classification tasks. Unlike unsupervised methods such as principal component analysis (PCA), which prioritize variance maximization without leveraging class labels, LDA explicitly maximizes interclass separability while minimizing intraclass variability. This property is crucial for improving diagnostic accuracy in LC detection, where clear distinctions between healthy and diseased samples are essential. Besides, while nonlinear methods like genetic algorithm (GA) might better capture complex sensor interactions, their computational demands make them less suitable for our framework. LDA's linearity guarantees computational efficiency, making it ideal for resource-constrained applications like real-time e-nose systems. Although LDA assumes Gaussian class distributions and equal covariance matrices—conditions that may not always hold in real-world data—our experiments here demonstrate its effectiveness, achieving a 2.35% point increase in classification accuracy while reducing the sensor count from 22 to 8. The 58% reduction in training time and 28% decrease in testing time (see Table V) further validate LDA's efficiency in this resource-constrained application. These performance and efficiency results of LDA in SAO align with our team's prior studies on sensor optimization [21]. In this prior work, we experimentally compared five SAO algorithms: LDA, PCA, GA, Wilks' Λ-statistic, and Mahalanobis distance. We demonstrated that LDA consistently outperformed the other algorithms in terms of classification accuracy and computational efficiency. For more details, please refer to [21]. Another key advantage of LDA lies in its feature ranking via its weight vector $\mathbf{W}$, which directly quantifies sensor contributions. This contrasts sharply with GA's opaque stochastic processes or Wilks' Λ-statistic's abstract hypothesis-driven metrics. The interpretability of LDA enables actionable insights into sensor relevance, critical for refining diagnostic frameworks. While future work may explore hybrid or nonlinear extensions, LDA's balance of performance, efficiency, and interpretability—validated by both this study and [21]—justifies its use here.

E. Evaluation of SIR

This section details the experiments to evaluate the effectiveness of the SIR method on the LC dataset. The experiments used optimized sensors selected by the SAO algorithm. These data were converted into 2-D images, referred to as MTMIs, and subsequently arranged for optimal representation using the SIA method. For a fair comparison, all experiments employed the same classifier, C3D-Light, along with consistent hyperparameters. Consequently, any variations in classification performance can be attributed solely to changes in MTMI arrangements.

Table VI provides an evaluation of the classification performance across four distinct MTMI arrangements: p_2 is sorted by descending DoC order in Table IV, p_3 is sorted by ascending Index order, and p_1 and p_4 represent permutations with the minimum and maximum SIFMI values, respectively. Among these, p_4 is identified as the optimal permutation. Table VI details the corresponding indices for these permutations (based on the index) and their respective SIFMI values, with notable variations in sensitivity, specificity, accuracy, and AUC. The progression from permutation p_1 to p_4 demonstrates a clear trend of improvement across all metrics, culminating in p_4 achieving the highest SIFMI value and optimal performance. This suggests that the optimal arrangement of MTMIs plays a crucial role in enhancing model accuracy and reliability.

Fig. 6 provides an intuitive visual representation of the MIMTS $\mathbf{U}$ for the training set under different arrangements of MTMI. The numbers along the matrix's main diagonal represent the sample count, indicating that the NMI value between an image and itself is 1. As evident from Fig. 6, varying arrangements of MTMI lead to distinct MIMTSs and, consequently, different SIFMIs. This arrangement plays a crucial role in the classification performance of 3DCNN. It's important to note that darker shades within the matrix signify higher NMI values between two MTMIs. Typically, arrangements with higher SIFMI correspond to an MIMTS with darker shades in the center of the MIMTS, aligning with the calculation principle outlined in Fig. 2.

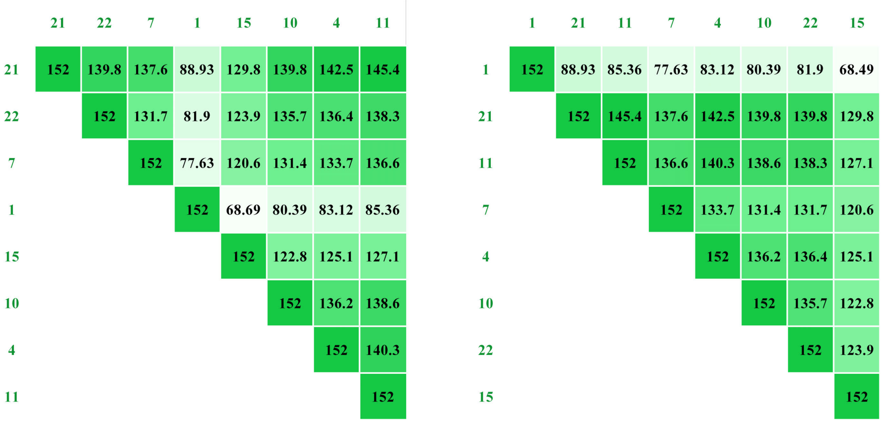
F. Evaluation of C3D-Light

First, we conducted a comprehensive comparison of our proposed classifier, C3D-Light, with other SOTA classifiers. Regarding the classification stage, the choice of classifier has a significant impact on the performance of our method. Since the training set is transformed into a 5-D tensor with the shape (samples, channels, frames, height, and width) prior to the classification stage, it is imperative that the classifier is able to process 5-D inputs. Given that 3DCNNs are well-suited for handling 5-D tensors, we opted to select 3DCNN models as our classifier. Specifically, in the "Classification" subsection of Section IV, we developed a 3DCNN model, named C3D-Light, tailored for our proposed framework. To validate the effectiveness of C3D-Light, we compared its performance with four well-known 3DCNN classification models: C3D [26], R3D [33], R(2 + 1)D [33], and CSN [26]. C3D was selected for its proficiency in maintaining spatial

TABLE VI

COMPARISON OF THE CLASSIFICATION PERFORMANCE WHEN DIFFERENT MTMI ARRANGEMENTS ARE APPLIED

MTMI arrangement	**SIFMI**	**Sens (%)**	**Spec (%)**	**Acc (%)**	**AUC (%)**
p_1 = [21,22,7,1,15,10,4,11]	816.81	83.64	53.33	72.94	88.79
p_2 = [4,22,10,11,15,7,1,21]	824.98	83.64	60.00	75.29	86.36
p_3 = [1,4,7,10,11,15,21,22]	883.57	98.18	76.67	90.59	92.73
p_4 = [1,21,11,7,4,10,22,15]	**899.38**	**100.00**	**80.00**	**92.94**	**94.85**



(a) (b)

Fig. 6. Visualizations of MIMTS **U** for the training set under different MTMI arrangements. Green numbers in the figure indicate sensor indices. (a) Visualization under arrangement p_1. (b) Visualization under arrangement p_4.

TABLE VII

HYPERPARAMETER SETTING FOR MODELS COMPARED

Hyperparameter	**Value**
Optimizer	SGD
Loss function	Cross-entropy
Learning rate	0.0005
Momentum	0.9
Batch size	8
Epoch	200

TABLE VIII

PERFORMANCE COMPARISON WHEN ADOPTING DIFFERENT 3DCNNS AS THE CLASSIFIER FOR OUR METHOD ON THE LC DATASET

Classifier	**Sens (%)**	**Spec (%)**	**Acc (%)**	**AUC (%)**	**Tr (min)**	**Te (s)**	**P (M)**	**F (G)**
C3D	96.36	73.33	88.24	**94.85**	28.25	1.09	119.94	386.92
R3D$_{34}$	65.45	70.00	67.06	72.42	16.65	0.87	63.46	207.80
R(2+1)D$_{34}$	87.27	63.33	78.82	86.97	19.35	1.02	33.17	422.95
CSN$_{152}$	94.55	**80.00**	89.41	84.24	65.07	1.33	32.16	**107.90**
C3D-Light (ours)	**100.00**	**80.00**	**92.94**	**94.85**	**16.28**	**0.57**	**26.40**	155.68

and temporal information across multiple frames, thereby enhancing the spatial–temporal relationships in the resultant feature maps [33]. R3D, an improvement on C3D based on the ResNet network, improves performance through residual connections. R(2 + 1)D is an advanced model which combines both 2-D and 3-D convolutions. This model separates spatiotemporal convolutions into temporal and spatial components, reducing computational demand while maintaining effective spatiotemporal modeling [33]. CSN employs channel-separated convolutions to process different channel information in the input, effectively extracting spatial and temporal features to boost classification performance [26]. In our experiments, we employed the LC dataset, which was preprocessed using the SAO algorithm and arranged in permutation p_4 based on the SIR method. All models were evaluated using consistent hyperparameters, as shown in Table VII, with no fine-tuning applied to ensure a fair comparison.

Table VIII presents a comparative analysis of different 3DCNN classifiers applied to our method on the LC dataset, where the subscript indicates the number of convolution layers used in each model. It should be noted that the training and testing times reported here solely include the classification stage. The results reveal that C3D-Light achieves the best performance in all metrics except GFLOPS. Even in terms of GFLOPS, its value ranks second among all the classifiers. Especially, its perfect sensitivity (100%) highlights its effectiveness in identifying true positives. Notably, it achieves optimal results with the fewest parameters (26.40 M), indicating the performance improvement stems from the efficient network architecture rather than increased model complexity.

Second, parameter sensitivity analysis was conducted to show that the proposed C3D-Light can achieve excellent results over a wide range of hyperparameter values. Fig. 7 illustrates that the accuracy exhibits minimal variation within a wide range as the learning rate varies, underscoring the model's resilience. Please note that since the learning rate spans multiple orders of magnitude, the horizontal axis has been set to a logarithmic scale to better display the data. Fig. 7 indicates that the proposed model is not sensitive to changes in

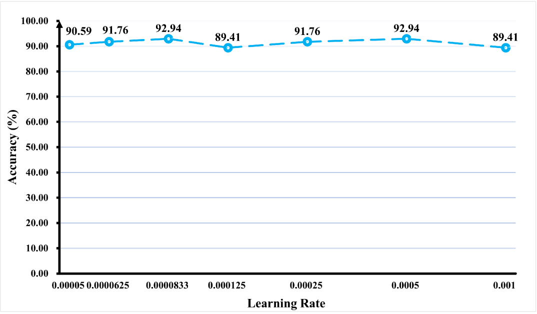


Fig. 7. Impact of varying learning rates on the performance of the C3D-Light model on the LC dataset. The learning rate varied within a wide range from 0.00005 to 0.001.

TABLE IX
PERFORMANCE OF THE C3D-LIGHT WITH DIFFERENT LOSS FUNCTIONS

Loss Function	Sens (%)	Spec (%)	Acc (%)	AUC (%)	Tr (min)	Te (s)
PolyLoss	**100.00**	73.33	90.59	93.33	17.19	0.70
Hinge Loss	**100.00**	16.67	67.06	64.39	17.59	0.80
Focal Loss	98.18	83.33	**92.94**	93.64	17.36	0.76
Cross-entropy	**100.00**	**80.00**	**92.94**	**94.85**	**16.28**	**0.57**

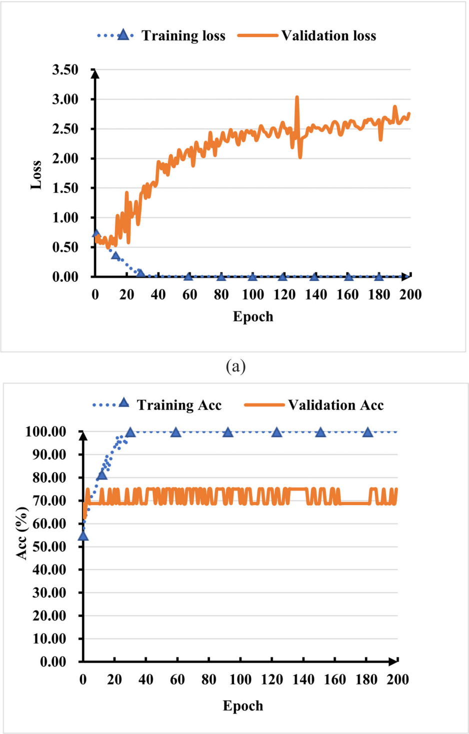
TABLE X
PERFORMANCE OF THE C3D-LIGHT WITH DIFFERENT OPTIMIZERS

Optimizer	Sens (%)	Spec (%)	Acc (%)	AUC (%)	Tr (min)	Te (s)
SGD	**100.00**	**80.00**	**92.94**	**94.85**	**16.28**	**0.57**
RMSProp	96.36	13.33	67.06	56.67	18.20	0.92
AdaGrad	**100.00**	60.00	85.89	92.73	18.19	0.92
AdamW	96.36	53.33	81.76	82.12	18.01	0.85

learning rate. Similar conclusions can be obtained for other key hyperparameters such as Momentum. Table IX compares the performance of our model using different loss functions [34]. SGD was used as the default optimizer for these experiments. From Table IX we can see that cross-entropy achieved the best results on all metrics. Accordingly, we prefer cross-entropy as the loss function for our model. Table X compares the performance of the model using different optimizers [35]. Cross-entropy was used as the default loss function for these experiments. From Table X, we can see that the SGD algorithm achieved the best results on all metrics. Thus, we prefer SGD as the optimizer for our model.

Third, we presented the loss and accuracy curves to visually demonstrate our model's performance. Fig. 8 shows the training and validation losses (a), as well as the training and validation accuracies (b).

It can be seen from Fig. 8 that the model training converges quickly, around epoch 30, based on the stabilization of both training and validation accuracies. From epoch 31 onward, the training accuracy remains consistently at 1.0, indicating a high level of stability in the model's performance on the training data. The validation accuracy, while fluctuating between 68.75% and 75.00%, also shows a degree of stability after epoch 30, as it does not exhibit large swings or continuous declines. Specifically, the training loss consistently decreased as the number of epochs increased, while the training accuracy correspondingly increased until it reached 1.00. This pattern aligned with our expected outcomes of gradient descent optimization. However, the validation loss and accuracy did not follow this trend. The validation loss reached its minimum at the ninth epoch, whereas after the ninth epoch, the validation accuracy fluctuated. There's a large gap between training and validation metrics. This suggested the onset of overfitting, where beyond the ninth epoch, the model became overly specialized to the training data and failed to generalize to new data. To mitigate the likelihood of overfitting, we carefully chose suitable hyperparameters for our model (see Table VII). In addition, techniques such as dropout were incorporated into the training process. As a result, the performance of our method can still surpass other SOTA methods.



(a)

(b)

Fig. 8. Loss and accuracy curves for C3D-Light over 200 epochs. (a) Training and validation losses. (b) Training and validation accuracies.

Finally, we illustrate the time-series-to-image conversion and visualize the class activation map (CAM) of C3D-Light in Fig. 9. Fig. 9(a) intuitively displays the positional relationship between the data points of the time series and the pixels in MTMI. The MTMI preserves the temporal relationship of the original sensor signals; specifically, the data points from the initial transient phase are placed in the top-left corner of the image. Consequently, the top-left corner of the image corresponds to the initial transient phase. Note that although Fig. 9(a) only illustrates the method of converting UTS into images, it also applies to MTS. MTMI will convert each UTS in MTS into an image, so one MTS will generate multiple MTMIs with the same resolution, which will be input into C3D-Light to generate a CAM. As depicted in Fig. 9(b), we plot the CAM of C3D-Light to illustrate the model's

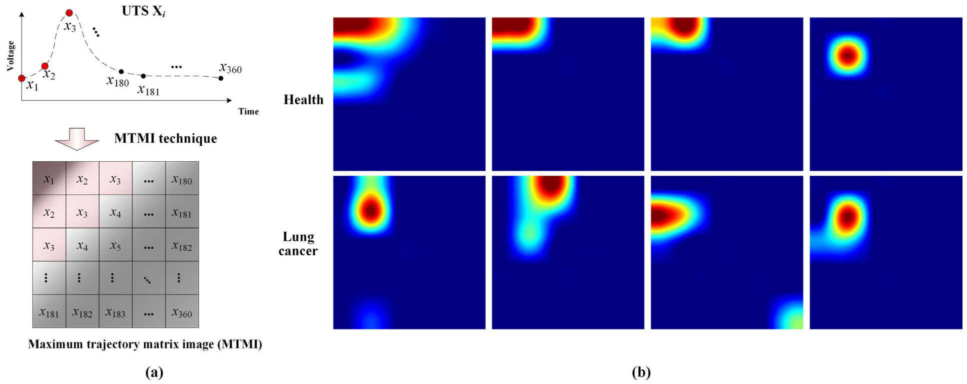

UTS X_i
Voltage
x_1
x_2
x_3
x_{180}
x_{181}
x_{360}
Time
MTMI technique
x_1 x_2 x_3 ... x_{180}
x_2 x_3 x_4 ... x_{181}
x_3 x_4 x_5 ... x_{182}
x_{181} x_{182} x_{183} ... x_{360}
Maximum trajectory matrix image (MTMI)
(a)
Health
Lung cancer
(b)

Fig. 9. (a) Illustration of the conversion of time series to images using the MTMI technique. For an UTS $\mathbf{X_i} = [x_1, x_2, \ldots, x_d]^T$ (where $d = 360$) sampled by Sensor i, our MTMI technique converts it into a 181 × 180 image. The data points from the initial transient phase, marked in red (x_1, x_2, x_3), are placed in the top-left corner of the image. Consequently, the top-left corner of the image corresponds to the initial transient phase. (b) CAMs of C3D-Light, with the first row showing samples from a healthy population and the second-row representing LC patients. The pixels in the CAM correspond to the pixels at the same position in the MTMIs; therefore, the top-left corner of the CAM also corresponds to the initial transient phase of the time series.

TABLE XI
COMPARISON OF CLASSIFICATION PERFORMANCE AMONG VARIOUS MTSC METHODS. METHODS ABOVE THE DASHED LINE ARE TRADITIONAL METHODS, WHILE THOSE BELOW ARE DEEP-LEARNING METHODS

Methods	**Sens (%)**	**Spec (%)**	**Acc (%)**	**AUC (%)**	**Pre (%)**	**F1 (%)**	**Tr (min)**	**Te (s)**
DTW	72.73	33.33	58.82	53.03	66.67	69.57	0.20	5.54
TSF	90.91	50.00	76.47	72.73	76.92	83.33	0.04	1.25
CIF	81.82	50.00	70.59	80.30	75.00	78.26	27.77	6.91
DrCIF	81.82	50.00	70.59	85.45	75.00	78.26	33.65	5.81
ROCKET	**100.00**	66.67	88.24	83.33	84.62	91.67	1.50	33.02
HIVE-COTE2.0	**100.00**	53.33	83.53	89.39	79.78	88.73	282.97	94.72
FreshPRINCE	90.91	56.67	78.82	68.94	79.49	84.78	1.04	14.16
ResNet	94.55	60.00	82.35	77.27	81.43	87.37	0.04	6.16×10^{-3}
FCN	94.55	66.67	84.71	80.61	83.80	88.75	**0.02**	2.46×10^{-3}
InceptionTime	94.55	63.33	83.53	78.94	82.64	88.11	0.04	8.73×10^{-3}
TapNet	80.00	73.33	77.65	88.18	85.96	82.33	124.78	1.30
TST	78.18	70.00	75.29	74.09	82.67	80.02	0.23	1.57×10^{-2}
FormerTime	87.27	60.00	77.65	73.64	77.53	78.59	0.55	1.99×10^{-2}
TodyNet	84.00	50.00	71.25	67.00	70.49	71.50	21.74	2.49×10^{-2}
SIR-3DCNN (ours)	**100.00**	**80.00**	**92.94**	**94.85**	**90.26**	**94.86**	17.70	0.98

focus areas that impact its predictions, offering insights into its interpretability. Note that the pixels in the CAM correspond to the pixels at the same position in the MTMIs. Therefore, the CAM visualization reveals that the model predominantly focuses on the upper-left region of the MTMI when making predictions, which corresponds to the initial transient phase of the time series, known for its richness in transient features. This observation indicates that transient features in electronic nose signals are crucial for classification, corroborating findings from [36] and [37].

G. Comparison of SOTA Methods

This section evaluates the proposed SIR-3DCNN performance against leading MTSC methods. These methods were selected for their exemplary performance across various datasets and represent a wide array of cutting-edge techniques recently introduced for MTSC. Among them, DTW [20], TSF [6], CIF [11], DrCIF [7], ROCKET [18], HIVE-COTE 2.0 [7], and FreshPRINCE [8] employ traditional techniques, whereas ResNet [12], FCN [12], InceptionTime [14], TapNet [13], TST [15], FormerTime [16], and TodyNet [17] are recent deep-learning methods. It's worth noting that DTW, a traditional method, is specifically used for computing similarity between time series, with classification based on this similarity. Traditional methods like TSF, CIF, DrCIF, ROCKET, HIVE-COTE 2.0, and FreshPRINCE depend on feature engineering, requiring the transformation of time series data into a set of features that encompass various statistical, trend, and other forms of features. TSF extracts simple features such as mean, standard deviation, and slope. Alongside these simple features, CIF also extracts the Catch22 features [38]. DrCIF extends CIF by incorporating features from periodograms and first-order differences. ROCKET is a kernel-based method that generates a large number of random convolutional features. HIVE-COTE 2.0 is an ensemble that combines multiple feature-based classifiers, including the Shapelet Transform, bag-of-words SFA, and random interval spectral ensemble (RISE). Notably, FreshPRINCE employs the entire set of TSFresh features for classification. In addition, we considered the diversity of model architectures during selection, ensuring the chosen methods cover various architectures, including Shapelet Transforms (CIF), tree-based classifiers (DrCIF), kernel-based methods (ROCKET), meta-ensembles (HIVE-COTE 2.0), CNNs (TapNet), and more. Consequently, this comparison offers valuable insights into the relative strengths and tradeoffs of each technique.

Table XI compares the classification performance of various MTSC methods. Our proposed SIR-3DCNN framework demonstrates comprehensive superiority, achieving perfect sensitivity (100%) alongside the highest scores in specificity (80%), accuracy (92.94%), AUC (94.85%), precision (90.26%), and F1-score (94.86%) across all comparative methods. While the training time of SIR-3DCNN remains moderate at 17.70 min, its testing time proves exceptionally efficient at under one second—a critical advantage for clinical deployment. This pattern of dominance across all primary classification metrics, coupled with real-time inference capabilities, establishes SIR-3DCNN as the effective and efficient solution for electronic nose-based LC detection, outperforming both conventional machine learning approaches and contemporary DL architectures.

V. Conclusion

This article proposes SIR-3DCNN, a novel and advanced framework for MTSC aimed at LC detection using electronic nose data. A key feature of SIR-3DCNN is the implementation of the LDA-based SAO algorithm to the sensor array, effectively reducing the number of sensors from 22 to 8. This reduction not only decreases processing time but also improves classification performance. In addition, we propose an SIR method that transforms the optimized MTS into images and determines their optimal arrangement by maximizing the SIFMI, resulting in a substantial increase in classification accuracy. Furthermore, we introduce C3D-Light, a more efficient model that outperforms traditional 3DCNN models across various metrics. In comparative assessments with SOTA methods, SIR-3DCNN consistently demonstrates superior performance on the LC dataset, leading in sensitivity, specificity, accuracy, and AUC. The framework also exhibits relatively low testing times (less than one second), highlighting its practical applicability in real-time LC detection scenarios.

As for the clinical implementation of SIR-3DCNN, its high performance (e.g., Sens 100%, Spec 80%) suggests its potential as a reliable adjunct tool for LC screening in clinical settings. By leveraging noninvasive breath analysis, this framework could reduce reliance on costly and invasive procedures such as biopsies or CT scans, particularly in resource-limited clinical environments. The lightweight design of C3D-Light further supports deployment on portable e-nose devices, enabling point-of-care diagnostics. However, several challenges must be addressed before widespread clinical adoption. First, patient variability—such as differences in comorbidities or metabolic profiles—may affect sensor responses, necessitating validation across diverse patient populations. Second, maintaining sensor array consistency (e.g., addressing calibration drift and clinical environmental interference) is critical for reproducible results. Third, to facilitate research on early diagnosis of LC, more confirmed early-stage LC cases should be collected. Finally, the integration with existing hospital IT systems and the alignment of screening results with those from other clinical diagnostic approaches should be proactively addressed to ensure practical clinical implementation.

Note that the data collection in this study strictly adhered to the requirements outlined in the Declaration of Helsinki and the protocol approved by the IRB of Chongqing University Cancer Hospital. All data were fully anonymized before analysis to protect patient privacy. While this study demonstrates the potential of our proposed method for this binary classification (LC versus healthy individuals), several limitations related to the medical data must be acknowledged. The dataset comprises only 169 cases, which is relatively small compared to datasets in other domains. This constraint is common in medical research, as obtaining large-scale, biopsy-confirmed medical data is often limited by ethical, logistical, and financial challenges. In addition, the dataset predominantly includes only two early-stage cases, which may limit the model's applicability to early-stage malignancies. Furthermore, while the demographic heterogeneity of the dataset (detailed in Table II) enhances the representativeness of our sample within the LC population, the small number of cases may not fully capture the diversity of the broader population, a limitation we aim to address in future work.

As stated in Section IV-F, the patterns evident in the curves illustrated in Fig. 8 indicate that our model experienced overfitting during training on this dataset. After thorough analysis, we believe there are two primary reasons for this issue. First, the significant difference between our training and validation performance suggests a potential discrepancy between the training and validation datasets. Our model attains nearly perfect accuracy on the training set, yet the validation accuracy oscillates between 68% and 75%. Second, the small size of our training set may have also contributed to the overfitting problem. While we have already implemented specific strategies to alleviate overfitting (as detailed in Section IV-F), further exploration of more effective methods to prevent overfitting is still necessary. In future research, we intend to adopt several strategies to address the overfitting issue and enhance the model's performance. First, we will investigate simpler model architectures that are more appropriate for the size and complexity of our dataset. Furthermore, we will explore data augmentation techniques to enhance the diversity of our training data. For instance, generative adversarial networks (GANs) and diffusion models [39] are widely used methods that can generate new training samples synthetically, thereby expanding the dataset and exposing the model to a broader range of variations.

Although our study with the LC dataset shows promising results, it is limited by its focus on a single dataset. Future research should evaluate the performance of SIR-3DCNN across diverse time series datasets to establish its broader applicability. Further exploration of the OPS could also provide deeper insights into time series analysis for medical diagnostics.

References

[1] H. Sung et al., "Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries," *CA, Cancer J. Clinicians*, vol. 71, no. 3, pp. 209–249, Feb. 2021.

[2] D. Mathios et al., “Detection and characterization of lung cancer using cell-free DNA fragmentomes,” *Nature Commun.*, vol. 12, no. 1, p. 5060, Aug. 2021.

[3] Y. Adiguzel and H. Kulah, “Breath sensors for lung cancer diagnosis,” *Biosensors Bioelectron.*, vol. 65, pp. 121–138, Mar. 2015.

[4] M. Parnas et al., “Precision detection of select human lung cancer biomarkers and cell lines using honeybee olfactory neural circuitry as a novel gas sensor,” *Biosensors Bioelectron.*, vol. 261, Oct. 2024, Art. no. 116466.

[5] A. Z. Temerdashev, E. M. Gashimova, V. A. Porkhanov, I. S. Polyakov, D. V. Perunov, and E. V. Dmitrieva, “Non-invasive lung cancer diagnostics (don’t short) through metabolites in exhaled breath: Influence of the disease variability and comorbidities,” *Metabolites*, vol. 13, no. 2, p. 203, Jan. 2023.

[6] H. Deng, G. Runger, E. Tuv, and M. Vladimir, “A time series forest for classification and feature extraction,” *Inf. Sci.*, vol. 239, pp. 142–153, Aug. 2013.

[7] M. Middlehurst, J. Large, M. Flynn, J. Lines, A. Bostrom, and A. Bagnall, “HIVE-COTE 2.0: A new meta ensemble for time series classification,” *Mach. Learn.*, vol. 110, nos. 11–12, pp. 3211–3243, Dec. 2021.

[8] M. Middlehurst and A. Bagnall, “The FreshPRINCE: A simple transformation based pipeline time series classifier,” in *Proc. Int. Conf. Pattern Recognit. Artif. Intell.*, Chengdu, China, Jan. 2022, pp. 150–161.

[9] S. Hao, Z. Wang, A. D. Alexander, J. Yuan, and W. Zhang, “MICOS: Mixed supervised contrastive learning for multivariate time series classification,” *Knowl.-Based Syst.*, vol. 260, Jan. 2023, Art. no. 110158.

[10] R. Liu et al., “MTMI-DCNN: A PSR-based method for time series sensor data classification,” *IEEE Sensors J.*, vol. 22, no. 7, pp. 6806–6817, Apr. 2022.

[11] M. Middlehurst, J. Large, and A. Bagnall, “The canonical interval forest (CIF) classifier for time series classification,” in *Proc. IEEE Int. Conf. Big Data*, Dec. 2020, pp. 188–195.

[12] Z. Wang, W. Yan, and T. Oates, “Time series classification from scratch with deep neural networks: A strong baseline,” in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 1578–1585.

[13] X. Zhang, Y. Gao, J. Lin, and C.-T. Lu, “TapNet: Multivariate time series classification with attentional prototypical network,” in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 4, pp. 6845–6852.

[14] H. Ismail Fawaz et al., “InceptionTime: Finding AlexNet for time series classification,” *Data Mining Knowl. Discovery*, vol. 34, no. 6, pp. 1936–1962, Nov. 2020.

[15] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, “A transformer-based framework for multivariate time series representation learning,” in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2021, pp. 2114–2124.

[16] M. Cheng, Q. Liu, Z. Liu, Z. Li, Y. Luo, and E. Chen, “FormerTime: Hierarchical multi-scale representations for multivariate time series classification,” in *Proc. ACM Web Conf.*, Apr. 2023, pp. 1437–1445.

[17] H. Liu et al., “TodyNet: Temporal dynamic graph neural network for multivariate time series classification,” *Inf. Sci.*, vol. 677, Aug. 2024, Art. no. 120914.

[18] A. Dempster, F. Petitjean, and G. I. Webb, “ROCKET: Exceptionally fast and accurate time series classification using random convolutional kernels,” *Data Mining Knowl. Discovery*, vol. 34, no. 5, pp. 1454–1495, Sep. 2020.

[19] N. Ilakiyaselvan, A. N. Khan, and A. Shahina, “Deep learning approach to detect seizure using reconstructed phase space images,” *J. Biomed. Res.*, vol. 34, no. 3, pp. 240–250, 2020.

[20] A. P. Ruiz, M. Flynn, J. Large, M. Middlehurst, and A. Bagnall, “The great multivariate time series classification bake off: A review and experimental evaluation of recent algorithmic advances,” *Data Mining Knowl. Discovery*, vol. 35, no. 2, pp. 401–449, Mar. 2021.

[21] H. Sun et al., “Sensor array optimization of electronic nose for detection of bacteria in wound infection,” *IEEE Trans. Ind. Electron.*, vol. 64, no. 9, pp. 7350–7358, Sep. 2017.

[22] M. A. Akbar et al., “An empirical study for PCA- and LDA-based feature reduction for gas identification,” *IEEE Sensors J.*, vol. 16, no. 14, pp. 5734–5746, Jul. 2016.

[23] Y. Zhao, Z. Li, X. Guo, and Y. Lü, “Alignment-guided temporal attention for video action recognition,” in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2022, pp. 13627–13639.

[24] U. Agrawal, V. Rohatgi, and R. Katarya, “Normalized mutual information-based equilibrium optimizer with chaotic maps for wrapper-filter feature selection,” *Expert Syst. Appl.*, vol. 207, Nov. 2022, Art. no. 118107.

[25] G. Castellano and G. Vessio, “A deep learning approach to clustering visual arts,” *Int. J. Comput. Vis.*, vol. 130, no. 11, pp. 2590–2605, Nov. 2022.

[26] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3D convolutional networks,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2015, pp. 4489–4497.

[27] S. X. Antoniou, E. Gaude, M. Ruparel, M. P. van der Schee, S. M. Janes, and R. C. Rintoul, “The potential of breath analysis to improve outcome for patients with lung cancer,” *J. Breath Res.*, vol. 13, no. 3, Apr. 2019, Art. no. 034002.

[28] F. S. Moninuola et al., “Early detection of lung cancer via breath analysis utilising electronic nose,” in *Proc. Int. Conf. Artif. Intell., Big Data, Comput. Data Commun. Syst. (icABCD)*, Durban, South Africa, Aug. 2023, pp. 1–6.

[29] B. Liu et al., “Lung cancer detection via breath by electronic nose enhanced with a sparse group feature selection approach,” *Sens. Actuators B, Chem.*, vol. 339, Jul. 2021, Art. no. 129896.

[30] B. Liu et al., “Sparse unidirectional domain adaptation algorithm for instrumental variation correction of electronic nose applied to lung cancer detection,” *IEEE Sensors J.*, vol. 21, no. 15, pp. 17025–17039, Aug. 2021.

[31] X. Chen, L. Yi, and R. Liu, “FEDA: A nonlinear subspace projection approach for electronic nose data classification,” *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–11, 2023.

[32] A. Jovic, K. Brkic, and N. Bogunovic, “A review of feature selection methods with applications,” in *Proc. 38th Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, Opatija, Croatia, May 2015, pp. 1200–1205.

[33] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, “A closer look at spatiotemporal convolutions for action recognition,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6450–6459.

[34] Q. Wang, Y. Ma, K. Zhao, and Y. Tian, “A comprehensive survey of loss functions in machine learning,” *Ann. Data Sci.*, vol. 9, no. 2, pp. 187–212, Apr. 2022.

[35] R. Abdulkadirov, P. Lyakhov, and N. Nagornov, “Survey of optimization algorithms in modern neural networks,” *Mathematics*, vol. 11, no. 11, p. 2466, May 2023.

[36] N. Nimsuk, “Improvement of accuracy in beer classification using transient features for electronic nose technology,” *J. Food Meas. Characterization*, vol. 13, no. 1, pp. 656–662, Mar. 2019.

[37] A. U. Rehman, S. B. Belhaouari, M. Ijaz, A. Bermak, and M. Hamdi, “Multi-classifier tree with transient features for drift compensation in electronic nose,” *IEEE Sensors J.*, vol. 21, no. 5, pp. 6564–6574, Mar. 2021.

[38] M. Middlehurst, P. Schäfer, and A. Bagnall, “Bake off redux: A review and experimental evaluation of recent time series classification algorithms,” *Data Mining Knowl. Discovery*, vol. 38, no. 4, pp. 1958–2031, Jul. 2024.

[39] A. Kazerouni et al., “Diffusion models in medical imaging: A comprehensive survey,” *Med. Image Anal.*, vol. 88, Aug. 2023, Art. no. 102846.