applied sciences

MDPI

Article

An Improved Algorithm of Drift Compensation for Olfactory Sensors

Siyu Lu [1], Jialiang Guo [1], Shan Liu [1], Bo Yang [1], Mingzhe Liu [1,2,*], Lirong Yin [3] and Wenfeng Zheng [1,*]

[1] School of Automation, University of Electronic Science and Technology of China, Chengdu 610054, China
[2] School of Data Science and Artificial Intelligence, Wenzhou University of Technology, Wenzhou 325000, China
[3] Department of Geography and Anthropology, Louisiana State University, Baton Rouge, LA 70803, USA
[*] Correspondence: liumz@cdut.edu.cn (M.L.); winfirms@uestc.edu.cn (W.Z.)

Abstract: This research mainly studies the semi-supervised learning algorithm of different domain data in machine olfaction, also known as sensor drift compensation algorithm. Usually for this kind of problem, it is difficult to obtain better recognition results by directly using the semi-supervised learning algorithm. For this reason, we propose a domain transformation semi-supervised weighted kernel extreme learning machine (DTSWKELM) algorithm, which converts the data through the domain and uses SWKELM algorithmic classification to transform the semi-supervised classification problem of different domain data into a semi-supervised classification problem of the same domain data.

Keywords: semi-supervised learning; extreme learning machine; sensor drift compensation

check for updates

Citation: Lu, S.; Guo, J.; Liu, S.; Yang, B.; Liu, M.; Yin, L.; Zheng, W. An Improved Algorithm of Drift Compensation for Olfactory Sensors. *Appl. Sci.* **2022**, *12*, 9529. https://doi.org/10.3390/app12199529

Academic Editor: Kourosh Kalantar-Zadeh

Received: 30 August 2022
Accepted: 20 September 2022
Published: 22 September 2022

1. Introduction

Machine olfaction is widely used in gas classification and calibration of accurate concentration estimation. For example, in terms of food safety, it is used to detect the purity and quality of food [1–3]. In terms of environmental protection, it is used for a wide range of air quality monitoring [4,5]. In medical cases, it is used to detect diseases [6]. The research on sensor drift compensation can effectively improve detection accuracy.

In the face of sensor drift, in order to avoid tedious calibration tasks and save costs, many researchers have studied drift compensation algorithms for many years, proposing different solutions [7–12]. The most important methods can be divided into three types: the first is the component correction method; the second is the adaptive method; the third is the machine learning method.

For example, Wold S et al. proposed an orthogonal signal correction method (orthogonal signal correction, OSC). This method makes the corrected signal retain as much useful information as possible by removing the linearly irrelevant part of the domain target matrix in the original signal [13]. Feng et al. used the OSC method to preprocess the data, and then optimized the RBF network through the particle swarm optimization algorithm. In order to detect wound infection, good results were obtained [14]. Artursson et al. proposed a component correction principal components analysis (CCPCA) algorithm based on the OSC algorithm [15]. This algorithm first assumes that the drift has a preferred direction in the measurement space rather than a random distribution and finds the direction of the drift through the method of principal component analysis (PCA). The drift direction is removed by the measurement matrix, and irrelevant information in the data is removed, thereby increasing the stability and increasing the generalization ability of the classification model.

The adaptive method is a passive drift compensation method that matches the trained model to the current sensor output by modifying the parameters in the classification algorithm [16,17]. There are two main methods: adaptive resonance theory (ART) and self-organizing feature mapping (SOM). Distante C. et al. combined adaptive resonance theory with neural networks for gas identification. For overlapping clusters and non-overlapping clusters, different approaches to the drift problem have been proposed [18]. Distante C. et al.

proposed a novel mSom neural network approach to improve gas classification for multi-sensor systems. The algorithm adapts to changes in the data distribution brought about by the drift problem by repeating the self-training process by using multiple self-organizing graphs that approximate the statistical distribution of a single odor set [19].

Machine learning methods can also be used to address sensor drift [17,20,21]. This approach automatically adapts the classification to drift, thereby reducing the impact of drift problems, rather than explicitly describing or computing drift. Vergara A. et al. proposed an ensemble method based on support vector machines [22], a method of weighted summation of classifiers trained at different time points. In order to reduce the impact of cost and sensor drift, semi-supervised algorithms with both the advantages of supervised learning and unsupervised learning are introduced into the field of machine olfactory. Combining the domain adaptation algorithm with the domain semi-supervised algorithm, in [23], Liu et al. proposed a method to construct a classifier using a weighted geodesic flow kernel (GFK) combined manifold regularization. Combined kernels are defined on multiple curves between source domain data and target domain data using unlabeled data.

Domain adaptation has become a better method to solve sensor drift. Domain adaptation is a type of transfer learning that utilizes informative source domain samples to improve the performance of target domain models [12,24–26]. Its idea is to map data features from different domains (such as two different datasets) to the same feature space, so that data from other domains can be used to enhance target domain training. Zhang proposed a framework called domain adaptive extreme learning machine [27], proposing two types of domain adaptive extreme learning machines (DAELM). One is the source domain adaptive learning machine (DAELM-S) and the target domain adaptive learning machine (DAELM-T). The framework uses limited target domain labeled data and labeled source domain data to train a classifier with good generalization ability. This algorithm has achieved good results in the problem of sensor drift.

However, DAELM requires a certain amount of labeled data in the target domain, and in many cases, it is relatively difficult to obtain labeled data. In this paper, we have proposed an algorithm named DTSWKELM, which can effectively avoid such problem. In this study, we first introduce the specific implementation process of the DTSWKELM and the principle of solving the sensor drift problem. The algorithm effect test is carried out on a public dataset, and the comparison experiment and analysis with different sensor drift compensation algorithms are carried out. This algorithm transforms the semi-supervised classification problem of different domain data into a semi-supervised classification problem of the same domain data through the method of domain transformation. Compared with the DAELM algorithm, it improves the problem that requires a certain amount of labeled target domain data and the random hidden layer mapping brings instability problems.

2. Datasets

The dataset used in this study is publicly available data collected by A. Vergara et al. using sensors and published on the UCI Machine Learning Repository [22,28]. This dataset has 6 different gases and 13,910 data samples, which is very suitable for the study of related algorithms for classification. In addition, the most important point is that the data collection work is divided into different batches at different times. Due to the characteristics of the sensor, it is prone to aging, poisoning, and other factors, resulting in the drift problem of the sensor. Therefore, the collected 10 batches of data are prone to different data distributions. The algorithm DTSWKELM introduced in this study is to solve the problem of long-term drift of the sensor, so this dataset is selected to test the effect of the DTSWKELM algorithm. Table 1 shows the details of the dataset.

Table 1. Data volume for different sample gases in 10 batches.

Batch ID	Month	Acetone	Acetaldehyde	Ethanol	Ethylene	Ammonia	Toluene	Total
Batch 1	1–2	90	98	83	30	70	74	445
Batch 2	3–10	164	334	100	109	532	5	1244
Batch 3	11–13	365	490	216	240	275	0	1586
Batch 4	14,15	64	43	12	30	12	0	161
Batch 5	16	28	40	20	46	63	0	197
Batch 6	17–20	514	574	110	29	606	467	2300
Batch 7	21	649	662	360	744	630	568	3613
Batch 8	22,23	30	30	40	33	143	18	294
Batch 9	24,30	61	55	100	75	78	101	470
Batch10	36	600	600	600	600	600	600	3600

3. Materials and Methods

3.1. Maximum Average Discrepancy

Maximum mean discrepancy (MMD) is a very efficient measure of the distance between two distributions. We mainly use it to compare the difference of target domain data and source domain data, and then find a new domain by minimizing MMD. The maximum mean discrepancy is based on the idea that we need to identify a function that takes distinct assumptions about two different distributions. By looking for a continuous function $\mathcal{F}$ in the sample space, finding the mean of the function values of samples with different distributions on $\mathcal{F}$, and by taking the difference of the two means, the average deviation of the two distributions corresponding to $\mathcal{F}$ can be obtained. Find an $\mathcal{F}$ such that the deviation has a maximum value, and the MMD is obtained. Finally, MMD is taken as the test statistic to judge whether the two distributions are the same. If this value is small enough, the two distributions are considered the same. At the same time, this value is also used to judge the similarity between the two distributions. In transfer learning, this $\mathcal{F}$ is generally used as the RBF kernel function, and MMD can be expressed by Equation (1) [29].

$$MMD^2(\mathcal{F}, p, q) := \overset{sup}{f \in \mathcal{F}} E_{x\sim p} \| [f(x)] - E_{y\sim q}[f(y)] \| \tag{1}$$

where $\mathcal{F}$ is the desired function, x, y is the sample of two random variables, p is the distribution of x, and q is the distribution of y. If and only if $p = q$, $MMD^2(\mathcal{F}, p, q) = 0$.

For the unsupervised domain adaptation problem, two different domains are considered: the source domain S and the target domain T, whose probability distributions are P_S and P_T, respectively. The source domain data $X_S = [x_1, x_2, \ldots, x_{Si}]$ and the source domain label $Y_S = [y_1, y_2, \ldots, y_{Si}]$ and the unlabeled target domain data $X_T = [x_1, x_2, \ldots, x_{Tj}]$, where N_S and N_T are the number of samples in the source domain and the number of samples in the target domain, respectively. Generally speaking, the probability distributions P_S and P_T are different. The Euclidean distance between the source domain and the target domain after a specific function $\varphi(\cdot)$ is mapped to the reproducing kernel Hilbert space (RKHS), as shown in Equation (2):

$$MMD^2(X_S, X_T) = \| \frac{1}{Ns} \sum_{i=1}^{Ns} \varphi(x_{Si}) - \frac{1}{N_T} \sum_{j=1}^{N_T} \varphi(x_{Tj}) \|_{\mathcal{H}}^{2} \tag{2}$$

3.2. Sensor Drift Compensation Algorithm

This section proposes the DTSWKELM algorithm, which transforms the source domain data and the target domain data so that the two sets of data distributions are close. The semi-supervised classification problem of different domain data is converted into a semi-supervised classification problem of the same domain, and then the semi-supervised classification task is carried out through the SWKELM algorithm. The algorithm has the advantages of a good classification effect, strong generalization ability, and no need for labeled target domain data.

In the dataset, labeled source domain data are accessible, so only unlabeled target domain data and labeled source domain data can be used to build data reconstruction models. Through data transformation, it is desirable to keep the source domain data unchanged as much as possible, while making the distribution of the drifting target data close to the distribution of the source data. Figure 1 is a flow chart of the DTSWKELM algorithm:

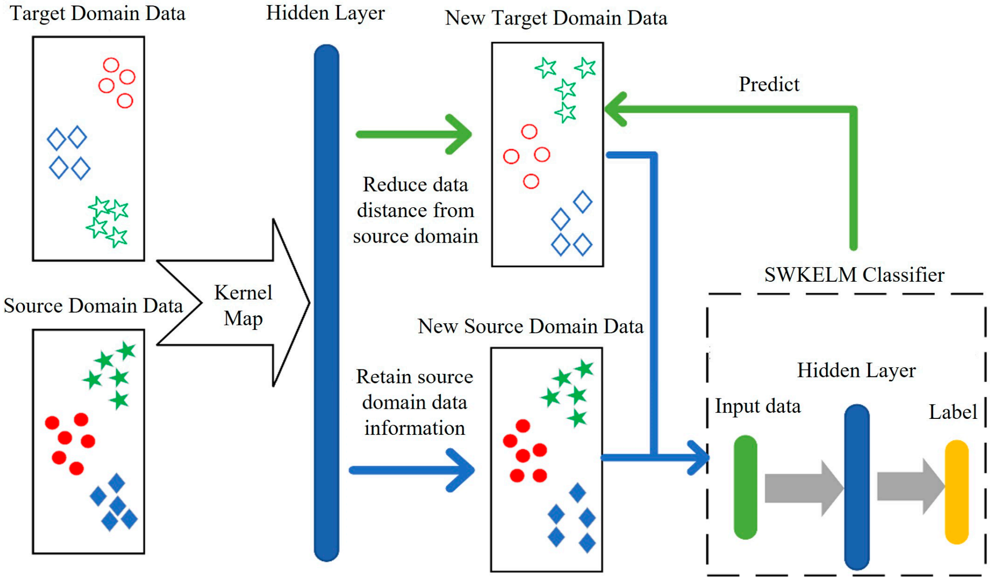

Figure 1. DTSWKELM algorithm flow chart.

As can be seen from Figure 1, the source domain data and the target domain data are obtained through kernel mapping to obtain the hidden layer. Under the constraints of two conditions, two sets of new domain data are obtained, which are sent to the SWKELM classifier for model training, and finally the new target domain data are predicted. The calculation flow of the specific algorithm is given below. The algorithm in this paper can be defined as the following optimization problem, as shown in Equation (3):

$$\underset{f\epsilon\mathcal{F}}{min}\, DD(\varnothing(X_S),\varnothing(X_T)) + L(X_S,\varnothing(X_S)) + \|f\|_H^2 \tag{3}$$

The first term in the formula is used to represent the distribution difference between the source domain data and the target domain data, and $\varnothing(\cdot)$ represents the correlation mapping. The second term is the loss function, which is used to prevent the loss of useful information in the source domain data in the process of data transformation. The third normal form is the regularization term used to avoid overfitting.

This paper chooses the maximum average difference to describe the distribution difference between the source domain and the target domain, as shown in Equation (4):

$$DD(\varnothing(X_S),\varnothing(X_T)) = MMD^2(X_S,X_T) \tag{4}$$

The second loss function can be expressed as Equation (5):

$$L(X_S,\varnothing(X_S)) = \sum_{i=1}^{N_S} \|\varnothing(x_{Si}) - x_{si}\|^2 \tag{5}$$

The domain transformation algorithm can be defined as Equation (6):

$$\begin{array}{c} \underset{\beta}{min}\frac{\lambda}{2}MMD^2(X_S,X_T)+\frac{C}{2}\sum_{i=1}^{N_S}\xi_{Si}{}^2+\frac{1}{2}\|\beta\|^2 \\ \text{s.t. } h(x_{Si})\beta = x_{Si}^T-\xi_{Si}^T,\ i=1,\cdots,N_S \\ h(x_{Si})\beta=\varnothing(x_{Si}),\ i=1,\cdots,N_S \\ h(x_{Ti})\beta=\varnothing(x_{Ti}),\ j=1,\cdots,N_T \end{array} \tag{6}$$

where β represents the output layer matrix, C and λ are the trade-off parameters for adjusting the model, and $h(x_{Si})$ is the $i-$ th point obtained by the source data through a single hidden layer neuron. $h(x_{Ti})$ is the $i-$ th point obtained by the source data through a single hidden layer neuron. x_{Si}^T represents the transpose of the source domain data samples, N_S and N_T are the number of source domain samples and the number of target domain samples, respectively.

Transforming the constrained optimization problem in Equation (6) into an unconstrained optimization problem, Equation (7) can be obtained

$$\underset{\beta}{min}\frac{\lambda}{2}MMD^2(X_S,X_T)+\frac{1}{2}Tr\Big[(X_S-H_S\beta)^T\Lambda_C(X_S-H_S\beta)\Big]+\frac{1}{2}\|\beta\|^2 \tag{7}$$

In the formula Λ_C is a diagonal matrix, the element on the main diagonal is the parameter C, and Tr represents the trace of a matrix.

Domain adaptation algorithms that minimize reconstruction error are different from traditional autoencoders or Boltzmann machines, which use backpropagation to update parameters to learn the weights of the input and output layers. Domain adaptation algorithms focus on the mapping of source and target domains to new domains rather than feature extraction.

Equation (4) can be calculated from Equation (8):

$$MMD^2(X_S,X_T)=\|\frac{1}{Ns}\sum_{i=1}^{Ns}\varnothing(x_{Si})-\frac{1}{N_T}\sum_{j=1}^{N_T}\varnothing(x_{Tj})\|_{\mathcal{H}}^{2} \tag{8}$$

Defining $H=\begin{bmatrix}H_S\\H_T\end{bmatrix}$, Equation (8) can be rewritten as Equation (9):

$$MMD^2(X_S,X_T)=Tr\Big[\beta^TH^TDH\beta\Big] \tag{9}$$

where $D\in R^{(N_S+N_T)\times(N_S+N_T)}$ is the matrix of MMD, which can be defined in the form of Equation (10):

$$D_{ij}=\begin{cases}\frac{1}{N_T{}^2} & if\ i,j>N_S\\ \frac{1}{N_S{}^2} & if\ i,j\le N_S\\ -\frac{1}{N_SN_T} & others\end{cases} \tag{10}$$

To sum up, Equation (7) can be rewritten as Equation (11):

$$\underset{\beta}{min}\frac{\lambda}{2}Tr\Big[\beta^TH^TDH\beta\Big]+\frac{1}{2}Tr\Big[(X_S-H_S\beta)^T\Lambda_C(X_S-H_S\beta)\Big]+\frac{1}{2}\|\beta\|^2 \tag{11}$$

For convenience, the source domain data and the target domain data are combined as X^*. The first N_S is the source domain data X_S, and the last N_T is 0. $\Lambda=dig(C,C,\cdots,C,0,\cdots,0)$, where the number of C is the number of source domain data, and the number of 0 is the number of target domain data. In this way, Equation (11) can be transformed into Equation (12):

$$\underset{\beta}{min}\frac{\lambda}{2}Tr\Big[\beta^TH^TDH\beta\Big]+\frac{1}{2}Tr\Big[(X^*-H\beta)^T\Lambda(X^*-H\beta)\Big]+\frac{1}{2}\|\beta\|^2 \tag{12}$$

Obviously, Equation (12) is a convex optimization problem, finding its gradient and letting it equal to 0, we can get:

$$\lambda H^T D H \beta + H^T \Lambda H \beta - H^T \Lambda X^* + \beta = 0 \tag{13}$$

Finally, we get the output layer matrix, as shown in Equation (14):

$$\beta = H^T \left(I_{N_S+N_T} + (\Lambda + \lambda D) H H^T \right)^{-1} \Lambda X^* \tag{14}$$

Because the input data are mapped to the hidden layer using the kernel function, the mapped data H cannot be directly and explicitly obtained, and the output layer matrix β cannot be directly calculated. However, since the kernel matrix $K = HH^T$, it can be directly calculated. The data after domain transformation, as shown in Equations (15) and (16):

$$X_{IS} = K(X_S, X) \left((\Lambda + \lambda D) K + I_{N_S+N_T} \right)^{-1} \Lambda X^* \tag{15}$$

$$X_{IT} = K(X_T, X) \left((\Lambda + \lambda D) K + I_{N_S+N_T} \right)^{-1} \Lambda X^* \tag{16}$$

where $X = \left[X_S^T, X_T^T\right]^T$ is the combination of source domain data and target domain data. The obtained new source domain data and target domain data are sent as input into the SWKELM model, the semi-supervised classifier is trained, and the trained classifier is used to predict the X_IT data, and finally the accuracy is calculated.

4. Results

The algorithm DTSWKELM introduced in this paper is to solve the problem of long-term drift of the sensor, so this dataset is selected to test the effect of the DTSWKELM algorithm. This part can be mainly divided into three experimental analyses. The first comparative experiment is the analysis of data distribution, comparing the distribution before and after data conversion, and observing its changes. The second experiment is to compare the recognition effects of different algorithms in the dataset. The third experiment is to analyze the hyperparameters. The whole experiment was carried out in a Window10 system, and Pycharm2020.1.3 was selected as the platform for algorithm implementation.

4.1. Experimental Data Distribution Analysis

The difference in the data distribution of different data is the best manifestation of the sensor drift problem. In order to more intuitively reflect the different distributions of different batches of data, the PCA method is used to reduce the dimensionality of the data. This part reduces the data to two dimensions and presents the dimensionality-reduced data points by means of a dot plot. Figure 2 is a dot plot of all batches of data in the dataset after PCA dimensionality reduction:

It can clearly be seen from Figure 2 that the uneven distribution of data in different batches of datasets is caused by sensor drift; especially when comparing Batch1 with Batch4, Batch5, Batch8, and Batch9, it is found that this situation is more obvious. In addition, we also found that there is no certain law in the change of distribution. It can be seen that the sensor drift is random, not in a fixed direction. It is for these reasons that the classification model that has been trained on one dataset often has a poor recognition effect on the new dataset.

The main idea of the DTSWKELM algorithm proposed in this study is to find a new domain for mapping between the source domain data and the target domain data by calculating the MMD. Make the new source domain data more similar to the target domain data and then perform model training and recognition on the new data. Figure 3 shows the distribution of different batches after domain transformation, where Batch1 is selected as the source domain data, and other batch data are respectively used as the target domain data.

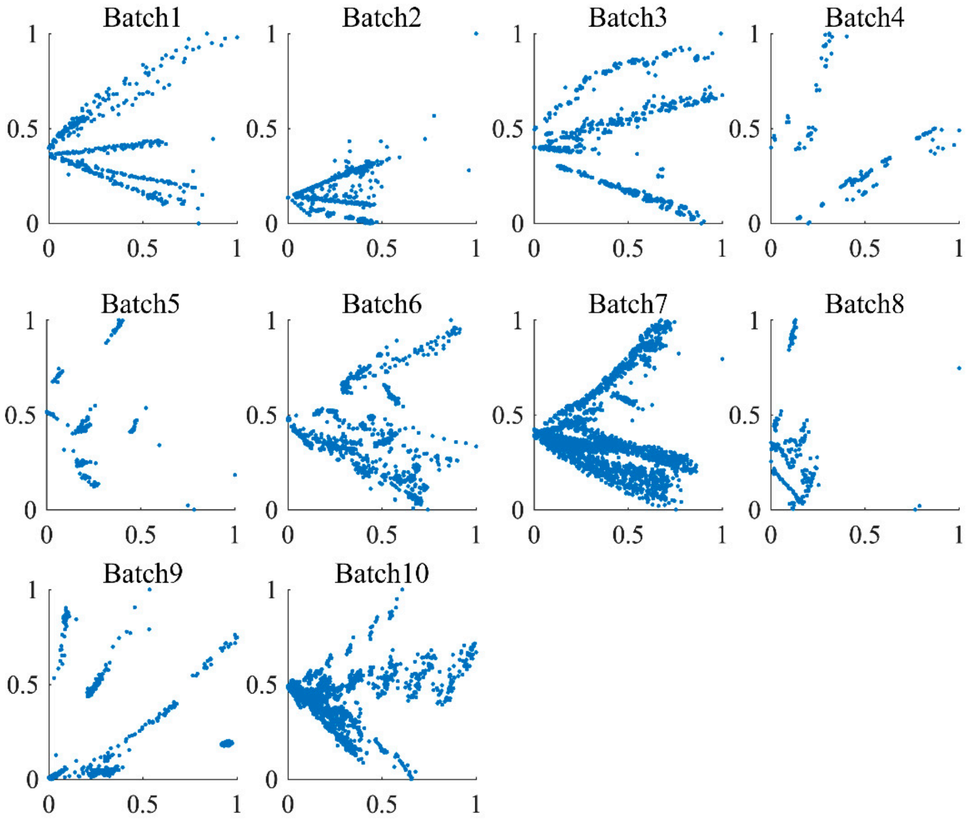

Figure 2. Dot diagram of 10 batches of data after PCA dimensionality reduction.

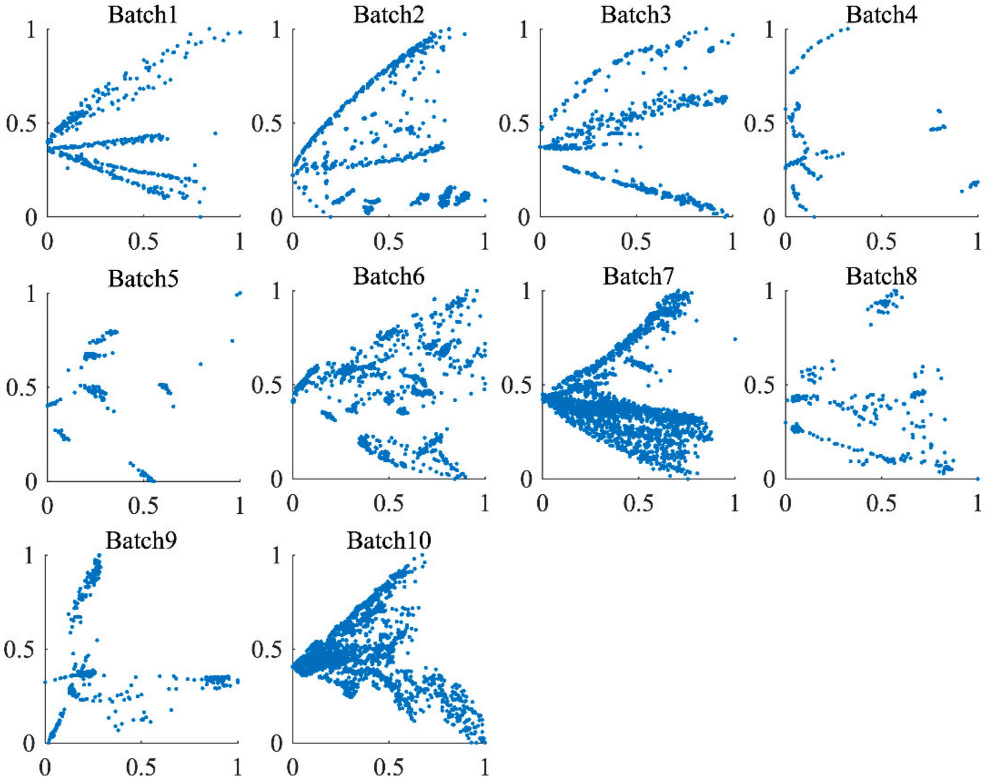

Figure 3. Dot diagram of 10 batches of data after PCA dimensionality reduction after domain transformation.

In the DTSWKELM algorithm, in the process of domain conversion, the information of the source domain data is preserved as much as possible. In this experiment, Batch1 is all used as the source domain data. Therefore, its distribution has not changed much, and the above figure only shows a diagram of Batch1. From the figure, we can see that the distribution of Batch2–10 data changed significantly after domain transformation, and it is closer to the distribution of Batch1, in which Batch2 and Batch8 are more obvious. It can be seen from this that the domain transformation part of the algorithm plays a role. It can effectively reduce the distribution difference between different batches of data, so that the new source domain data are more similar to the target domain data. Thus, the semi-supervised learning problem in different domains caused by the sensor drift problem is transformed into a semi-supervised learning problem in the same domain.

4.2. Sensor Drift Algorithm Comparison Experiment

In the comparative experiment, the comparison of the recognition effect of the DTSWKELM algorithm with other algorithms on this dataset is shown to verify the effectiveness of the DTSWKELM algorithm. Two different sets of comparative experiments are set up: the first set of experiments uses Batch1 data as the source domain data, and Batch2–9 data as the target domain data. The second set of experiments uses adjacent batch data as two datasets, that is, using Batch N−1 data as the source domain data and Batch N data as the target domain data. At the same time, seven commonly used sensor drift compensation algorithms were selected for the comparison of recognition effects, namely SVM-rbf algorithm, SVM-comgfk algorithm, ML-comgfk algorithm, ELM-rbf algorithm, DAELM-S (5) algorithm, domain transfer broad learning system, SWKELM algorithm, DTBLS algorithm, and TDACNN algorithm [30,31].

4.2.1. Experiment 1

Table 2 shows the recognition effects of eight different algorithms in Experiment 1. The bold data is the highest recognition effect of each Batch. In order to better display the comparison results between different algorithms, the data in Table 2 are converted into a histogram in Figure 4.

Table 2. The recognition effect of each algorithm in Experiment 1 (%).

Task	1–>2	1–>3	1–>4	1–>5	1–>6	1–>7	1–>8	1–>9	1–>10	AVG
SVM-rbf	74.36	61.03	50.93	18.27	28.26	28.81	20.07	34.26	34.48	38.94
SVM-comgfk	74.47	70.15	59.78	75.09	73.99	54.59	55.88	**70.23**	41.85	64
ML-comgfk	80.25	74.99	**78.79**	67.41	77.82	71.68	49.96	50.79	53.79	67.28
ELM-rbf	70.63	66.44	66.83	63.45	69.73	51.23	49.76	49.83	33.5	57.93
DAELM-S(5)	72.66	75.72	61.3	86.29	53.45	59.4	31.16	66.85	44.39	61.25
DTBLS	78.67	**96.36**	74.6	85.23	83.2	**81.53**	58.67	56.19	63.1	75.28
TDACNN	**89.56**	83.83	77.64	75.63	74.36	62.08	**75.1**	60.85	50.88	72.21
SWKELM	76.13	86.88	73.29	81.82	89.73	71.88	43.57	59.78	53.31	70.71
DTSWKELM	88.26	90.66	77.01	**89.85**	**96.31**	74.29	55.78	62.77	**68.33**	**78.14**

Observing Table 1 and Figure 4, first compare the recognition effects of DTSWKELM and 6 common sensor drift compensation algorithms. Under the conditions set in Experiment 1, the DTSWKELM proposed in this paper achieves the best recognition effect in the four groups of tasks and has the highest recognition accuracy. Especially when the target domain data is Batch6 data, the recognition accuracy of DTSWKELM reaches 96.31%, which is 13.11% higher than DTBLS and 68.05% higher than SVM-rbf. Although DTSWKELM achieves the highest recognition accuracy only in Batch5, Batch6, and Batch10, it is not too different from the algorithm with the best recognition effect. In addition, from the average of the recognition accuracy in the 9 tasks, DTSWKELM has the best average recognition accuracy, so it can be seen that on the whole, DTSWKELM performs better. Next, we observe SWKELM, from which we can see that SWKELM also achieved relatively good results, and the overall average recognition accuracy is lower than that of TDACNN, DTBLS, and DTSWKELM. Moreover, it shows better results than DTBLS and TDACNN in Batch6. It can be seen that in some scenarios, traditional semi-supervised learning algorithms can also achieve better results in semi-supervised classification problems between different domain data.

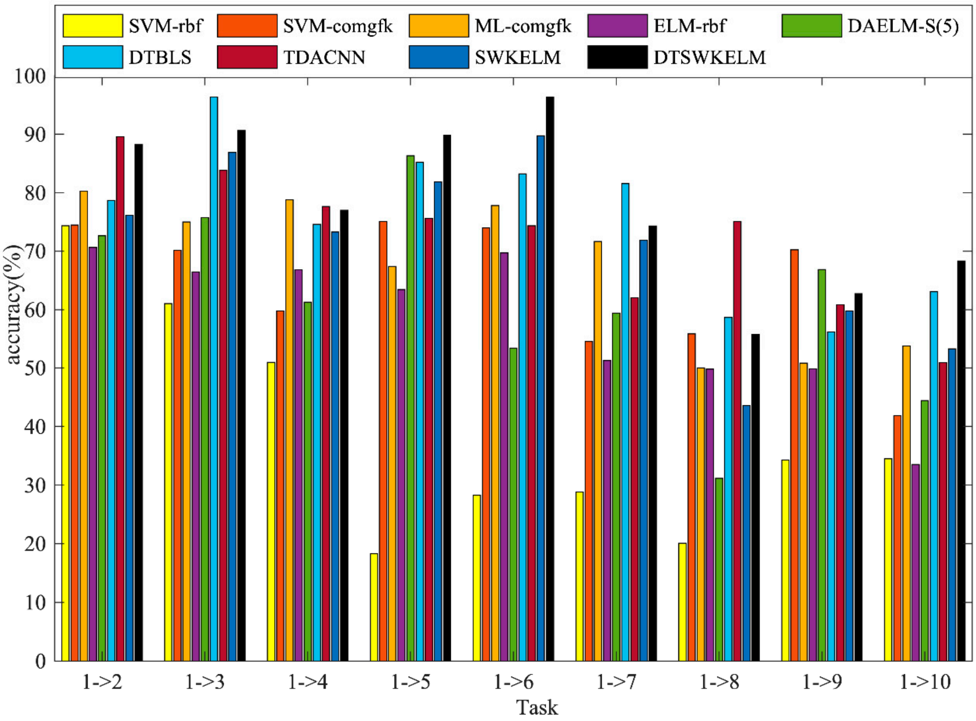

Figure 4. The histogram of the recognition effect of each algorithm in Experiment 1.

4.2.2. Experiment 2

Table 3 below is the recognition effect of eight different algorithms in Experiment 2. Bold data also represents the highest recognition effect of each Batch. In order to better display the comparison results between different algorithms, the data in Table 3 are also converted into a histogram in Figure 5.

Table 3. The recognition effect of each algorithm in Experiment 2 (%).

Task	1–>2	2–>3	3–>4	4–>5	5–>6	6–>7	7–>8	8–>9	9–>10	AVG
SVM-rbf	74.36	87.83	90.06	56.35	42.52	83.53	91.84	62.98	22.64	68.01
SVM-comgfk	74.47	73.75	78.51	64.26	69.97	77.69	82.69	85.53	17.76	69.40
ML-comgfk	80.25	**98.55**	84.89	89.85	75.53	91.17	61.22	95.53	39.56	79.62
ELM-rbf	70.63	40.44	64.16	64.37	72.7	80.75	88.2	67	22	63.36
DAELM-S(5)	72.66	69.99	72.61	79.54	52.93	87.18	91.36	56.66	29.05	68
DTBLS	78.67	97.65	79.88	67.01	75.34	90.44	**95.1**	68.09	54.47	78.52
TDACNN	**89.56**	97.46	87.58	94.68	73.9	80.18	78.43	83.19	47.64	81.48
SWKELM	76.13	90.73	90.68	93.4	73.43	83.73	87.07	92.55	41.69	81.05
DTSWKELM	88.26	97.23	**93.17**	**98.98**	**78.43**	**93.99**	93.19	**96.38**	**55.08**	**88.30**

Looking at Table 3 and Figure 5, similar conclusions can be drawn as in Experiment 1. Compared with the seven commonly used sensor drift compensation algorithms, DTSWKELM performs better overall, with an average accuracy of 88.30%, which is 6.82% higher than TDACNN. Compared with the SWKELM algorithm, it has a better recognition effect, and the average accuracy is 7.25% higher, which reflects the effectiveness of the domain conversion process. In addition, it can be seen that the recognition effect of each algorithm in different tasks in Experiment 2 is generally higher than that in Experiment 1. This is mainly because the data of adjacent batches are relatively less affected by sensor drift, and the distribution difference between the data is relatively small.

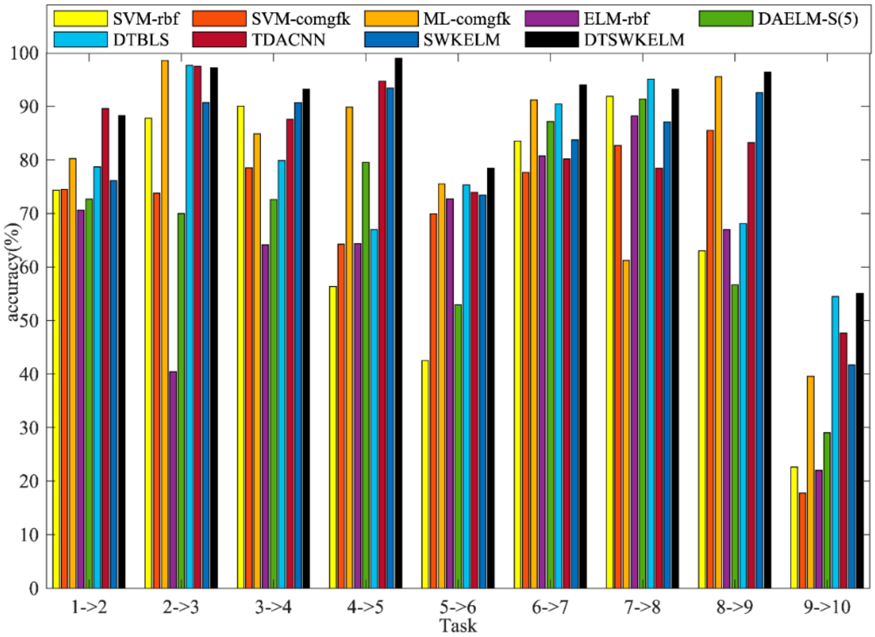

Figure 5. The histogram of the recognition effect of each algorithm in Experiment 2.

In summary, through two different sets of experiments verify the effect of the DTSWKELM algorithm, the same conclusion is obtained, and DTSWKELM shows the best recognition effect.

4.3. Parameter Influence and Analysis

In the DTSWKELM algorithm, MMD is used to describe the distance between two sets of data distributions, and popular regularization is used to correlate labeled data with unlabeled data. These two parts play a crucial role in this algorithm. In this section, the trade-off parameters λ_1 and λ_2 of these two parts in the optimization problem are analyzed and discussed. In this paper, the random search method is used to determine the optimal hyperparameters in the DTSWKELM model, and then the two hyperparameters are analyzed while other hyperparameters are fixed. Figure 6 below shows the influence of two hyperparameters on the recognition effect of the algorithm under the conditions of Experiment 1.

Figure 6a shows the influence of the trade-off parameter λ_1 of the MMD part on the recognition effect of the algorithm when other hyperparameters are fixed. Take $lg(\lambda_1) = [-4, -3, -2, -1, 0, 1, 2, 3, 4]$. As can be seen from the figure, $lg(\lambda_1)$ in $[-4, 0]$ is relatively stable in this range, and the trade-off parameter λ_1 should be selected within this range. However, when λ_1 increases, the recognition effect of the algorithm decreases. We speculate that this may be due to the fact that this part accounts for too much in the optimization problem, resulting in the loss of too much information in the domain-transformed data.

Figure 6b is the influence of the trade-off parameter λ_2 of the popular regularization part on the recognition effect of the algorithm when other hyperparameters are fixed. Similarly, taking $lg(\lambda_2) = [-4, -3, -2, -1, 0, 1, 2, 3, 4]$, the trade-off parameter λ_2 is not as stable as λ_1. However, it can be seen that $lg(\lambda_2)$ can achieve better results on $[-2, 0]$. When the trade-off parameter λ_2 is too large, it can be found that the accuracy of the algorithm is low. This is because when the manifold regularization part occupies a large proportion in the optimization problem, the useful label information will be weakened, and semi-supervised learning will degenerate into unsupervised learning, resulting in low recognition accuracy.

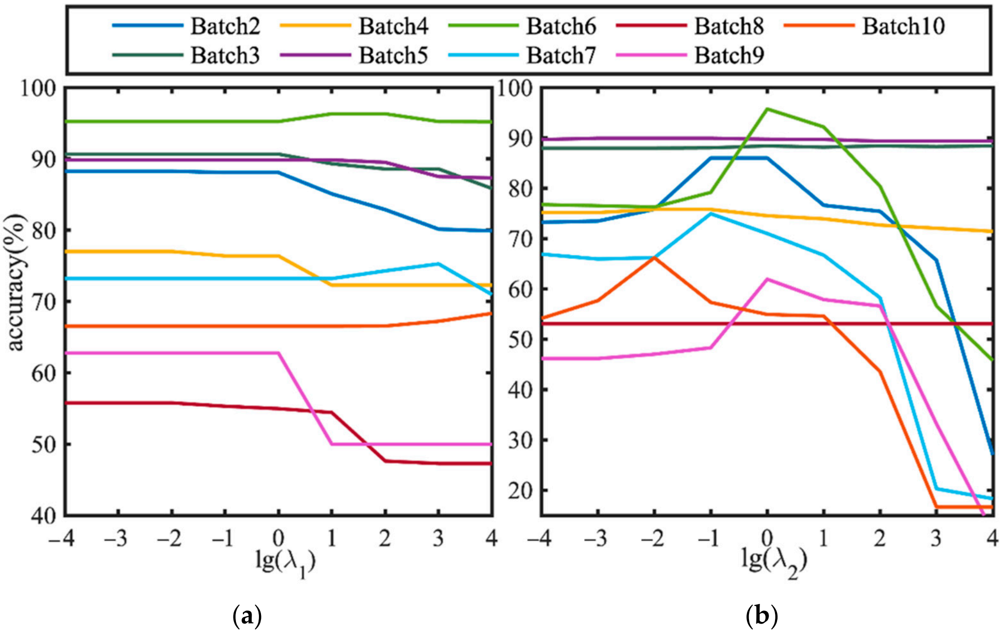

Figure 6. The impact of the trade-off parameters λ_1 and λ_2 on the algorithm recognition effect. (**a**) The influence of λ_1 on the algorithm; (**b**) the influence of λ_2 on the algorithm.

5. Discussion

The sensor drift is caused by the sensor's own material, processing method, or external environment. In this case, the semi-supervised classification problem of different domain data, that is, the sensor drift compensation problem, is studied. This paper proposes a domain shift semi-supervised weighted kernel extreme learning machine (DTSWKELM) algorithm, which defines the benchmark dataset as the source domain data and the drift dataset as the target domain data. By mapping the source domain data and the target domain data to the new domain, and finally performing semi-supervised learning on the new domain data set, the target domain data are predicted. The algorithm transforms the semi-supervised classification problem of different domain data into a semi-supervised classification problem of the same domain data through the method of domain transformation. Compared with the DAELM algorithm, the problem of requiring a certain amount of labeled target domain data and the instability problem caused by random hidden layer mapping is improved. Experiments show that the proposed algorithm can effectively compensate for the long-term sensor drift problem.

The DTSWKELM algorithm is a sensor compensation algorithm for single-source domain data. Although it has achieved good results, in some cases, there will be multiple source domains, and the algorithm cannot combine multiple source domains together. Reasonable and effective use of multiple source domain data can better learn the characteristics of the data and solve the problem of sensor drift, which is also an important problem in future research on olfactory machines.

6. Conclusions

Inspired by the DAELM algorithm, this study combines the domain transformation algorithm with the semi-supervised learning algorithm and proposes the DTSWKELM algorithm to compensate for sensor drift. First, by using MMD to represent the distance between two distributions, by minimizing MMD, a new domain is found, and the source domain data and the target domain data are mapped, thereby reducing the source domain data and the target domain data. The distribution difference between the data, the obtained new domain data is sent to the SWKELM model, the semi-supervised classifier is trained, and finally the target domain data is identified.

In the analysis stage of the experimental results, three groups of comparative experiments are mainly set up. First, the PCA method is used to compare and analyze the distribution of different batches of data before and after domain conversion, which more intuitively shows the impact of the sensor drift problem and verifies the effectiveness of the domain conversion process. Next, when testing the performance of the DTSWKELM algorithm, two experiments were set up. The first experiment is to set Batch1 as the source domain data, the data of Batch2–Batch10 are set as the target domain data and predict it. The second experiment is to set BatchN−1 as the source domain data and BatchN as the target domain data and make predictions on it. In these two groups of experiments, seven commonly used sensor drift compensation algorithms and SWKELM algorithm are used as control algorithms. Compared with other algorithms, the DTSWKELM algorithm proposed in this study has better recognition effect and can better deal with the long-term sensor drift problem. The last part is an analysis of the hyperparameter settings in the model. By setting different hyperparameters for comparative experiments, it shows the importance of hyperparameters to the model.

Author Contributions: Conceptualization, W.Z. and S.L. (Shan Liu); methodology, B.Y. and L.Y.; software, J.G.; validation, S.L. (Shan Liu); formal analysis, J.G. and L.Y.; investigation, B.Y.; resources, J.G. and S.L. (Shan Liu); data curation, J.G.; writing—original draft preparation, S.L. (Siyu Liu), M.L. and L.Y.; writing—review and editing, S.L. (Siyu Liu), M.L., S.L. (Shan Liu) and L.Y.; visualization, S.L. (Siyu Liu) and J.G.; supervision, B.Y.; project administration, W.Z.; funding acquisition, W.Z. All authors have read and agreed to the published version of the manuscript.

Funding: Support by the Sichuan Science and Technology Program, 2021YFQ0003.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are publicly available. These data can be found here: https://archive.ics.uci.edu/ml/datasets/Gas+Sensor+Array+Drift+Dataset+at+Different+Concentrations (accessed on 4 March 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wakhid, S.; Sarno, R.; Sabilla, S.I. The effect of gas concentration on detection and classification of beef and pork mixtures using E-nose. *Comput. Electron. Agric.* **2022**, *195*, 106838. [CrossRef]
2. Oates, M.J.; González-Teruel, J.D.; Ruiz-Abellon, M.C.; Guillamon-Frutos, A.; Ramos, J.A.; Torres-Sánchez, R. Using a Low-Cost Components e-Nose for Basic Detection of Different Foodstuffs. *IEEE Sens. J.* **2022**, *22*, 13872–13881. [CrossRef]
3. Huang, C.; Gu, Y. A Machine Learning Method for the Quantitative Detection of Adulterated Meat Using a MOS-Based E-Nose. *Foods* **2022**, *11*, 602. [CrossRef] [PubMed]
4. Alagoz, B.B.; Simsek, O.I.; Ari, D.; Tepljakov, A.; Petlenkov, E.; Alimohammadi, H. An Evolutionary Field Theorem: Evolutionary Field Optimization in Training of Power-Weighted Multiplicative Neurons for Nitrogen Oxides-Sensitive Electronic Nose Applications. *Sensors* **2022**, *22*, 3836. [CrossRef] [PubMed]
5. Ari, D.; Alagoz, B.B. An effective integrated genetic programming and neural network model for electronic nose calibration of air pollution monitoring application. *Neural. Comput. Applic.* **2022**, *34*, 12633–12652. [CrossRef]
6. Sarno, R.; Inoue, S.; Ardani, M.S.; Purbawa, D.P.; Sabilla, S.I.; Sungkono, K.R.; Fatichah, C.; Sunaryono, D.; Bakhtiar, A.; Prakoeswa, C.R. Detection of Infectious Respiratory Disease Through Sweat from Axillary Using an E-Nose With Stacked Deep Neural Network. *IEEE Access* **2022**, *10*, 51285–51298. [CrossRef]
7. Holmberg, M.; Artursson, T. *Drift Compensation, Standards, and Calibration Methods*; Wiley-VCH Verlag GmbH & Co. KGaA: Weinheim, Germany, 2004.
8. Schöberl, M.; Fößel, S.; Kaup, A. Fixed pattern noise column drift compensation (CDC) for digital moving picture cameras. In Proceedings of the IEEE International Conference on Image Processing, Hong Kong, 26–29 September 2010; IEEE: New York, NY, USA, 2010.
9. Ahmadou, D.; Laref, R.; Losson, E.; Siadat, M. Reduction of drift impact in gas sensor response to improve quantitative odor analysis. In Proceedings of the 2017 IEEE International Conference on Industrial Technology (ICIT), Toronto, Canada, 22–25 March 2017; IEEE: New York, NY, USA, 2017.

10. Hang, L.; Chu, R.; Jian, R.; Xia, J. Long-term drift compensation algorithms based on the kernel-orthogonal signal correction in electronic nose systems. In Proceedings of the International Conference on Fuzzy Systems & Knowledge Discovery, Zhangjiajie, China, 15–17 August 2015; IEEE: New York, NY, USA, 2016.
11. Whslén, J.; Orhan, I.; Sturm, D.; Lindh, T. Performance evaluation of time synchronization and clock drift compensation in wireless personal area networks. In Proceedings of the 7th International Conference on Body Area Networks, Oslo, Norway, 24–26 September 2012.
12. Tao, Y.; Zeng, K.; Liang, Z. Drift compensation algorithm based on Time-Wasserstein dynamic distribution alignment. In Proceedings of the 2020 IEEE/CIC International Conference on Communications in China (ICCC), Xiamen, China, 28–30 July 2020; pp. 130–135. [CrossRef]
13. Wold, S.; Antti, H.; Lindgren, F.; Öhman, J. Orthogonal signal correction of near-infrared spectra. *Chemom. Intell. Lab. Syst.* **1998**, *44*, 175–185. [CrossRef]
14. Feng, J.; Tian, F.; Jia, P.; He, Q.; Shen, Y.; Fan, S. Improving the performance of electronic nose for wound infection detection using orthogonal signal correction and particle swarm optimization. *Sens. Rev.* **2014**, *2014*, 34. [CrossRef]
15. Artursson, T.; Eklöv, T.; Lundström, I.; Mårtensson, P.; Sjöström, M.; Holmberg, M. Drift correction for gas sensors using multivariate methods. *J. Chemom. A J. Chemom. Soc.* **2000**, *14*, 711–723. [CrossRef]
16. Yan, J.; Chen, F.; Liu, T.; Zhang, Y.; Peng, X.; Yi, D.; Duan, S. Subspace alignment based on an extreme learning machine for electronic nose drift compensation. *Knowl. Based Syst.* **2022**, *235*, 107664. [CrossRef]
17. Ma, Z.; Luo, G.; Qin, K.; Wang, N.; Niu, W. Online Sensor Drift Compensation for E-Nose Systems Using Domain Adaptation and Extreme Learning Machine. *Sensors* **2018**, *18*, 742.
18. Distante, C.; Siciliano, P.; Vasanelli, L. Odor discrimination using adaptive resonance theory. *Sens. Actuators B Chem.* **2000**, *69*, 248–252. [CrossRef]
19. Zuppa, M.; Distante, C.; Siciliano, P.; Persaud, K.C. Drift counteraction with multiple self-organising maps for an electronic nose. *Sens. Actuators B: Chem.* **2004**, *98*, 305–317. [CrossRef]
20. Liang, Z.; Zhang, L.; Tian, F.; Wang, C.; Yang, L.; Guo, T.; Xiong, L. A novel WWH problem-based semi-supervised online method for sensor drift compensation in E-nose. *Sens. Actuators B Chem.* **2021**, *349*, 130727. [CrossRef]
21. Das, P.; Manna, A.; Ghoshal, S. Gas sensor drift compensation by ensemble of classifiers using extreme learning machine. In Proceedings of the International Conference on Renewable Energy Integration into Smart Grids: A Multidisciplinary Approach to Technology Modelling and Simulation (ICREISG), Bhubaneswar, India, 14–15 February 2020.
22. Vergara, A.; Vembu, S.; Ayhan, T.; Ryan, M.A.; Homer, M.L.; Huerta, R. Chemical gas sensor drift compensation using classifier ensembles. *Sens. Actuators B Chem.* **2012**, *166*, 320–329. [CrossRef]
23. Liu, Q.; Li, X.; Ye, M.; Ge, S.S.; Du, X. Drift compensation for electronic nose by semi-supervised domain adaption. *IEEE Sens. J.* **2013**, *14*, 657–665. [CrossRef]
24. Jian, Y.; Lu, K.; Deng, C.; Wen, T.; Yan, J. Drift compensation for e-nose using qpso-based domain adaptation kernel elm. In *International Symposium on Neural Networks (ISNN2018)*; Springer: Cham, Switzerland, 2018.
25. Guo, T.; Yu, K.; Cheng, X.; Bashir, A.K. Robust electronic nose in industrial cyber physical systems based on domain adaptive subspace transfer model. In Proceedings of the 2021 IEEE International Conference on Communications Workshops (ICC Workshops), Montreal, QC, Canada, 14–23 June 2021; IEEE: New York, NY, USA, 2021.
26. Liu, R.; Chen, X.; Tian, F.; Qian, J.; Wang, F.; Yi, L. MCSP-SSS: A Domain Adaptive Framework for High-Accuracy Sensor Data Classification. *IEEE Sens. J.* **2021**, *21*, 25995–26005. [CrossRef]
27. Zhang, L.; Zhang, D. Domain adaptation extreme learning machines for drift compensation in E-nose systems. *IEEE Trans. Instrum. Meas.* **2014**, *64*, 1790–1801. [CrossRef]
28. Rodriguez-Lujan, I.; Fonollosa, J.; Vergara, A.; Homer, M.; Huerta, R. On the calibration of sensor arrays for pattern recognition using the minimal number of experiments. *Chemom. Intell. Lab. Syst.* **2014**, *130*, 123–134. [CrossRef]
29. Borgwardt, K.M.; Gretton, A.; Rasch, M.J.; Kriegel, H.P.; Schölkopf, B.; Smola, A.J. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics* **2006**, *22*, e49–e57. [CrossRef]
30. Liu, B.; Zeng, X.; Tian, F.; Zhang, S.; Zhao, L. Domain transfer broad learning system for long-term drift compensation in electronic nose systems. *IEEE Access* **2019**, *7*, 143947–143959. [CrossRef]
31. Zhang, Y.; Xiang, S.; Wang, Z.; Peng, X.; Tian, Y.; Duan, S.; Yan, J. TDACNN: Target-domain-free domain adaptation convolutional neural network for drift compensation in gas sensors. *Sens. Actuators B Chem.* **2022**, *361*, 131739. [CrossRef]