

Research Article

A Deep Transfer NO_x Emission Inversion Model of Diesel Vehicles with Multisource External Influence

Zhenyi Xu ^{1,2} Ruibin Wang,^{1,3} Yu Kang ^{1,4,5,6} Yujun Zhang,⁷ Xiushan Xia,⁸ and Renjun Wang^{1,3}

¹Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei 230088, China

²School of Information Science and Technology, University of Science and Technology of China, Hefei 230026, China

³School of Computer Science and Technology, Anhui University, Hefei 230601, China

⁴Department of Automation, University of Science and Technology of China, Hefei 230026, China

⁵Institute of Advanced Technology, University of Science and Technology of China, Hefei 230088, China

⁶Key Laboratory of Technology in GeoSpatial Information Processing and Application System, Chinese Academy of Sciences, Beijing 100192, China

⁷Key Laboratory of Environmental Optics & Technology, Anhui Institute of Optics and Fine Mechanics, Chinese Academy of Sciences, Hefei 230031, China

⁸Institute of Advanced Technology, University of Science and Technology of China, Hefei 230088, China

Correspondence should be addressed to Zhenyi Xu; xuzhenyi@mail.ustc.edu.cn and Yu Kang; kangduyu@ustc.edu.cn

Received 20 August 2021; Accepted 9 December 2021; Published 30 December 2021

Academic Editor: Rakesh Mishra

Copyright © 2021 Zhenyi Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

By installing on-board diagnostics (OBD) on tested vehicles, the after-treatment exhaust emissions can be monitored in real time to construct driving cycle-based emission models, which can provide data support for the construction of dynamic emission inventories of mobile source emission. However, in actual vehicle emission detection systems, due to the equipment installation costs and differences in vehicle driving conditions, engine operating conditions, and driving behavior patterns, it is impossible to ensure that the emission monitoring data of different vehicles always follow the same distribution. The traditional machine learning emission model usually assumes that the training set and test set of emission test data are derived from the same data distribution, and a unified emission model is used for estimation of different types of vehicles, ignoring the difference in monitoring data distribution. In this study, we attempt to build a diesel vehicle NO_x emission prediction model based on the deep transfer learning framework with a few emission monitoring data. The proposed model firstly uses Spearman correlation analysis and Lasso feature selection to accomplish the selection of factors with high correlation with NO_x emission from multiple sources of external factors. Then, the stacked sparse AutoEncoder is used to map different vehicle working condition emission data into the same feature space, and then, the distribution alignment of different vehicle working condition emission data features is achieved by minimizing maximum mean discrepancy (MMD) in the feature space. Finally, we validated the proposed method with the diesel vehicle OBD data that were collected by the Hefei Environmental Protection Bureau. The comprehensive experiment results show that our method can achieve the feature distribution alignment of emission data under different vehicle working conditions and improve the prediction performance of the NO_x inversion model given a little amount of NO_x emission monitoring data.

1. Introduction

With the rapid development of China's urbanization and social economy, China's motor vehicle fleet is growing rapidly and has become the world's largest production and marketing

of motor vehicles for eleven consecutive years. At the same time, the air pollution problem caused by motor vehicle emissions is becoming increasingly serious, and it has become an important source of air pollution in large and medium-sized cities in China, as well as the important cause of fine

particulate matter and photochemical smoke pollution. As an important tool for quantitative accounting of mobile source pollution, emission inventory can be used for air pollution control measures and traceability analysis. Because the current mobile source pollution emission regulation mainly relies on the annual vehicle inspection, the single vehicle emission testing takes a long time, and the test results cannot fully reflect the actual emissions of vehicles on the road, making it difficult to realize dynamic regulation of mobile source emissions. By installing on-board diagnostics (OBD) on tested vehicles [1], the after-treatment exhaust emissions can be monitored in real time to provide data support for the construction of dynamic emission inventories of mobile source emission. However, due to security of data privacy and equipment installation costs, it is not possible to install monitoring equipment on all road-running vehicles for emission detection, while a series of problems such as human data tampering and equipment failure often leads to missing monitoring values, which greatly limits the application of OBD monitoring data in mobile source emission management. Therefore, it is significant to improve the application efficiency of OBD monitoring data through reliable analysis of features affecting emission detection and accurate prediction of missing monitoring data for mobile source emission precise regulation.

Existing emission estimation methods for mobile sources are mainly divided into average-speed based models and actual driving cycle based models. The former method usually builds a statistical regression model of pollution emissions based on the average speed of a fleet of vehicles and is usually used to estimate the macrolevel traffic pollution emissions within a specific region (administrative district or city) for a specific time period (usually a quarter or a year). The typical models are the MOBILE model developed by the US Environmental Protection Agency (US EPA) [2], the EMFAC model developed by the California Air Resources Board (CARB) [3], and the COPERT model developed by the European Commission (EC) [4]. These models obtain emission features from standard bench test cycles and characterize vehicle emission characteristics in terms of mean values such as average speed and average emission features, while ignoring the effects of actual road operating conditions, driving behavior, and vehicle dynamics on vehicle emissions. The driving cycle based emission models analyze the emissions of vehicles under different driving cycles through the complete driving process based on the multidimensional working condition characteristics data such as instantaneous speed and acceleration obtained when the vehicle is driving, which are suitable for the tasks of analyzing single-vehicle emissions or emissions calculations for a specific number of roads. The main models in this category are the IVE model [5] and the CMEM model [6] developed by the University of California, Riverside (UCR), the MOVES model [7] developed by the EPA (which can estimate emissions based on both average speed and driving cycles), and the EMIT model [8] developed by the Massachusetts Institute of Technology (MIT). Due to the lack of basic vehicle testing data, domestic research on emission factor models started late, and default foreign model values were used directly in assessing local vehicle pollution

emissions, resulting in large estimation errors. In recent years, with the development and application of vehicle emission monitoring system, it can obtain the actual road emission features used to make corrections to foreign emission models [9]. Quirama et al. [10] used PEMS to construct an energy-based microtrip operating model and estimate the actual energy consumption and exhaust emissions of a fleet in a given region. Tsinghua University developed an emission factor model for the Beijing Vehicle fleet (EMBEV) based on mature foreign emission models, which integrates the average speed and driving cycle to achieve macro- and microemission factor acquisition [11]. Wang et al. [12] used a sequential decision strategy based on vehicle low-frequency GPS trajectories to achieve roadway speed estimation and combined with a microscopic emission model to estimate vehicle CO₂ emissions. Wang et al. [13] considered the influence of vehicle historical operating state and constructed a microscopic emission model based on BP neural network using short-time driving cycle. The traditional driving cycle-based emission model uses artificially designed parameters such as vehicle speed and acceleration to characterize the relationship between vehicle driving cycle and pollution emissions, but it ignores the vehicle engine operating state information and inadequate representation of vehicle driving cycle characteristics, which makes it difficult to effectively estimate the exhaust emissions of monitoring missing vehicles under different driving conditions.

With the boom in machine learning and deep learning research, some scholars began to introduce artificial intelligence techniques into the research of mobile source emission estimation. Chen et al. [14] used quantile regression forest on vehicle remote sensing data for NO prediction. Xu et al. [15] established a spatiotemporal map convolutional multifusion network for effective prediction of regional vehicle emissions in Hefei city. Xu et al. [16] combined deep moment residual early-late fusion network with semisupervised geographical weighted regression to predict regional emissions as a spatiotemporal series data. Altug and Kucuk [17] trained XGBoost with engine speed, engine torque, pedal position, and vehicle speed data as inputs to predict NO_x emissions and compared it with elastic network and LSTM, showing its high accuracy. Fei et al. [18] proposed a multicomponent fusion time network to predict the emissions of CO considering multiple complex features. Xu et al. [19] constructed a mobile source emission prediction model based on deep neural network to realize the relationship mapping between vehicle transient operating conditions and pollution emissions and further proposed a deep correction model for remote emission sensing data based on COPERT emission features by constructing a three-layer AutoEncoder network to realize the feature extraction of multisource heterogeneous data, such as meteorological data, road network data, traffic flow data, and urban functional areas [20].

In actual vehicle emission detection systems, due to differences in vehicle driving conditions, engine operating conditions, and driving behavior patterns, it is impossible to ensure that the emission monitoring data of different vehicles always follow the same distribution. However, in a traditional

machine learning emission model, it is usually assumed that the training set and testing set of emission testing data are derived from the same data distribution, and a unified emission model is used for estimation of different types of vehicles, ignoring the difference in monitoring data distribution. As shown in Figure 1, Label_S represents the NO_x emission values of diesel vehicle used for training to obtain the NO_x prediction model on one kind of diesel vehicle, which is the label of the training set in the regression model. Label_T represents the NO_x emission values of diesel vehicle that are expected to make use of the knowledge of the prediction model obtained on the Label_S dataset, which is the label of the data set on another kind of diesel vehicle in the regression model. The NO_x emission distribution of source domain and target domain is different, which will cause the performance degradation in the supervised model based on independent identical distribution assumption. The transfer learning technique [21] can provide a solution for the construction of exhaust emission prediction models under different driving conditions by transferring the data-complete source domain knowledge to the data-sparse target domain.

Inspired by the insight of transfer learning, a novel NO_x emission inversion prediction method for diesel vehicles is proposed in this paper. Specifically, it is a deep transfer learning (DTL)-based model which firstly uses Spearman correlation analysis and Lasso feature selection to accomplish the selection of factors with high correlation with NO_x emission from multiple influence factors (e.g., throttle state and engine-related states). Then, the stacked sparse AutoEncoder is used to map different vehicle working condition emission data into the same feature space, and then the distribution alignment of different vehicle working condition emission data features is achieved by minimizing maximum mean discrepancy (MMD) in the feature space. Finally, we validated the proposed method on the real-world diesel vehicle OBD data, and the comprehensive results show that the proposed DTL model outperforms several deep learning (DL) methods, indicating that DTL based on multiple sources of external influences has great potential for diesel vehicle NO_x emission prediction in the case of insufficient monitoring data.

The rest of the article is organized as follows. Section 2 discusses the related works. Section 3 describes the construction of the DTL model. In Section 4, several experiments are conducted. The conclusions and future research are drawn in Section 5.

2. Related Works

2.1. Lasso. Using features unrelated to the prediction as input variables will increase the complexity and reduce the explanatory power of the regression model, so it is necessary to select the relevant initial features. Least absolute shrinkage and selection operator (Lasso) proposed by Tibshiran [22] is a commonly used variable selection method in terms of the machine learning field. It achieves variable selection by adding the L1 norm so that some of the variable coefficients in the input variables are trained to be set to 0. The loss function is as follows:

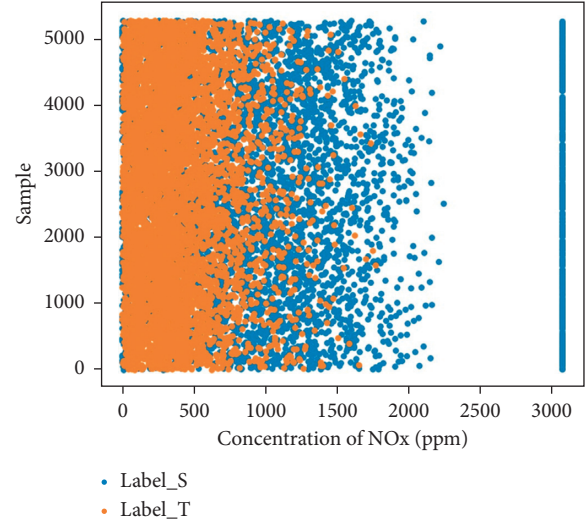


FIGURE 1: Emission distribution of different driving conditions.

$$\beta_{\text{Lasso}} = \frac{1}{m} \sum_{i=1}^m (y_i - wx_i) + \lambda \|w\|_1, \quad (1)$$

where λ is the penalty coefficient and the larger its value, the fewer variables are retained. The cross-validation method is usually used to determine its optimal value.

2.2. SAE. AutoEncoder (AE) is a symmetric single hidden layer neural network [23]. It consists of an encoding module and a decoding module, where the encoding module is represented by the input layer to the hidden layer, and the decoding module is represented by the hidden layer to the output layer. After training, it is able to copy the input to the output to the maximum extent possible, and the features of the hidden layer represent an abstract representation of the input features in the feature space. The AE structure is shown in Figure 2, where the input i th sample $X_i = [x_{i,1}, x_{i,2}, x_{i,3}, \dots, x_{i,m}]$ contains m features, where the features h of the hidden layer is specifically expressed as

$$h = f(w_1 \cdot x + b_1). \quad (2)$$

The formula w_1 is the weight from the input layer to the hidden layer, b_1 is the bias from the input layer to the hidden layer, and f is an activation function, and in this paper, we choose sigmoid. The reconstructed feature \hat{x} can be expressed as

$$\hat{x} = f(w_2 \cdot h + b_2), \quad (3)$$

where w_2 is the weight of the hidden layer to the output layer and b_2 is the bias of the hidden layer to the input layer. In order to ensure that \hat{x} can be restored to the maximum extent, the loss function is used as follows:

$$J(w, b) = \sum_{i=1}^m \left(\frac{1}{2} \|x_i - \hat{x}_i\|^2 \right). \quad (4)$$

When the number of hidden layer neurons is smaller than the number of inputs, the AutoEncoder can achieve data compression.

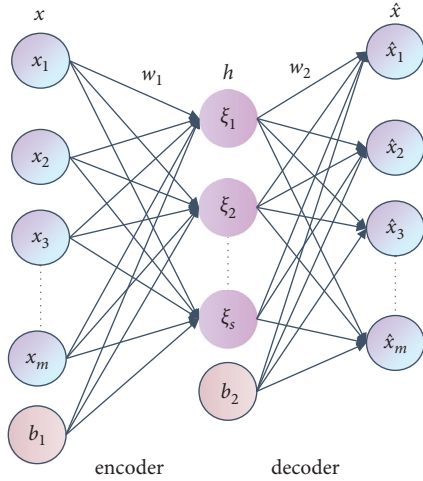


FIGURE 2: The structure of AutoEncoder.

The AutoEncoder simply copies the input to the output in training, which makes it difficult to obtain meaningful feature representations. Nowadays, research compensates for this drawback by adding constraints to traditional AutoEncoder, resulting in various novel AutoEncoder, such as Denoising AutoEncoder (DAE) [24], Sparse AutoEncoder (SAE) [25], and Variational AutoEncoder (VAE) [26].

In SAE, KL divergence is added as a sparse penalty term to force only some of the neurons in the hidden layer to be activated. The KL divergence is expressed as follows:

$$D_{KL} = \rho \log \frac{\rho}{\rho_j} + (1 - \rho) \log \frac{1 - \rho}{1 - \rho_j}, \quad (5)$$

where ρ represents the probability of the hidden layer neuron being activated, which is generally taken as a value close to 0. ρ_j is the actual activation probability of the j th neuron in the hidden layer, which is expressed as follows:

$$\rho_j = \frac{1}{m} \sum_{i=1}^m f_j(x_i), \quad (6)$$

where $f_j(x_i)$ represents the activation probability of the hidden layer neuron j when the input data is the i th sample.

In addition, to prevent the network from overfitting, the L2 norm is added to the loss function and α and β are the penalty coefficients of the sparse and weight terms. In summary, the loss function of SAE is as follows:

$$J_{SAE}(w, b) = \frac{1}{2} \sum_{i=1}^m (\|x_i - \hat{x}_i\|^2) + \alpha \sum_{j=1}^{j=s} D_{KL}(\rho \| \rho_j) + \frac{\beta}{2} (\|w_1\|^2 + \|w_2\|^2). \quad (7)$$

2.3. Stacked AutoEncoder. Compared to a normal AutoEncoder, a stacked AutoEncoder can obtain hidden features that are more suitable for complex regression tasks. The stacked sparse AutoEncoder uses layer-wise unsupervised pretraining [27]; specifically, after the simple sparse

AutoEncoder is trained, the features of the hidden layer are used as a new input to train a new sparse AutoEncoder, which can be described as $n \rightarrow m \rightarrow n \Rightarrow m \rightarrow k \rightarrow m \Rightarrow \dots \Rightarrow s \rightarrow t \rightarrow s$, and when the required number of layers is reached, all hidden layers are combined in order to form a stacked sparse self-encoder.

2.4. Domain Adaptation. Domain adaptation (DA) [28] is a more popular transfer learning method, which aims to map source features with different distributions and target features into the same space and draw the distributions of the two close in the feature space, thus achieving distribution alignment, and then the objective function obtained by training using the source data in the feature space can be transferred to the target domain.

There are three main types of DA methods in deep learning, which are discrepancy-based domain adaptation, adversarial-based domain adaptation, and reconstruction-based domain adaptation.

Discrepancy-based domain adaptation focuses on measuring the difference between the source and target domains by adding a certain metric and achieving alignment between the source and target domains by minimizing this metric. In deep domain adaptation, Tzeng et al. [29] proposed a new CNN structure that performs domain adaptation by adding an adaptive layer and an MMD-based loss function and has excellent performance on vision domain tasks; Werner et al. [30] proposed central moment difference (CMD), which performs domain adaptation by aligning the central moments of each order between domains; Li et al. [31] proposed a DTN based on MMD for adaptation of edge distribution and conditional distribution, which has superiority in image classification and recognition as well as text classification.

Adversarial-based domain adaptation is mainly achieved through adversarial with discriminators, where the generator aligns source and target data on the feature space. Eric et al. [32] combined discriminative model, weight sharing, and GAN loss to propose Adversarial Discriminative Domain Adaptation (ADDA); Judy et al. [33] proposed Cyclic Consistent Adversarial Domain Adaptation (CyCADA) to perform cross-domain adaptation at both pixel level and feature level while ensuring semantic consistency. Shen et al. [34] proposed WGDRL metric and optimized feature extraction network to reduce Wasserstein distance in an adversarial manner.

Reconstruction-based domain adaptation mainly focuses on domain adaptation by reconstructing the data to ensure that the learned features remain unchanged. Glorot et al. [35] proposed domain adaptation based on stacked AutoEncoder SDAs to extract higher-order semantic information; Bousmalis et al. [36] proposed a DSN framework to decode the source and target domains with a common decoder for each of the three encoder outputs that extract the common features between different domains and use the shared features for transferring.

2.5. MMD. Maximum mean discrepancy (MMD) is used more frequently in transfer learning as a common means of

measuring the difference between two domains. It maps the original data into Hilbert space and then measures the distribution between the two domains, which is a kernel learning method [37]. The specific metric formula is as follows:

$$\text{MMD}[F, X, Y] = \left\| \sum_{i=1}^{n_1} k(x_i) - \sum_{j=1}^{n_2} k(y_j) \right\|^2, \quad (8)$$

where $k(\cdot)$ is the mapping for mapping the original data into the Reproducing Kernel Hilbert Space (RKHS), X, Y denote the samples of two distributions, and F is the set of mapping functions.

3. Methodology

In this section, we mainly introduce the details of the model in this paper. As shown in Figure 3, we propose a deep transfer learning (DTL) model for NO_x emissions from diesel vehicles based on multisource external influences, using Spearman for correlation analysis and Lasso-based feature selection to find out the features with strong correlation with diesel vehicle NO_x emission. After that, the stacked sparse AutoEncoder is designed to extract the common hidden features in the source and target domains. The data alignment of different vehicle models is achieved by minimizing the MMD distance between the source and target domains. Finally, the transfer of NO_x emission models between vehicles with different data distribution is obtained.

3.1. Data Description. The OBD data of diesel vehicles collected in Hefei in 2020 includes license plate, terminal number, data date, engine speed, actual output torque percentage, water temperature of engine, oil temperature of engine, after-treatment downstream NO_x value, after-treatment downstream oxygen percentage, atmospheric pressure, environmental temperature, after-treatment waste mass flow rate, urea tank level percentage, temperature of urea tank, vehicle speed, gas pedal opening, word trip mileage, total mileage, engine instantaneous fuel injection, engine instantaneous fuel consumption rate, average engine fuel consumption, engine fuel consumption for a single trip, cumulative engine fuel consumption, battery voltage mailbox level, cumulative engine running time, longitude, latitude, SCR upstream temperature, and SCR downstream temperature.

Table 1 shows the comparison of the detailed parameters of the source domain diesel vehicle and the target domain diesel vehicle. In order to improve the data quality, we preprocessed the data, including data deduplication, outlier removal, and removal of irrelevant features. After preprocessing, the data statistics of source domain diesel vehicle and target domain diesel vehicle are shown in Tables 2 and 3.

3.2. Relevant Features Selection. There are many features affecting NO_x monitoring downstream of diesel vehicle after-treatment, and for the source data after pretreatment, we calculate Spearman correlation coefficients between NO_x

and many features downstream of after-treatment, such as oxygen percentage, engine speed, and temperature of engine water, and subject these coefficients to hypothesis testing at $p = 0.05$ ($t = 1.645$) and remove the uncorrelated NO_x emission external features as new characteristics. The specific values of Spearman coefficient and T value are shown in Table 4, from which it is easy to know that the temperature of engine oil, the temperature of ambient, temperature of urea tank, and percentage of urea tank level are not related to NO_x emission under the condition of $t = 1.645$.

After finding out the new features, the Lasso algorithm was used to calculate the correlation coefficients of each feature with NO_x , and then the features whose coefficients were not 0 were taken as the final features. Among them, the Lasso coefficients of each feature with NO_x after Spearman correlation analysis are shown in Table 5, where the Lasso coefficients of vehicle speed and NO_x are 0, and they are removed from the final features.

The new source data consisting of Spearman and Lasso processed features are denoted as n_X_S , and their features $\text{feature}_{n_X_S}$ are subdivided by source into vehicle engine-related, vehicle throttle-related, and vehicle after-treatment system-related, and their specific classification is shown in Table 6.

In order to ensure that the source features and the target features are the same, we take the feature $\text{feature}_{n_X_S}$ of n_X_S as the benchmark and make the feature feature_{X_T} in the target domain intersect with it, and the obtained feature $\text{feature}_{n_X_T}$ forms the new target domain data n_X_T for which the visualization is expressed as

$$\begin{aligned} \text{feature}_{n_X_T} &= \text{feature}_{n_X_S} \cap \text{feature}_{X_T}, \\ \text{feature}_{n_X_S} &\in \text{feature}_{X_T}. \end{aligned} \quad (9)$$

Since the monitoring elements for diesel vehicle emissions are consistent, $\text{feature}_{n_X_S}$ is a subset of feature_{X_T} .

3.3. DTL. After screening the external correlates of diesel vehicle NO_x emissions, we obtained homotypic source and target data highly correlated with vehicle NO_x emissions, whose characteristics contain engine speed, actual output torque percentage, temperature of engine water, gas pedal opening, after-treatment downstream oxygen percentage, and after-treatment exhaust gas mass flow rate. Domain adaptation is achieved by minimizing the MMD distance representing the difference in distribution between the source and target data through a deep transfer of network projection to the common space and a high-dimensional sparse representation.

3.3.1. Stacked Sparse AutoEncoder. We take n_X_S as the input of the first layer of the stacked sparse AutoEncoder, the number of hidden layer neurons is set to 5 times the number of input features, $\alpha = 2, \beta = 0.01$, and the probability of the hidden layer neurons being activated ρ is 0.05, and we optimize the loss function J_{SAE} by backpropagation and save

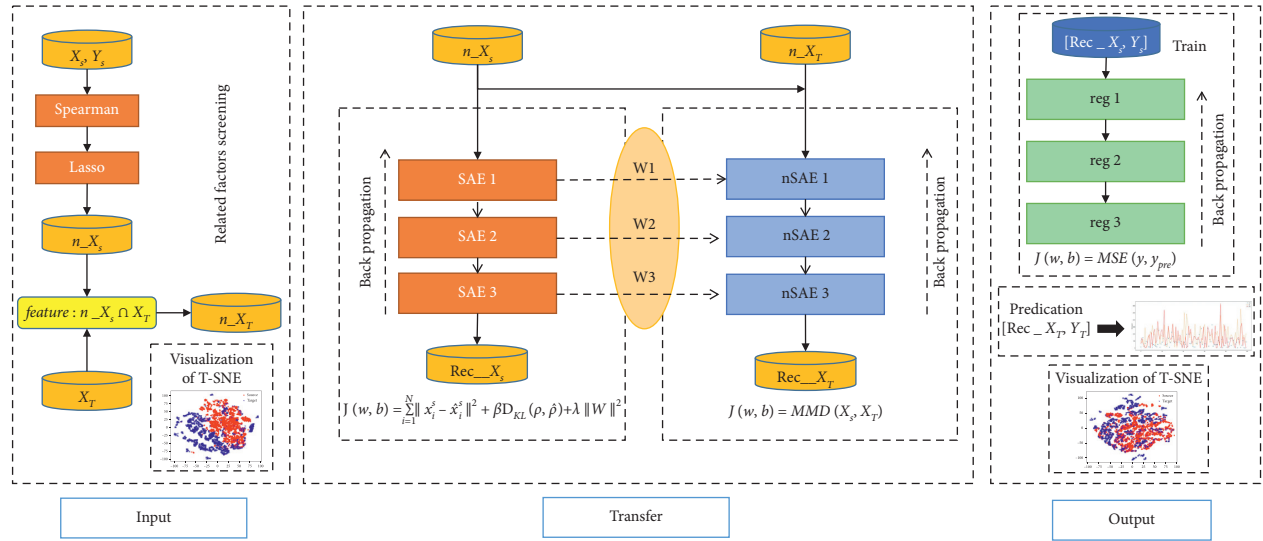


FIGURE 3: The architecture of the proposed method.

TABLE 1: The detailed vehicle and engine parameters.

	Source	Target
Vehicle brand	HIGER	SHACMAN
Vehicle type	KLQ6115K	SX3315DR366
Engine type	WP10.336E40	WP12.375E50
Number of cylinders	6	6
Displacement	9.726 L	11.596 L
Rated rotation speed	1900 rpm	1900 rpm
Maximum output power	247 kW	276 kW
Maximum horsepower	336	375
Maximum torque	1500 Nm	1800 Nm
Maximum torque speed	1300–1600 rpm	1000–1400 rpm

TABLE 2: Descriptive statistics of the source domain.

	Count	Mean	Std	Min	Max
Engine speed	5229	1194.87	297.60	0	2095.0
Actual output torque percentage	5229	23.64	21.16	0	99.0
Water temperature of engine	5229	78.10	3.59	67.0	88.0
Oil temperature of engine	5229	86.69	4.17	69.0	97.0
After-treatment downstream NO _x value	5229	997.92	820.01	0	3076.0
After-treatment downstream oxygen percentage	5229	13.69	4.76	0	21.0
Environmental temperature	5229	29.90	1.37	26.0	32.0
After-treatment waste mass flow rate	5229	443.13	149.48	0	1008
Urea tank level percentage	5229	90.99	1.98	82.0	100.0
Temperature of urea tank	5229	30.31	0.52	29.0	32.0
Vehicle speed	5229	35.41	18.42	0	59.0
Gas pedal opening	5229	19.19	19.36	0	100.0

the hidden layer weights after the network converge. Then, we use the hidden layer feature data as input and train the new sparse AutoEncoder according to the above steps, and when the required number of stacked layers k is reached, the saved hidden layers are stacked, and Table 7 shows the hidden feature dimensions of different stacked layers.

3.3.2. Weight Sharing. In order to learn the common hidden features of the source and target domains more quickly

and efficiently, we use weight sharing as a means to transfer the weights of each layer of the stacked sparse AutoEncoder trained with the source data to the final deep transfer network.

Weight sharing is common means in deep transfer learning [38–40]. After pretraining a stacked sparse AutoEncoder with source data, the weights W_i and bias b_i of each layer need to be shared to the new stacked sparse AutoEncoder to complete the weight transferring, described as

TABLE 3: Descriptive statistics of the target domain.

	Count	Mean	Std	Min	Max
Engine speed	5229	1182.85	269.88	534.0	1997.0
Actual output torque percentage	5229	26.47	24.89	0	99.0
Water temperature of engine	5229	75.01	0.79	73.0	78.0
Oil temperature of engine	5229	72.41	3.49	35.0	77.0
After-treatment downstream NO _x value	5229	387.85	288.98	0	1769.0
After-treatment downstream oxygen percentage	5229	13.27	3.37	2.0	21.0
Environmental temperature	5229	29.64	1.48	26.0	32.0
After-treatment waste mass flow rate	5229	464.85	177.72	171.0	1142.0
Urea tank level percentage	5229	56.89	2.25	35.0	82.0
Temperature of urea tank	5229	30.01	0.88	29.0	31.0
Vehicle speed	5229	37.56	19.45	0	80.0
Gas pedal opening	5229	24.38	23.67	0	100.0

TABLE 4: Spearman coefficients and corresponding T values.

Feature	ρ	T
Engine speed	0.3035	21.95
Actual output torque percentage	0.4749	34.34
Temperature of engine water	0.0281	2.03
Temperature of engine oil	0.0178	1.29
After-treatment downstream oxygen percentage	-0.4292	-31.04
Urea tank level percentage	0.0169	1.22
Environmental temperature	0.0093	0.67
After-treatment exhaust gas mass flow rate	0.4868	35.20
Temperature of urea tank	-0.0008	-0.055
Vehicle speed	0.1482	10.72
Gas pedal opening	0.4847	35.05

TABLE 5: Lasso coefficients of relevant features.

Engine speed	0.5630
Actual output torque percentage	12.5938
Temperature of engine water	-5.6142
After-treatment downstream oxygen percentage	-9.4329
After-treatment exhaust gas mass flow rate	1.8042
Vehicle speed	0
Gas pedal opening	13.9244

$$\begin{aligned} w_i^n &= w_i, \\ b_i^n &= b_i, \end{aligned} \quad (10)$$

where w_i^n is the i th hidden layer weight of the new stacked sparse AutoEncoder and b_i^n is the i th hidden layer bias of the new stacked sparse AutoEncoder; w_i and b_i are the i th hidden layer weight and bias of the trained stacked sparse AutoEncoder, respectively.

3.3.3. Feature Transfer Learning. In order to mix the source domain and the target domain into the same domain in the feature space, we put $n_{-}X_S$, $n_{-}X_T$ into the new sparse AutoEncoder together as inputs and use MMD as the loss function. Therefore, the loss function of the stacked sparse AutoEncoder based deep transfer network is as follows:

$$\begin{aligned} J(w, b) &= \text{Di} f. (n_{-}X_S, n_{-}X_T) \\ &= \text{MM} D[n_{-}X_S, n_{-}X_T] \\ &= \left[\frac{1}{m^2} \sum_{i,j=1}^m k(n_{-}x_{S,i}, n_{-}x_{S,j}) - \frac{2}{mn} \sum_{i,j=1}^{m,n} k(n_{-}x_{S,i}, n_{-}x_{T,j}) + \frac{1}{n^2} \sum_{i,j=1}^n k(n_{-}x_{T,i}, n_{-}x_{T,j}) \right]^{\frac{1}{2}}. \end{aligned} \quad (11)$$

By continuously minimizing the MMD, the distribution of target and source domain can be effectively brought closer together in new feature space. By continuously minimizing

the MMD, the distributions of the target domain and source domain can be effectively approximated in the new feature space. With the help of back propagation, the gradient

TABLE 6: Final characteristic features.

Vehicle engine-related	Vehicle throttle-related	Vehicle after-treatment system-related
Engine speed	Gas pedal opening	After-treatment downstream oxygen percentage
Actual output torque percentage	—	After-treatment exhaust gas mass flow rate
Temperature of engine water	—	—

TABLE 7: Number of hidden features of SAE.

Number of stacked layers of SAE	Number of hidden features
1 layer	6 * 5
2 layers	6 * 5 * 5
3 layers	6 * 5 * 5

descent updates the weights and biases until convergence, and the output is used as new features. The new features in the source and target domains are denoted as Rec_X_S, Rec_X_T , respectively.

3.4. Target Domain. NO_x prediction. The source data and the target data are projected onto the feature space by the deep transfer network, and the domain adaptation is completed. The transformation of the original features through the deep transfer network can be described in detail by the following equation:

$$x_i \longrightarrow [x_{1,i}, x_{2,i}, \dots, x_{s,i}], \quad (12)$$

where x_i is the original feature in column i , $[x_{1,i}, x_{2,i}, \dots, x_{s,i}]$ is the feature representing x_i on the feature space, where s is an artificially set parameter.

Since the NO_x values downstream of the after-treatment are affected by nonlinear features, such as engine speed, oxygen percentage downstream of the after-treatment, and engine water temperature, we chose to use a BP neural network to build a regression prediction model.

After feature transferred, we divide $[Rec_X_S, Y_S]$ into training and validation sets by 8:2, and $[Rec_X_T, Y_T]$ is used as the test set to construct a double hidden layer BP neural network model. The mean square error (MSE) is chosen for the loss function of the whole regression network, and the mean absolute error (MAE) is chosen for the evaluation index, and Adam is used as the optimization function, and after the whole network converges, it is tested on the test set.

3.5. Evaluation Metrics. We use mean absolute error (MAE) and root mean squared error (RMSE) to effectively evaluate the prediction effectiveness of NO_x emissions. They are calculated as follows:

$$\begin{aligned} MAE &= \frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i), \\ RMSE &= \sqrt{\frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2}, \end{aligned} \quad (13)$$

where m is the number of samples, y_i is the true value of the label, and \hat{y}_i is the predicted value of the label.

4. Experiments

4.1. MMD Settings. The arrangement of MMD is diverse. We try to add MMD at different layers for domain adaptation based on stacking sparse AutoEncoder layers and compare the MAE and RMSE of their predicted values to select the optimal position as the final model setup.

Table 8 shows the MAE, RMSE comparison of the predicted values after selecting sparse AutoEncoders with different layers and trying to add MMD at different layers, where SAE (n), n represents the number of stacked sparse AutoEncoder layers, and the number in the DTL term represents the addition of MMD at that layer. Since the number of features dimensions increases exponentially as the layers are stacked, which also increases the training time, we only compare the stacked sparse AutoEncoder up to 3 layers. From the results in Table 8, we choose the sparse AutoEncoder with three stacked layers and add MMD in the second layer for domain adaptation.

4.2. Model Performance. In order to verify the effectiveness of the model proposed in this paper, we compare the traditional deep learning model with the results of our model. The traditional deep learning (DL) model defaults to the source domain and the target domain belonging to the same distribution and uses the source data as the training set and the validation set to train a BP neural network and the target data as the test set. Figure 4 shows the prediction effect of 100 randomly selected data points, in which the DTL model proposed in this paper has a smaller prediction error and a better fit with the true value compared with the traditional DL model.

To further validate the effectiveness of the model in this paper, we conducted experiments using the DTL and DL on the dataset without relevant feature screening (the effect is shown in Figure 5), and the training set, validation set, and test set were kept consistent with the previous experiments. In the regression prediction section, Random Forest, Support Vector Regression (SVR), and AdaBoost were tried as regressors, and Table 9 shows the comparison of the prediction results of the models, where DL represents the prediction model with deep learning without considering the influence of external features on NO_x emissions, nDTL represents the prediction model with deep transfer learning without considering the influence of external features on NO_x emissions, nDL represents the prediction model with deep learning with considering the influence of external features on NO_x emissions, and nDTL represents the prediction model with deep transfer learning with considering

TABLE 8: Comparison of model predictions based on MMD settings.

	DTL	MAE	RMSE
SAE(1)	1	449.40	584.62
SAE(2)	1	365.07	455.09
	2	374.74	458.43
	1 + 2	380.30	479.37
SAE(3)	1	345.68	453.65
	2	314.65	403.04
	3	387.59	481.68
	1 + 2	364.42	476.69
	1 + 3	372.75	482.47
	2 + 3	399.71	510.59
	1 + 2 + 3	368.28	489.83

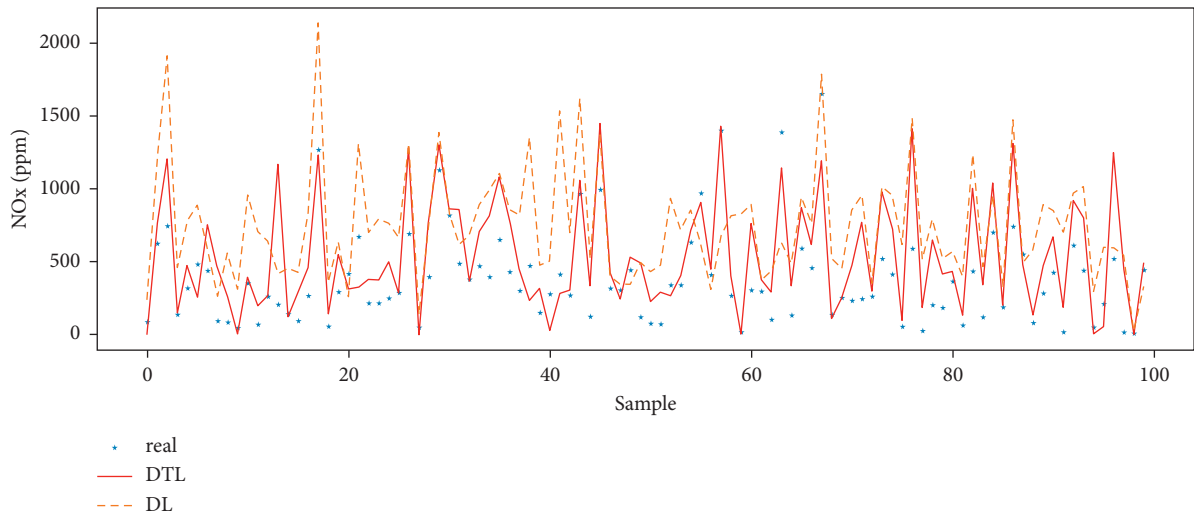


FIGURE 4: Comparison of DTL predictions and DL based on multiple sources of external features.

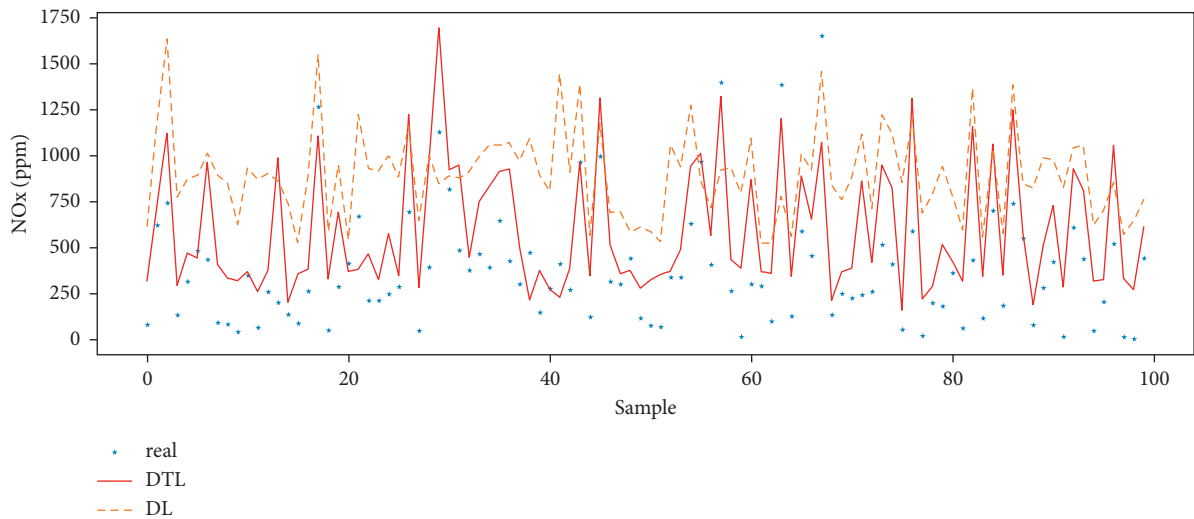


FIGURE 5: DTL prediction and DL comparison without considering multiple sources of external features.

the influence of external features on NO_x emissions. The data comparison in Table 9 shows that the performance of the DL without feature transfer is significantly higher than that of the DTL model, which proves the effectiveness of

feature transfer in unsupervised prediction, and it can be clearly concluded that the data after feature screening is more favorable for the prediction of diesel vehicle NO_x concentration, and in this experiment, the model of the

TABLE 9: Comparison of NO_x prediction values under different regressors.

Model	MAE	RMSE
SVR	389.45	430.59
RandomForest	430.24	489.56
AdaBoost	444.43	496.10
DL	573.95	823.85
DTL	372.60	566.20
nDL	570.08	635.35
nDTL (ours)	314.65	403.04

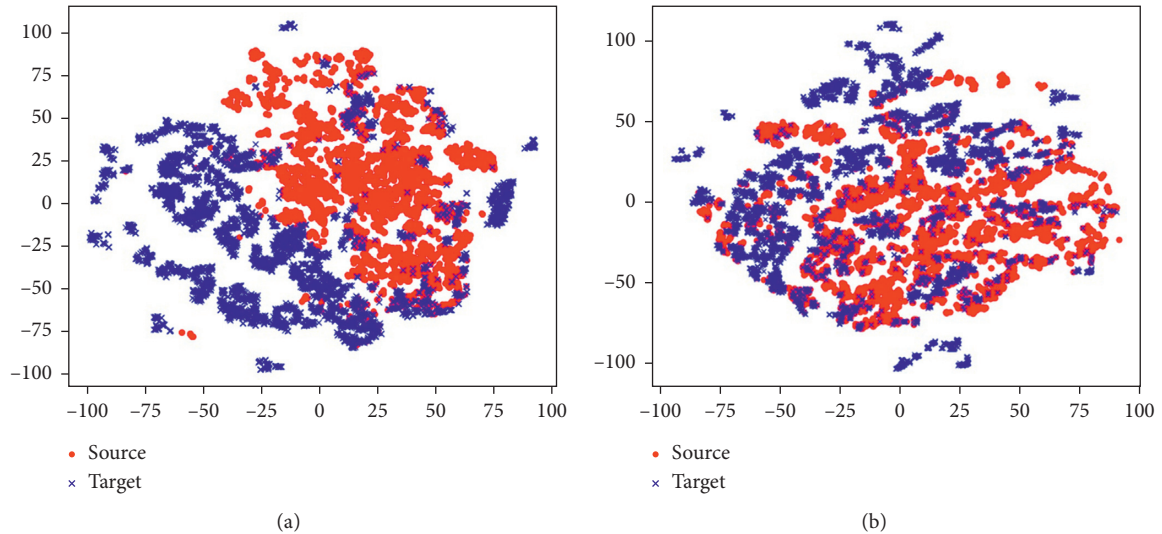


FIGURE 6: Vehicle driving condition distribution of source and target domains by T-SNE: (a) before feature alignment; (b) after feature alignment.

neural network is better than the models of traditional machine learning.

Figure 6 shows the visualization effect of the source diesel vehicle and target diesel vehicle before and after feature transferring by t-sne dimensionality reduction. Figure 6(a) shows the distribution of engine speed, real-time output torque percentage, temperature of engine water, gas pedal opening, after-treatment downstream oxygen percentage, and after-treatment exhaust gas mass flow rate features on the source diesel vehicle and the target diesel vehicle after t-sne dimensionality reduction. Figure 6(b) shows the distribution of the above features on the source diesel vehicle and the target diesel vehicle after t-sne dimensionality reduction by the reconstructed features after the deep transfer learning framework proposed in this paper. It is obvious from the figure that the data distribution of the source diesel vehicles and the target diesel vehicles after the training of the DTL model can be basically mixed in one domain, and domain adaptation is achieved.

4.3. Exploring the Influencing Features of NO_x Emission.

The above experiments selected relevant features that have significant effect on NO_x and predicted them effectively on the DTL model. In order to further investigate which specific aspect of features has more influence on NO_x concentration, we trained DTL with each type of attribute distribution

TABLE 10: Prediction under single-source external factors.

	MAE	RMSE
Engine-related	580.13	642.39
Throttle-related	419.19	474.71
Postprocessing system-related	618.96	665.192

according to the source division of influencing features in Table 6, predicted NO_x , and obtained MAE of predicted data, RMSE, as shown in Table 10.

From the indicators in Table 10, it is easy to know that the throttle-related feature for DTL is better predicted; that is, the degree of opening and closing of the throttle pedal has a great influence on the NO_x emissions of diesel vehicles during driving. In the real world, the acceleration sensitivity of diesel cars is poor. If the engine suddenly increases the fuel supply when the gas pedal is stepped on sharply at low speed of the diesel car, the circulating fuel supply will increase sharply, and due to its poor sensitivity, the diesel engine speed will not increase much, resulting in relatively weak air turbulence, prolonging the combustion process and increasing incomplete combustion, which eventually leads to increased NO_x emission; when the throttle is released sharply, it will cause the engine combustion conditions to deteriorate and work unstably due to the sudden closing of the throttle, and the NO_x emission will increase. Therefore, the driver should operate

the throttle smoothly when driving, not emergency pedal or emergency release of the throttle pedal.

Both engine-related and throttle-related features can be controlled in real time by the driver during driving. Proper driving behavior can greatly reduce pollutant emissions, that is, when road conditions and environmental conditions permit, a steady speed should be maintained and frequent speed changes should not be made, and NO_x emissions from diesel vehicles can be effectively reduced through after-treatment systems.

5. Conclusion

In this paper, we propose a deep AutoEncoder transferring inversion model for NO_x emission prediction of diesel vehicles under the integration of multiple sources of external influences to perform NO_x emission pattern transferring among different diesel vehicles and then effectively improve the accuracy of diesel vehicle NO_x emission prediction. For the OBD data of diesel vehicles, the features related to NO_x emissions are selected by using Spearman correlation analysis and Lasso feature selection, and the selected features of engine speed, actual output torque percentage, temperature of engine water, gas pedal opening, after-treatment downstream oxygen percentage, and after-treatment exhaust gas mass flow rate have strong correlation with NO_x emissions, and the designed DTL learning framework with distribution alignment relies on diesel vehicles containing the above strong correlation features with corresponding NO_x emission values and diesel vehicles with only the above strong correlation features to jointly train the network model so that the diesel vehicle data of different categories converge to the same distribution in the feature space and then train the objective function in the feature space using the diesel vehicle data containing NO_x emission values, and transfer to the diesel vehicles without NO_x emission values to achieve NO_x prediction of diesel vehicles without NO_x emission values, which provides an effective prediction method for the prediction of unlabeled diesel vehicle data. Based on the analysis of diesel vehicle NO_x based on external features from different sources, vehicle throttle-related features have large impact on diesel vehicle NO_x emissions, and reasonable control of throttle state during driving is an important means to effectively control diesel NO_x emissions.

Future research can be extended in the following ways.

(1) In potential feature extraction, other methods can be tried to find the abstract representation of the original features in the feature space. (2) In feature transfer, the MMD metric is used to measure the difference in distribution between two domains, and a more appropriate metric can be selected in later studies based on the dataset.

Data Availability

The data used to support the findings of this study have not been made available because of data ownership issues.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (62103124, 62033012, and 61725304), Major Special Science and Technology Project of Anhui, China (201903a07020012 and 202003a07020009), China Postdoctoral Science Foundation (2021M703119), and the National Key R&D Program of China under Grant 2018YFE0106800.

References

- [1] M. M. Oluwaseyi and M. S. Abolarin, "Specifications and analysis of digitized diagnostics of automobiles: a case study of on board diagnostic (OBD II)," *International Journal of Engineering Research*, vol. 9, 2020.
- [2] Us Environmental Protection Agency, *User's Guide to MOBILE6.1 and MOBILE6.2*, Mobile source emission factor model (EPA report EPA-420-R-03-010), Office of Transportation and Air Quality, U.S., 2003.
- [3] California Air Resources Board, *EMFAC2021 User's Guide*, California Environmental Protection Agency, California, U.S., 2021.
- [4] L. Ntziachristos, D. Gkatzoflias, C. Kouridis, and Z. Samaras, *COPERT: A European Road Transport Emission Inventory Model*, Springer, Berlin, Heidelberg, pp. 491–504, 2009.
- [5] J. Lents and N. Davis, *IVE Model User's Guide, Model and Data Files*, Technical Report submitted to the US Environmental Protection Agency, 2009.
- [6] G. Scora and M. Barth, *Comprehensive Modal Emissions Model (Cmem), Version 3.01[J]. User Guide*, p. 1070, Centre for environmental research and technology. University of California, Riverside, 2006.
- [7] S. Vallamsundar and J. Lin, "Overview of US EPA new generation emission model: MOVES," *International Journal on Transportation and Urban Development*, vol. 1, no. 1, p. 39, 2011.
- [8] A. Capiello, I. Chabini, E. K. Nam, A. Lue, and M. Abou Zeid, "A statistical model of vehicle emissions and fuel consumption," in *Proceedings of the IEEE 5th International Conference on Intelligent Transportation Systems*, pp. 801–809, IEEE, Singapore, Sept 2002.
- [9] P. Sderena, J. Laurikko, C. Weber et al., "Monitoring Euro 6 diesel passenger cars NOx emissions for one year in various ambient conditions with PEMS and NOx sensors[J]," *The Science of the Total Environment*, vol. 746, Article ID 140971, 2020.
- [10] L. F. Quirama, M. Giraldo, J. I. Huertas, and M. Jaller, "Driving cycles that reproduce driving patterns, energy consumptions and tailpipe emissions," *Transportation Research Part D: Transport and Environment*, vol. 82, Article ID 102294, 2020.
- [11] D. Yang, S. Zhang, T. Niu et al., "High-resolution mapping of vehicle emissions of atmospheric pollutants based on large-scale, real-world traffic datasets," *Atmospheric Chemistry and Physics*, vol. 19, no. 13, pp. 8831–8843, 2019.
- [12] S. Wang, Z. Li, J. Tan, W. Guo, and L. Li, "A method for estimating carbon dioxide emissions based on low frequency

- GPS trajectories,” in *Proceedings of the 2017 Chinese Automation Congress (CAC)*, pp. 1960–1964, IEEE, Jinan, China, Oct 2017.
- [13] W. Zhongqi, G. Wei, B. I. Jun et al., “Microscopic emission model of motor vehicle based on short-time real driving cycle,” *Chinese Journal of Environmental Engineering*, vol. 10, no. 10, pp. 5803–5807, 2016, in Chinese.
- [14] J. Chen, G. Dobbie, S. K. Yun, E. Somervell, and G. Olivares, “Vehicle emission prediction using remote sensing data and machine learning techniques,” in *Proceedings of the 33rd Annual ACM Symposium*, ACM, New York, NY, United States, Apr 2018.
- [15] Z. Xu, Y. Kang, Y. Cao, and Z. Li, “Spatiotemporal graph convolution multifusion network for urban vehicle emission prediction,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 99, pp. 1–13, 2021.
- [16] Z. Xu, Y. Cao, and Y. Kang, “Deep spatiotemporal residual early-late fusion network for city region vehicle emission pollution prediction,” *Neurocomputing*, vol. 355, pp. 183–199, 2019.
- [17] K. B. Altug and S. E. Kucuk, “Predicting tailpipe NOx emission using supervised learning algorithms,” in *Proceedings of the 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, IEEE, Ankara, Turkey, October 2019.
- [18] X. Fei, F. Long, F. Li, and Q. Ling, “Multi-component fusion temporal networks to predict vehicle exhaust based on remote monitoring data,” *IEEE Access*, vol. 9, no. 99, pp. 42358–42369, 2021.
- [19] Z. Xu, Y. Kang, and W. Lv, “Analysis and prediction of vehicle exhaust emission using ann,” in *Proceedings of the 2017 36th Chinese Control Conference (CCC)*, pp. 4029–4033, IEEE, Dalian, China, July 2017.
- [20] Z. Xu, Y. Kang, Y. Cao, and L. Zerui, “Deep amended COPERT model for regional vehicle emission prediction,” *Science China Information Sciences*, vol. 64, pp. 1–3, 2021.
- [21] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.
- [22] R. Tibshirani, “Regression shrinkage and selection via the lasso: a retrospective,” *Journal of the Royal Statistical Society: Series B*, vol. 73, no. 3, pp. 267–288, 2011.
- [23] P. Baldi, G. Guyon, and V. Dror, “Autoencoders, unsupervised learning and deep architectures,” in *Proceedings of the ICML Workshop on Unsupervised and Transfer Learning*, JMLR.org, London U.K, July 2011.
- [24] P. Vincent, H. Larochelle, Y. Bengio, and P. A. Manzagol, “Extracting and composing robust features with denoising autoencoders[C]//machine learning,” in *Proceedings of the Twenty-Fifth International Conference (ICML 2008)*, ACM, Helsinki, Finland, June 2008.
- [25] M. Ranzato, C. Poultney, S. Chopra, and Y. Lecun, “Efficient learning of sparse representations with an energy-based model,” in *Proceedings of the Twentieth Advances in Neural Information Processing Systems (NIPS 2006)*, ACM, Vancouver, Canada, January 2007.
- [26] D. P. Kingma and M. Welling, *Auto-Encoding Variational Bayes*, ICLR, 2014, <https://arxiv.org/abs/1312.6114>.
- [27] H.-I. Suk, S. W. Lee, S.-W. Lee, and D. Shen, “Latent feature representation with stacked auto-encoder for AD/MCI diagnosis,” *Brain Structure and Function*, vol. 220, no. 2, pp. 841–859, 2015.
- [28] M. Wang and W. Deng, “Deep visual domain adaptation: a survey,” *Neurocomputing*, vol. 312, pp. 135–153, 2018.
- [29] E. Tzeng, H. Judy, Z. Ning, S. Kate, and D. Trevor, *Deep Domain Confusion: Maximizing for Domain Invariance*, <https://arxiv.org/abs/1412.3474>, 2014.
- [30] Z. Werner, G. Thomas, L. Edwin, N. Thomas, and S.-P. Susanne, *Central Moment Discrepancy for Domain-Invariant Representation Learning EB/OL*, pp. 12–20, 2020, <https://arxiv.org/abs/1702.08811>.
- [31] J. Li, K. Lu, Z. Huang, L. Zhu, and H. T. Shen, “Transfer independently together a generalized framework for domain adaptation,” *J. IEEE Transactions*, vol. 49, no. 6, pp. 2144–2155, 2018.
- [32] T. Eric, H. Judy, S. Kate, and D. Trevor, “Adversarial discriminative domain adaptation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Washington D. C. USA, July 2017.
- [33] H. Judy, T. Eric, P. Taesung et al., “Cycadacyleconsistent adversarial domain adaptation,” in *Proceedings of the International Conference on Machine Learning*, IEEE, Washington D. C. USA, Nov 2017.
- [34] J. Shen, Q. Yanru, Z. Weinan, and Y. Yong, *Wasserstein Distance Guided Representation Learning for Domain Adaptation*, AAAI, 2018, <https://arxiv.org/abs/1707.01217>.
- [35] X. Glorot, T. George, S. Nathan, K. Dilip, and E. Dumitru, “Domain adaptation for large-scale sentiment classification: a deep learning approach,” ICML, 2011, <https://arxiv.org/abs/1608.06019>.
- [36] K. Bousmalis, T. George, S. Nathan, K. Dilip, and E. Dumitru, *Domain Separation Networks*, NIPS, 2016, <https://arxiv.org/abs/1608.06019>.
- [37] A. Gretton, K. M. Borgwardt, M. J. Rasch, S. Bernhard, and J. S. Alexander, “A kernel method for the two-sample problem,” *Advances in Neural Information Processing Systems*, p. 1, 2007, <https://arxiv.org/abs/0805.2368>.
- [38] L. Wen, L. Gao, and X. Li, “A new deep transfer learning based on sparse auto-encoder for fault diagnosis,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 1, pp. 136–144, 2019.
- [39] C. Sun, M. Ma, Z. Zhao, T. Shaohua, Y. Ruqiang, and C. Xuefeng, “Deep transfer learning based on sparse auto-encoder for remaining useful life prediction of tool in manufacturing,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, p. 1, 2018.
- [40] K. Chetak, M. S. Luís, A. A. Luís, S. Ricardo, M. S. Jorge, and M. d S. Joaquim, “Improving transfer learning accuracy by reusing Stacked Denoising Autoencoders,” in *Proceedings of the IEEE International Conference on Systems*, IEEE, San Diego, CA, USA, October 2014.