DPC-IINK: A Framework for Drift Prediction and Compensation with Inter and Intra Node Knowledge Transfer

Zhang Yu, Tanaya Chaudhuri, Liu Pan, Wang Lu, Wu Min and Li Xiaoli

Abstract—A gas sensor provides a system with signals to monitor and respond to events which matter. However, some widely deployed sensors behave with drift, leading to inaccurate measurement. The system could malfunction due to the drift. Even more, if the system is not robust enough, the mistaken measurement could trigger some serious consequences. A drift prediction and compensation framework is necessary to contain this malfunction. On the other hand, when sensors are deployed with large quantity, it limits the maintaining cost of them, including the calibration cost. Data-driven frameworks show good potential for cheap calibration. But, there are some challenges. First, sensor drift keeps occurring. The drift models need to be retrained over time. Second, the drift models for the gas sensors of same type cannot be one-for-all way because they vary due to small manufacturing discrepancy. Third, drift models for a sensor could even be different only because the sensor is under different scenarios. In this paper, we propose a new framework for drift prediction and compensation with inter and intra node knowledge transfer (DPC-IINK) for widely deployed low-cost gas sensors. Knowledge transfer is employed for inter- and intra-node way, as well as for continuous and discontinuous drift. Our evaluation shows that our proposed framework with knowledge transfer improves drift prediction performance with less computing cost.

Index Terms—sensor drift, gas sensor,deep learning, knowledge transfer.

1 Introduction

Metal-oxide (MOX) semiconductors have attracted much interest to detect toxic gases in ambient environment due to their high sensitivity, relative inexpensiveness and portable size compatibility [1] [2]. Their mechanism is well known- the absorption and desorption from the surface of metal oxide by a toxic gas vary the resistivity of the material, which was first demonstrated by Seiyama et al. in the 1960s [3]. The sensor instability, known as the sensor drift, becomes a significant quality requirement in the development of gas sensors for the IoT market [4]. Based on the sources of reaction on the surface material, sensor drift that results in the decrease of sensors selectivity and sensitivity could be classified into two main factors, the intrinsic and the environmental. The intrinsic factor is attributed to sensor poisoning from the ambient gas at the sensing microstructure surface, where the chemical reaction could cause the metal oxide initial conductor to gradually and continuously drift over time. The other factor is the external surrounding condition of environment like ambient temperature, humidity and pressure [5] [6]. This drift phenomenon is inevitable after the fabrication of chemical sensors; therefore, drift detection and compensation are necessary to obtain reliable and accurate measurement for long term usage [7] [8] [9]. Various methods to deal with the sensor drift have been studied via signal processing and machine learning approaches recently [6] [10] [11] [12].

In an industrial system, gas sensors detect presence or concentrations of gases. However, the gas sensors, especially low-cost ones always behave with drift, leading to unknown consequences. To prevent the potential problems, a sensor drift containing method is necessary. Furthermore, when the gas sensors are widely deployed, it is not practical to use an expensive drift compensation. A data-driven method collects time-series data and compensates drift with models. It is a both cost-effective and relatively accurate way. In this paper, we proposed a data-driven framework for drift prediction and compensation with inter and intra node knowledge transfer.

There are three challenges for modeling the drift of gas sensors. First, sensor drift occurs all the time. Compensations must be conducted repeatedly. Second, each sensor of the same manufacturing model requires an individual drift model due to the fabricating discrepancy, which is inherent. But the sensors still share some inherent similarities, and the drift models could share some common knowledge among them. Third, drift models could differ even when a sensor is under different scenarios.

In this paper, we employ knowledge transfer to keep the modeling process as flexible as possible, so that our framework deals with discrepancies among sensors or scenarios with acceptable computing cost. Our contributions are threefold:

- An accelerating aging test board was designed and implemented to collect data for model evaluation. A dataset with diversified drift scenarios has been generated.
- DPC-IINK have been proposed to predict drift for sensors of same manufacturing model. It employs a feature layer extracting the similarities and a transfer layer sharing knowledge among sensors and scenar-

- *Zhang Yu, Tanaya Chaudhuri, Liu Pan, Wang Lu, Wu Min and Li Xiaoli are with Institute for Infocomm Research (I2R), Agency for Science, Technology and Research (A*STAR), Singapore, 138632.*

ios, so that the knowledge transfer can be conducted both in intra-node and inter-node way. With knowledge transfer, the prediction models can be retrained with as less data as possible but still achieve good performance.
- A framework employing a backend server and edge servers to continuously track the deployed sensors has been proposed to compensate drift for a cluster of sensor nodes. The edge server collects data and "refines" the drift models, while the backend server trains base model with aggregated datasets from the edge servers.

The rest of the paper is arranged as the following. Section 2 presents the literature review of transfer learning over gas sensors, while section 3 formulates our problem and proposes the methodology. Section 4 and 5 evaluate and conclude our proposal.

2 Literature Review

Based on data relations between the source and target domains, transfer learning methods can be divided into three major categories: 1) Instance-based transfer learning- it combines partial instances from source domain which are similar to the target domain with target domain instances to improve the specific weights for target task [13] [14] [15], 2) Mapping-based transfer learning- new instances from source domain and target domain are generated by similarity mapping even when the instances from the two domains are different; they are fit into the neural network for training to improve the task domain [16], and 3) Network-based transfer learning- it partially utilizes layer of neuron network in source domain; the partial pre-trained network structure and parameters are transferred to the target domain and are treated as a feature extractor. The network-based transfer learning is widely used in language processing and image classification [17] [18]. Based on the relationship between network structure and transferability, LeNet, AlexNet, VGG, Inception, ResNet are good choices for this type [19].

Transfer learning has been applied to gas sensor area recently. Marshall *et. al* proposed a fully convolutional network with transfer learning to segment gas leakage images for gas detection and smoke video which shared similar characteristics with the gas leakage visually [20]. Transfer learning could help supplement the limited available data. Zhong et. al proposed a feature mapping model to compute the feature representations based on transfer learning with CNN for gas turbine fault diagnosis under small sample conditions [12]. Transfer learning technique was also introduced into the field of soft sensor development for multi-grade chemical processes [21]. However, most of the existing approaches are used to transfer the knowledge to improve the accuracy of the target with limited data. The environmental change and sensor to sensor variation would be the obstacle for knowledge transfer.

3 Methodology

3.1 Problem Formulation

It is challenging to adequately learn a drift model for MOX sensors. This is due to the difficulty in mapping the variability among sensors into an adaptive versatile model. For example, in Fig. 1, all three sensors namely, Node1, Node2 and Node3 are of the same manufacturing model (Bosch BME 680) and are located under the same scenario. Yet, they drift with time in different ways as seen in the figure. Both manufacturing discrepancy and disruption in sensor measurement may lead to the difference in drift illustrated by Fig. 1. Such diversity is ubiquitous in MOX sensors. However, no matter how diversifying the drift response may be, there are still common patterns that could be learned. All our deployed BME 680 sensors share the same physical principles of measurement.

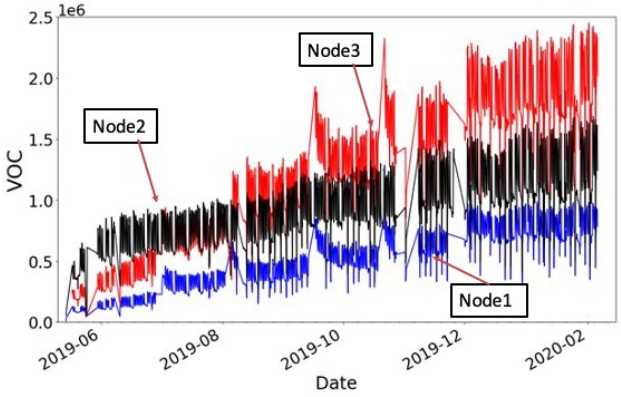

Fig. 1: Drift over time under same scenario for different MOX sensors of same type. 1) *Node2* and *node3* drift with different slopes, indicating that some sensors may drift faster than others. 2) Drift pattern might change after sensor measuring is interrupted as *node1* and *node3*.

Transfer learning improves a learner in one domain by sharing knowledge from a related domain. Considering a sensor as a domain, intuitively, transfer learning is capable of not only sharing the similarity among the MOX sensors of same manufacturing model, but also capturing difference of any individual sensor by retraining. The knowledge sharing saves computing cost for a cluster of low cost sensors under the same scenario, while the retraining refines the drift models with better prediction performance.

In our proposed DPC-IINK framework, we employ a neural network with (a) feature layers and (b) transfer layers. Let the linear operation in a layer l of the network be parameterized by w^l. With $x^l \in \mathbb{R}^d$ and $y^l \in \mathbb{R}$ being the input vector of layer l and the output label, respectively, in a forward network, $x^l = y^{l-1}$. The feature and the transfer layers of our network can be concisely represented by

$$y^l = \sigma^l(f(x^l; w^l)), \tag{1}$$

where $f(x^l; w^l)$ is a function of x^l and parameterized by the weight vector w^l. And σ^l is the activation function for the lth layer. Suppose we have a training data denoted as $(\mathbf{x}_i, y_i)$, where $i = \{1, ..., N\}$, the goal of learning is to minimize

$$\arg\min_{\mathbf{w}} \sum_{i=0}^{N} \mathscr{L}(y_i, \theta(\mathbf{x}_i; \mathbf{w})), \tag{2}$$

where $\theta(\mathbf{x}_i; \mathbf{w})$ is a function of $\mathbf{x}_i$, and the function is parameterized by $\mathbf{w}$, which denotes the set of all parameters.

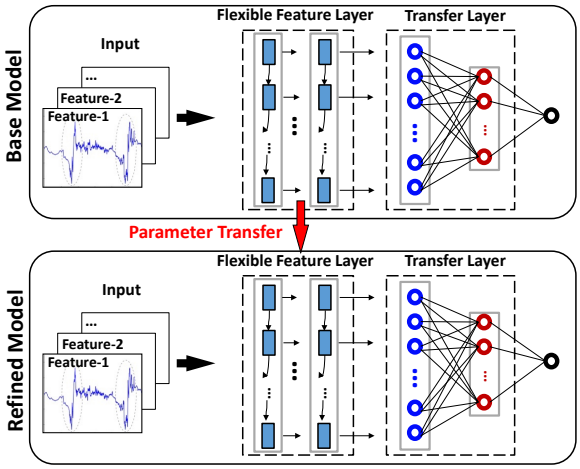

Fig. 2: Drift model architecture in DPC-IINK: interaction between base model and refined model.

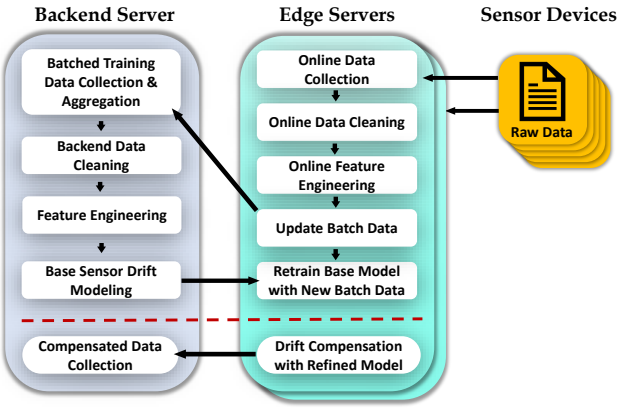

Fig. 3: DCP-IINK framework for large scale deployment of sensor drift prediction and compensation.

The loss function is denoted by $\mathscr{L}(\cdot)$. A typical choice is the least squares for regression.

According to the prior work by Yosinski *et. al* [19], the generic features tend to be transferable in lower layers, and the learned features become slightly domain-biased in upper layers. Therefore, our feature layers (lower layers) can be employed to extract generic features, while our transfer layers (upper layers) can be used to capture difference in the models. $\mathbf{w}^f$ and $\mathbf{w}^t$ denote the parameters for the feature layers and the transfer layers, respectively.

3.2 Transfer Learning Architecture for Drift Prediction

Developing and training an individual drift model for each MOX sensor is computationally expensive and infeasible for large-scale deployment. Unlike it, our proposed DPC-IINK predicts drifts for MOX sensors using transfer learning architecture as exhibited in Fig. 2. Since the lower layers tend to be transferable, the $\mathbf{w}^f$ in the feature layers can be used to learn the generic features. On the other hand, the $\mathbf{w}^t$ in the transfer layers is employed for the unique features of a MOX sensor. Unlike the data used by Yosinski *et. al*, our drift data is temporal time-series data. Although the CNN employed in [19] is not suitable for our data, intuitively we still could explore to extract generic features with other networks. In our architecture, the feature layers are quite flexible, where any deep learning network that works well on time-series data could be employed. As for the transfer layers, fully connected layers are employed.

With transfer learning, sensors could share generic features as illustrated in Fig. 2 with a relatively "stable" base model. The base model may be trained over any sensor under the same scenario, since all of our sensors are of same manufacturing model. Any of our sensors could be used to extract generic knowledge. Our architecture transfers the parameters of $\mathbf{w}^f$ in the feature layers by freezing the $\mathbf{w}^f$ in the base model, and retrain the $\mathbf{w}^t$ in the transfer layers to generate a refined model which is fine-tuned for drift prediction over each sensor. The retraining actually happens only over the $\mathbf{w}^t$ in the transfer layers. Our method experiences lower training cost as the training of full parameters happens on only the base model. It makes sense when MOX sensors are deployed with large quantity, and the edge server has limited computing capacity.

In our architecture, we also transfer knowledge among domains. A domain stands for a continuous measurement by a gas sensor. The drift upon continuous measurement could be modeled very well by deep learning algorithms as seen in our previous work in [22]. However, a discontinuous measurement due to external operation factors could worsen the performance of such specifically trained model. Our proposed approach could solve this problem through knowledge transfer between the domains, making it an adaptive and standalone method.

3.3 Feature Layers and Transfer Layers

Considering our time-series data, the feature layers are currently comprised of a long short-term memory (LSTM) or gated recurrent unit (GRU) [23] network with different configurations, instead of the CNN network used by Yosinski *et. al* [19]. The $\mathbf{w}^f$ differs when the feature layers employ different networks. Taking GRU as an example, the $\mathbf{w}^f$ includes the parameter set presented in the following computation of gates $\mathbf{z}_t$ and $\mathbf{r}_t$, cell value $\tilde{\mathbf{h}}_t$, and hidden value $\mathbf{h}_t$:

$$\begin{aligned}
\mathbf{z}_t &= \sigma(\mathbf{w}^{xz}\mathbf{x}_t + \mathbf{w}^{hz}\mathbf{h}_{t-1}) \\
\mathbf{r}_t &= \sigma(\mathbf{w}^{xr}\mathbf{x}_t + \mathbf{w}^{hr}\mathbf{h}_{t-1}) \\
\tilde{\mathbf{h}}_t &= \tanh(\mathbf{w}^{zh}\mathbf{x}_t + \mathbf{w}^{hh}(\mathbf{h}_{t-1} \odot \mathbf{r}_t)) \\
\mathbf{h}_t &= (1 - \mathbf{z}_t) \odot \tilde{\mathbf{h}}_t + \mathbf{z}_t \odot \mathbf{h}_{t-1},
\end{aligned} \tag{3}$$

where $\mathbf{h}_t$ is the hidden state. $\mathbf{w}^{xz}$, $\mathbf{w}^{hz}$, $\mathbf{w}^{xr}$, $\mathbf{w}^{hr}$, $\mathbf{w}^{zh}$ and $\mathbf{w}^{hh}$ are weight parameter set for GRU [23]. $\tilde{\mathbf{h}}_t$ is the candidate hidden state. σ represents sigmoid function and $\odot$ denotes Hadamard production.

In the refined models, the trainable $\mathbf{w}^t$ in the transfer layers stands for parameters in full connected network. The full connected networks capture individual features for each sensor or domain.

3.4 Drift Model Deployment

In the case of large scale deployment of low-cost ambient sensors, the maintenance cost of calibration is a deep con-

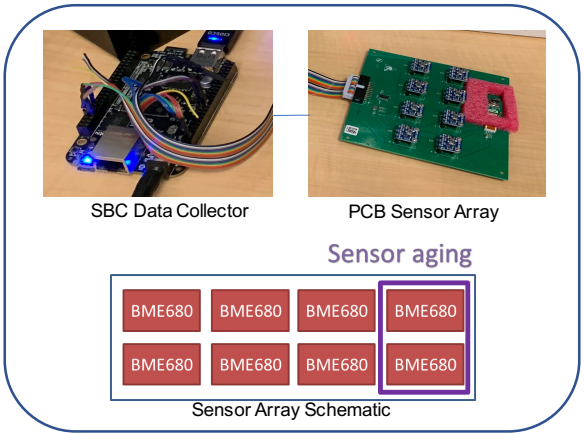

Fig. 4: Sensor Array for Aging Experiments.

cern. A drift prediction model driven by data may reduce this cost. Drift consistently occurs gradually, but it is significant over a long period of time. Although the performance of a drift model deteriorates over time, the drift model is not updated frequently due to the smaller gradual changes [24]. When a significant drift has occurred over a long time, the drift model is retrained to adapt. Our proposed DPC-IINK framework shown in Fig. 3 is designed to cost-effectively maintain such large scale deployed low-cost sensors.

The framework employs a backend server and multiple edge servers to transfer knowledge among sensors with a base model. On one hand, the powerful backend server trains and updates a base model with the merged new data provided by the edge servers. On the other hand, the edge servers employ the base model to partially retrain the drift models for specific sensors with smaller datasets from each sensor. Finally, the edge servers calibrate the deployed sensors with newly retrained drift models. With transfer learning, our framework compensates the drift with relatively low cost and considerable accuracy.

4 Evaluation

In our DPC-IINK framework, the sensor drift is compensated using a drift prediction model, so the model is crucial for the framework. In this section, we evaluate the performance of our proposed drift models. We have designed and implemented a PCB mounted with an array of BME680 sensors. The PCB supports to plug in/out them seamlessly and inter-collects data between the sensors and a background server. An aging experiment is also designed to accelerate sensor drifts. With some sensors working under either ambient environment or oven baking, we collected data for the discontinuous drift, which is different from the data of continuous drift collected in CGDA [22]. We evaluate both accuracy and computing complexity of the drift models over both the continuous and discontinuous data.

4.1 Discontinuous Drift

4.1.1 Data Collection

Our designed sensor array is depicted in Fig. 4. It is equipped with 8 Bosch-BME680 sensors, and they are numbered from 601 to 608. The array monitors measurements of VOC, air temperature, air pressure and humidity. A single-board computer (SBC) connects with our PCB by a cable. The PCB is capable of simultaneously collecting data from the array at different sampling rates. The SBC pushes the collected data to an edge server.

Although the BME680 sensors keep drifting naturally even under ambient environment, an artificial aging experiment is designed to create more drifting scenarios for performance evaluation. Unlike the naturally continuous drift in CGDA [22], the aging experiment in our study both accelerates and disrupts the natural drift process. Using our flexible sensor array, 2 out of 8 sensors are plugged out irregularly. They are put into an oven for baking, while the other 6 sensors always work normally for comparison. The sensor drifts in Fig. 5 validates that the artificial aging experiments make the sensors achieve comparatively larger and disrupted drifts.

TABLE 1: Oven-baking settings for artificial aging experiments.

Testing frequency:	once more than 1 month
Operating time:	3 days
Min oven temperature:	30°C
Max oven temperature:	80°C
Recess operation period:	1 week

The technical settings for the accelerating aging experiments are described in TABLE 1. The 2 sensors in oven usually go through 3 temperature cycles each day for more than 3 days. The air pressure of the oven is kept the same as environment. After oven-baking, the 2 sensors are mounted back onto the PCB. There would be a cooling down procedure before the data collection on board is resumed.

4.1.2 Data Description

The discontinuous data from the array was collected for around 8 months. With multiple nodes in the array and oven-baking experiments, it provides more scenarios for drift prediction.

Fig. 5 illustrates the responses from the aging experiments. Firstly, it shows that the sensor with oven-baking encounters comparatively more significant drift than the normal sensors. Secondly, as illustrated in Fig. 5, the monitored period for the aging experiments by oven-baking is divided into 5 stages (domains) namely, a) baseline stage, b) source stage, c) stage 1, d) stage 2 and, e) stage 3. The first 5 days before the first oven-baking is considered as the baseline stage. Oven-baking experiments are performed before each of the subsequent stages. To describe the influence of sensor drift, we computed drift ratio of VOC responses, which were based on the detected circuit resistance of the BME680 sensor. Firstly, we computed the hourly VOC response during the baseline stage, so there are 24 responses per day. Next, we averaged the hourly responses over the 5 days of the baseline stage, resulting in a single sequence of 24 mean responses, which we consider as our baseline for calculating drift ratio. Thereafter, for the subsequent stages, the drift ratio is computed as the ratio of VOC responses between the current stage and the baseline stage. Along the last 4 stages, 24 drift ratios are generated each day by dividing the baseline samples accordingly. Our

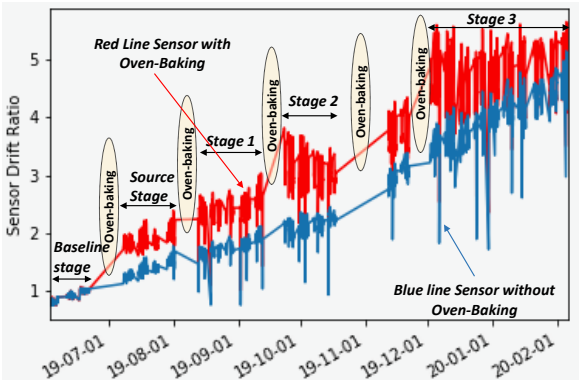

Fig. 5: Drift ratio of VOC responses for the sensors with (red) and without (blue) artificial aging experiments.

drift prediction model employs the drift ratio as the model's output.

Except the baseline stage, all the stages in each sensor are considered as the scenarios or domains for prediction models. In CGDA [22], drifting models are required to be updated over time even for a continuous drift. The disruption along stages in Fig. 5 indicates that an individual drift model is required for each stage. The whole duration is segmented into 4 scenarios. A stage could be either a source or target domain for our transfer learning models. In our case, the source stage is used for base model training, while stages 1, 2, 3 are used for retraining and testing (Fig. 6).

4.2 Prediction for Discontinuous Drift

Drift may behave in either continuous or discontinuous way due to intrinsic changes, but in either way it still has some shared features which are useful for a cluster of models for sensors under similar scenarios. Transfer learning is used to take advantage of shared knowledge among drift models.

Considering the discontinuous drift, Fig. 6 provides an illustration for intra-node and inter-node transfer learning. The intra-node transfer learning is within the same sensor, node 601, while the inter-node transfer learning is from source domain of node 601 to all stages in the nodes other than node 601. With transfer learning, we have a base model trained in source domain and refined models retrained in target domains. The evaluation compares model performance with 3 different flexible feature layers, namely single-LSTM, dual-LSTM and GRU.

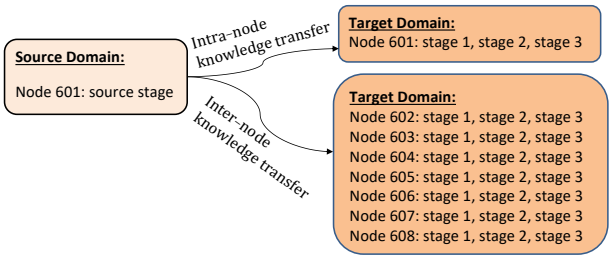

Fig. 6: Illustration of intra-node and inter-node transfer learning between source and target domains for our evaluation.

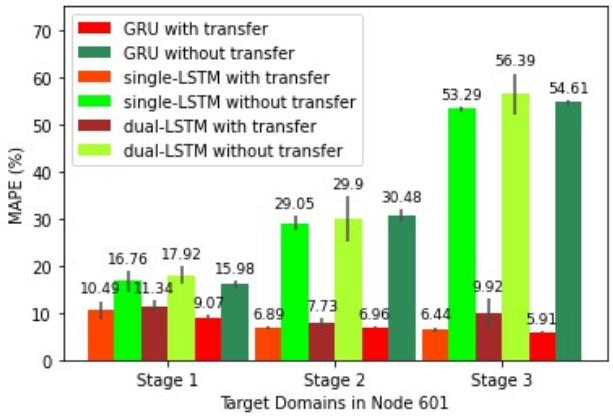

Fig. 7: Comparison of MAPE (%) for intra-node transfer learning models.

4.2.1 *Intra-node Transfer Learning*

The intra-node drift prediction performance for node 601 is presented in Fig. 7. It shows the mean absolute percentage errors (MAPE) for the models equipped with single-LSTM, dual-LSTM and GRU feature layers. The predictions are conducted with the same deep learning structure shown in Fig. 2, either with or without transfer learning.

As seen in Fig. 7, our proposed transfer learning approaches consistently outperform those without transfer learning. The mean MAPEs for the transfer learning models with single-LSTM, dual-LSTM and GRU feature layers in stage 1 are 10.49%, 11.34% and 9.07%, respectively. From stage 1 to stage 3, where the node 601 is oven-baked three times for accelerating aging and the drift is abruptly increased with oven-baking, the mean MAPE gaps between the transfer learning and non transfer learning approaches are expanding from 6.28% ($15.58\% - 10.3\%$) to 47.34% ($54.76\% - 7.42\%$). The results indicate that our proposed transfer learning approaches successfully capture the intrinsic changes inside the sensor regardless of the oven-baking experiments described in section 4.1.1 and the discontinuous drift. Our proposal exhibits great flexibility among all kinds of scenarios.

As for the feature layers, GRU performs best for the drift in node 601, with a mean MAPE of 7.31% among the 3 stages. The results suggest that a single layer GRU or LSTM is enough to capture the shared knowledge among domains, a deeper feature layer does not help in this case.

4.2.2 *Inter-node Transfer Learning*

Besides intra-node transfer learning, Table 2 presents the MAPE performances of inter-node transfer learning when the drift models transfer knowledge from the source stage of node 601 to the other nodes on board. Similar to the intra-node results, the transfer learning approaches consistently outperform the non-transfer ones. For each of the nodes, the performance gaps are getting larger. Taking node 603 with the GRU feature layer as an example, the gap increases from (9.32% vs. 16.09%) to (5.95% vs. 54.72%), since the non-transfer approaches are not capable of capturing the intrinsic drift changes.

Table 2 also shows that among all nodes, our approaches achieve the lowest mean MAPEs of 9.25%, 6.95% and 5.87% for stage 1, 2 and 3, respectively. It indicates that our models have good drift prediction capacity even when the models work on discontinuous drift and the knowledge is shared from other sensor node. Our transfer learning approaches can be "adapted" to other nodes flexibly.

Comparing different feature layers, the inter-node transfer learning indicates that the multi-layered LSTM may not work better than the single layered one.

TABLE 2: Comparison of MAPE (%) among stages in different nodes for inter-node transfer learning.

Target	602	603	604	**604**	606	607	608	Mean
Stage	Feature Layer: Single-LSTM							
S1	8.83	9.71	9.96	9.40	9.69	9.25	9.86	9.53
without	15.85	16.67	16.29	16.10	16.53	16.03	17.07	16.36
S2	6.72	7.02	6.72	6.72	6.67	6.86	6.78	**6.78**
without	28.26	29.08	27.82	27.82	28.31	29.61	29.79	28.67
S3	6.04	6.08	6.15	6.15	6.10	6.35	6.18	6.15
without	53.69	53.26	53.91	53.91	53.62	53.79	53.00	53.60
Feature Layer: Dual-LSTM								
S1	13.40	17.25	12.21	12.21	12.13	15.02	14.23	13.78
without	19.90	26.21	19.92	19.92	19.25	23.83	21.72	21.54
S2	7.87	8.70	8.73	8.54	8.68	8.10	8.17	8.40
without	30.34	34.42	32.24	33.57	41.66	31.83	32.31	33.77
S3	7.51	6.79	7.71	7.71	7.58	7.14	7.08	7.36
without	53.71	53.41	54.95	54.95	54.92	54.53	54.38	54.41
Feature Layer: GRU								
S1	9.42	9.32	8.87	8.91	9.70	9.37	9.15	**9.25**
without	15.49	16.09	16.12	15.96	15.18	15.28	15.79	15.70
S2	6.94	6.87	7.09	7.09	6.80	6.86	7.03	6.95
without	30.12	30.37	29.55	29.55	30.44	29.74	30.40	30.02
S3	5.82	5.95	5.83	5.83	5.93	5.90	5.83	**5.87**
without	54.71	54.72	54.78	54.78	54.63	54.50	54.77	54.70

4.3 Prediction for Continuous Drift

In addition to discontinuous drift, we also employed our proposed method on the case of continuous drift used in CGDA [22] for comparison purpose. For a lot of problems, normally more data provides more information and better performance. But it is not necessarily the case for drift problem where the intrinsic aging keeps occurring over time. The "in-time" data is more important.

Table 3 presents chronological results of 10 months for the 3 feature layers using DPC-IINK. The base model employs around 2 months' data, while each retraining only employs 5 days data for "adaptation". The results in Table 3 shows that our method achieves a mean MAPE of 5.42% with the feature layer of GRU, which outperforms the best result of CGDA [22] at 5.50%. Compared with CGDA, the models used less days' data for training, but it still outperforms CGDA. Less data means less resource consumption, which is important for edge computing.

Compared among the 3 feature layers, GRU performs better than the other two. It again indicates that a multi-layered LSTM does not necessarily have more benefits for this study.

Computing time is am important aspect in real-life where multiple sensors are deployed across the network. It is highly desirable that the drift prediction model is able to predict the sensor drift using both minimum computation and minimum training time. While the CGDA model requires that a new separate model is trained for each sensor node and needs several months of data for training, our method uses transfer learning where each retraining requires only 5 days data for "adaptation". Thus, our model has lesser computational cost as well as better predictive accuracy. This is beneficial for the edge computing as well.

TABLE 3: Chronological comparison of MAPE (%) among different feature layers for continuous drift with transfer learning.

Test month	Feature layer		
	GRU	S-LSTM	D-LSTM
M1	6.65	6.7	9.16
M2	6.77	5.35	5.87
M3	6.3	8.57	7.11
M4	4.59	6.75	7.9
M5	5.4	6.49	5.79
M6	6.44	7.66	8.51
M7	3.61	4.58	4.38
M8	4.0	7.98	6.47
M9	4.29	6.9	6.42
M10	6.18	11.0	13.01
Mean	**5.42**	7.20	7.46

5 Conclusion

Drift in MOX sensors is an inherent challenge and leads to undesirable consequences. This paper has proposed DPC-IINK, a novel framework to predict drift with transfer learning for the gas sensors of same manufacturing model. The deep learning mechanism with knowledge transfer can be conducted in both intra-node and inter-node ways. The transfer learning employs a feature layer which extracts the similarities among the sensors and a transfer layer which transfers knowledge among them. With the knowledge transfer, our models can be retrained with less data and in turn less computing efforts. Our proposed framework also comprises a backend server and edge servers to continuously collect and aggregate data from large-scale deployed sensors. With the knowledge transfer, the edge server retrains the drift models and "adapts" the base model for each specific sensor, while the backend server only trains the base model. The experimental results show that our transfer learning method achieves better performance both in the intra-node and inter-node ways, as well as for the continuous and discontinuous data, compared with methods without knowledge transfer.

Acknowledgments

The authors would like to thank our colleague Liu Guimei for her discussion regarding AutoML. We are also grateful to Raymond Keh and Ngai Kim Hoong who help with implementing and scrutinising the PCB boards regarding data collection.

References

[1] A. M. Collier-Oxandale, J. Thorson, H. Halliday, J. Milford, and M. Hannigan, "Understanding the ability of low-cost MOx sensors to quantify ambient VOCs," *Atmos. Meas. Tech.*, vol. 12, no. 3, pp. 1441–1460, 2019.

[2] C. Wang, L. Yin, L. Zhang, D. Xiang, and R. Gao, "Metal oxide gas sensors: Sensitivity and influencing factors," *Sensors*, vol. 10, no. 3, pp. 2088–2106, 2010.

[3] T. Seiyama, A. Kato, K. Fujiishi, and M. Nagatani, "A new detector for gaseous components using semiconductive thin films," *Analytical Chemistry*, vol. 34, pp. 1502–1503, 1962.

[4] D. Kumar, S. Rajasegarar, and M. Palaniswami, "Automatic sensor drift detection and correction using spatial kriging and kalman filtering," in *Proceedings of the 2013 IEEE International Conference on Distributed Computing in Sensor Systems*.

[5] S. Di Carlo and M. Falasconi, *Drift correction methods for gas chemical sensors in artificial olfaction systems: Techniques and challenges*. INTECH Open Access Publisher, 2012.

[6] E. Hines, E. Llobet, and J. Gardner, "Electronic noses: A review of signal processing techniques," *IEE Proceedings-Circuits, Devices and Systems*, vol. 146, no. 6, pp. 297–310, 1999.

[7] S. Brahim-Belhouari, A. Bermak, and P. C. Chan, "Gas identification with microelectronic gas sensor in presence of drift using robust gmm," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, 2004, pp. V–833.

[8] Q. Liu, X. Li, M. Ye, S. S. Ge, and X. Du, "Drift compensation for electronic nose by semi-supervised domain adaption," *IEEE Sensors Journal*, vol. 14, no. 3, pp. 657–665, 2013.

[9] Q. Liu, X. Hu, M. Ye, X. Cheng, and F. Li, "Gas recognition under sensor drift by using deep learning," *International Journal of Intelligent Systems*, vol. 30, no. 8, pp. 907–922, 2015.

[10] K. Sothivelr, F. Bender, F. Josse, E. E. Yaz, A. J. Ricco, and R. E. Mohler, "Online chemical sensor signal processing using estimation theory: Quantification of binary mixtures of organic compounds in the presence of linear baseline drift and outliers," *IEEE Sens. J.*, vol. 16, no. 3, pp. 750–761, 2015.

[11] S. Marco and A. Gutierrez-Galvez, "Signal and data processing for machine olfaction and chemical sensing: A review," *IEEE Sensors Journal*, vol. 12, no. 11, pp. 3189–3214, 2012.

[12] S. sheng Zhong, S. Fu, and L. Lin, "A novel gas turbine fault diagnosis method based on transfer learning with cnn," *Measurement*, vol. 137, pp. 435–453, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0263224119300405

[13] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proceedings of the 24th International Conference on Machine Learning*, ser. ICML '07. New York, NY, USA: Association for Computing Machinery, 2007, p. 193200. [Online]. Available: https://doi.org/10.1145/1273496.1273521

[14] Y. Yao and G. Doretto, "Boosting for transfer learning with multiple sources," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1855–1862.

[15] C. Wan, R. Pan, and J. Li, "Bi-weighting domain adaptation for cross-language text classification," in *IJCAI*, 2011.

[16] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ser. ICML'15. JMLR.org, 2015, p. 97105.

[17] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1717–1724.

[18] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. Red Hook, NY, USA: Curran Associates Inc., 2016, p. 136144.

[19] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems*, vol. 27. Curran Associates, Inc., 2014.

[20] Marshall, J.-S. Park, and J.-K. Song, "Fcn based gas leakage segmentation and improvement using transfer learning," in *2019 IEEE Student Conference on Electric Machines and Systems (SCEMS 2019)*, 2019, pp. 1–4.

[21] Y. Liu, Y. Chao, L. Kaixin, B. Chen, and Y. Yao, "Domain adaptation transfer learning soft sensor for product quality prediction," *Chemometrics and Intelligent Laboratory Systems*, vol. 192, p. 103813, 07 2019.

[22] T. Chaudhuri, M. Wu, Y. Zhang, P. Liu, and X. Li, "An attention-based deep sequential gru model for sensor drift compensation," *IEEE Sensors Journal*, vol. 21, no. 6, pp. 7908–7917, 2021.

[23] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *arXiv:1406.1078*, 2014.

[24] S. Hoi, D. Sahoo, J. Lu, and P. Zhao, "Online learning: A comprehensive survey," *ArXiv*, vol. abs/1802.02871, 2018.