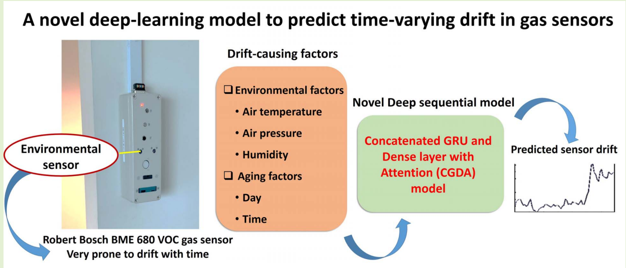
An Attention-Based Deep Sequential GRU Model for Sensor Drift Compensation

Tanaya Chaudhuri, Min Wu, *Senior Member, IEEE*, Yu Zhang, Pan Liu, and Xiaoli Li, *Senior Member, IEEE*

***Abstract*—Sensor accuracy is vital for the reliability of sensing applications. However, sensor drift is a common problem that leads to inaccurate measurement readings. Owing to aging and environmental variation, chemical gas sensors in particular are quite susceptible to drift with time. Existing solutions may not address the temporal complex aspect of drift, which a sequential deep learning approach could capture. This article proposes a novel deep sequential model named Concatenated GRU & Dense layer with Attention (CGDA) for drift compensation in low-cost gas sensors. Concatenation of a stacked GRU (Gated Recurrent Unit) block and a dense layer is integrated with an attention network, that accurately predicts the hourly drift sequence for an entire day. The stacked GRU extracts useful temporal features layer by layer capturing the time dependencies at a low computational expense, while the dense layer helps in retention of handcrafted feature knowledge, and the attention mechanism facilitates adequate weight assignment and elaborate information mapping. The CGDA model achieves a significant mean accuracy over 93%, outperforming several state-of-the-art shallow and deep learning models besides its ablated variants. It can greatly enhance the reliability of sensors in real-world applications.**


A novel deep-learning model to predict time-varying drift in gas sensors
Environmental sensor
Robert Bosch BME 680 VOC gas sensor
Very prone to drift with time
Drift-causing factors
❑ Environmental factors
• Air temperature
• Air pressure
• Humidity
❑ Aging factors
• Day
• Time
Novel Deep sequential model
Concatenated GRU and Dense layer with Attention (CGDA) model
Predicted sensor drift

***Index Terms*—Attention network, deep learning, drift compensation, gas sensor drift, gated recurrent unit.**

I. Introduction

GAS sensors are substantially significant given their widespread applications ranging from environmental monitoring to air quality and pollution checks, biometrics, food and agriculture, medical diagnosis and robotics. However, chemical changes by aging and environmental variation commonly lead to sensor drift with the passage of time. This results in inaccurate measurement readings and deterioration of sensor reliability [1]. Low-cost metal-oxide gas sensors in particular are attractive owing to their cost-effectiveness, operational ease and spatial coverage, but they get problematic with time by their susceptibility to drift [2], [3]. Although several methods have been developed over the past decades, drift compensation still remains a challenge.

Drift compensation approaches can be categorised as (1) univariate approaches, (2) multivariate approaches and (3) adaptive approaches [4], [5]. Univariate approaches comprise sensor signal processing such as baseline manipulation, frequency domain filtering, multiplicative correction or estimation theory [6]–[8]. These are simplified methods where compensation is applied to each sensor independently. Their drawbacks are that firstly, they may not be adequate when the drift is complex in nature which is often the real case, and secondly, they are sensitive to sample rate changes.

Multivariate approaches comprise cluster analysis such as self-organizing maps [9], signal correction and deflation by dimension reduction methods [10], or system identification methods [11]. Their drawbacks are that firstly, they need frequent sampling and secondly, these approaches may not accurately separate the undesired component from the useful components in the case of complex drift effect with noise [12].

On the other hand, the adaptive approaches comprise machine learning algorithms. These approaches are popular because they allow the flexibility to add non-linear relationships in the drift model, without making any prior assumptions of the drift signal. Some of the related works on machine learning based drift correction are discussed in the next section.

A. Related Works on Adaptive Approaches

An electronic-nose (e-nose) in the machine olfaction system serves the purpose of odor recognition and has wide applications. However, drift in the gas sensors affects their

Manuscript received October 9, 2020; revised December 5, 2020; accepted December 6, 2020. Date of publication December 14, 2020; date of current version February 17, 2021. This work was supported by the Agency for Science, Technology and Research (A*STAR) under the Republic of Singapore's Research Innovation Enterprise (RIE2020) Advanced Manufacturing and Engineering Industry Alignment Fund - Pre-Positioning Programme (IAF-PP) under Grant A1789a0024. The associate editor coordinating the review of this article and approving it for publication was Dr. Irene Taurino. *(Corresponding author: Min Wu.)*

The authors are with the Institute for Infocomm Research (I2R), A*STAR, Singapore 138632 (e-mail: tanaya_chaudhuri@i2r.a-star.edu.sg; wumin@i2r.a-star.edu.sg; zhang_yu@i2r.a-star.edu.sg; liu_pan@i2r.a-star.edu.sg; xlli@i2r.a-star.edu.sg).

Digital Object Identifier 10.1109/JSEN.2020.3044388

measurements and thereby hampers the reliability of the e-nose predictions. Marco *et. al* provides an excellent review of the machine learning pattern recognition methods used for gas discrimination in sensor arrays [4]. Many of these methods seek to overcome the time-dependent drift while classifying gases. In this respect, Support Vector Machine (SVM) classifiers have been widely used [13], [14]. Vergara *et. al* used a weighted combination of SVMs trained at different points of time which helped to counteract the drift [13]. Verma *et. al* [14] added regularization to this weighted ensemble which further improved accuracy and reduced over-fitting as well. Rehman *et. al* [15] proposed heuristic Random Forest (RF) to classify gases where RF learning was embedded with particle swarm optimization to compensate the drift. Brahim *et. a*l [16] adopted Gaussian Mixture Models (GMM) to develop a gas classifier which counteracts the drift by extracting robust features using a simulated drift.

Recently, deep learning has been explored for gas recognition with drift suppression. Tian *et. al* [17] designed a gaussian Deep Belief Network (DBN) to identify gases under sensor drift. It used the DBN as a non-linear function to learn the drift based differences between the source and target domains. Liu *et. al* [18] adopted DBN and stacked sparse autoencoder (sSAE) to extract deep features, and these features were later used to train a gas classifier that could reduce the drift. Luo *et. al* [19] adopted DBN to extract depth characteristics of the drifted gas sensor data. Then, the DBN model was coupled with an SVM which improved gas recognition under drift. Altogether, several such drift correcting machine learning classifiers have progressed the odor recognition arena over the last decade.

The drifted data has a different projected distribution than the clean data. Capturing this difference helps to separate the two types of data and reduce the drift through subspace learning. Zhang *et. al* [20] proposed an unsupervised subspace projection approach that reduced the drift by projecting the data onto a new common subspace using principal component analysis (PCA). Such projection approaches were extended to transfer learning based feature adaptation in [21]. The drift was compensated by aligning the principal component subspace between the clean and the drifted data. Similarly, these subspace learning capabilities were extended to cross-domain discriminative learning in [22]. Odor recognition models could be transferred between different e-noses using this method. Such dimensionality reduction methods are often incorporated as a pre-processing technique in gas classifiers.

In this respect, several transfer learning based domain adaptation methods have been proposed. Yan *et. al* [23] proposed a drift-correcting autoencoder using transfer learning while Zhang *et. al* [24] proposed domain-adaptation Extreme Learning Machines (ELM) to suppress drift. Liu *et. al* proposed a semi-supervised domain adaption method to compensate drift using a weighted geodesic flow kernel (GFK) and a classifier with manifold regularization [25].

In most of these adaptive methods, it is assumed that the drift trend can be traced through its direction in projected subspace or its distribution. However, long-term drift do not always have a consistent direction trend or a regular pattern [12]. Very few studies such as [26] have considered this aspect and explored the time-series prediction of drift signal. In this study [26], Zhang *et. al* developed a drift prediction model using chaotic time series prediction method based on phase space reconstruction (PSR) and single-layered neural networks (SNN). Mumyakmaz *et. al* [27] employed two neural networks with the first as a classifier to identify the gases, and the second to predict the concentration ratios of the gases in a sensor array. De Vito *et. al* [28] applied time delay support vector regressor (SVR) and time delay neural network to predict the real-time gas concentrations in sensor array. However, the sensor drift was not studied in both these works.

B. Contributions of This Paper

Although a great many excellent works have been done on adaptive approaches to compensate drift, there are a few concerns. Firstly, most of the existing machine learning approaches are pattern recognition algorithms aimed at classification of gases in sensor arrays. Very few studies have explored direct prediction of the drift values. In this study, we aim at building a model focused on the prediction of sensor drift. The predicted drift can be used to correct sensor readings, which can make applications like gas recognition more reliable.

Secondly, the parameters that cause sensor drift have rarely been incorporated in the classification or prediction models. Based on the causes, sensor drift is categorised into two types: (a) The 'first-order' or 'real' drift, results from sensitivity changes in chemical sensors due to aging and poisoning. It often occurs over long periods of time. (b) The 'second-order' drift, may result from the slow changes in the external environment such as ambient temperature, pressure etc. It may occur over short time periods, but the fluctuations can significantly alter sensor response [1], [6], [9], [13]. Moreover, practically it is extremely difficult to empirically differentiate between these two drifts [13]. In this study, we incorporate these aging and environmental drift-causing factors into the feature space of our drift prediction model.

Thirdly, most of the current machine learning based drift solutions are shallow learning methods. The few methods that have explored deep learning are focused on gas classification as well. Deep learning has definitive power to capture the complexities in data and therefore would be more suitable for handling complex drift signals. Furthermore, given the 'temporal' nature of 'time'-varying drift, sequential neural networks such as the Gated Recurrent Unit (GRU) network [29] could be apt for modelling the drift. Several advantages of the GRU such as the ability to capture time dependencies, temporal feature learning and low computational cost make it suitable for modelling the drift in our study. However, modelling longer sequences can suffer from information loss. The attention mechanism [30] can prevent such loss, besides providing better interpretability and appropriate information mapping. Attention mechanism is a recent revolutionary concept in deep learning. It selectively pays 'attention' to the most relevant information in deep neural networks while ignoring the non-relevant parts, by assigning appropriate weights. It can help

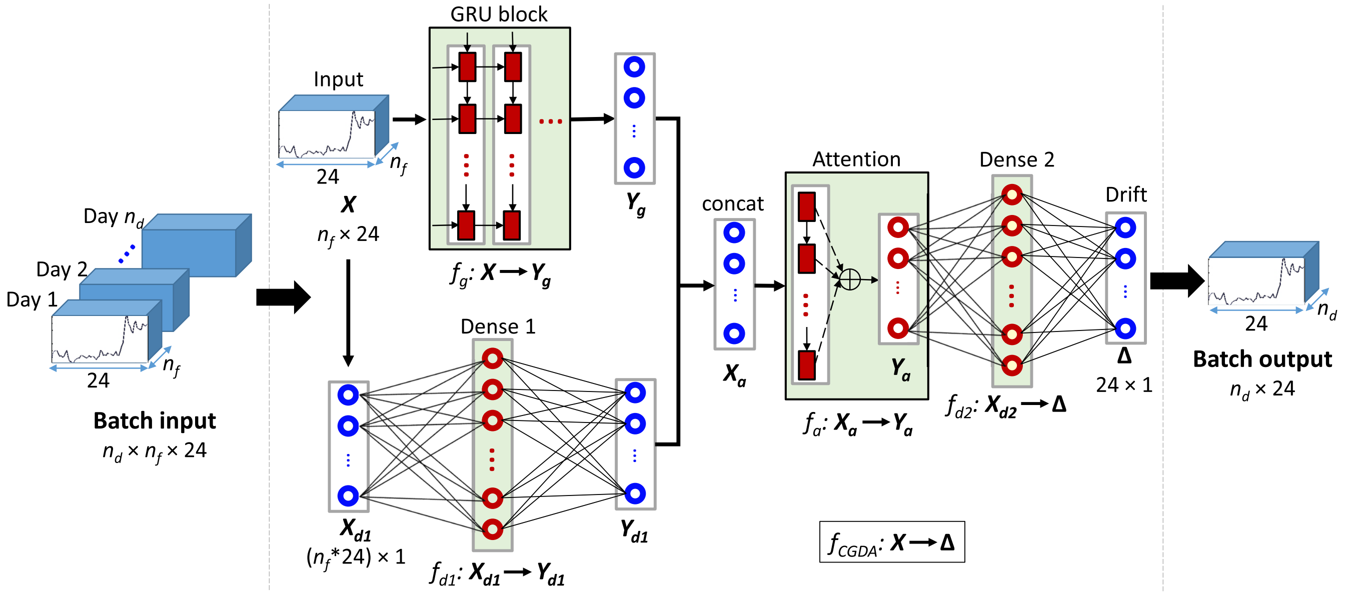


Fig. 1. Architecture of the proposed CGDA model for drift prediction. Here, f_{CGDA}: the overall CGDA model, X: the CGDA model input, $\mathbf{\Delta}$: the CGDA model drift output, n_d: the number of days (batch size), n_f: the number of input features, and '24' represents the number of hours in a day. Enclosed dotted portion illustrates a single day (sample) for simplicity purpose, where f_g: the GRU network, f_{d1}: the first dense network, f_a: the attention network, and f_{d2}: the second dense network. $\boldsymbol{X}, \boldsymbol{X_{d1}}, \boldsymbol{X_a}, \boldsymbol{X_{d2}}$ are the inputs and $\boldsymbol{Y_g}, \boldsymbol{Y_{d1}}, \boldsymbol{Y_a}, \mathbf{\Delta}$ are the outputs of f_g, f_{d1}, f_a and f_{d2} networks, respectively.

our model focus on the most relevant hidden states of the input sequences for drift prediction.

In this article, we propose a deep learning based sequential model for drift compensation in low cost gas sensors, incorporating the aging and environmental drift-causing factors. The specific contributions are:

- We propose a novel deep sequential model termed as the Concatenated GRU and Dense layer with Attention (CGDA) model for drift compensation in low-cost gas sensors. Concatenation of a stacked GRU block and a dense layer is integrated with attention mechanism, that accurately predicts the drift sequence for an entire day.
- The stacked GRU block extracts useful temporal features layer by layer capturing time dependencies at a low computational expense, which enhances the drift prediction.
- The dense layer helps in better retention of feature information by supporting handcrafted features, while the attention network facilitates adequate weight assignment and prevents information loss along the input sequence; this aids in appropriate representation of the drift and further enhances its prediction.
- Intensive feature extraction by the GRU and dense layers, and elaborate information mapping by the attention network improve the model's drift prediction ability. It's efficacy is validated through a comprehensive chronological comparison with 8 different state-of-the-art shallow and deep learning models including an ablation study.
- The input feature space addresses the environmental and aging factors of drift. The study findings are corroborated through an experimental dataset generated by our institute by deploying a robust sensor-network that collected data over 4 to 14 months duration at indoor and semi-indoor locations within office premises.

The rest of the paper is organized as follows: Section II introduces the proposed CGDA model's framework. Section III describes the experimental settings and the sensor dataset followed by the results and discussions in section IV. Finally, section V highlights the concluding remarks and future work.

II. Proposed CGDA Model

This section describes the framework of the **C**oncatenated **G**RU and **D**ense layer with **A**ttention (CGDA) model.

A. Overall Framework

The architecture of the proposed CGDA model is illustrated in Fig. 1. Broadly, the CGDA model f_{CGDA} comprises a GRU block that is concatenated with a dense layer, and followed by an attention network.

The input feature space comprises of a batch of data over n_d number of days. Each day is a sample in the data. Each sample (day) consists of a time-series sequence of n_f different features. The length of each sequence is 24, considering there are 24 hourly points in a day corresponding to time-slots 1, 2, .. 24. Considering n_d = number of days in the data (batch size) and n_f = number of features, the input feature tensor set $\boldsymbol{X} \in \mathbb{R}^{n_d \times n_f \times 24}$ is denoted as:

$$\boldsymbol{X}^{1,2,..n_d}_{1,2,..n_f}[t] \text{ where } t = \{1, 2, ..24\}$$

We consider the target drift $\mathbf{\Delta} \in \mathbb{R}^{n_d \times 24}$ as a time sequence of hourly drift values along a day which is denoted as:

$$\mathbf{\Delta}^k = \{\delta^k_1, \delta^k_2, \ldots \delta^k_{24}\}, \text{ where } k = \{1, 2, ..n_d\}$$

The CGDA model can be symbolically represented as f_{CGDA} : $\boldsymbol{X} \rightarrow \mathbf{\Delta}$.

B. GRU Block

The feature input tensor is fed to the GRU network f_g. Note that the GRU may be single or multi-layered (stacked) depending upon hyperparameter optimization. For a given time-step t, the input is $\boldsymbol{X}[t] \in \mathbb{R}^{n_d \times h}$ and the computations are:

$$\boldsymbol{R}[t] = \sigma\left(\boldsymbol{X}[t]\boldsymbol{W}_{xr} + \boldsymbol{H}[t-1]\boldsymbol{W}_{hr} + \boldsymbol{b}_r\right) \quad (1)$$

$$\boldsymbol{Z}[t] = \sigma\left(\boldsymbol{X}[t]\boldsymbol{W}_{xz} + \boldsymbol{H}[t-1]\boldsymbol{W}_{hz} + \boldsymbol{b}_z\right) \quad (2)$$

$$\tilde{\boldsymbol{H}}[t] = \tanh\left(\boldsymbol{X}[t]\boldsymbol{W}_{zh} + (\boldsymbol{R}[t] \odot \boldsymbol{H}[t-1])\boldsymbol{W}_{hh} + \boldsymbol{b}_h\right) \quad (3)$$

$$\boldsymbol{H}[t] = \boldsymbol{Z}[t] \odot \boldsymbol{H}[t-1] + \left(1 - \boldsymbol{Z}[t] \odot \tilde{\boldsymbol{H}[t]}\right) \quad (4)$$

where, $\boldsymbol{H}[t-1] \in \mathbb{R}^{n_d \times h}$ is the hidden state of the last time-step, $\boldsymbol{R}[t] \in \mathbb{R}^{n_d \times h}$ is the reset gate, $\boldsymbol{Z}[t] \in \mathbb{R}^{n_d \times h}$ is the update gate, $\boldsymbol{W}_{xr}, \boldsymbol{W}_{xz} \in \mathbb{R}^{n_d \times h}$ and $\boldsymbol{W}_{hr}, \boldsymbol{W}_{hz} \in \mathbb{R}^{h \times h}$ are weight parameters, $\boldsymbol{b}_r, \boldsymbol{b}_z, \boldsymbol{b}_h \in \mathbb{R}^{1 \times h}$ are the biases, and h is the number of features in the layer ($h = n_f$ for first layer). $\tilde{\boldsymbol{H}}[t]$ is the candidate hidden state and $\boldsymbol{H}[t]$ is the new state. σ denotes sigmoid function and $\odot$ denotes Hadamard product. In case of a multi-layered GRU network, the input of a particular layer is the hidden state $\boldsymbol{H}[t]$ of the previous layer. No dropout has been used between the layers.

C. Concatenation With Dense Layer

Simultaneously, the feature tensor is reshaped to $\boldsymbol{X_{d1}} \in \mathbb{R}^{n_d \times (n_f * 24)}$) vector which is fed to a single dense layer network f_{d1}. Rectified Linear Unit (ReLU) is used as the activation function. Assuming that $(\boldsymbol{X_{d1}})^k$ is the input vector and $(\boldsymbol{Y_{d1}})^k$ is the output vector for the k^{th} training sample, the dense layer can be formulated as:

$$V_i^k = ReLU\left(\sum w_{ij}(x_{d1})_j^k - bh_i\right) \quad (5)$$

$$(Y_{d1})_p^k = \sum V_i^k w_{pi} - bo_p \quad (6)$$

where, V_i^k is the output of hidden neuron i, w_{ij} is weight parameter from input layer neuron j to hidden layer neuron i, bh_i is the bias of hidden neuron i, w_{pi} is the weight parameter from hidden neuron i to output neuron p, and bo_p is the bias of output neuron p. The output $\boldsymbol{Y_g}$ from the last layer of GRU and the output $\boldsymbol{Y_{d1}}$ from the dense layer are concatenated.

D. Attention Mechanism

The concatenated layer $\boldsymbol{X_a}(\boldsymbol{Y_g}, \boldsymbol{Y_{d1}})$ is fed to an attention network f_a. Soft attention is used. The attention mechanism is explained below.

Firstly, the input concatenated sequence is encoded into a set of internal states $h_1, h_2, \ldots h_m$. Alignment scores are calculated for each encoded state by training a feedforward network that learns to recognize relevant states by creating higher scores for states that deserve attention, and vice-versa. Attention weights $\alpha_1, \alpha_2, ..\alpha_m$ are generated by applying *softmax* function to the scores. Note that the attention weight vector gives a probabilistic interpretation i.e. $\alpha \in [0, 1]$ and $\sum \alpha = 1$. Next, the context vector is computed:

$$\boldsymbol{C} = \alpha_1 * h_1 + \alpha_2 * h_2 + \ldots + \alpha_m * h_m \quad (7)$$

$\boldsymbol{C}$ is concatenated with the output generated from the previous time step. The process repeats for all time steps. Finally, this is followed by a second dense layer network f_{d2} also with ReLu as the activation function (similarly using equations 5-6), to map the attention layer output $\boldsymbol{Y}_a$ to the target drift sequence $\boldsymbol{\Delta}$. The CGDA model $f_{CGDA} : \boldsymbol{X} \rightarrow \boldsymbol{\Delta}$ can thus be summarized as:

$$\begin{aligned} \text{GRU } f_g &: \boldsymbol{X} \rightarrow \boldsymbol{Y_g} && (8)\\ \text{Dense1 } f_{d1} &: \boldsymbol{X_{d1}} \rightarrow \boldsymbol{Y_{d1}} && (9)\\ \text{Attention } f_a &: \boldsymbol{X_a}(\boldsymbol{Y_g}, \boldsymbol{Y_{d1}}) \rightarrow \boldsymbol{Y_a} && (10)\\ \text{Dense2 } f_{d2} &: \boldsymbol{X_{d2}}(\boldsymbol{Y_a}) \rightarrow \boldsymbol{\Delta} && (11) \end{aligned}$$

The equations of all activation functions used in the model are summarized below:

$$\begin{aligned} \sigma(x) &= \frac{1}{(1 + e^{-x})} && (12)\\ \tanh(x) &= \frac{(e^x - e^{-x})}{e^x + e^{-x}} && (13)\\ ReLU(x) &= max(x, 0) && (14)\\ softmax_i(\vec{x}) &= \frac{e^{x_i}}{\sum_{j=1}^{m} e^{x_j}} \quad \text{for } i = 1, \ldots, m && (15) \end{aligned}$$

E. Rationale Behind Model Structure

The advantages of the GRU are manifold. First, its suitability for sequence modelling is apt for our time-varying drift which is a time-series data. Secondly, its ability to capture time dependencies is apt for our fixed-length drift sequences. The reset gate and the update gate help capture the short-term and long-term dependencies, respectively. Moreover, the GRU extracts important temporal features layer-by-layer that improves model robustness. Additionally, a GRU unit has few parameters leading to faster training which is an important requirement for the stacked layers in our model. The dense layer helps in better retention of the feature information. While the GRU generates deep features, the dense layer retains the handcrafted features. The placement of the attention network in the CGDA model has multiple benefits:

- Without the attention layer, equal weights would be assigned to all the states in the concatenated output from the GRU and the dense layer. With attention network, we have weighted attention to the states i.e., assigning adequate weightage to each state in the concatenated output. It also allows us to better interpret the contribution of the two participating models i.e. GRU vs. Dense1.
- It gives attention to the entire input sequence instead of just the last state. This prevents information loss for long input sequences unlike a *seq2seq* network where information tends to get lost towards the end.
- For each time step, a separate context vector is computed by computing the attention weights. Thus, through this mechanism, our model can discover interesting mappings between various segments of the input sequence and their corresponding parts in the output sequence.

F. Feature Space

A 3-stage approach is used for feature refinement: firstly, the initial feature space is selected based on domain knowledge of drift-causing factors; secondly, filter method is used to statistically evaluate the relevant subset, and lastly, intrinsic method is used by the deep layers of the CGDA model. The following parameters are selected for the initial feature set, that addresses the factors causing drift:

- Environmental factors: air temperature (T_a), air pressure (P_a), relative humidity (RH), particulate matter ($PM_{2.5}$)
- Aging factors: elapsed days (*elap-day*), hourly time-slot in the day (*time-slot*)

The *elap-day* feature denotes the number of days elapsed since the sensor deployment. Note that *elap-day* is not the same as the sample number; there may be missing samples in the data. In Fig. 1, the days 1 to n_d are samples, each containing *elap-day* as a feature. The *time-slot* feature denotes the time-slot among the 24 hours in a day. While the aging effect is predominantly represented by *elap-day*, the *time-slot* can represent the daily cyclic influence.

Thereafter, a filter method namely a simple Pearson Product-Moment correlation analysis is performed to evaluate the linear relationships. A confidence interval of 90% ($p < 0.1$) is used for the significance tests. Finally, an intrinsic method performs automatic feature selection during the model's training process, which in this case is facilitated by the deep layers of the CGDA model.

III. Experimental Settings and Data

A. Experimental Settings

1) Hyperparameter Optimization: For each experiment, a hyperparameter optimization is performed to select the best model during training. The ranges covered are- learning rate: $[10^{-6}, 10^{-4}]$, recurrent layers in the GRU block: [1, 5], hidden GRU units in a layer: [12, 400], hidden neurons in the dense layer: [12, 200]. For each experiment, the number of neurons in the output layer of GRU (Y_g) and Dense1 (Y_{d1}) are kept equal before concatenation. Early-stopping regularization based on the validation error is used, with train:validation size ratio set at 80:20, and the number of epochs limited to 50,000. The Adam optimizer is implemented and the loss function of the model is set as root-mean-squared-error (RMSE).

2) Performance Metric: We evaluate the model's performance using the drift-prediction-accuracy (DPA) based on the mean-absolute-percentage-error (MAPE):

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\delta_i - \hat{\delta}_i}{\delta_i}\right| \tag{16}$$

$$DPA = (1 - MAPE) \times 100\ (\%) \tag{17}$$

where, δ_i and $\hat{\delta}_i$ are the true and the predicted drifts, respectively, and N is the number of samples.

3) Ablation Study: An ablation study is important for understanding the causality in the model. The ablated variants of the CGDA model are implemented which are, GRU (a single/stacked GRU model) [29], SNN (a single-layered neural network model) [26] and CGD (concatenated GRU block and dense layer model without an attention mechanism).

4) Comparison With State-of-the-Art: To comprehensively evaluate our proposed CGDA model, we compare against several state-of-the-art machine learning methods. Shallow models namely, decision tree regression (DTR) [31], support vector regression (SVR) [32], and random forest (RF) [15] are implemented besides the SNN. Deep models namely, long short-term memory (LSTM) [33] and 1D convolutional neural network (CNN) [34] are employed besides the deep ablated variants (GRU, CGD). LSTM is a popular gated sequential network, and 1D-CNN is suitable for time-series regression with sensor data. Therefore, 4 shallow models (DTR, SVR, RF, SNN) and 4 deep models (LSTM, CNN, GRU, CGD) are implemented besides CGDA for the comparative analysis.

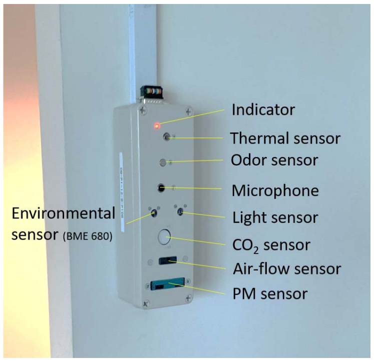


Fig. 2. A labelled sensor node deployed at measurement site.

It is to be noted that although both GRU and LSTM keep long-term dependencies while handling the exploding/vanishing gradient problems, GRU does not require memory units unlike LSTM. Since our input sequence length is fixed (24 time-steps), GRU is a better choice compared to LSTM whose benefits weigh up mainly for longer sequences. Moreover, GRU uses less training parameters thereby having lower computational cost and faster response. Therefore, we do not compare a model with LSTM in place of GRU.

B. Sensor Data Collection

Several sensors namely, an environmental sensor, a particulate matter sensor, an air-flow sensor, a CO_2 sensor, an ambient light sensor, an odor sensor, a thermal sensor and a microphone, are embedded together using a TI AM335x BeagleBone Black Robotics Cape, and encased into a compact box referred to as a 'sensor node' or simply 'node' as labelled in Fig. 2. The sensor models are listed in Table I. The BME680 environmental sensor is a widely used low-cost MOX based gas sensor meant for monitoring volatile organic compounds (VOC). It measures three environmental parameters namely, air temperature, air pressure and humidity. Freshly manufactured new BME680 sensors are cased. A node continuously senses the environment, performs edge processing, and sends the analysed data to a backend server where it is stored and viewed real-time through a web portal.

Several sensor nodes were deployed within the office premises of the Institute for Infocomm Research, ASTAR, Singapore, for collection of long-time drift performance as described in Table II. Nodes A and B were placed in indoor locations, while nodes C and D were deployed in semi-indoor locations. The indoor locations are rooms in the interior of

TABLE I
SUMMARY OF THE ENCASED SENSORS WITHIN A NODE

Sensor	Model	Measuring parameters
Environmental sensor	BME680, Robert Bosch, Germany	Air temperature, Air pressure, Humidity, VOC resistance
Particulate matter sensor	PMS7003M, Plantower, China	PM 1.0, PM 2.5, PM 10
Air-flow sensor	D6F-W01A1, Omron, Japan	Air-flow
CO_2 sensor	GC-0027, CozIR, GSS, UK	CO_2 concentration
Ambient light sensor	NOA1305, On Semiconductor, USA TSL2561, Adafruit, USA	Light intensity
Odor sensor	TGS2603, Figaro, Japan	Detection of odor and air contaminants
Thermal sensor	D6T-44L-06, Omron, Japan	Object temperature

TABLE II
DEPLOYMENT LOCATIONS AND DURATIONS OF THE SENSOR NODES USED FOR COLLECTION OF DRIFT PERFORMANCE

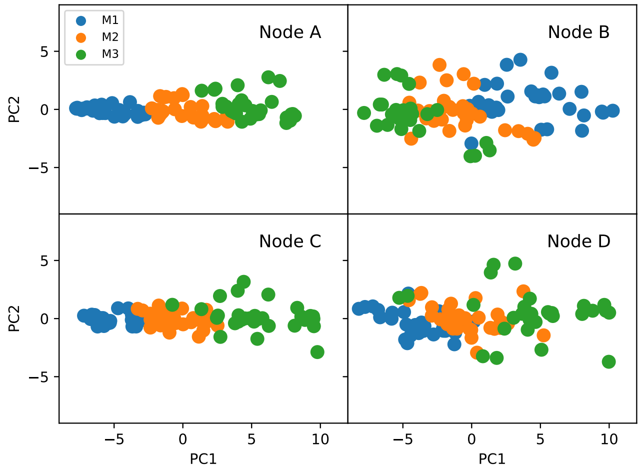
Location Type	Node	Location Description	Duration (months)
Indoor	A	Software laboratory	14
	B	Meeting room	6
Semi-Indoor	C	Pantry 1	4
	D	Pantry 2	4

building with no access to outdoors, while the semi-indoor locations have an enclosed boundary with direct access to the outdoors. The data collection duration spanned 4 to 14 months.

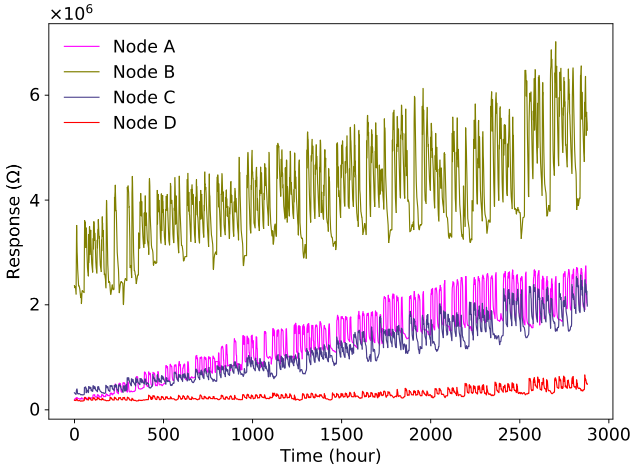
Multiplicative drift is used in this study and is denoted by δ (unitless) as the ratio between the current (drifted state) resistance to the actual (non-drifted state) resistance response. Similar to previous works [13], [35], [36], the multiplicative drift is based on the assumption that the sensors are calibrated before being deployed and therefore the response during the initial period post deployment can be considered drift-free. The data was logged at intervals of 3 seconds. For this study, we averaged the data per hour. Besides the hourly drift, the hourly mean of features air temperature, pressure, relative humidity, and PM2.5 were computed. Note that air-flow is removed from consideration as a feature, due to its almost constant value. The CO_2, light intensity, odor and thermal sensors, and the microphone were meant for monitoring purposes only.

C. PCA for Time-Varying Drift Illustration

Principal component analysis (PCA) is used to inspect the drift across the months as shown in Fig. 3a. PCA is applied to project the samples to a 2-D subspace, where each day is a sample consisting of 24 timeslot (hourly) drifts. The plots depict the first two principal components (PC) and the first component PC1 accounts for the majority of variance in the drift: 0.957 (node A), 0.806 (node B), 0.958 (node C) and 0.839 (node D). For simplicity, only the first three months are analysed. Firstly, it is evident from the visual inspection that there is an obvious drift with time across months, which are separable while sliding horizontally along PC1. Secondly, different data distributions among the nodes indicate that the drift varies from one sensor to another. For visual clarity, the sensor response (VOC resistance) of only the first 4 months are illustrated in Figure 3b. The scale of response varies between the four nodes as they are placed in different environments. Therefore, it is vital to develop separate drift model for each node. While the basic model architecture may be same, the hyperparameters have been tuned per node for best performance.



(a)



(b)

Fig. 3. (a) Illustration of the drift across months M1 to M3. (b) Sensor response along first 4 months M1-M4.

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

A. Feature Significance by Filter Method

The results of the Pearson correlation and the significance tests between the sensor drift and the features are summarized in Fig. 4. Here, *hs*, *ls*, and *ns* denote high significance ($p < 0.05$), low significance ($p < 0.1$), and no significance ($p \geq 0.1$), respectively. Most features reveal significant correlation with the sensor drift. Aging as the major cause of first-order drift is validated by the highest correlations with *elap-day*. All features show significance except for RH in

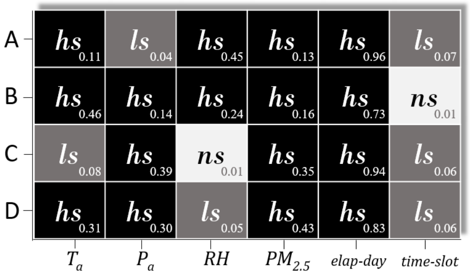


Fig. 4. Significance of Pearson correlation between the sensor drift and the initial features. *hs* = high significance ($p < 0.05$), *ls* = low significance ($p < 0.1$), *ns* = no significance ($p \geq 0.1$).

node C and *time-slot* in node D, which could be attributed to the comparatively smaller sample sizes. Interestingly, the significance of *time-slot* feature supports our speculation that the time of the day can have some catalytic effect on the drift. The movements and the occupancy in office spaces usually tend to have a daily as well as a weekly pattern, which the combination of *elap-day* and *time-slot* may represent. The collective effect of these factors have complex non-linearities that the non-linear CGDA as an intrinsic method can adequately model.

B. Drift Prediction Performance

The chronological drift prediction performance of all the models are presented in Table III and Fig. 5. Table III lists the test MAPE values and Fig. 5 graphically summarizes the DPA values. The first column in Table III depict the test month. Note that the table lists the test error; a model trained on the first month (M1) is tested on the second month (M2), next a model trained on the first two months (M1-2) is tested on the third month (M3) and so on.

It is observed from Table III that the proposed CGDA model achieves the best performance consistently across most of the chronological experiments. It exhibits mean MAPE of 5.5% to 7.59% only, as compared to much larger errors in the traditional shallow models. It predicts the drift in sensor nodes A, B, C and D with excellent mean accuracies of 94.50%, 93.66%, 92.41% and 92.58%, respectively. As seen for node A in Fig. 5, DTR, SVR, and RF perform poorly due to low training data in the first experiment (M2). However, CGDA performs well even with just one month training data. From Fig. 5, it is further evident that the CGDA model consistently achieves high DPA (above 90%) across most months. This is a remarkable advantage of the CGDA model in terms of reliability, as compared to the remaining models.

As for ablation study, GRU is better able to predict the drift as compared to SNN. The integration of the GRU layers with a dense layer in CGD further enhances its ability to formulate the drift. While the dense layer uses the supplied handcrafted features, the deep GRU layers generate important temporal features. The addition of the attention layer to this concatenated integration helps to compute the weights appropriately, thereby preventing loss of information. This weight assignment in CGDA allows extracting only the most relevant and adequate information, which further enhances the ability to capture the drift.

The results are further categorised based on the type of the sensor location: indoor vs. semi-indoor, and the type of the learning model: deep learning vs. shallow learning.

TABLE III
CHRONOLOGICAL COMPARISON OF MAPE (%) AMONG VARIOUS MODELS. BEST PERFORMANCES ARE HIGHLIGHTED IN BOLD

	Shallow learning models				Deep learning models				Proposed
Test month	DTR [31]	SVR [32]	RF [15]	SNN [26]	LSTM [33]	CNN [34]	GRU [29]	CGD -	CGDA
					Node A				
M2	74.07	82.47	62.87	29.79	34.58	35.68	27.46	18.73	**11.53**
M3	49.02	32.09	31.63	9.52	11.80	5.48	7.91	6.98	**5.12**
M4	21.96	15.98	19.96	8.02	**4.73**	8.97	7.70	6.95	5.07
M5	19.77	18.44	21.71	9.94	5.94	5.38	5.23	4.37	**4.11**
M6	24.08	11.26	10.69	17.40	6.49	7.26	7.26	6.79	**5.40**
M7	32.30	14.77	19.38	7.06	6.21	15.46	6.18	5.98	**5.10**
M8	20.32	19.99	19.55	12.84	6.87	**5.22**	7.24	6.14	5.56
M9	17.55	15.80	11.97	10.03	3.09	4.99	4.20	2.91	**2.80**
M10	14.86	12.85	10.72	13.64	**4.24**	6.06	6.53	5.48	4.62
M11	20.78	10.28	15.18	19.40	7.37	7.29	7.67	6.81	**5.29**
M12	14.46	9.86	12.47	14.66	6.37	4.60	7.47	6.06	**3.72**
M13	13.85	10.71	9.00	30.09	6.28	**6.27**	8.93	7.29	6.40
M14	11.38	9.99	7.43	11.15	10.48	8.78	10.12	9.36	**6.75**
Mean	25.72	20.35	19.43	14.89	8.80	9.34	8.76	7.22	**5.50**
					Node B				
M2	14.59	19.35	13.22	16.51	26.57	19.06	12.59	10.17	**8.07**
M3	17.97	17.34	13.60	8.01	9.55	7.77	6.25	5.09	**4.67**
M4	18.59	16.31	15.63	12.65	15.31	8.99	8.92	7.75	**7.11**
M5	17.39	8.72	9.87	10.82	6.14	6.78	8.05	6.47	**5.83**
M6	13.63	11.65	12.55	13.42	8.78	6.3	9.53	8.32	**6.04**
Mean	16.43	14.67	12.97	12.28	13.27	9.78	9.07	7.56	**6.34**
					Node C				
M2	35.72	31.04	33.83	11.40	9.26	9.47	9.62	9.12	**8.98**
M3	32.39	35.60	35.79	22.83	24.88	9.8	10.20	9.37	**8.77**
M4	26.99	12.74	24.89	9.27	6.08	7.56	6.61	5.75	**5.02**
Mean	31.70	26.46	31.50	14.50	13.41	8.94	8.81	8.208	**7.59**
					Node D				
M2	34.60	13.94	25.68	16.03	17.89	27.07	12.80	9.45	**6.74**
M3	29.85	24.70	29.53	15.07	**7.68**	9.26	13.27	11.35	8.73
M4	39.80	38.48	30.96	23.92	12.89	10.46	15.22	9.16	**6.80**
Mean	34.75	25.71	28.72	18.34	12.82	15.60	13.76	9.99	**7.42**

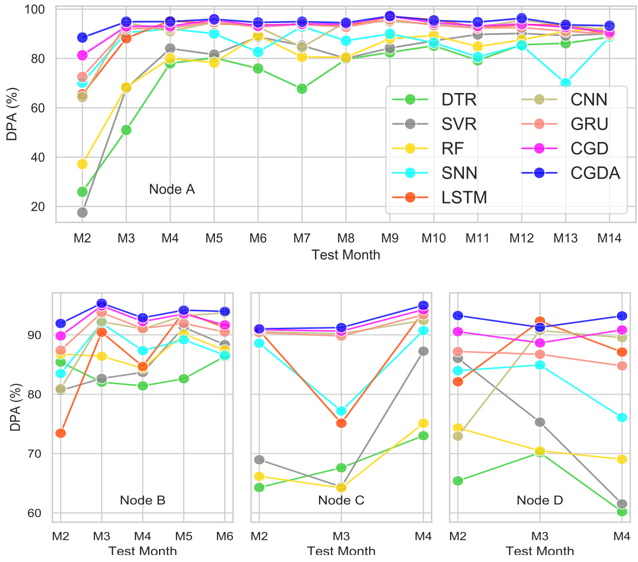


Fig. 5. Comparison of the chronological DPA of all models.

TABLE IV
SUMMARY OF CATEGORICAL MEAN DPA (%)

		Indoor		Semi-indoor		Mean	
		A	B	C	D		
Shallow	DTR	74.28	83.57	68.30	65.25	72.85	78.22
	SVR	79.65	85.33	73.54	74.29	78.20	
	RF	80.57	87.03	68.50	71.28	76.84	
	SNN	85.11	87.72	85.50	81.66	85.00	
Deep	LSTM	91.20	86.73	86.59	87.18	87.92	**90.40**
	CNN	90.66	90.22	91.06	84.40	89.08	
	GRU	91.24	90.93	91.19	86.24	89.90	
	CGD	92.78	92.44	91.92	90.01	91.79	
	CGDA	94.50	93.66	92.41	92.58	**93.29**	
Mean		**87.64**		82.33			

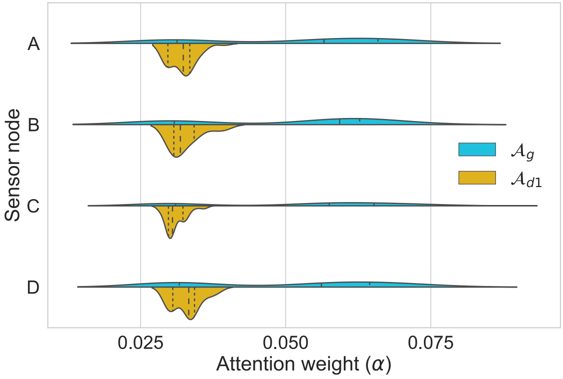


Fig. 6. Split-violin plot of the attention weights.

Table IV summarizes the mean DPAs for this categorical analysis. CGDA achieves an overall mean DPA of 93.29% across all nodes, outperforming the rest while DTR attains the least mean DPA at 72.85%. Firstly, the deep learning models with a mean DPA of 90.40% perform significantly better than the shallow learning models that could attain only 78.22% mean DPA. This reaffirms the ability of deep layers to better decipher temporal features and is suggestive of its suitability for sensor drift estimation which is a temporal dependency problem. Secondly, the models are better able to predict drift in the indoor sensors (87.64%) as compared to the semi-indoor sensors (82.33%). This behaviour could be attributed to the more stable environment in the indoor rooms. Prediction would be further challenging for nodes located outdoors due to the varied external factors. It will probably require an enhanced model considering features like CO_2 concentration, rainfall status, pollution levels, air velocity etc.

C. Attention Network Analysis and Drift Compensation

The attention weights α's are assigned to the states derived from $\boldsymbol{X_a}$ which is the concatenation of $\boldsymbol{Y_g}$ and $\boldsymbol{Y_{d1}}$ as was described in Fig. 1. Fig. 6 presents a split violin plot of the attention weights categorised by $\boldsymbol{\mathcal{A}_g}$ and $\boldsymbol{\mathcal{A}_{d1}}$, which are the attention weight vectors ($\boldsymbol{\mathcal{A}} = \{\alpha_1, \alpha_2, \ldots\}$) corresponding to GRU-derived $\boldsymbol{Y_g}$, and Dense1-derived $\boldsymbol{Y_{d1}}$, respectively, such that the context vector is:

$$C = \boldsymbol{\mathcal{A}_g Y_g} + \boldsymbol{\mathcal{A}_{d1} Y_{d1}} \tag{18}$$

The plot in Fig. 6 reveals that the $\boldsymbol{Y_g}$ weights cover a much wider range with greater probability of larger values, while the $\boldsymbol{Y_{d1}}$ weights are limited with probability of smaller values. The larger weights for GRU indicate that the GRU block makes a greater contribution as compared to Dense1. It denotes that the deep features created by the GRU have more significance than the handcrafted features fed to the dense layer.

V. CONCLUSION

In this article, we proposed a novel deep-sequential model termed as the CGDA model for drift compensation in low-cost gas sensors. Concatenation of a stacked GRU block and a dense layer is integrated with attention mechanism, that predicts the drift sequence for an entire day, through intensive feature extraction and elaborate information mapping. The stacked GRU layers extract useful deep temporal features layer by layer capturing time dependencies at a low computational expense, the dense layer helps retain handcrafted feature information, while the attention network facilitates adequate weightage assignment and prevents information loss. In addition, the feature space addresses the environmental and aging effects on the sensor. The efficacy of the CGDA model is validated through its superior performance with over 93% mean DPA as compared to 8 state-of-the-art shallow and deep learning models across multiple nodes at varied locations.

The CGDA model can help realize remote sensor calibration and greatly enhance the reliability of gas sensors in real-world applications. Its utility can be extended to other sensors as well through appropriate improvisations such as a suitable feature-space. In future, we plan to address the issue of noisy data [37] for sensor drift prediction. We will also continue to work advanced machine learning methods (e.g., ensemble deep learning [38] and deep transfer learning [39]) for this task. In particular, we will extend the study to larger number of sensors covering more varied locations, and then explore cross-node and cross-location model transfer mechanisms for better robustness.

ACKNOWLEDGMENT

The authors would like to express their sincere appreciation to the Communications & Networks department, I2R, for providing the data, and Eugene, Dr. Liu Guimei, Robert Bosch Singapore, Institute of Microelectronics and National Metrology Centre for their insightful suggestions.

REFERENCES

[1] S. Di Carlo and M. Falasconi, *Drift Correction Methods for Gas Chemical Sensors in Artificial Olfaction Systems: Techniques and Challenges*. Rijeka, Croatia: InTech, 2012.

[2] 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. Techn.*, vol. 12, no. 3, pp. 1441–1460, Mar. 2019.

[3] 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, Mar. 2010.

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

[5] S. De Vito, G. Fattoruso, M. Pardo, F. Tortorella, and G. Di Francia, "Semi-supervised learning techniques in artificial olfaction: A novel approach to classification problems and drift counteraction," *IEEE Sensors J.*, vol. 12, no. 11, pp. 3215–3224, Nov. 2012.

[6] M. J. Wenzel, A. Mensah-Brown, F. Josse, and E. E. Yaz, "Online drift compensation for chemical sensors using estimation theory," *IEEE Sensors J.*, vol. 11, no. 1, pp. 225–232, Jan. 2011.

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

[8] 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 Sensors J.*, vol. 16, no. 3, pp. 750–761, Feb. 2016.

[9] M. Z. Abidin, A. Asmat, and M. N. Hamidon, "Temperature drift identification in semiconductor gas sensors," in *Proc. IEEE Conf. Syst., Process Control (ICSPC)*, Dec. 2014, pp. 63–67.

[10] A. Perera, N. Papamichail, N. Barsan, U. Weimar, and S. Marco, "On-line novelty detection by recursive dynamic principal component analysis and gas sensor arrays under drift conditions," *IEEE Sensors J.*, vol. 6, no. 3, pp. 770–783, Jun. 2006.

[11] M. Holmberg, F. Winquist, I. Lundström, F. Davide, C. DiNatale, and A. D'Amico, "Drift counteraction for an electronic nose," *Sens. Actuators B, Chem.*, vol. 36, nos. 1–3, pp. 528–535, Oct. 1996.

[12] A. C. Romain and J. Nicolas, "Long term stability of metal oxide-based gas sensors for e-nose environmental applications: An overview," *Sens. Actuators B, Chem.*, vol. 146, no. 2, pp. 502–506, Apr. 2010.

[13] A. Vergara, S. Vembu, T. Ayhan, M. A. Ryan, M. L. Homer, and R. Huerta, "Chemical gas sensor drift compensation using classifier ensembles," *Sens. Actuators B, Chem.*, vols. 166–167, pp. 320–329, May 2012.

[14] M. Verma, S. Asmita, and K. K. Shukla, "A regularized ensemble of classifiers for sensor drift compensation," *IEEE Sensors J.*, vol. 16, no. 5, pp. 1310–1318, Mar. 2016.

[15] A. U. Rehman and A. Bermak, "Heuristic random forests (HRF) for drift compensation in electronic nose applications," *IEEE Sensors J.*, vol. 19, no. 4, pp. 1443–1453, Feb. 2019.

[16] S. Brahim-Belhouari, A. Bermak, and P. C. H. Chan, "Gas identification with microelectronic gas sensor in presence of drift using robust GMM," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 5, May 2004, pp. 833–835.

[17] Y. Tian *et al.*, "A drift-compensating novel deep belief classification network to improve gas recognition of electronic noses," *IEEE Access*, vol. 8, pp. 121385–121397, 2020.

[18] Q. Liu, X. Hu, M. Ye, X. Cheng, and F. Li, "Gas recognition under sensor drift by using deep learning," *Int. J. Intell. Syst.*, vol. 30, no. 8, pp. 907–922, Aug. 2015.

[19] Y. Luo, S. Wei, Y. Chai, and X. Sun, "Electronic nose sensor drift compensation based on deep belief network," in *Proc. 35th Chin. Control Conf. (CCC)*, Jul. 2016, pp. 3951–3955.

[20] L. Zhang, Y. Liu, Z. He, J. Liu, P. Deng, and X. Zhou, "Anti-drift in E-nose: A subspace projection approach with drift reduction," *Sens. Actuators B, Chem.*, vol. 253, pp. 407–417, Dec. 2017.

[21] L. Zhang and D. Zhang, "Efficient solutions for discreteness, drift, and disturbance (3D) in electronic olfaction," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 2, pp. 242–254, Feb. 2018.

[22] L. Zhang, Y. Liu, and P. Deng, "Odor recognition in multiple E-Nose systems with cross-domain discriminative subspace learning," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 7, pp. 1679–1692, Jul. 2017.

[23] K. Yan and D. Zhang, "Correcting instrumental variation and time-varying drift: A transfer learning approach with autoencoders," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 9, pp. 2012–2022, Sep. 2016.

[24] L. Zhang and D. Zhang, "Domain adaptation extreme learning machines for drift compensation in E-Nose systems," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 7, pp. 1790–1801, Jul. 2015.

[25] Q. Liu, X. Li, M. Ye, S. S. Ge, and X. Du, "Drift compensation for electronic nose by semi-supervised domain adaption," *IEEE Sensors J.*, vol. 14, no. 3, pp. 657–665, Mar. 2014.

[26] L. Zhang, F. Tian, S. Liu, L. Dang, X. Peng, and X. Yin, "Chaotic time series prediction of E-nose sensor drift in embedded phase space," *Sens. Actuators B, Chem.*, vol. 182, pp. 71–79, Jun. 2013.

[27] B. Mumyakmaz, A. Özmen, M. A. Ebeoğlu, and C. Taşaltın, "Predicting gas concentrations of ternary gas mixtures for a predefined 3D sample space," *Sens. Actuators B, Chem.*, vol. 128, no. 2, pp. 594–602, Jan. 2008.

[28] S. De Vito *et al.*, "Gas concentration estimation in ternary mixtures with room temperature operating sensor array using tapped delay architectures," *Sens. Actuators B, Chem.*, vol. 124, no. 2, pp. 309–316, Jun. 2007.

[29] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*. [Online]. Available: http://arxiv.org/abs/1406.1078

[30] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*. [Online]. Available: http://arxiv.org/abs/1409.0473

[31] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.

[32] R. Laref, E. Losson, A. Sava, and M. Siadat, "Support vector machine regression for calibration transfer between electronic noses dedicated to air pollution monitoring," *Sensors*, vol. 18, no. 11, p. 3716, Nov. 2018.

[33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[34] J. Yang, M. N. Nguyen, P. P. San, X. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Proc. IJCAI*, vol. 15, 2015, pp. 3995–4001.

[35] Y. Wang, A. Yang, X. Chen, P. Wang, Y. Wang, and H. Yang, "A deep learning approach for blind drift calibration of sensor networks," *IEEE Sensors J.*, vol. 17, no. 13, pp. 4158–4171, Jul. 2017.

[36] Y. Wang, A. Yang, Z. Li, X. Chen, P. Wang, and H. Yang, "Blind drift calibration of sensor networks using sparse Bayesian learning," *IEEE Sensors J.*, vol. 16, no. 16, pp. 6249–6260, Jun. 2016.

[37] G. Mustafa, H. Li, J. Zhang, and J. Deng, "ℓ1-regression based subdivision schemes for noisy data," *Comput.-Aided Des.*, vol. 58, pp. 189–199, Jan. 2015.

[38] X. Qiu, L. Zhang, Y. Ren, P. Suganthan, and G. Amaratunga, "Ensemble deep learning for regression and time series forecasting," in *Proc. IEEE Symp. Comput. Intell. Ensemble Learn. (CIEL)*, Dec. 2014, pp. 1–6.

[39] S. M. Salaken, A. Khosravi, T. Nguyen, and S. Nahavandi, "Seeded transfer learning for regression problems with deep learning," *Expert Syst. Appl.*, vol. 115, pp. 565–577, Jan. 2019.

Tanaya Chaudhuri received the B.Tech. degree in electrical engineering from the National Institute of Technology (NIT) at Silchar, India, in 2013, and the Ph.D. degree from Nanyang Technological University (NTU), Singapore, in 2019. She had served in software development at Oracle. She is currently a Scientist with the Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore. Her current research interests include deep learning, machine learning, data mining, and bioinformatics.

Min Wu (Senior Member, IEEE) received the B.S. degree in computer science from the University of Science and Technology of China (USTC), in 2006, and the Ph.D. degree in computer science from Nanyang Technological University (NTU), Singapore, in 2011. He is currently a Senior Scientist with the Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore. His current research interests include machine learning, data mining, and bioinformatics. He received the Best Paper Awards in InCoB 2016 and DASFAA 2015. He was also a recipient of the IJCAI competition on repeated buyers prediction in 2015.

Yu Zhang received the B.S. and M.E. degrees from the Huazhong University of Science and Technology (HUST), in 2000 and 2003, respectively, and the Ph.D. degree in computer science from Nanyang Technological University (NTU), Singapore, in 2010. He is currently a Scientist with the Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore. His current research interests include machine learning, data mining, and big data.

Pan Liu received the Ph.D. degree from Nanyang Technological University, in 2015. He had served as a Senior R&D Engineer for device development in semiconductor industry. He is currently a Scientist with the Institute for Infocomm Research, A*STAR, Singapore. He has led several industry projects in collaboration with partners across the semiconductor sector. His research interests include TFT and memory device development, data analytics in predictive maintenance, and machine learning with industry application.

Xiaoli Li (Senior Member, IEEE) received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, in 2001. He is currently a Principal Scientist with the Institute for Infocomm Research, A*STAR, Singapore. He is also an Adjunct Full Professor with Nanyang Technological University. He has published more than 220 high quality articles with more than 10 000 citations and received six best paper awards. He has led more than ten research projects in collaboration with industry partners across a range of sectors. His research interests include AI, data mining, machine learning, and bioinformatics. He has been serving as an Area Chair and a Senior PC Member for leading AI and data mining related conferences (including KDD, ICDM, SDM, PKDD/ECML, WWW, IJCAI, AAAI, ACL, and CIKM).