This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3503616

Enhancing IoT Sensors Precision through Sensor Drift Calibration with Variational Autoencoder

Md Kamal Hossain, Iftekhar Ahmad, Daryoush Habibi, and Muhammad Waqas

***Abstract*—IoT sensors are made of physical materials, and due to natural decay in materials, sensor data drifts over time. Even though sensors are calibrated after deploying at the site, the accumulation of errors in sensor measurements due to sensor drifts renders the data progressively irrelevant, creating significant issues for end applications. In this paper, we propose a software-driven drift detection and calibration framework based on probabilistic observation in latent space using Variational Autoencoders (VAEs). The proposed method utilizes the latent distribution of the generative model from sampled observational data, which are collected during the calibration phase of the deployed sensors. Variational inference in VAEs is employed to approximate the true posterior distribution for detecting sensor drifts, incorporating metrics such as Kullback-Leibler (KL) divergence. Additionally, reconstruction loss is utilized for calibrating the sensors. Both simulated and real-world sensor data are used to evaluate the proposed method. Experimental results demonstrate significant improvement over existing drift detection and calibration techniques.**

***Index Terms*—Sensor drift, variational autoencoder, soft calibration, latent distribution, Kullback-Leibler divergence**

I. Introduction

TECHNOLOGIES have become integral for managing diverse facets of daily life, encompassing public services, business operations, and even grocery shopping. Governments, societies, and companies increasingly depend on these interconnected technological platforms, tools, and interfaces, as the Internet evolves rapidly toward a more decentralized version. Looking ahead, the interconnectivity and convergence of these digital tools are poised to deepen, particularly as society embraces the upcoming generation of the Internet, often referred to as Internet 3.0 [1], [2].

Amidst this technological evolution, the seamless functioning of these interconnected IoT systems, which depend on sensing accurate readings and sending these data to a central server, has become paramount. One critical challenge arises from the accumulation of errors in data readings, particularly due to sensor drift, posing a significant issue for end applications. Traditionally, this problem is addressed by comparing measurements from faulty sensors with those from accurately calibrated standard sensors. However, the calibration process is labor-intensive and loses effectiveness when dealing with numerous sensors that require frequent calibration [3], [4], [5]. In the context of remote and large-scale implementations of IoT sensors, the cost and impracticality of frequent manual calibration for inexpensive sensors create a substantial demand for soft calibration methods [6], [7], [8].

One simple technique for calibration involves applying a known stimulus to the sensor network and measuring its response [9]. Then, comparing the ground truth input to the response will result in finding the gain and offset for the linear drifts case [10], [5]. The calibration problem of the sensor network was also tackled by [6] using Bayesian framework. In their work, the researchers proposed that, once deployed, sensor measurements would differ linearly from the actual values by certain gains and offsets unique to each sensor, despite initial calibration at the factory. To address this issue, they developed a subspace matching technique for estimating these gains and offsets using routine sensor measurements, without the need for comparison to ground truth values. The estimated gains and offsets were assumed to be constant for each sensor and were used to calibrate future readings to the true values. Although this method was effective in controlled environments, it was less successful in the presence of noise and other disruptions.

Over time, some sensor nodes may exhibit drift in their readings, which, if left uncorrected, can lead to incorrect conclusions in downstream applications. When the unreliability level surpasses a certain threshold, the sensor-readings become unreliable and the applications using these data become irrelevant, as it is infeasible to manually re-calibrate the sensors. Soft calibration of sensor data at the central server is the most viable solution to address the drift issue and enable wider adoption of the technology. This is because the data from the same sensors will not be correlated if installed in different environments, and the faults or drift instances are likely to be correlated as well.

To address the computational limitations of IoT devices, our approach centralizes the data processing and model training on a powerful server. The IoT sensors are responsible only for data collection and transmission, while the central server performs the complex sensor drift detection and calibration using the Variational Autoencoder (VAE) model. Consequently, performing soft calibration of the sensor data at some central server and addressing their drifts separately helps to extend the effective and useful lifetime of the sensor nodes [10], [5], [7]. Traditional sensor drift calibration methods often rely on manual interventions, linear assumptions, or extensive historical data, which can be impractical and ineffective for large-scale IoT deployments. VAEs offer a promising alternative by leveraging deep learning to model complex, non-linear relationships in sensor data. VAEs can learn latent representations that

Corresponding author: Md Kamal Hossain is with School of Enginering, Edith Cowan University, Joondalup, Australia (email: mdkamal.hossain@ecu.edu.com.au http://www.kamalhossain.org/contact.html).

Iftekhar Ahmad, Daryoush Habibi and Muhammad Waqas are also with School of Enginering, Edith Cowan University, Joondalup, Australia

Manuscript received April 15, 2024.

capture the underlying distribution of sensor measurements, enabling the detection and calibration of drift without manual recalibration or reliance on simplistic assumptions [8], [5]. This probabilistic framework allows for robust handling of uncertainties and variations inherent in sensor data, making VAEs particularly suitable for real-world applications where sensor behavior may be unpredictable [8]. In this paper, our main contributions are summarized as follows:

- We propose a sensor drift calibration framework based on VAEs. Kullback-Leibler (KL) divergence is used both in VAEs loss function and also for sensor drift detection.
- We compare the performance of our proposed methods using both simulated and real-world datasets. The proposed method calibrates sensors based on calibrated sensor observations. Kalman filter and some existing techniques use predicted value as ground truth for subsequent estimations, leading to error accumulation. Experimental results show that our proposed technique is more robust than existing techniques as it does not accumulate errors.

The rest of this paper is organized as follows. We first present some related work in Section II. In Section III, we formulate the drift calibration problem. In Section IV, we describe our proposed sensor drift detection and soft calibration algorithm. In Section V, the proposed algorithm is evaluated and compared with both simulated and real-world datasets. We conclude and discuss possible future work in Section VI.

II. Related Work

Sensors are made of materials and usually left unattended for long periods of time in the field, sensor data drifts are inevitable due to natural decay in materials. Additionally, often in defence applications hundreds to thousands of sensor nodes deployed in nearly inaccessible locations under harsh environment, which can eventually accelerate sensor drift. As a result, sensor data would become inaccurate and unreliable after a while. For such applications, it is not practical to unmount and re-calibrate these sensors individually because of the enormous number and their remote locations [11], [12], [13].

When a sensor fails due to malfunctions or wear and tear, it may transmit faulty or dirty data rather than just stopping data transmission [14], [15]. In the uncontrolled environment where sensors are deployed, it is difficult to assess the accuracy of data without additional contextual information or sensor redundancy [16].

In [16], authors mainly focused on data-centric approaches such as rule-based or anomaly detection to identify faulty data, but these have limitations including the potential for faulty data to mimic non-faulty data and the high false positives and negatives that can result from temporal and spatial dependency [16]. Additional contextual data is required to detect anomalies, but the approach is not always feasible due to the high cost and battery requirements, particularly in deployments with hundreds or thousands of sensors [17], [18], [19]. A sensor redundancy approach using two sensors in each device to validate abnormal data is presented in [20], but that is not practical for large-scale deployments. However, this approach is expensive and requires high battery consumption, making it unsuitable for large-scale deployments with hundreds or thousands of sensors.

Current research has focused on data-centric methods such as rule-based or anomaly detection to identify faults by analysing historical sensor data spanning from days to years [16]. However, this does not indicate faulty data, particularly in hyper-local air pollution contexts where variations are high [17]. Anomaly detection relies on integrating contextual information and these dependencies can introduce high rates of false positives and negatives due to localised data fluctuations[19], [21], [18].

Several studies have shown that sensors begin to experience drift after a few months of deployment due to factors such as wear and tear, ageing, and semiconductor impurity effects [22], [23]. Periodic calibration of the sensor is a common approach used to correct drift, where the deployed sensor is placed alongside a high-end sensor at intervals. However, this method is laborious, costly, and necessitates bringing the sensor back for co-location [22]. In [24], authors have proposed blind calibration techniques using learning algorithms to calibrate sensors based on the assumption that nearby sensors' data should be highly correlated. However, these learning algorithms perform poorly in cases where nearby sensors record differing pollution levels due to diverse emission sources, leading to hyper-local variations. Such approaches depend on substantial historical sensor data (ranging from a few hours to years) to model the behaviour of the sensed data.

IoT sensor data arrives continuously and tends to change over time. Generally, a concept or sensor drift is defined as the data distribution over a certain period when the sensor is under calibration. In the real world, concepts in a data stream often change with time rather than staying static due to various real-life scenarios, such as changes in operating load conditions, ageing, environmental conditions, and several other factors. In data streams, the concepts of interest are often dependent on unknown context and can change in unexpected ways, a phenomenon referred to as concept drift in the field of machine learning and predictive analytics. Concept drifts can have a negative impact on the accuracy of data analysis and decision-making systems, causing predictions to become less precise as time goes on. A practical illustration of concept drift is in a smart factory environment, where numerous IIoT devices and sensors gather information on the status of machines and factory operations [9]. These data are transmitted to a cyber-physical system (CPS) which will then be used for health checks and different analytical purposes. Concept drifts can often occur in unpredictable ways, making it difficult to identify when, where, and why the changes in concepts have occurred.

To calibrate the sensors, traditionally a known stimulus is triggered to the sensor and measure the response or physically visit the site to unmount and calibrate [25]. The process of identifying the gain and offset in cases of linear drift involves comparing the initial input with the corresponding output. It has been noted in research that sensors calibrated to factory settings upon deployment will exhibit linear deviations from the actual ground truth, characterised by specific gains and

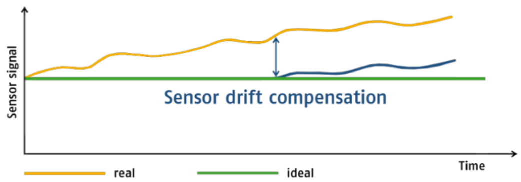


Fig. 1: Representation of sensor drift in temperature sensor

offsets for each sensor [26]. A method has been developed to calculate these gains and offsets through the use of subspace matching. While the method showed promising results in a controlled setting, it was less effective in the presence of noise and other interference. In literature, multiple statistical techniques were reported to calibrate sensors data but mostly those were designed considering a specific environmental condition or network [27], [28]. These methods can struggle with non-linear drift behaviours and are sensitive to noise, limiting their effectiveness in complex environments. In contrast, VAEs capture non-linearities in data by learning a probabilistic latent distribution. By modelling the true data distribution more accurately through this latent space, VAEs provide better generalisation and robustness to noise. This makes VAEs a more powerful tool for sensor drift calibration in dynamic and uncertain environments [8], [5]. Researchers have shown that the effective life of the network can be extended by detecting sensors that are drifting and correcting their measurements [6], [28]. Hence, a more generalised solution is required to adopt dynamically for different environment and varying sensors readings [29].

III. PRELIMINARIES

Sensor drift is the result of a change in the properties of one or more sensor components, often due to wear and tear or degradation. This change is gradual and can go unnoticed for a long time. Usually, a sensor is considered to have drifted when there is a significant difference in the data between the deployed sensor and a reference sensor.

As shown in Figure 1, the sensor drift is monitored by a regular comparison routine. The sensors are calibrated before being deployed on-site. With respect to time, the actual sensor value start to deviate gradually which might continue to transmit faulty or dirty data [14]. Additional contextual information or on-site calibration is required to assess the accuracy of the data, as the environment in which these sensors are deployed is uncontrolled. In some cases, there could be hundreds of sensors need inspection for calibration which might not be always feasible, frequent manual calibration is impractical and cost prohibitive. Hence, there is a significant need for soft calibration of sensors or sensor drift compensation. A intelligent soft calibration system eventually will increase the effective lifetime of the sensors.

In our approach, the drift detection will be handled by "Memory" module as shown in Figure 2. The memory module is a real-time streaming and data validation platform which will receive data from sensors and keep track of the individual sensor data distribution for validation purposes. To address the computational limitations of IoT devices, we centralise the data processing and model training on a central server where the Memory module resides. The IoT sensors are responsible solely for data collection and transmission, minimising their processing burden and energy consumption. The central server performs the complex sensor drift detection and calibration using the VAE model. By offloading the computationally intensive tasks to the server, our system ensures efficient operation of IoT devices while providing accurate and timely calibration of sensor data before any decision-making processes. In the drift detection module, KL divergence algorithm is utilised to ensure new streams of data from sensors follow the expectational distribution. The expectational distribution in the memory component will utilise incremental update of distribution to make the system adaptive for any environment. If the new streams of data fail to satisfy at expectational distribution, data will be passed to third component of the system which is error/loss estimation and drift compensation of the sensor data [30].

Sensor data drift can be explained, given a sample instance $X\epsilon C_i$ which can be classified as -

$$p(C_i|X) = \frac{P(c_i)P(X|c_i)}{p(X)} \tag{1}$$

To rephrase, while the input dis tribution $p(c|X)$ undergoes changes, $p(X)$ stays unchanged.

Imagine an environment equipped with n sensors. For every discrete moment t, define $\mathbf{x_t} = [x_{1,t}, x_{2,t}, \ldots, x_{n,t}]^T$ as the ideal signal that the sensors aim to measure, where $x_{i,t}$ indicates the true signal value of sensor i at time t devoid of any drift or noise. Clearly, $\mathbf{x}$ exists within an n-dimensional measurement space, identified as M.

Define y as the actual reading from the sensors. Given that each sensor is subject to unknown drifts and noise, we propose $y_{i,t} = x_{i,t} + d_{i,t} + \nu_{i,t}$, where $y_{i,t}$ represents the observed measurement, $x_{i,t}$ the actual, albeit unknown, true value, $d_{i,t}$ the unknown sensor drift, and $\nu_{i,t}$ the noise in measurement. This relationship can be succinctly expressed in vector form:

$$\mathbf{y}_t = \mathbf{x}_t + \mathbf{d}_t + \nu_t. \tag{2}$$

The challenge in soft drift calibration lies in deducing the original or true values $\mathbf{x}$ from the drifted and noisy data $\mathbf{y}$.

In equation 2, $\mathbf{y}_t$ is the only directly measurable component, necessitating additional constraints within the equation to effectively estimate the drift. This process aims to estimate and compensate for unknown sensor drift and noise in sensor measurements, ultimately enhancing the accuracy of the recorded data. It is postulated that the true signal resides within a subspace of M, termed S, which is of lower dimensionality. Let the dimension of S be represented by r, with r being less than n.

In the VAE framework, the representation of sensor signals, $\mathbf{x}_t$, is modelled through a latent distribution. This latent distribution is denoted as $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_t)$, where $\mathbf{z}$ represents the latent variables and $\boldsymbol{\phi}$ are the parameters of the encoder network. The encoding process, parameterised by $\boldsymbol{\phi}$, transforms the input sensor signal $\mathbf{x}_t$ into a distribution over latent variables.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3503616

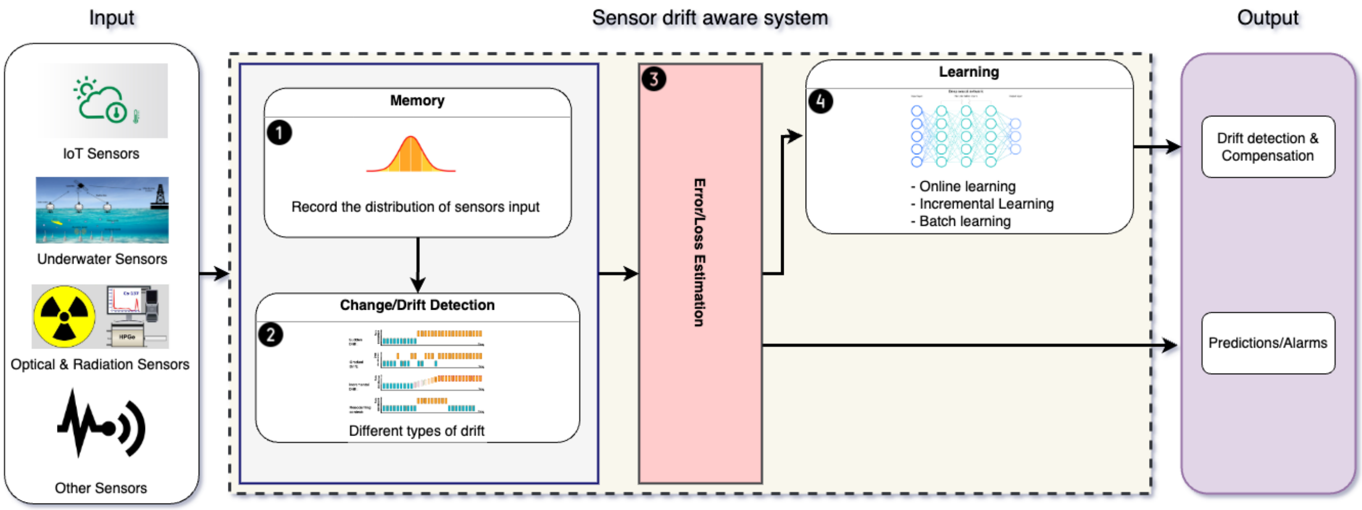


Fig. 2: Sensor drift aware soft calibration system

This provides a probabilistic representation that captures the underlying structure and variability within the sensor signals as shown in Figure 3.

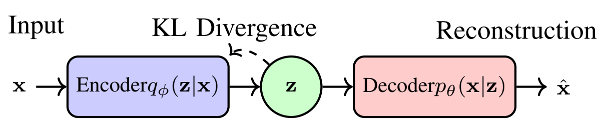


Fig. 3: Illustration of the VAE model. The input $\mathbf{x}$ is encoded into a latent variable $\mathbf{z}$ via the encoder $q_\phi(\mathbf{z}|\mathbf{x})$. The decoder $p_\theta(\mathbf{x}|\mathbf{z})$ reconstructs the input, while the KL divergence regularises the latent space during training.

Consider the linear source-sensor model. In a sensing environment influenced by several signal sources denoted by $\mathbf{s} = [s_1, s_2, \ldots, s_r]^T$, the signal detected at each sensor is derived from a linear mix of these sources. Consequently, $x_i = \sum_{j=1}^{r} a_{ij} s_j$, where a_{ij} represents the influence of source j on sensor i. This model is redefined in vector form as $\mathbf{x} = A\mathbf{s}$, where A is an $n \times r$ matrix and the entry at the i-th row and j-th column is a_{ij}. The true signal vector $\mathbf{x}$ therefore occupies the r-dimensional column space of A, with the signal space S being defined as $S = \text{col}(A)$. This framework has been applied in studies such as [31] and [32], which discuss temperature and light sensors, respectively.

For systems with non-linear characteristics, kernel techniques are useful to project the non-linear signal subspace into a space of higher dimension.

Define $S^\perp$ as the orthogonal complement in M of the signal subspace S, forming a p-dimensional subspace where $p = n - r$. Let $\{\phi_1, \ldots, \phi_p\}$ be the orthonormal basis for $S^\perp$. For any true sensor measurement $\mathbf{x}_t \in S$,

$$\phi_i^T \mathbf{x}_t = 0, \quad i = 1, 2, \ldots, p. \tag{3}$$

Consider $\Phi = [\phi_1, \phi_2, \ldots, \phi_p]^T$ as a $p \times n$ matrix where each row corresponds to a basis vector of $S^\perp$, leading to

$$\Phi \mathbf{x}_t = \Phi(\mathbf{y}_t - \mathbf{d}_t - \nu_t) = 0. \tag{4}$$

From Eq. 4, it follows that

$$\Phi \mathbf{y}_t = \Phi \mathbf{d}_t + \mathbf{v}_t. \tag{5}$$

Here, $\mathbf{v}_t = \Phi \nu_t$ indicates that $\Phi \mathbf{y}_t$ is primarily influenced by sensor drift. The matrix Φ serves as the observation matrix, and assuming its known values, we can observe p-dimensional sensor drifts.

Let $\mathbf{z}_t = \Phi \mathbf{y}_t$, substituting in Eq. 5 gives:

$$\mathbf{z}_t = \Phi \mathbf{d}_t + \mathbf{v}_t. \tag{6}$$

In this, $\Phi \in \mathbb{R}^{p \times n}$ (where $p < n$) is identified as the observation matrix, $\mathbf{d}_t$ is the unknown sensor drift, $\mathbf{v}_t$ represents the random noise, and $\mathbf{z}_t$ transforms the sensor measurement.

Hence, we convert the problem of sensor drift calibration into two subproblems from the underdetermined linear system represented by Eq. 6

a) Constructing the Observation Matrix Φ

To construct the observation matrix Φ that is orthogonal to the subspace S where the sensor measurements $\mathbf{x}_t$ reside:

1) Identify the subspace S as the column space of matrix A, which maps the signal sources $\mathbf{s}$ to the sensor readings $\mathbf{x}$.
2) Compute the orthogonal complement $S^\perp$ of S with dimension $p = n - r$.
3) Use Singular Value Decomposition (SVD) of A to find the basis vectors for $S^\perp$ from the vectors corresponding to zero singular values.
4) Form the matrix Φ by stacking these basis vectors as rows, resulting in a $p \times n$ matrix.

$$\Phi = \begin{bmatrix} \phi_1^T \\ \phi_2^T \\ \vdots \\ \phi_p^T \end{bmatrix}$$

b) Estimating the Drift Vector $\mathbf{d}_t$

Given the under-determined system represented by:

$$\mathbf{z}_t = \Phi \mathbf{d}_t + \mathbf{v}_t$$

The steps to estimate $\mathbf{d}_t$ are:

1) Formulate the problem as a noisy linear system.
2) Apply least squares estimation:
$$\hat{\mathbf{d}}_t = (\Phi^T\Phi)^{-1}\Phi^T\mathbf{z}_t$$
3) If $\Phi^T\Phi$ is not invertible, use Tikhonov regularization:
$$\hat{\mathbf{d}}_t = (\Phi^T\Phi + \lambda I)^{-1}\Phi^T\mathbf{z}_t$$
where λ is a regularisation parameter.
4) Consider iterative methods for large systems.
5) Validate the model by comparing $\mathbf{x}_t = \mathbf{y}_t - \hat{\mathbf{d}}_t$ against true values or use statistical error metrics.

By solving this optimisation problem, we can estimate the drift vector $\hat{\mathbf{d}}_t$ that minimises the difference between the observed transformed sensor measurements $\mathbf{z}_t$ and the product of the observation matrix Φ and the drift vector $\mathbf{d}_t$.

A. Theoretical Foundations of the VAE-Based Approach

1) Why VAEs are Superior for Sensor Drift Calibration: VAEs are generative models that learn a probabilistic mapping from input data to a latent space and back to the data space. The encoder network $q_\phi(\mathbf{z}|\mathbf{x})$ maps input data $\mathbf{x}$ to a latent representation $\mathbf{z}$, while the decoder network $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ reconstructs the data from the latent variables. The VAE optimises the evidence lower bound (ELBO), which consists of the reconstruction loss and the KL divergence between the approximate posterior and the prior distribution [33]:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})\right] - D_{\mathrm{KL}}\left(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \,\|\, p(\mathbf{z})\right). \tag{7}$$

By minimising this loss, the VAE learns to estimate the drift $\mathbf{d}_t$ that explains the observed deviations in the sensor measurements.

This framework allows the VAE to capture complex, non-linear relationships in the data, making it well-suited for modelling sensor measurements that may exhibit intricate drift patterns. Unlike traditional methods, VAEs do not assume linearity and can handle multi-modal data distributions. The probabilistic nature of VAEs enables the model to quantify uncertainty, which is crucial for reliable drift detection and calibration [8].

2) Assumptions Underlying the VAE Approach: Our VAE-based sensor drift calibration relies on several key assumptions:

- **Low-Dimensional Drift Representation:** The drift $\mathbf{d}_t$ can be effectively captured in a lower-dimensional latent space.
- **Orthogonality to True Signal Subspace:** The observation matrix Φ is orthogonal to the true signal subspace S, ensuring that $\Phi\mathbf{x}_t = \mathbf{0}$.
- **Gaussian Noise Assumption:** The measurement noise $\boldsymbol{\nu}_t$ is assumed to be Gaussian, aligning with common VAE likelihood models.
- **Representative Calibration Data:** The data collected during the calibration phase is representative of the normal sensor behaviour without drift.

3) Advantages Over Traditional Methods: By integrating the sensor drift modelling with VAEs, our method offers several advantages over traditional calibration techniques:

- **Non-Linear Drift Modelling:** VAEs can capture complex, non-linear relationships in drift patterns that linear methods may miss.
- **Probabilistic Framework:** The VAE's probabilistic nature allows for uncertainty quantification in drift estimation, improving robustness.
- **Latent Space Representation:** The VAE learns a latent representation of the drift, facilitating better generalisation and adaptability.
- **No Need for Manual Intervention:** The method operates without requiring manual recalibration or handcrafted features.

Once the drift $\mathbf{d}_t$ is estimated using the VAE, we can compensate for it in the sensor measurements:

$$\hat{\mathbf{x}}_t = \mathbf{y}_t - \hat{\mathbf{d}}_t, \tag{8}$$

where $\hat{\mathbf{d}}_t$ is the estimated drift from the VAE. This provides an estimate $\hat{\mathbf{x}}_t$ of the true sensor values and a mechanism to estimate $\mathbf{d}_t$ without relying on traditional least squares or regularisation methods, offering improved handling of noise and non-linearities.

IV. Drift Detection & Calibration

A. Assumptions and Overview

As was discussed in section III, by assuming that the ground-truths of sensor measurements are in a lower dimensional signal subspace compared to the measurement space, we convert the drift calibration problem into constructing and solving the linear system represented by Eq. (5). The proposed drift calibration method has three phases, learning phase, drift detection phase and calibration phase. We assume that sensors are calibrated before deployment, so we can infer that: a) within a short period after sensors are deployed, the drift should be zero or insignificant; b) within a reasonably long period, a few (less than p) sensors are drifted. We further assume that the signal subspace is time invariant. This assumption is appropriate in many applications, although the signal value changes a lot, the signal subspace is decided by the environmental structure and sensors' geographic locations, which change very slowly over time. In the training phase, leveraging sensor data acquired during a brief period post-deployment with minimal drift, we employ principal component analysis (PCA) to derive the orthonormal basis for both the signal subspace and its orthogonal complement. Subsequently, the VAE latent distribution is harnessed to model the underlying distribution of sensor signals. This distribution, denoted as $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_t)$, captures the latent variables given the input sensor signal $\mathbf{x}_t$.

During the calibration phase, assuming the signal space remains temporally invariant, Eq. (5) still holds true, while a sparse number of sensors may exhibit drift. This sparsity property of the sensor drift vector $\mathbf{d}_t$ is exploited. Through the subtraction of the estimated drift from the sensor readings, a soft calibration of sensors becomes achievable.

B. Learning Observation Matrix

The process of transforming the input sensor signal $\mathbf{x}_t$ into a distribution over latent variables $\mathbf{z}$ involves both encoding and sampling steps. In 9-

- $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_t)$ represents the conditional distribution of the latent variables $\mathbf{z}$ given the input sensor signal $\mathbf{x}_t$.
- $\boldsymbol{\phi}$ represents the parameters of the encoder network, which are learned during the training of the VAE.
- The encoder network transforms the input sensor signal $\mathbf{x}_t$ into a distribution over latent variables $\mathbf{z}$.

1) Encoding Step: The encoder network parameterised by $\boldsymbol{\phi}$ produces the parameters of the distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_t)$, which is typically assumed to be Gaussian. In mathematical terms, this can be expressed as follows:

$$q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_t) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}_t), \mathrm{diag}(\boldsymbol{\sigma}_{\boldsymbol{\phi}}(\mathbf{x}_t))^2). \tag{9}$$

where $\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}_t)$ and $\boldsymbol{\sigma}_{\boldsymbol{\phi}}(\mathbf{x}_t)$ are the mean and standard deviation vectors computed by the encoder.

2) Reparameterisation Trick: To make the sampling process differentiable, the reparameterisation trick is often employed. Instead of directly sampling from $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_t)$, we sample from a simpler distribution, typically a standard Gaussian, and then transform the samples to match the parameters of $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_t)$. This is done as follows:

$$\mathbf{z} = \boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}_t) + \boldsymbol{\sigma}_{\boldsymbol{\phi}}(\mathbf{x}_t) \odot \boldsymbol{\epsilon}. \tag{10}$$

where $\boldsymbol{\epsilon}$ is a sample from a standard Gaussian distribution and $\odot$ denotes element-wise multiplication.

In summary, the encoder network transforms the input sensor signal $\mathbf{x}_t$ into a distribution over latent variables $\mathbf{z}$ by computing the mean and standard deviation parameters through neural network transformations and then utilising the reparameterisation trick to sample from this distribution in a differentiable manner [34].

C. Drift Model

Before introducing the drift estimation algorithm, let's define the prior probabilistic model for sensor drift within the framework of a Variational Autoencoder (VAE). We assume that the drifts of different sensors are independent, and the increment of each sensor's drift at each time instant follows an independent Gaussian distribution:

$$d_{i,t} = d_{i,t-1} + \delta_{i,t}, \quad \delta_{i,t} \sim \mathcal{N}(0, \sigma^2). \tag{11}$$

Here, $d_{i,t}$ represents the drift value for sensor i at time instant t, and $\delta_{i,t}$ is the increment of sensor i's drift at time t.

As the proposed algorithm serves as a general-purpose calibration algorithm, we make the suitable assumption that the increment of sensors' drifts follows a zero-mean Gaussian distribution.

Consistent with the assumption that sensors are calibrated before deployment, we initialise the drift values to be zero at the beginning:

$$d_{i,0} = 0$$

. At any time instant t, the drift values are the sum of a series of Gaussian increments. Consequently, the accumulated drifts remain Gaussian [33]. The expectation of sensor drift can be expressed as:

$$d_{i,t} = d_{i,t-1} + \delta_{i,t}, \quad \delta_{i,t} \sim \mathcal{N}(0, \sigma^2). \tag{12}$$

The expectation of the sensor drift at any time instant t can be expressed as the sum of the expectations of the increments up to that time:

$$\mathbb{E}[d_{i,t}] = \sum_{k=1}^{t} \mathbb{E}[\delta_{i,k}]. \tag{13}$$

Since each $\delta_{i,k}$ follows a zero-mean Gaussian distribution ($\mathcal{N}(0, \sigma^2)$), the expectation of each increment is zero:

$$\mathbb{E}[\delta_{i,k}] = 0. \tag{14}$$

Therefore, the expectation of sensor drift simplifies to:

$$\mathbb{E}[d_{i,t}] = 0. \tag{15}$$

This result is consistent with the assumption that the increments of the sensor drifts follow a zero-mean Gaussian distribution.

The sensors' drift vector $\mathbf{d}_t$ at time instant t and the corresponding diagonal covariance matrix $\boldsymbol{\Sigma}_t$ can be expressed as follows:

$$\mathbf{d}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_t). \tag{16}$$

Here, $\mathbf{d}_t$ represents the sensors' drift vector at time t and $\boldsymbol{\Sigma}_t$ is a diagonal covariance matrix.

This notation indicates that the sensors' drift vectors at time t follow a multivariate normal (Gaussian) distribution with a mean vector of zeros ($\mathbf{0}$) and a diagonal covariance matrix $\boldsymbol{\Sigma}_t$. The diagonal covariance matrix $\boldsymbol{\Sigma}_t$ characterises the variances of individual components of the sensors' drift vector at time t. The covariance between different components is assumed to be zero, reflecting independence.

$$\boldsymbol{\Sigma}_t = \begin{bmatrix} \sigma_{1,t}^2 & 0 & \dots & 0 \\ 0 & \sigma_{2,t}^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_{n,t}^2. \end{bmatrix} \tag{17}$$

Here, $\sigma_{i,t}^2$ is the variance of sensor-i's drift at time instant t.

D. Drift Detection

We leverage the VAE's ability to learn the underlying distribution of sensor readings in lower dimensions. The idea is to train a VAE on a dataset representing normal sensor operation, and then use the trained model to reconstruct sensor readings in real-time [35]. An increase in the reconstruction error may indicate sensor drift or anomalous behaviour.

- Train a VAE for Normal Operation Train a VAE using data from the normal operation of the sensor. This is the baseline model.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3503616

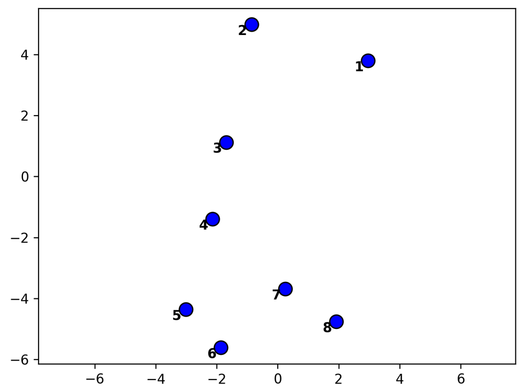


Fig. 4: Positions of the sensors in the simulated environment.

$$\Theta = \arg\max_{\Theta} \sum_{i=1}^{N} \mathbb{E}_{z\sim q_\phi(z|x^{(i)})}\Big[\log p_\Theta(x^{(i)}|z) - D_{KL}(q_\phi(z|x^{(i)})||p(z))\Big]. \tag{18}$$

where Θ represents the parameters of the VAE, $x^{(i)}$ is a data point, z is a latent variable, $p(z)$ is the prior on the latent space, and $q_\phi(z|x^{(i)})$ is the approximate posterior.

- **Calculate Reconstruction Loss:** For each new sensor reading x_t, calculate the reconstruction loss using the trained VAE:

$$\text{Reconstruction Loss}_t = \text{Loss}(x_t, \text{VAE Reconstructed}(x_t)) + \text{Additional Terms} \tag{19}$$

The specific form of the loss function depends on the nature of the data. For our work, Mean Squared Error is used as performance metric.

- **Define Drift Threshold:** Establish a threshold for the reconstruction loss that indicates a significant deviation from the normal operation. This threshold (ϵ) can be determined through analysis of the reconstruction loss distribution on normal data.
- **Detect Drift:** When the reconstruction loss ($\text{Reconstruction Loss}_t$) exceeds the threshold (ϵ), it suggests that drift is occurring.

The decoder module on VAE output a normalised data. We need to scale this difference by the range or standard deviation of the normal data to get a actual measure of drift magnitude.

This approach allows us to detect sensor drift by leveraging the VAE's ability to learn the normal distribution of sensor readings and identify deviations through reconstruction errors.

E. Drift Estimation

The problem is modelled using VAE for sensor drift detection. In VAE, two main components, encoder and decoder is trained using sample observations during calibration phase of the deployed sensors. Let's assume sensor data is represented as X with N samples and D dimensions (features).

1. Encoder: The encoder maps sensor data x_i to a latent variable z_i with a Gaussian distribution in the latent space:

$$z_i \sim \mathcal{N}(\mu_i, \sigma_i^2). \tag{20}$$

where, μ_i and $\log(\sigma_i^2)$ are outputs of the encoder neural network.

2. Reparameterisation Trick: To ensure differentiability for backpropagation, we sample from a standard Gaussian distribution $\epsilon_i \sim \mathcal{N}(0,1)$ and transform it using the predicted parameters:

$$z_i = \mu_i + \sigma_i \cdot \epsilon_i. \tag{21}$$

3. Decoder: The decoder maps z_i back to the data space:

$$\hat{x}_i = \text{Decoder}(z_i). \tag{22}$$

4. Loss Function: The VAE loss function consists of two components:

- Reconstruction Loss: Measures the difference between the original and and reconstructed data using a suitable loss metric ($L(x_i, \hat{x}_i)$).
- KL Divergence Loss: Measures how close the approximate posterior $q(z_i|x_i)$ is to the prior $p(z_i)$ in the latent space. The total loss is the sum of the reconstruction and KL divergence losses.

For example, using mean squared error (MSE) for the reconstruction loss and the KL divergence formula:

$$\text{Loss}(x_i, \hat{x}_i, \mu_i, \sigma_i) = L(x_i, \hat{x}_i) + \frac{1}{2} \sum_{j=1}^{D} (\sigma_i^2 + \mu_i^2 - \log(\sigma_i^2) - 1). \tag{23}$$

5. Training: Train the VAE by optimising the model's parameters to minimise the loss function using gradient descent-based methods.

6. Sensor Drift Detection: After training, for a new sensor reading x_{new}:

- Encode x_{new} to get μ_{new} and σ^2_{new}.
- Sample z_{new} using the reparameterisation trick.
- Decode z_{new} to get the reconstructed $\hat{x}_{\text{new}}$.
- Compute the reconstruction error $L(x_{\text{new}}, \hat{x}_{\text{new}})$.
- If the error exceeds a predefined threshold, classify it as a drift as mentioned in previous drift detection section.

The relationship between the exact value and the reconstruction loss in a VAE can be represented mathematically as follows:

Let's denote, x as the exact input value (data point) of sensor measurement. $x_{\text{reconstructed}}$ as the output generated by the VAE's decoder, which is the reconstruction of x. $\mathcal{L}_{\text{reconstruction}}$ as the reconstruction loss, calculated using mean squared error (MSE). The mathematical equation showing the relationship between the exact value and reconstruction loss is:

$$\mathcal{L}_{\text{reconstruction}} = \text{Loss}(x, x_{\text{reconstructed}}). \tag{24}$$

Here, Loss represents the specific loss function used to measure the difference between the exact value x and the reconstructed value $x_{\text{reconstructed}}$. In our case, mean squared error (MSE) is used as the loss function, the equation 24 becomes:

$$\mathcal{L}_{\text{reconstruction}} = \frac{1}{N}\sum_{i=1}^{N}(x_i - x_{\text{reconstructed}_i})^2. \qquad (25)$$

N is the total number of elements in the data point x. x_i and $x_{\text{reconstructed}_i}$ are the i-th elements of the exact value x and the reconstructed value $x_{\text{reconstructed}}$, respectively. In summary, the reconstruction loss quantifies the difference between the exact input value and the generated output of the VAE's decoder [36], [37].

As illustrated, Figure 5 shows the end-to-end system integrating the VAE model into the sensor data processing pipeline for drift detection and calibration. By modelling the normal behaviour of sensor data in the latent space during training, the VAE becomes sensitive to deviations caused by drift. The reconstruction error serves as an indicator of such deviations. When the error surpasses a certain threshold, the system identifies the presence of drift and calibrates the sensor readings accordingly. This automated process enhances the robustness and reliability of sensor networks in dynamic and uncertain environments.

Algorithm 1 VAE-Based Sensor Drift Calibration

Require: Historical sensor data $\{\mathbf{x}_t\}_{t=1}^{T}$
Ensure: Calibrated sensor readings

```
Training Phase:
for each epoch do
    for each batch x do
        Encode: z ~ q_φ(z|x)
        Decode: x̂ ~ p_θ(x|z)
        Compute Loss: L = D_KL(q_φ(z|x) || p(z)) + L_rec
        Update Parameters: φ, θ ← Optimizer(φ, θ, L)
    end for
end for
Application Phase:
for each new data point x_t do
    Encode: z_t ~ q_φ(z|x_t)
    Decode: x̂_t ~ p_θ(x|z_t)
    Compute Reconstruction Error: e_t = ||x_t − x̂_t||
    if e_t > ϵ then
        Calibrate sensor reading: x_t^calibrated = x̂_t
    else
        x_t^calibrated = x_t
    end if
end for
```

V. EVALUATION

In this section, we have used the simulated data to evaluate the performance of the proposed algorithm. Based on the performance of the algorithm, the optimal technique is selected to test on real-world data. We also compare the proposed drift calibration techniques with other methods. The performance of the algorithms is compared against recovery rate and mean square error (MSE).

For an individual sensor with T samples, let $D \in \mathbb{R}^T$ represent the drift vector, D_i denote the i-th element or the drift of the sensor, and $\hat{D}$ be the estimated sensor drift vector. The MSE of drift estimation is given by $\frac{\|\hat{D}-D\|_F^2}{\|D\|_F^2}$, where $\|\cdot\|_F$ denotes the Frobenius norm [38] of a vector.

Before introducing the term recovery rate, we first define a successful recovery. If the sensor is drifted, let D represent the drifted sensor, and in the estimated sensor drift vector $\hat{D}$, let $\hat{D}$ denote the element with the largest l_2 norm. A successful drift recovery occurs only when $D = \hat{D}$. During the simulation, we systematically test numerous scenarios involving drifted sensors, applying various drift estimation techniques to each scenario. Define T as the set of experiments for a single algorithm across different scenarios, and let $\hat{T}_s = \{T \mid D = \hat{D}\}$ represent the subset of these experiments where recovery is successful. The recovery rate is then calculated as $\frac{\|\hat{T}_s\|}{\|T\|}$, indicating the proportion of successful recoveries relative to the total number of trials.

The proposed drift calibration method are compared with the following techniques, including:

- Support Vector Regression and Kalman Filter (SVR-KF-oracle): a prediction-based drift calibration algorithm proposed in [6]
- Signal Space Projection and Kalman filter (SSP-KF-oracle): a modified version of SSP-KF proposed in [10], [25], where the drift detection algorithm is replaced by some certainty and model knows which data-point is drifted.
- Dynamic Residual Projection: a statistical technique known for effectively detecting concept drift in IoT scenarios proposed in [3], which employs dynamic residual projection for detecting and calibrating sensor drift.
- Blind Drift Calibration with Deep Learning: an ML/AI-based approach presented in [39], utilizing deep learning models to perform blind calibration of sensor drift without prior knowledge of the drift characteristics.
- Self-Calibration (Interpolation & Autoregression): An advanced self-healing and blind/self drift calibration technique using interpolation and autoregression for low-cost wireless sensor networks, as discussed in [40].
- VAE-SC: Our proposed technique, a Variational Autoencoder based Soft Calibration (VAE-SC) technique which uses lower dimension latent distributions of the sensors data while in calibration phases to train and estimate the drift quantity.

In this work, our focus is the performance of recovery rate and MSE of the drift estimation. Even though we have implemented KL Divergence based drift detection [41], [42] for our solution, in this work we will focus only recovery rate and MSE of the drift estimation.

The SVR-KF-oracle and SSP-KF-oracle are considered as ideal scenarios, as it removes the uncertainty associated with drift sensor detection. In this context, an oracle is a hypothetical drift detector capable of consistently and accurately identifying the subset of sensors affected by drift.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3503616

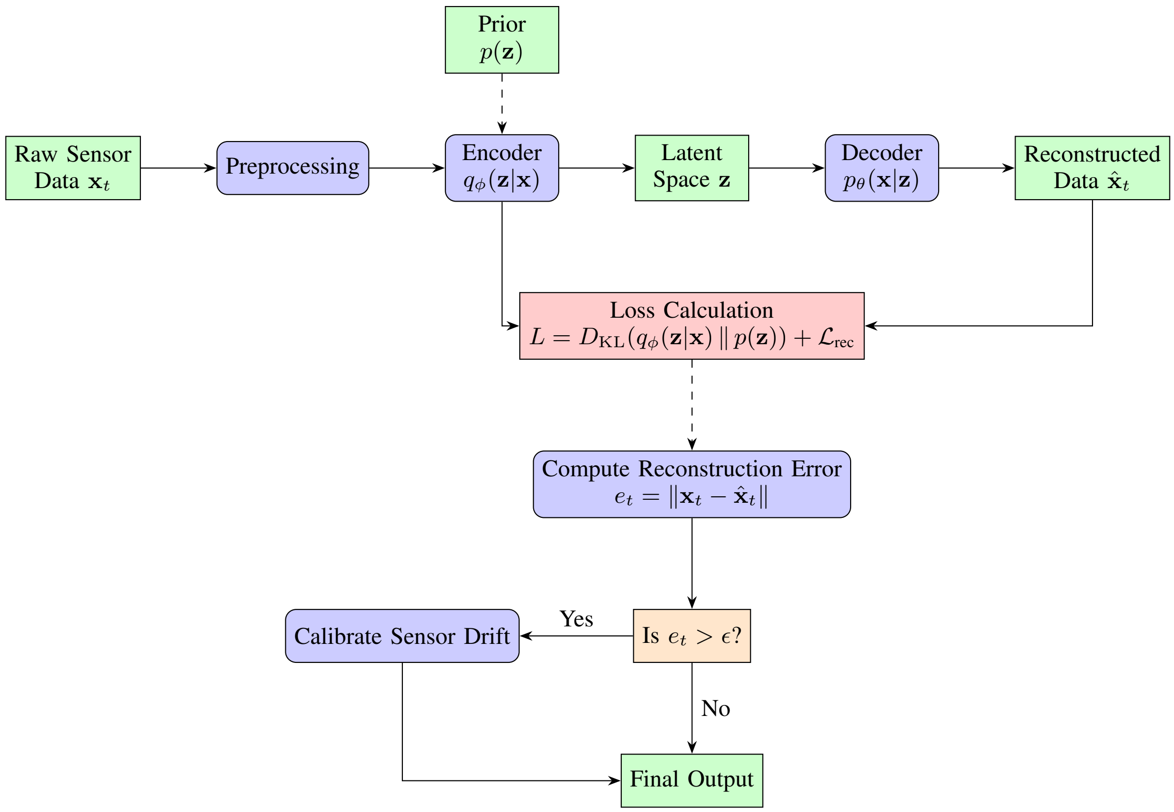


Fig. 5: VAE model training and application for sensor drift calibration.

The experiments were conducted using Python along with its numerical computation libraries, including SciPy [43] and PyTorch [44].

A. Datasets

While several open sensor datasets are available, their limitation lies in providing sensor measurements without accompanying ground truth information. The verification of calibration accuracy becomes challenging in the absence of a known reference.

In our experimental approach, we initially use simulated sensors data with diverse deployments to rigorously evaluate the performance of calibration algorithms across varying conditions. Additionally, we setup a sensor test-bed inside our research lab. This test-bed yields a dataset characterised by minimal drift, thereby enabling a robust validation of calibration algorithms.

To evaluate the performance of the our algorithm, we choose n = 8 simulated sensors data which are being transmitted to a central cloud server from the source locations. The sensors are randomly placed in a 6x6 radius area as shown in Figure 4. Initially, the sensors are assume to be drift free and there are no noise in the signal. Over time, the the sensor readings start to drift.

Each sensor's true measurement is a linear blend of each signal source. At any given moment t, the actual measurement for sensor j is

$$x_{j,t} = \sum_{j=1}^{r} s_{j,t}. \tag{26}$$

where $s_{j,t}$ is the signal value of source j at the time instant t. We assume after initial installation the sensors are deployed as in calibrating phase for t= S_{j,t_i} where t_i is i number of data points at time instance t. Hence, the cloud system server has sensors reading without any drift for t_i data points which is used to to generate latent distribution for our variational autoencoders method [45].

Algorithm 2 Generate Random Sensor Data with Drift Simulation

Require: Number of Sensors n, Drift Magnitude d
Ensure: Sensor Measurements x
  for $i \leftarrow 1$ to n **do**
    $x_i \leftarrow$ GenerateRandomValue() ▷ Generate a random value within a range for sensor i
    $x_i \leftarrow x_i + \text{RandNoise}(d)$ ▷ Add random noise for drift simulation
  end for

B. Real-world Dataset

We have deployed eight HTS221 temperature sensors using Arduino Nano [46] in our lab. The test-bed setup has two controlled chambers - in one chamber, the sensors were

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3503616

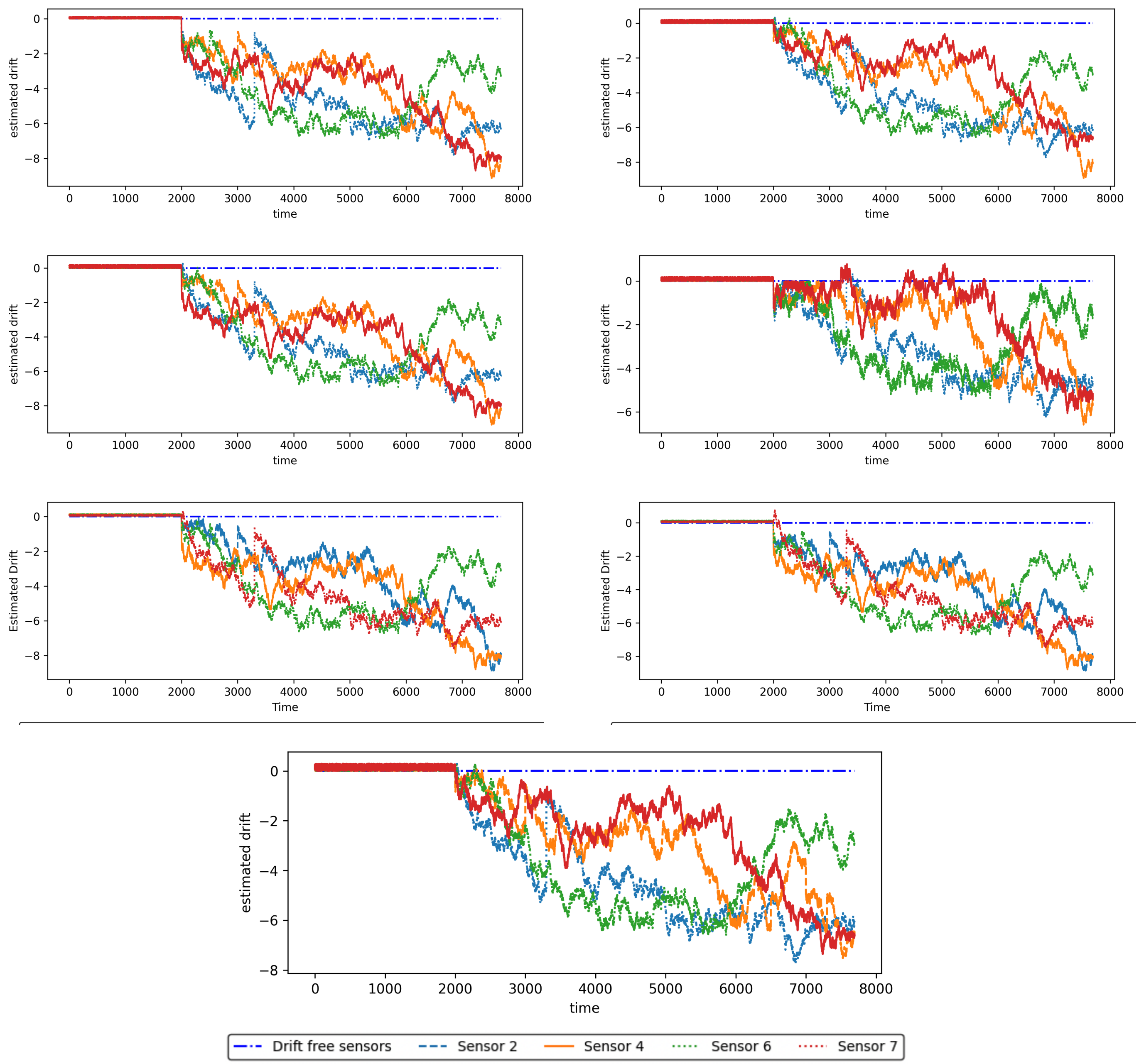


Fig. 6: Demonstration of calibration results from different algorithms where sensors 3, 4, 5, and 7 are drifted. The images represent: (a) simulated sensor drift, (b) SVR-KF-oracle, (c) SSP-KF-oracle, (d) VAE-SC, (e) Dynamic Residual Projection, (f) Blind Drift Calibration with Deep Learning, and (g) Self-Calibration Interpolation Autoregression.

exposed to high temperature and humidity, and in another chamber, sensors are in room temperature. We collect the sensor readings every 30 seconds using an open-source IoT data platform named ThingsBoard [47]. Before the installation of the sensors, all the sensors were placed together in the same environment to capture the differences in the readings. We found that the difference in temperature is less than 0.7°C, therefore, there is no significant difference in readings. We used the sensor data collected over six months and pushed all the data to a central cloud server, which is running on ThingsBoard. The calibration phase was 29 Oct, 2023 to 31 Dec, 2023 to collect the data. Then the sensors were deployed into two separate chambers in our control environment. We re-sample the data for both of these phases- 2000 data points from calibration phase and 10000 data points sampled from 1 Jan, 2024 - 31 March, 2024 where we observed some data drift in senor readings. In real world dataset, a latent distribution of the measurements during calibration phase is derived and use it as training data for VAE-SC.

C. Analysing Results from Simulated Experiments

In the simulated dataset, a set of m sensors is randomly selected to emulate drift. A random walk process, simulating the drift, is initiated starting at the time instance 2000. As a

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3503616

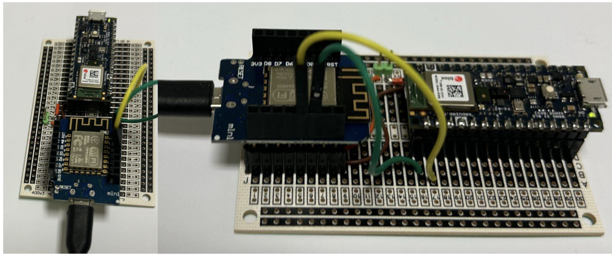


Fig. 7: Experimental setup for sending sensor reading to cloud

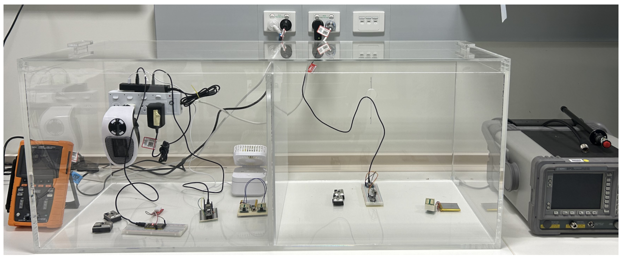


Fig. 8: Controlled chamber: sensors and data collection

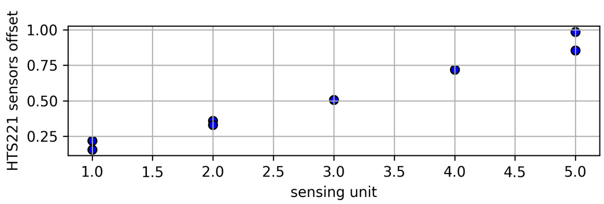


Fig. 9: Offsets of HTS221 sensors varies from 0.1 °C to 0.9 °C, so measurements from HTS221 sensor without calibration is not accurate

result, the initial 2000 samples of measurements remain drift-free and are utilised during the learning phase to estimate the observation matrix $\mathbf{\Phi}$ and train the SVR model employed by the SVR-KF method. The random walk process is generated as follows:

$$\begin{aligned} d_{i,1:2000} &= 0 \\ d_{i,t} &= d_{i,t-1} + \delta_{i,t}, \quad \delta_{i,t} \sim \mathcal{N}(0, \sigma^2). \end{aligned} \tag{27}$$

In the simulation setup, the drift increment variance, denoted by σ^2, is established at 0.03, determined through trial and error to optimally model the variability in sensor drift. This variance level helps model the variability of sensor drift over time. Throughout the learning phase, the threshold for PCA (Principal Component Analysis) is fixed at 0.99, which signifies that the PCA model retains 99% of the variance of the data, ensuring a comprehensive capture of the data's characteristics. In the observation matrix Φ, sensor readings from the calibration phase are utilised to extract the latent representation of the signal subspace, aiding in the identification and correction of any drift in the sensor measurements.

We let the number of drifted sensors m vary from 4 sensors. For each drift number m, there exist 8 combinations of m different drifted sensors and we randomly generate $8+4\times(m-1)$ different combinations of drifted sensors with independent drift. For each trial, different methods are used to estimate the sensor drift, and their performance is compared.

For our technique, we leverage VAE for drift detection, characterised by a latent space dimension of 16. The encoder and decoder architectures consist of LeakyReLU activations, batch normalisation, and sequential linear layers. numerical variables are embedded using two layers with input dimensions 8, outputting 16 dimensions. The noise term, sampled from a normal distribution ($\delta_{i,t} \sim \mathcal{N}(0, \sigma^2)$), is a crucial component for introducing variability in the latent space. This comprehensive set of parameters forms the foundation of our VAE model which are obtain based on lowest training loss during learning, tailored for effective drift detection in our experimental setting.

The comparative analysis of various methods in a simulated field is illustrated in Figure 11. In Figure 11(a), we present a comparison of the recovery rates for VAE-SC, SVR-KF-oracle, SSP-KF-oracle, Dynamic Residual Projection, Blind Drift DL, and Self-Calibration. Additionally, Figure 11(b) depicts the average Mean Squared Error (MSE) for each trial conducted with VAE-SC, SVR-KF-oracle, SSP-KF-oracle, Dynamic Residual Projection, Blind Drift DL, and Self-Calibration.

The experimental results on the simulated dataset demonstrate the superior performance of the proposed VAE-based Soft Calibration (VAE-SC) method compared to existing techniques. Key observations from these experiments include: (1) Across all methods, there is a marginal increase in the Mean Squared Error (MSE) as the number of drifted sensors increases, accompanied by a decrease in the recovery rate. This trend indicates that calibration becomes slightly more challenging as more sensors experience drift. (2) The VAE-SC method consistently achieves the highest recovery rate after drift, averaging approximately 91%, outperforming all other methods. (3) Both the Self-Calibration and SSP-KF-Oracle methods exhibit strong performance, with recovery rates of about 87% and 88% respectively after drift, but they do not match the consistency and accuracy of VAE-SC.

Furthermore, the VAE-SC method maintains relatively low and stable MSE values even as the number of drifted sensors grows, showcasing its robustness and resilience to multiple simultaneous drifts. In contrast, the SVR-KF-Oracle method shows a significant decline in recovery rate as the number of drifted sensors increases, dropping to around 71%, and exhibits the highest MSE after drift at approximately 0.21. This decline is primarily due to the accumulation of prediction errors inherent in SVR-KF-Oracle. The Dynamic Residual Projection and Blind Drift Deep Learning methods perform comparably, with recovery rates around 84% and MSE after drift around 0.15, indicating effective but slightly lower performance compared to VAE-SC.

These results highlight the efficacy of the proposed VAE-SC approach in successfully recovering sensor drift across various scenarios. Its ability to model complex, non-linear relationships in the sensor data through the probabilistic latent space allows for better generalisation and robustness to noise. The consistent performance of VAE-SC, regardless of the number of drifted sensors, underscores its potential as a powerful tool for sensor drift calibration in dynamic and uncertain environments.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3503616

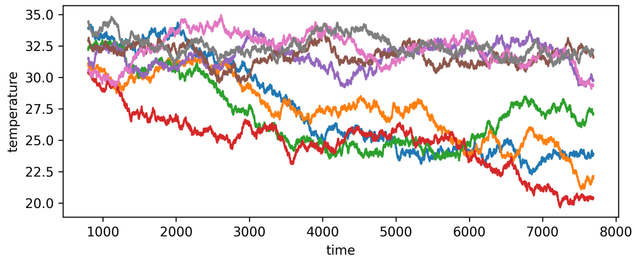
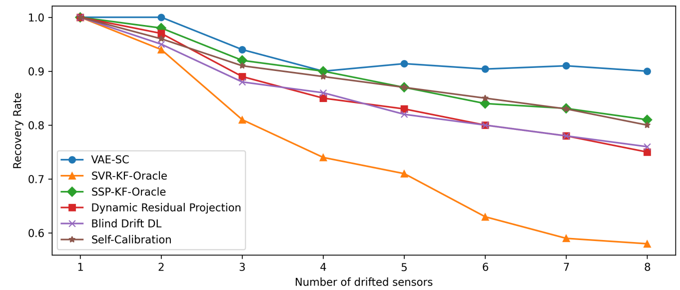
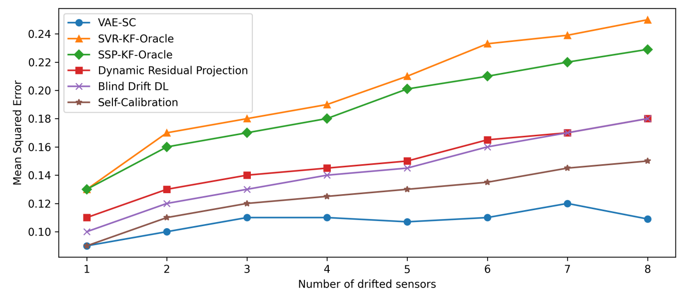


Fig. 10: Samples of dataset for evaluation: readings from each sensor are different.



(a)



(b)

Fig. 11: Simulated data when the number of drifted sensors varies from 1 to 8. Failed recoveries are eliminated in MSE calculation. (a) Recovery rate of VAE-SC and other techniques. (b) Mean square error of different methods.

D. Analysing Results from real-world Experiments

In this study, we assess the performance of various algorithms using real world datasets. The initial 2000 samples as mentioned in previous section, the measurements serve as the drift learning set for estimating the observation matrix ϕ and training the Support Vector Regression (SVR) model and VAE model.

We conducted an experiment using eight sensors. Among these sensors, four were subjected to extreme weather conditions created by a humidifier and a heater over a period of four weeks. We began to observe discrepancies in sensor readings and signs of drift.

Subsequently, we apply various calibration methods in each trial. For the calibration using VAE, we select hyperparameters, number of layers are three, the size of the latent space is 10 dimensions, the learning rate is set to 0.001 and ADAM Optimiser for performance validation. This hyperparameters were selected based on based on trial and error; and empirical evidence, aiming to optimize the performance and accuracy of the calibration process using the VAE technique [37], [34]. The resulting recovery rates and Mean Squared Errors (MSEs) are presented in Figure 13

TABLE I: Comparison of Recovery Rate and MSE for Different Techniques with Simulated Data

Technique	RR Before	RR After	MSE Before	MSE After
VAE-SC	1.00	0.92	0.09	0.11
SVR-KF	1.00	0.71	0.13	0.21
SSP-KF	1.00	0.88	0.13	0.20
Dynamic Residual Projection	1.00	0.84	0.11	0.15
Blind Drift-DL	1.00	0.84	0.10	0.15
Self-Calibration	1.00	0.87	0.09	0.13

Figure 13(a) illustrates that Variational Auto-encoder based soft calibration (VAE-SC) exhibits a superior recovery rate compared to Kalman based filter. The real-world dataset comprises 8 sensors organised in the controlled chambers with each group having 4 sensors, leading to relatively low inter-location correlation.

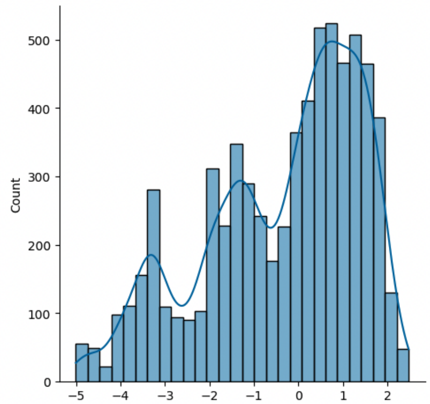
Figure 12 illustrates the distribution of sensor readings before and after drift occurred in the real-world dataset. The distribution plot of the real-world dataset reveals a noticeable change: initially, the data exhibited a relatively uniform distribution, with a smaller standard deviation and variance. However, following the onset of sensor drift, the distribution became right-skewed, indicating significant changes in sensor behaviour. These drifts occur gradually and at a slower pace, making them challenging for traditional approaches to detect.

Table II presents a comparison of the Mean Squared Error (MSE) and Recovery Rate before and after drift for different calibration techniques using real-world data. The experimental evaluation using real world data validates the effectiveness of the proposed VAE-based Soft Calibration (VAE-SC) method in practical sensor calibration scenarios. As shown in Table II, VAE-SC achieves the lowest Mean Squared Error (MSE) after drift (0.08) and the highest recovery rate after drift (0.98), outperforming all other methods tested. This indicates that VAE-SC is highly precise in correcting sensor readings and minimizing residual errors, even in the presence of real-world sensor noise and drift patterns.

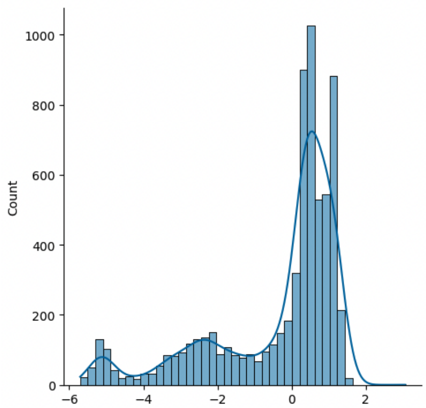
Compared to other techniques such as Dynamic Residual Projection and Blind Drift Deep Learning, which also demonstrate strong performance, VAE-SC provides superior accuracy and consistency. The minimal increase in MSE after drift for VAE-SC showcases its robustness against drift, maintaining reliable sensor readings over time. These findings underscore the potential of VAE-SC for practical deployment in dynamic environments, where reliable sensor data is critical for system performance and decision-making.

The Mean Squared Error (MSE) of different calibration results is presented in Figure 13. It is evident from the figure that Support Vector Regression with Kalman Filter (SVR-KF-oracle) and Sparse State Prediction with Kalman Filter (SSP-KF-oracle) achieve the lowest MSE. Notably, in the real-world

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3503616



(a)



(b)

Fig. 12: Distribution plots depicting sensor readings before and after drift in a real-world dataset, revealing significant changes in sensor behaviour. (a) Distribution of real-world data without drift (b) Distribution of real-world data with drift

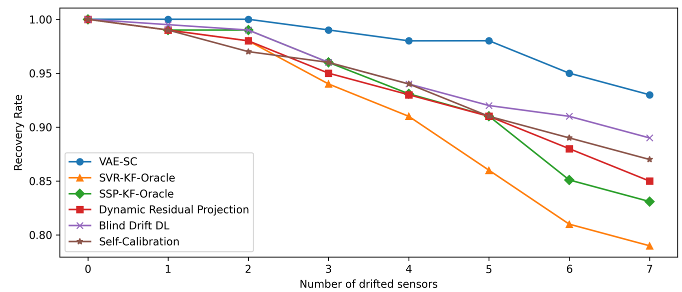
TABLE II: Comparison of Mean Squared Error and Recovery Rate for Different Techniques with Real-World Data

Technique	MSE Before	MSE After	RR Before	RR After
Proposed VAE-SC	0.06	0.08	1.00	0.98
SVR-KF-Oracle	0.07	0.09	1.00	0.90
SSP-KF-Oracle	0.13	0.12	1.00	0.92
Dynamic Residual Projection	0.09	0.08	1.00	0.93
Blind Drift Deep Learning	0.07	0.09	1.00	0.94
Self-Calibration	0.08	0.09	1.00	0.93

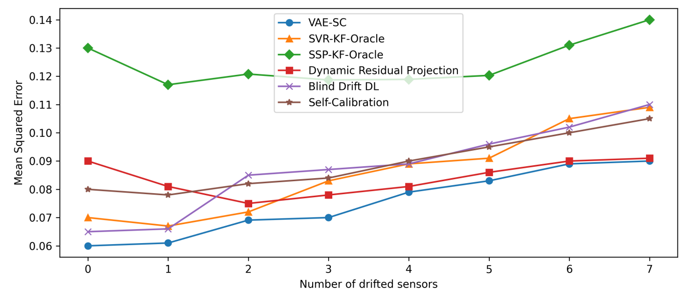
dataset, VAE-SC outcomes existing Kalman based techniques.

In addition, we test the generalisation ability of VAE-SC. As we generate sensor drifts using Algorithm 2 during training, we need to test VAE-SC performance using random data. In addition to the random drift, we also simulate four other types of drift: bilateral linear drift, positive linear drift, positive square-root drift, and sine drift. To simulate these kinds of drift, for each sensor, we first generate a random number called the end value e_i as the largest drift value it can reach. Next, we generate the linear drift by

$$d_{i,t} = e_i \times \frac{t}{T} \tag{27}$$



(a)



(b)

Fig. 13: Recovery rate and MSE of different methods in real-world data, regular field when the number (a) Recovery rate of VAE-SC and existing techniques. (b) Mean square error of different methods.

the square-root drift by

$$d_{i,t} = e_i \times \sqrt{\frac{t}{T}} \tag{28}$$

and the sine drift by

$$d_{i,t} = e_i \times \sin\left(r_i \pi \frac{t}{T}\right) \tag{29}$$

Figure 14 shows the recovery rate of VAE-SC under different drift levels.

Here, $d_{i,t}$ is the drift value of sensor i at time instant t; e_i is the end value; r_i is a random number sampled from $U(3, 4)$; and T is the total time length of the simulation.

We conducted extensive simulations to evaluate the generalisation ability of the proposed VAE-based Soft Calibration (VAE-SC) method against various types of drift, including linear (both positive and bilateral), non-linear (sine and positive square root), and stochastic (random walk) patterns. The recovery rates of VAE-SC under these drift types were analysed across a range of drift magnitudes, measured by the Root Mean Square Error (RMSE) of the drift.

As illustrated in Figure 14, the recovery rate of VAE-SC generally increases with the magnitude of the drift across all drift types. For small drift magnitudes (low RMSE values), the recovery rates are initially low, indicating that minor drifts are more challenging to detect and correct due to their similarity to normal sensor noise. However, as the drift magnitude increases, the recovery rates improve significantly, often exceeding 90% for higher RMSE values.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3503616

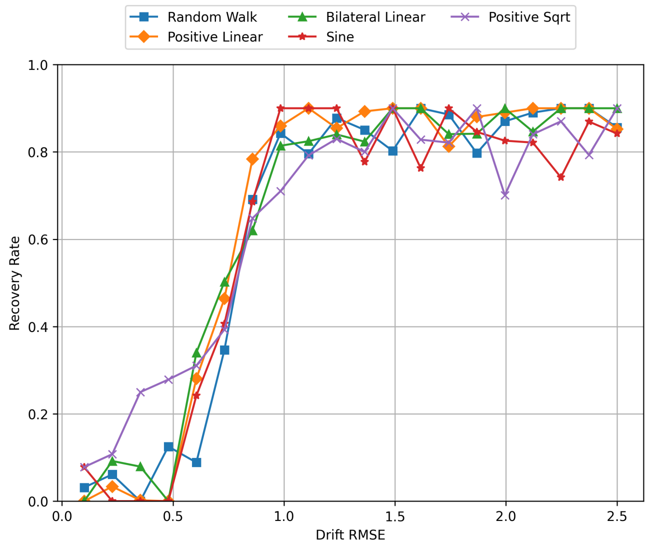


Fig. 14: Recovery rates of VAE-SC on different types of drift.

Specifically, for the *Random Walk* drift, the recovery rate improves from near zero at low RMSE to approximately 90% as the RMSE approaches 2.5. Similar trends are observed for the *Positive Linear* and *Bilateral Linear* drifts, where recovery rates exceed 85% for higher drift magnitudes. The *Sine* and *Positive Square Root* drifts, representing non-linear patterns, also show substantial increases in recovery rates with larger drift magnitudes, demonstrating the capability of VAE-SC to handle complex drift behaviours.

These results highlight that VAE-SC is effective at detecting and correcting larger drift magnitudes across various drift patterns. The method's ability to model non-linearities and capture the underlying data distribution enables it to generalise well to different types of drift. Additionally, we tested the noise tolerance of VAE-SC by introducing varying levels of measurement noise. The method maintained robust performance despite the added noise, further demonstrating its suitability for real-world applications where sensor measurements are often noisy.

Overall, the simulations confirm that VAE-SC effectively generalises to different drift types and maintains high recovery rates remains above 80%, for significant drift magnitudes, reinforcing its potential as a versatile and reliable tool for sensor drift calibration in dynamic and uncertain environments.

E. Drift Detection & Soft Calibration

Our experimental findings indicate that when the drifted sensors are known, techniques such as SSP-KF and SVR-KF can achieve highly accurate drift estimation. However, in the absence of accurate drift detection, methods like the original SVR-KF yield inaccurate estimations, emphasizing the substantial impact of uncertainty in drift sensor detection on drift value estimation.

The proposed VAE-SC) method demonstrates the capability to consistently detect and estimate drifted sensors, maintaining high recovery rates and low Mean Squared Error (MSE) regardless of the increasing number of drifted sensors. As shown in our experiments, VAE-SC achieves an average recovery rate of approximately 91% after drift, outperforming other methods. Similarly, the Self-Calibration technique exhibits strong performance in both drift detection and estimation, with recovery rates around 87% and MSE after drift of approximately 0.13, indicating reliable performance.

Other methods, such as Dynamic Residual Projection and Blind Drift Deep Learning (Blind Drift-DL), also show effective drift detection capabilities, achieving recovery rates of approximately 84% after drift. However, they tend to have slightly higher MSE values compared to VAE-SC and Self-Calibration, suggesting less precise calibration. These methods perform adequately but may not match the consistency and accuracy of VAE-SC, especially as the number of drifted sensors increases.

An existing approach is the Compressed Sensing Kalman Filter (CSKF) proposed in [48], but it suffers from cumulative estimation errors caused by detection errors. Similarly, while SSP-KF-Oracle can achieve accurate drift estimation when the drifted sensors are known, its performance decreases when this information is unavailable or inaccurate.

One challenge in drift detection with the proposed VAE-SC algorithm is that drift can only be detected if its magnitude is sufficiently large relative to the observation noise. Although VAE-SC exhibits superior noise robustness compared to other methods, the recovery rate suffers at low Signal-to-Noise Ratio (SNR) or low drift-to-noise ratio in our case [49]. This limitation is deemed acceptable, considering that sensor drift as small as the noise level is tolerable in many applications.

Overall, the inclusion of Dynamic Residual Projection, Blind Drift-DL, and Self-Calibration in our comparative study highlights the strengths and weaknesses of different approaches in drift detection and calibration. While VAE-SC and Self-Calibration maintain high recovery rates and low MSEs, especially in scenarios with multiple drifted sensors, other methods like SVR-KF-Oracle and SSP-KF-Oracle struggle with accurate drift estimation due to the accumulation of prediction errors and their sensitivity to the accurate identification of drifted sensors.

Our analysis suggests that VAE-SC offers a robust solution for drift detection and soft calibration in sensor networks, outperforming traditional methods and recent approaches in both simulated and real-world datasets. Its ability to model complex, non-linear relationships in the sensor data through the probabilistic latent space contributes to its superior performance, making it a powerful tool for maintaining the reliability of sensor networks in dynamic and uncertain environments.

VI. Discussion, Limitation and future work

A. Real-World Implementation and Scalability Considerations

1) Deployment Architecture: In our proposed system, the data acquisition and processing align with standard IoT deployment practices. IoT devices equipped with sensors collect data continuously and transmit it to a central server over the network. This transmission is typically can be achieved using wireless communication protocols such as Wi-Fi, LoRaWAN, or cellular networks, depending on the application's requirements and infrastructure availability.

We utilised *ThingsBoard*, an open-source IoT platform deployed on a public cloud service, to facilitate data collection, processing, and visualisation. ThingsBoard acts as middleware that handles device connectivity, data ingestion, and routing of data streams to the appropriate processing modules. It provides a scalable and flexible infrastructure capable of managing numerous devices and high data volumes.

In our architecture, IoT devices send their data to ThingsBoard using lightweight protocols like Message Queuing Telemetry Transport (MQTT) or HyperText Transfer Protocol (HTTP). ThingsBoard then forwards the data to the central server where the VAE-based sensor drift calibration model resides. This setup allows for centralised processing, which is advantageous for computational efficiency and ease of model updates [50].

2) Computational Costs: The computationally intensive tasks, such as training and inference of the VAE model, are centralized on the server. This design choice offloads the computational burden from the IoT devices, which often have limited processing power and energy resources. By leveraging powerful server hardware and cloud computing capabilities, we can efficiently handle the computational demands of the VAE model.

During real-time operation, the VAE processes incoming data to detect and correct sensor drift. The inference process is relatively lightweight, involving forward passes through the neural network, which can be optimized for performance. Techniques such as model quantisation and batching can further enhance efficiency, enabling real-time or near-real-time processing of data streams.

Centralising computational tasks also simplifies the deployment and maintenance of the VAE model. Updates and improvements to the model can be implemented on the server without the need to modify software on the IoT devices, reducing operational overhead.

3) Scalability: Scalability is a critical consideration in IoT applications, where the number of devices and the volume of data can grow rapidly. Our system leverages a serverless cloud architecture to achieve dynamic scaling. Serverless platforms automatically manage the allocation of computing resources, scaling up or down based on the workload in real-time.

By utilising cloud services, we can handle large-scale data volumes and a high number of devices without significant performance degradation. The architecture supports horizontal scaling, where additional computing instances are provisioned to manage increased loads. This elasticity ensures that the system maintains responsiveness and processing efficiency as the IoT deployment expands.

Moreover, the use of ThingsBoard facilitates scalability by providing features such as load balancing, distributed processing, and device management. It allows for seamless integration of new devices and supports multi-tenancy, which is beneficial for applications involving multiple users or organisational units.

B. Limitations of the VAE-Based Approach

Utilising the correlation among sensory data and leveraging statistical features or prior knowledge can significantly enhance the detection rate of sensor drifts. Even small drifts become detectable thanks to the distinctions in statistical features between drift and noise. Once the identification of drifted sensors is accomplished, obtaining an accurate estimation of the drift value becomes considerably more straightforward.

Furthermore, it's important to note that our study did not delve into investigating the sensitivity of the sensors or analysing the impact of different weather conditions on sensor readings; these aspects are beyond the scope of our research. Instead, our primary focus was on recovering the true values of the sensors, particularly when they exhibited drift in their readings. While the VAE-based approach offers significant advantages for sensor drift calibration, it is important to acknowledge its limitations:

1) Data Requirements:
   Training a VAE effectively requires a substantial amount of high-quality calibration data that accurately represents the normal operating conditions of the sensors. In situations where such data is scarce or expensive to obtain, the performance of the VAE may degrade.
2) Model Sensitivity and Hyperparameter Tuning:
   The performance of VAEs is sensitive to the choice of hyperparameters, such as the size of the latent space, learning rate, and network architecture. Selecting optimal hyperparameters often requires extensive experimentation and domain expertise.
3) Assumption of Gaussian Noise:
   The VAE model commonly assumes that the noise in the data is Gaussian. If the sensor noise deviates significantly from a Gaussian distribution, the model's assumptions may not hold, potentially affecting calibration accuracy.
4) Black-Box Nature:
   Deep learning models like VAEs are often considered black boxes due to their complex architectures, making it difficult to interpret the learned representations and understand the specific features contributing to drift detection.
5) Potential for Overfitting:
   Without proper regularisation and validation, VAEs can overfit the training data, leading to poor generalisation on new sensor measurements, especially in environments that differ from the calibration conditions.

These limitations highlight the challenges and considerations necessary for the practical deployment of our VAE-based sensor drift calibration method.

C. Future Works

In future work, potential strategies to mitigate these limitations along with exploring the sensitivity of different sensors to environmental conditions could be a valuable area of study.

- Our model demonstrates the capability to process multiple data points simultaneously and derive a compensated output. By taking into account n number of data points and comparing them with the distribution, our model can effectively identify and mitigate errors, resulting in improved accuracy and robustness.

- Furthermore, our technique is adaptable to function with multiple data points, enabling it to handle complex datasets efficiently. This versatility allows for broader applications across various domains, where the processing of multiple data points simultaneously is essential for accurate analysis and decision-making.
- Moreover, our model is designed to be versatile and applicable across multiple environments or locations. The same model can be deployed in various settings, offering a scalable solution that minimises the need for model customisation or adaptation for different scenarios.
- In contrast, traditional methods often rely on processing one data point at a time, limiting their efficiency and scalability. By leveraging advanced techniques such as parallel processing and distribution-based comparison, our model surpasses the capabilities of traditional methods, offering superior performance and flexibility in data analysis and compensation.

VII. Conclusion

Our model demonstrates a unique capability to process multiple data points simultaneously, enabling it to derive compensated outputs by comparing them with the distribution. This feature not only enhances the accuracy and robustness of our technique but also allows for broader applications across various domains where processing multiple data points is essential. While our model excels in detecting and compensating for drift, further research is needed to improve its ability to capture recurring errors effectively. Nevertheless, our model's strength lies in its versatility and scalability, as it can be deployed across multiple environments or locations without the need for extensive customisation. Unlike traditional methods that process one data point at a time, our model leverages advanced techniques such as parallel processing and distribution-based comparison to offer superior performance and flexibility in sensor drift detection in varying settings and environmental conditions.

References

[1] R. Chengoden, N. Victor, T. Huynh-The, G. Yenduri, R. H. Jhaveri, M. Alazab, S. Bhattacharya, P. Hegde, P. K. R. Maddikunta, and T. R. Gadekallu, "Metaverse for healthcare: a survey on potential applications, challenges and future directions," *IEEE Access*, vol. 11, pp. 12 765–12 795, 2023.

[2] J. M. Barcelo-Ordinas, M. Doudou, J. Garcia-Vidal, and N. Badache, "Self-calibration methods for uncontrolled environments in sensor networks: A reference survey," *Ad Hoc Networks*, vol. 88, pp. 142–159, 2019.

[3] H. Qiao, B. Novikov, and J. O. Blech, "Concept drift analysis by dynamic residual projection for effectively detecting botnet cyber-attacks in iot scenarios," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 6, pp. 3692–3701, 2021.

[4] Y. Wang, A. Yang, Z. Li, P. Wang, and H. Yang, "Blind drift calibration of sensor networks using signal space projection and kalman filter," in *2015 IEEE tenth international conference on intelligent sensors, sensor networks and information processing (ISSNIP)*. IEEE, 2015, pp. 1–6.

[5] N. U. Okafor and D. T. Delaney, "Missing data imputation on iot sensor networks: Implications for on-site sensor calibration," *IEEE Sensors Journal*, vol. 21, no. 20, pp. 22 833–22 845, 2021.

[6] M. S. Takruri, S. Challa, and R. Chakravorty, "Recursive bayesian approaches for auto calibration in drift aware wireless sensor networks," *Journal of Networks*, 2010.

[7] B.-T. Lee, S.-C. Son, and K. Kang, "A blind calibration scheme exploiting mutual calibration relationships for a dense mobile sensor network," *IEEE Sensors Journal*, vol. 14, no. 5, pp. 1518–1526, 2014.

[8] J. Loy-Benitez, S. Heo, and C. Yoo, "Imputing missing indoor air quality data via variational convolutional autoencoders: Implications for ventilation management of subway metro systems," *Building and Environment*, vol. 182, p. 107135, 2020.

[9] C. C. Lin, D. J. Deng, C. H. Kuo, and L. Chen, "Concept drift detection and adaption in big imbalance industrial iot data using an ensemble learning method of offline classifiers," *IEEE Access*, vol. 7, pp. 56 198–56 207, 2019.

[10] 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 Journal*, vol. 16, no. 16, pp. 6249–6260, 2016.

[11] L. Yang and A. Shami, "A lightweight concept drift detection and adaptation framework for iot data streams," *IEEE Internet of Things Magazine*, vol. 4, no. 2, pp. 96–101, 2021.

[12] T. A. Ahanger and A. Aljumah, "Internet of things: A comprehensive study of security issues and defense mechanisms," *IEEE Access*, vol. 7, pp. 11 020–11 028, 2018.

[13] P. Fraga-Lamas, T. M. Fernández-Caramés, M. Suárez-Albela, L. Castedo, and M. González-López, "A review on internet of things for defense and public safety," *Sensors*, vol. 16, no. 10, p. 1644, 2016.

[14] S. Marathe, A. Nambi, M. Swaminathan, and R. Sutaria, "Currentsense: A novel approach for fault and drift detection in environmental iot sensors," in *Proceedings of the International Conference on Internet-of-Things Design and Implementation*, 2021, pp. 93–105.

[15] T. Chakraborty, A. U. Nambi, R. Chandra, R. Sharma, M. Swaminathan, Z. Kapetanovic, and J. Appavoo, "Fall-curve: A novel primitive for iot fault detection and isolation," in *Proceedings of the 16th ACM conference on embedded networked sensor systems*, 2018, pp. 95–107.

[16] K. Ni, N. Ramanathan, M. N. H. Chehade, L. Balzano, S. Nair, S. Zahedi, E. Kohler, G. Pottie, M. Hansen, and M. Srivastava, "Sensor network data fault types," *ACM Transactions on Sensor Networks (TOSN)*, vol. 5, no. 3, pp. 1–29, 2009.

[17] J. Gama, *Knowledge discovery from data streams*. CRC Press, 2010.

[18] X. Fang and I. Bate, "Issues of using wireless sensor network to monitor urban air quality," in *Proceedings of the First ACM International Workshop on the Engineering of Reliable, Robust, and Secure Embedded Wireless Sensing Systems*, 2017, pp. 32–39.

[19] Y. Cheng, X. Li, Z. Li, S. Jiang, Y. Li, J. Jia, and X. Jiang, "Aircloud: A cloud-based air-quality monitoring system for everyone," in *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, 2014, pp. 251–265.

[20] F. M. Bulot, S. J. Johnston, P. J. Basford, N. H. Easton, M. Apetroaie-Cristea, G. L. Foster, A. K. Morris, S. J. Cox, and M. Loxham, "Long-term field comparison of multiple low-cost particulate matter sensors in an outdoor urban environment," *Scientific reports*, vol. 9, no. 1, pp. 1–13, 2019.

[21] M. E. H. Benbouzid, "A review of induction motors signature analysis as a medium for faults detection," *IEEE transactions on industrial electronics*, vol. 47, no. 5, pp. 984–993, 2000.

[22] Y. Cheng, X. He, Z. Zhou, and L. Thiele, "Ict: In-field calibration transfer for air quality sensor deployments," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 1, pp. 1–19, 2019.

[23] L. J. Chen, Y. H. Ho, H.-H. Hsieh, S. T. Huang, H. C. Lee, and S. Mahajan, "Adf: An anomaly detection framework for large-scale pm2. 5 sensing systems," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 559–570, 2017.

[24] M. Mueller, J. Meyer, and C. Hueglin, "Design of an ozone and nitrogen dioxide sensor unit and its long-term operation within a sensor network in the city of zurich," *Atmospheric Measurement Techniques*, vol. 10, no. 10, pp. 3783–3799, 2017.

[25] L. Balzano and R. Nowak, "Blind calibration of sensor networks," in *Proceedings of the 6th international conference on Information processing in sensor networks*, 2007, pp. 79–88.

[26] B. Hoadley, "A bayesian look at inverse linear regression," *Journal of the American Statistical Association*, vol. 65, no. 329, pp. 356–369, 1970.

[27] M. Takruri, K. Aboura, and S. Challa, "Distributed recursive algorithm for auto calibration in drift aware wireless sensor networks," in *Innovations and advanced techniques in systems, computing sciences and software engineering*. Springer, 2008, pp. 21–25.

[28] M. Takruri, S. Challa, and R. Chakravorty, "Auto calibration in drift aware wireless sensor networks using the interacting multiple model algorithm," in *2008 Mosharaka International Conference on Communications, Computers and Applications*. IEEE, 2008, pp. 98–103.

[29] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM computing surveys (CSUR)*, vol. 46, no. 4, pp. 1–37, 2014.
[30] H. Yu, T. Liu, J. Lu, and G. Zhang, "Automatic learning to detect concept drift," *arXiv preprint arXiv:2105.01419*, 2021.
[31] S. O. Buadhacháin and G. Provan, "A model-based control method for decentralized calibration of wireless sensor networks," in *2013 American Control Conference*. IEEE, 2013, pp. 6571–6576.
[32] J. Feng, S. Megerian, and M. Potkonjak, "Model-based calibration for sensor networks," in *SENSORS, 2003 IEEE*, vol. 2. IEEE, 2003, pp. 737–742.
[33] A. Asperti and M. Trentin, "Balancing reconstruction error and kullback-leibler divergence in variational autoencoders," *Ieee Access*, vol. 8, pp. 199 440–199 448, 2020.
[34] M. J. Kusner, B. Paige, and J. M. Hernández-Lobato, "Grammar variational autoencoder," in *International conference on machine learning*. PMLR, 2017, pp. 1945–1954.
[35] R. Xie, N. M. Jan, K. Hao, L. Chen, and B. Huang, "Supervised variational autoencoders for soft sensor modeling with missing data," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2820–2828, 2019.
[36] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special lecture on IE*, vol. 2, no. 1, pp. 1–18, 2015.
[37] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.
[38] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU press, 2013.
[39] 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 Journal*, vol. 17, no. 13, pp. 4158–4171, 2017.
[40] R. Ahmad, B. Rinner, R. Wazirali, S. K. Abujayyab, and R. Almajalid, "Two-level sensor self-calibration based on interpolation and autoregression for low-cost wireless sensor networks," *IEEE Sensors Journal*, 2023.
[41] M. Xie, J. Hu, S. Guo, and A. Y. Zomaya, "Distributed segment-based anomaly detection with kullback–leibler divergence in wireless sensor networks," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 101–110, 2016.
[42] T. Van Erven and P. Harremos, "Rényi divergence and kullback-leibler divergence," *IEEE Transactions on Information Theory*, vol. 60, no. 7, pp. 3797–3820, 2014.
[43] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright *et al.*, "Scipy 1.0: fundamental algorithms for scientific computing in python," *Nature methods*, vol. 17, no. 3, pp. 261–272, 2020.
[44] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.
[45] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.
[46] A. Nano, "Arduino nano," *A MOBICON Company*, 2018.
[47] ThingsBoard, "ThingsBoard: Open-source IoT Platform," https://github.com/thingsboard/thingsboard, accessed on December 6, 2024.
[48] N. Vaswani, "Kalman filtered compressed sensing," in *2008 15th IEEE International Conference on Image Processing*. IEEE, 2008, pp. 893–896.
[49] Z. Zhang and B. D. Rao, "Sparse signal recovery with temporally correlated source vectors using sparse bayesian learning," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 5, pp. 912–926, 2011.
[50] J. de Carvalho Silva, J. J. Rodrigues, A. M. Alberti, P. Solic, and A. L. Aquino, "Lorawan—a low power wan protocol for internet of things: A review and opportunities," in *2017 2nd International multidisciplinary conference on computer and energy science (SpliTech)*. IEEE, 2017, pp. 1–6.

Kamal Hossain Kamal Hossain is currently working as a Lead Data and AI Consultant in Australia while pursuing a Ph.D. at Edith Cowan University. Prior to his current role, he held multiple positions across various industries and served as a Machine Learning Researcher at Telekom Malaysia Research & Development (TM R&D). In that capacity, Kamal contributed to advanced machine learning projects and collaborated with academic institutions, bridging the gap between industry and academia. He received his Bachelor of Electronic Engineering degree in 2012 and his Master of Engineering Science degree in 2015 from Multimedia University in Cyberjaya, Malaysia. His research interests include IoT sensors, sensor drift, generative Bayesian statistics and cybersecurity for IoT devices.

Iftekhar Ahmad Iftekhar Ahmad is currently working as an Associate Professor with the School of Engineering, Edith Cowan University, Australia. He received his PhD from Monash University, Australia, in 2007. His research interests include green communications, integration of renewable energy, electric vehicle charging/discharging.

Daryoush Habibi Daryoush Habibi graduated with a Bachelor of Engineering (Electrical) with First Class Honours from the University of Tasmania in 1989 and a PhD from the same University in 1994. His employment history includes Telstra Research Laboratories, Flinders University, Intelligent Pixels Inc., and Edith Cowan University, where he is currently a Professor and the Head of the Centre for Green and Smart Energy Systems. His research interests include engineering design for sustainable development, renewable and smart energy systems, environmental monitoring technologies, and reliability and quality of service in engineering systems and networks. He has over 200 refereed publications in high-impact journals, conference proceedings and book chapters. He is a Fellow of Engineers Australia, a Fellow of the Institute of Marine Engineering, Science and Technology, and a Senior Member of the IEEE.

Muhammad Waqas Muhammad Waqas pursued his PhD degree with the Department of Electronic Engineering at Tsinghua University, Beijing, China, in 2019. From October 2019 to March 2022, he was a Research Associate at the Faculty of Information Technology, Beijing University of Technology, Beijing, China and also affiliated with GIK Institute of Engineering Sciences and Technology, Pakistan. From April 2022, he had been an Assistant Professor at the Computer Engineering Department, College of Information Technology, University of Bahrain, Bahrain. In the UK, he is currently a Senior Lecturer (Assistant Professor) at the School of Computing and Mathematical Sciences, Faculty of Engineering and Science, University of Greenwich, London, UK. He has also been an Adjunct Senior Lecturer (Assistant Professor) at the School of Engineering, Edith Cowan University, Australia, since November 2021.

Dr Waqas has more than 100 research publications in reputed Journals and Conferences with more than 3200 citations, an h-index of 25 and an i10 index of 63. He is an Associate Editor and Guest editors of several reputed journals and achieved more than 7 funded projects. He was also invited as a distinguished/invited speaker at several reputed conferences.

His current research interests are in the areas of Wireless Communication, vehicular networks, cybersecurity and Machine Learning. He is recognised as a Global Talent in the area of Wireless Communications by UK Research and Innovation and a Professional Member of Engineer Australia. He is a senior member of IEEE, a Professional Member of ACM, an IEEE Young Professional, a Member of the Pakistan Engineering Council and PhD approved supervisor by the Higher Education Commission of Pakistan.