sensors

MDPI

Article

An Edge Transfer Learning Approach for Calibrating Soil Electrical Conductivity Sensors

Yun-Wei Lin [1,*], Yi-Bing Lin [1,2,3,4,5,6,*], Ted C.-Y. Chang [7] and Bo-Xun Lu [1]

[1] College of Artificial Intelligence, National Yang Ming Chiao Tung University, Tainan 711, Taiwan; boxunlu.ai09@nycu.edu.tw
[2] College of Humanities and Sciences, China Medical University, Taichung 406, Taiwan
[3] Miin Wu School of Computing, National Cheng Kung University, Tainan 701, Taiwan
[4] College of Computer Science, National Yang Ming Chiao Tung University, Hsinchu 300, Taiwan
[5] Department of Computer Science and Information Engineering, Asia University, Taichung 413, Taiwan
[6] Research Center for Information Technology Innovation, Academia Sinica, Taipei 115, Taiwan
[7] Quanta Computer Co., Ltd., Taoyuan 333, Taiwan; ted.chang@quantatw.com
* Correspondence: jyneda@nycu.edu.tw (Y.-W.L.); liny@nctu.edu.tw (Y.-B.L.)

Abstract: Smart agriculture utilizes Internet of Things (IoT) technologies to enable low-cost electrical conductivity (EC) sensors to support farming intelligence. Due to aging and changes in weather and soil conditions, EC sensors are prone to long-term drift over years of operation. Therefore, regular recalibration is necessary to ensure data accuracy. In most existing solutions, an EC sensor is calibrated by using the standard sensor to build the calibration table. This paper proposes SensorTalk3, an ensemble approach of machine learning models including XGBOOST and Random Forest, which can be executed at an edge device (e.g., Raspberry Pi) without GPU acceleration. Our study indicates that the soil information (both temperature and moisture sensor data) plays an important role in SensorTalk3, which significantly outperforms the existing calibration approaches. The MAPE of SensorTalk3 can be as low as 1.738%, compared to the 7.792% error of the original sensor. Our study indicates that when the errors of uncalibrated moisture and temperature sensors are not larger than 8.3%, SensorTalk3 can accurately calibrate EC. SensorTalk3 can perform model training during data collection at the edge node. When all training data are collected, AI training is also finished at the edge node. Such an AI training approach has not been found in existing edge AI approaches. We also proposed the dual-sensor detection solution to determine when to conduct recalibration. The overhead of this solution is less than twice the optimal detection scenario (which cannot be achieved practically). If the two non-standard sensors are homogeneous and stable, then the optimal detection scenario can be approached. Conventional methods require training calibration AI models in the cloud. However, SensorTalk3 introduces a significant advancement by enabling on-site transfer learning in the edge node. Given the abundance of farming sensors deployed in the fields, performing local transfer learning using low-cost edge nodes proves to be a more cost-effective solution for farmers.

Keywords: artificial intelligence; electrical conductivity; farming sensors; Internet of Things (IoT); Random Forest; sensor calibration; XGBOOST

check for updates

Citation: Lin, Y.-W.; Lin, Y.-B.; Chang, T.C.-Y.; Lu, B.-X. An Edge Transfer Learning Approach for Calibrating Soil Electrical Conductivity Sensors. *Sensors* **2023**, *23*, 8710. https://doi.org/10.3390/s23218710

Academic Editor: Carlos M. Travieso-González

Received: 27 September 2023
Revised: 18 October 2023
Accepted: 23 October 2023
Published: 25 October 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The ability of soil to conduct electrical current indicates drainage, the amount of nitrogen usage, rooting depth, water holding and cation-exchange capacity in a farm field. Such ability is typically measured through electrical conductivity (EC) [1,2]. In smart agriculture [3], we utilize Internet of Things (IoT) technologies to enable low-cost EC sensors to support farming intelligence [4,5]. Figure 1a illustrates a three-in-one sensor for the temperature, the humidity and the EC. Several three-in-one sensors are deployed in a

strawberry greenhouse, as shown in Figure 1b. In this greenhouse, the fertilizer drippers are controlled through real-time AI prediction with inputs from the farming sensors.



(**a**) (**b**)

Figure 1. (**a**) A 3-in-one sensor and (**b**) the Bao strawberry greenhouse.

Because of aging and changing weather and soil conditions, EC sensors are prone to long-term drift over years of operation. Therefore, they must be recalibrated on a regular basis to guarantee data accuracy.

Most smart farming studies do not consider in-field EC sensor calibration; instead, they focus on offline (laboratory) calibration, which is expensive and interrupts smart farming operations. In our previous studies [6–8], we addressed this issue by employing a table lookup method.

Specifically, we have developed a series of techniques for in-field EC sensor calibration called SensorTalk generations 1–3. The generations of SensorTalk were transitioned in chronological order. In the first SensorTalk generation [6,7], we designed a failure detection and calibration framework. The SensorTalk framework automatically detects the failures of many farming sensors and actuators through mutual tests among them. SensorTalk also uses a lookup table mechanism to calibrate the aging sensors. Following the cloud computing paradigm, computing tasks are offloaded from the IoT devices to the SensorTalk server in the cloud to reduce energy consumption and improve the performance of the farming IoT devices. To be more specific, the first two generations of SensorTalk use a cloud-based dashboard to detect sensor/actuator failures and calibrate the aged sensors semi-automatically.

The second generation (SensorTalk2) [8] is an extension of SensorTalk in redundancy. In this approach, nine EC sensors installed in the Murata redundant-sensor package [9] are used in a voting procedure to generate the correct EC values. When a sensor in the package is aged, calibration is automatically conducted locally to fix the aged sensor through the use of other correct sensors through the same lookup table approach of SensorTalk. SensorTalk2 is an edge computing solution that performs calibration through local redundant sensors in the Murata package [10–13].

After optimization, the accuracy of the lookup table is limited to 96%. To address this main weakness and research gap, we propose SensorTalk3 to replace the lookup table mechanism in SensorTalk with AI mechanisms. In-field experiments indicate that SensorTalk3 significantly improves the accuracy of SensorTalk, achieving rates higher than 98%. Previous approaches need to train the calibration AI models in the cloud. A major contribution of SensorTalk3 is the ability to perform transfer learning in the edge node. Since there are many farming sensors in the farm field, it is more cost-effective to perform transfer learning locally with inexpensive edge nodes for the farmers. Moreover, the process of calibrating a sensor and achieving accurate values is further delayed when employing transfer learning in the cloud. This delay results in the sensor producing inaccurate readings during the calibration period. Thankfully, by leveraging SensortTalk3's transfer learning capabilities in the edge node, we can effectively address this problem.

The paper is organized as follows. Section 2 surveys the related work; Section 3 elaborates on the SensorTalk3 approach; Section 4 describes the AI models; Section 5 evaluates the prediction performance of SensorTalk3.

2. Related Studies

This section surveys the previous studies on EC sensors and sensor calibration.

2.1. EC Sensors

This subsection overviews the related work for EC sensor designs. An EC sensor is typically implemented with four electrodes constructed from conductive materials such as graphite, stainless steel, or platinum (Pt). In this design, two outer electrodes are built apart to pass a current produced by conductive ions (such as metals and salts in soil). The circuit uses a sinusoidal AC voltage source generated based on the theory of the Wien bridge oscillator. A low oscillation frequency (1 Hz) is selected so that the impedance depends mainly on resistance and less on capacitance or inductance. The circuit eliminates any DC offset to avoid electrode polarization. Two inner electrodes are constructed to measure the resulting current. The conductive ions between the outer electrodes create a current path, where the measured conductivity implies the ionic concentration of the soil. In this paper, conductivity is measured in Deci Siemens (dS). When using a contacting conductivity sensor, conductivity cell geometry affects the conductivity reading. In order to ensure the standardization of EC measurements, units of specific conductivity are used. Equation (1) gives the equation for specific conductivity σ_b measured in dS per meter (dS/m):

$$\sigma_b = \sigma_m \kappa = \frac{\sigma_m L}{A} \tag{1}$$

where A is the surface area of an outer electrode and L is the distance between the outer electrodes. The cell constant κ (i.e., the volume between the two electrodes) is computed as L/A. The measured conductivity σ_m (the measured current divided by the applied voltage) is multiplied by the cell constant to determine the specific bulk conductivity σ_b of the soil, i.e., the EC of the undisturbed matrix of soil, water and air.

An example of EC sensor package is the Murata Model LT5006. The Murata redundant sensor package also includes the sensors for moisture (Volumetric Water Content or VWC) and temperature, which can be used in both the soil and water at same time.

The σ_b range of the Murata EC sensor is 0–5 dS/m and the resolution is 0.001 dS/m. To protect the electrodes of the EC sensor from corrosion, it is essential to use low voltage and highly corrosive-resistant materials.

It is also important to know the soil pore water EC (σ_w; the EC of the water in the pore spaces of the soil) as an indicator of the solute concentration. The σ_w value is calculated using some parameters measured using the Murata moisture sensor. To achieve high accuracy, sensor calibration is compensated by considering the temperature dependence. The relationship between σ_w and σ_b is given in Equations (2) and (3).

$$\sigma_w = \frac{\epsilon_w \sigma_b}{\epsilon_b - \epsilon_b^*} \tag{2}$$

where ϵ_b is the bulk soil dielectric permittivity that can be measured through the application of Equation (1). In Equation (2), ϵ_b^* is an offset, which is the real portion of the dielectric permittivity when $\sigma_b = 0$. In [14], $\epsilon_b^* = 4.1$ is recommended as a generic offset. In Equation (2), ϵ_w is expressed as

$$\epsilon_w = 80.3 - 0.37(T_s - 20) \tag{3}$$

where T_s is the soil temperature (°C) measured by the soil sensor co-located with the bulk EC measurement in the Murata package.

There are several state-of-the-art IC designs for EC sensors. Based on Equation (2), the study in [15] fabricated a redundant sensor package for measuring pH, EC and temperature using compatible CMOS technology on a Si chip where the chip size is 5 mm × 5 mm. An EC sensing area using Pt electrodes is deposited on the Ti/Al electrode. The fabricated redundant sensor package showed good performance in a 6-day experiment. The study in [16] presents a self-sustained soil EC sensing IC using the concept of Equation (2). With a wide-range DC–DC converter, the system simultaneously measures the conductivity of the soil and harvests energy. This IC was fabricated with the 0.18µm CMOS technology. Using the carbon and zinc electrodes, the input voltage of the soil cell ranges from 0.8 to 1 V. The soil EC is measured from 0.114 to 0.744 dS/m in a resolution of 0.00568 dS/m per bit.

2.2. Sensor Calibration

Based on the lookup table, SensorTalk and SensorTalk2 use standard sensors to produce a calibration table for an aged sensor. Details of lookup table calibration can be found in [6–8] and the references therein.

In [17], the authors surveyed the machine learning techniques for the calibration of air quality monitoring sensors. They identify several open research challenges. Specifically, low-cost air quality sensors suffer from cross-sensitivities between different ambient pollutants, and their accuracy degrades over time. They are also affected by external factors, including weather changes, traffic, and human behavior. This survey indicated that periodic in-field recalibration using machine learning is promising to recover the accuracy of the sensors. SensorTalk3′s experience on EC sensor calibration has shown consistent conclusions.

In [4], in situ blind calibration for sensor network monitoring was proposed, requiring neither physical intervention nor identical ground-truth signals. A multioutput Gaussian process (MOGP) was used to model the spatial-temporal distribution of the measure and drift to remove irrelevant short-term fluctuations. MOGP also decomposes the drift from long-term trends. The algorithm was evaluated on a real-world dataset of strain sensors. This study did not take advantage of the knowledge of other types of sensors as SensorTalk3 does.

Based on extreme learning machine and projection onto convex sets, a method [18] was proposed to calibrate electronic nose drift under long-term working conditions for both recognition and regression applications specific to isopropanol and acetone gases. In [19], the authors proposed automatically calibrating a large number of barometer sensors, which uses a low-power barometer on a smartphone without requiring reference points or any manual operation. However, this work requires user encounter detection to conduct peer-to-peer calibration, which calibrates all barometers by solving a minimum dominating set problem. In a field experiment, this approach yields an accuracy of within 0.1 hpa in 82% of cases. User encounter detection cannot be guaranteed in general scenarios.

The accuracy of a gas sensor may be affected by other gases. Such interference is called cross-sensitivity. Cross-sensitivity can cause an unwanted effect on the sensor, which may be a positive response, negative response or inhibition. In [20], the authors designed a fast informed (Semi-) non-negative matrix factorization method to solve in situ calibration of cross-sensitive sensors. In these gas sensors, the readings of a sensor depend on the readings of other sensors. The approach was studied under simulation. It is not clear if the approach works in real environments.

In [21], the authors investigated metal oxide (MOx) sensors for detecting indoor Benzene, Toluene, Ethylbenzene, and Xylene gases. Since temperature and humidity can easily affect the MOx sensors, calibration is conducted through machine learning, including artificial neural network, non-linear curve fitting and linear regression. This approach was validated using three operating points of temperature/humidity in the laboratory. The experiments indicated that the proposed approach reduces 73% of the temperature and humidity impacts on the reading variation of the MOx sensors. Unlike the study in [21], SensorTalk3 does not eliminate the effects of temperature and humidity. Instead, we utilize the humidity (moisture) and temperature to improve the accuracy of soil EC sensors in

farming conditions without manually setting the operation points of temperature/humidity in the laboratory.

By using a metal oxide semiconductor gas sensor as an example, the study in [22] showed how to calibrate the readings in temperature cycled operation, which measure mixtures of artificial room air containing several volatile organic compounds and quantifying formaldehyde. The study attempted to minimize the calibration time through preprocessing of the training data. Through steady-state detection, the labeled valid data points are added to the dataset as compared to a time-consuming manual annotation. By reducing 50% of the original data, the preprocessed data can still train ResNet neural networks to produce errors 25% smaller than the errors defined by the WHO.

The study in [23] utilized deep learning to investigate the effects of weather in both drifting and sensor measurements. A procedure was designed to generate simulated emission and dispersion of PM 2.5 and PM 10. The study in [24] proposed a sensor calibration method for PM2.5, which uses a domain adaptation technique to reduce the calibration time. Temperature, PM10, and humidity are used as the features of the deep learning models for predicting the PM2.5 values. The results show that both proposed models in [23,24] reduce the calibration error. However, they were not validated against the real data.

Most of the above studies attempt to calibrate the gas sensors by eliminating the cross-sensitivity effects caused by humidity and temperature. Unlike these studies, SensorTalk3 takes advantage of humidity and temperature to improve the calibration accuracy of the EC sensors.

The study in [25] developed a multivariable model, employing radial basis function artificial neural network to estimate soil EC based on various factors. Laboratory tests yielded a high R2 of 0.99 and RMSE of 0.005 dS. This approach requires a special hardware layout for sensors, and the implementation cost may be high for commercial usage.

3. The SensorTalk3 Approach

In this section, we first describe the SensorTalk3 architecture and then show how the datasets are collected in this study.

3.1. The SensorTalk3 Architecture

Figure 2 illustrates the SensorTalk3 architecture. The SensorTalk server consists of three components. The IoTtalk engine (Figure 2 (1)) [26] is responsible for interaction with the IoT devices. The DataTalk module (Figure 2 (2)) is responsible for data preprocessing (feature extraction). The AItalk module (Figure 2 (3)) is responsible for AI prediction. An IoT device interacts with the IoTtalk Engine through the Device Application (DA). It is interesting to note that from the viewpoint of the IoTtalk Engine, both DataTalk and AItalk are managed as IoT devices.

The SensorTalk3 server can be deployed in the cloud or installed in an edge node. In the current implementation, the cloud-based SensorTalk3 is deployed in a virtual machine in a commercial cloud at Chunghwa Telecom, the largest Telecom company in Taiwan. The edge-based SensorTalk3 is deployed in a Raspberry Pi4, described in Appendix A.

The EC device under test (DUT EC1 in Figure 2 (4)) is calibrated using the standard EC sensor (STD EC1 in Figure 2 (5)), the standard temperature and moisture sensors (STD Sensors in Figure 2 (6)). In the training phase, the data from devices (4), (5) and (6) are sent to DataTalk through the IoTtalk Engine. After data preprocessing, the extracted features are sent to AItalk through the IoTtalk Engine. Note that the data received by DataTalk may also be sent to the AgriTalk Database (Figure 2 (7)) for archival purposes. Details of data preprocessing and AI modeling are given in the next section.

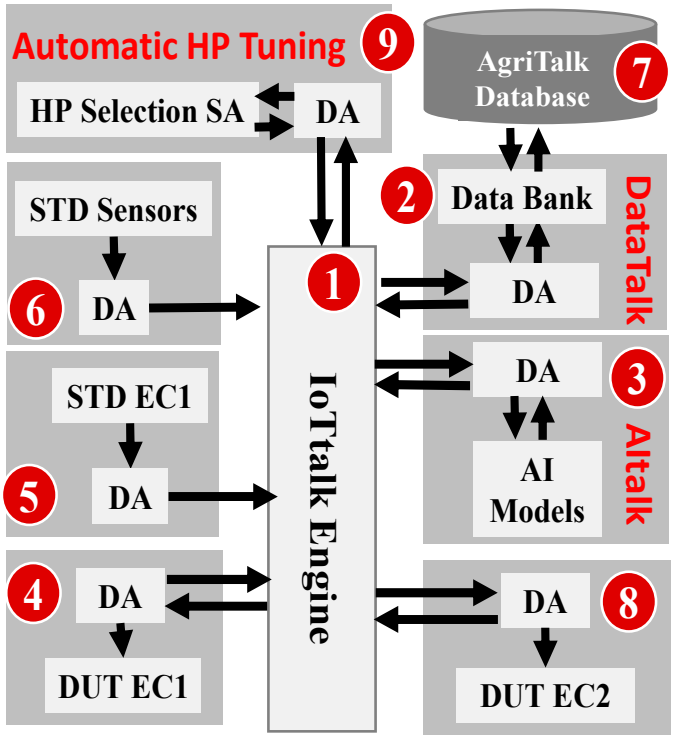


Figure 2. The SensorTalk3 Architecture.

3.2. The Datasets and Data Preprocessing

We have collected the greenhouse data. For every 20 s, we obtain a sample from each of the four sensors to generate a data item. The ith data item is a quadruplet $\langle e_i, E_i, T_i, M_i \rangle$, where $\{e_i\}$ are obtained from the DUT EC (Figure 2 (4)), $\{E_i\}$ are obtained from the STD EC sensor, $\{T_i\}$ are obtained from the STD temperature sensor, and $\{M_i\}$ are obtained from the STD moisture sensor (Figure 2 (5) and (6)). The set $\{E_i\}$ serves as the labels in the AI models. DataTalk (Figure 2 (2)) computed the mean absolute percentage error (MAPE) between the DUT EC and the STD EC and organized these collected data into 4 datasets:

Dataset 1 collected 20,000 data items during 1 October 2022–17 January 2022, where the EC ranges from 138 to 561 (µS/cm), the temperature ranges from 23.4 to 25.6 (°C), and the moisture ranges from 18.0 to 32.5 (%). The MAPE between the DUT EC and the STD EC is 7.624%.

Dataset 2 collected 11,372 data items during 22 January 2022–26 January 2022, where the EC ranges from 133 to 557 (µS/cm), the temperature ranges from 23.3 to 25.2 (°C), and the moisture ranges from 16.8 to 28.4 (%). The MAPE between the DUT EC and the STD EC is 5.716%.

Dataset 3 collected 44,076 data items during 27 January 2022–2 February 2022, where the EC ranges from 138 to 561 (µS/cm), the temperature ranges from 22.6 to 26.2 (°C), and the moisture ranges from 18.8 to 32.3 (%). The MAPE between the DUT EC and the STD EC is 11.158%.

Dataset 4 collected 36,096 data items during 24 February 2022–3 February 2022, where the EC ranges from 143 to 552 (µS/cm), the temperature ranges from 20.7 to 26.0 (°C), and the moisture ranges from 18.5 to 32.3 (%). The MAPE between the DUT EC and the STD EC is 7.794%.

In these datasets, the MAPEs of the EC without calibration (which is referred to as the "Original" method) range from 5.716% to 11.158%, and therefore, calibration is essential to enhance the accuracy of the EC readings.

We suspect that an EC value is not only affected by the temperature and the moisture but also their change rates. Therefore, DataTalk conducts data preprocessing to create three new features for our AI models. Let X_i represents the ith sample of sensor X, where

$X = e, E, T$, or M. Suppose that there are N data items in a dataset, where $N \geq i > w \geq 0$. Let $\delta_{X,i,w}$ denote the change rate of the ith sample of sensor X in the window w. Then, we have

$$\delta_{X,i,w} = \frac{|X_i - X_{i-w}|}{X_i} \text{ and } \delta_{X,w} = \frac{\sum_{i=1}^{N} \delta_{X,i,w}}{N} \quad (4)$$

Figure 3 shows that $\delta_{X,i,w}$ is amplified by w for both moisture and temperature. For EC, $\delta_{X,i,w}$ oscillates with w, which means that we need to choose a small w to reflect recent change trend. In Section 4, we show that $w = 2$ yields the best performance. Therefore, $\delta_{X,i,2}$ are included as the input features in our AI models.

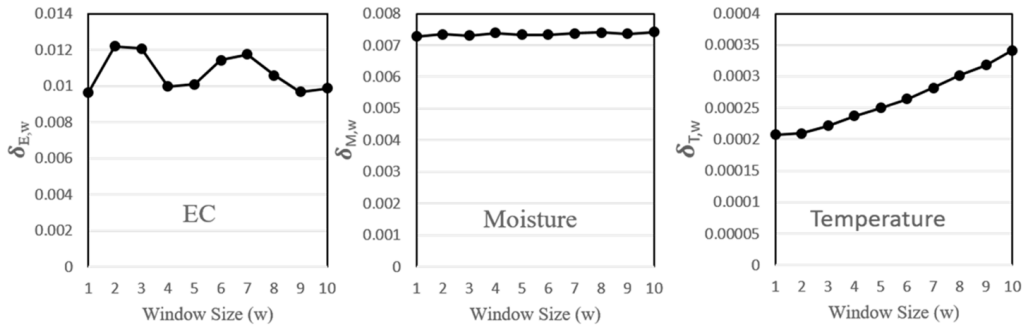


Figure 3. $\delta_{X,w}$ against w, where $X = (E, M, T)$.

4. The AI Models

As we described in Section 2, the EC prediction is affected by temperature, moisture and their change rates. This problem is more appropriately solved via classification-based machine learning models. We consider the following models: The SensorTalk2 model is based on the lookup table [6], XGBOOST [27,28] and Random Forest [29,30]. After collecting and analyzing the data, specifically examining the relationship between each data point of the DUT value and STD value, it becomes apparent that a certain degree of linear correlation exists. Initially, we attempted the conventional approach of using linear regression to calibrate the DUT with Dataset 3, resulting in an unsatisfactory MAPE value of 4.8723%. Upon further inspection of the dataset, it was observed that most data points exhibit a linear correlation, while a few do not show a significant linear relationship. Given the versatility of Random Forest for processing various types of features with linear relationships within the same dataset [29] and the capability of XGBoost in handling numerical features and non-linear relationships between features [28], we decided to employ both algorithms. The experimental results indicate that, in most cases, the MAPEs for XGBoost are generally smaller than those for Random Forest. However, it was noted that Random Forest outperforms XGBoost in specific instances, especially when the dataset comprises mostly linear data along with a few non-linear data points. Considering that Random Forest is adept at processing various types of features with linear relationships within the same dataset, and XGBoost is well-suited for handling numerical features and non-linear relationships between features, we have opted for these two algorithms. Based on these models, we create an ensemble model that utilizes the linear regression method to integrate XGBOOST and Random Forest, which take advantage of the individual machine-learning models to improve the prediction accuracy. The advantage of the classification model is its low time and space complexities in execution. Following the time series of the datasets, we use Dataset 1 for training, Dataset 2 for validation, and Datasets 3 and 4 for inferencing. The input features in our AI models are $\{e_i, T_i, M_i, \delta_{e,i,2}, \delta_{T,i,2}, \delta_{M,i,2}\}$, and the labels are $\{E_i\}$. The output measure of SensorTalk3 is MAPE expressed as

$$\text{MAPE} = \left(\frac{1}{N}\right) \sum_{i=1}^{N} \frac{|e_i - E_i|}{E_i} \quad (5)$$

We first select the loss function. The candidates include MSE (Mean Square Error) and RMSLE (Root Mean Squared Logarithmic error) expressed as

$$\text{MSE} = \left(\frac{1}{N}\right)\sum_{i=1}^{N}(e_i - E_i)^2 \tag{6}$$

and

$$\text{RMSLE} = \left(\frac{1}{N}\right)\sum_{i=1}^{N}(\log(e_i + 1) - \log(E_i + 1))^2 \tag{7}$$

Table 1 illustrates the MAPEs of XGBOOST and Random Forest using MSE and RMSLE as loss functions using the default hyperparameters. Since the outliers seldom occur in our sensors, the MSE loss function outperforms the RMSLE function, as indicated in Table 1. We chose MSE as the loss function in our AI Model.

Table 1. Loss function selection.

Loss Function	MAPE	
	XGBOOST	Random Forest
MSE	1.714%	2.981%
RMSLE	54.701%	3.180%

Then, we tune the hyperparameters for AItalk. The set of XGBOOST hyperparameters is expressed as Σ_{XGB} = {w, n_estimators, max_depth, learning_rate, gamma}. The impact of hyperparameter values on the performance of XGBoost is explained as follows. Increasing "n_estimators" can enhance the model's complexity and fitting capability, but it also results in higher computational costs and risk of overfitting. "max_depth" defines the maximum depth of each decision tree in a gradient boosting tree. Deeper trees can better capture complex relationships within the training data but are prone to overfitting, especially with limited data. Shallower trees restrict the model's complexity, helping to prevent overfitting, but may miss some important patterns within the data. The "learning_rate" is used to control the adjustment step size of model weights in each iteration. A lower learning rate makes the model learn more finely, ensuring it does not miss the optimal solution but may require more iterations to achieve peak performance. On the other hand, a larger learning rate speeds up model learning but increases the risk of missing the best solution. The "gamma" value impacts the model's complexity and helps prevent overfitting. A larger "gamma" value leads to more conservative splitting decisions, simplifying the tree and aiding in preventing overfitting. Smaller "gamma" values allow for more splits, increasing the tree's complexity, which helps the model fit the training data better but also raises the risk of overfitting [28].

The Random Forest hyperparameter set is expressed as Σ_{RF} = {w, n_estimators, max_depth, min_samples_split, max_sample}. The impacts of the "n_estimators" and "max_depth" values are similar to those in XGBoost. Additionally, a larger "min_samples_split" value results in a simplified tree, making it more effective in preventing overfitting but potentially at the cost of model flexibility. Conversely, smaller "min_samples_split" values enable more splits, offering better fitting capacity but increasing the risk of overly complex trees, which are more susceptible to overfitting. The "max_samples" hyperparameter in Random Forest controls the fraction of the original dataset assigned to each tree with an optimal performance fraction. This hyperparameter is instrumental in managing model variance and mitigating overfitting issues [29].

Let the default value for a hyperparameter σ be σ_d and the optimal value be σ_o, where $\sigma_{min} \leq \sigma, \sigma_d, \sigma_o \leq \sigma_{max}$. The hyperparameter selection (HP Selection) is designed in the following iterative procedure and is implemented as a SA of The automatic HP Tuning device in Figure 2 (9). Let the set of the default hyperparameter values in the AI model X

be $\Sigma_{X,d} = \{\sigma_d | \forall \sigma \in \Sigma_X\}$, where X = XGBOOST or Random Forest. The SA pseudo code is listed below:

The HP Selection SA

Line 1. $I = 0$;
Line 2. for (every hyperparameter $\sigma \in \Sigma_X$) do {
Line 3. $\Sigma_0 \leftarrow \Sigma_{X,d}$; $\Sigma_1 \leftarrow \varnothing$; $\sigma_{o,0} \leftarrow \sigma_{min}$;
Line 4. for ($\sigma_{min} \leq \sigma_0 \leq \sigma_{max}$) do {
Line 5. Execute the AI model X with the hyperparameter value set $\{\sigma_0\} \cup \Sigma_0 - \{\sigma_d\}$;
Line 6. If (the prediction result with σ_0 is better than the prediction result with $\sigma_{o,0}$) then
Line 7. $\sigma_{o,0} \leftarrow \sigma_0$;
}
Line 8. $\Sigma_1 \leftarrow \Sigma_1 \cup \{\sigma_{o,0}\}$; $I \leftarrow I + 1$;
}
Line 9. while ($\Sigma_I \neq \Sigma_{I-1}$) do {
Line 10. for (every hyperparameter $\sigma \in \Sigma_X$) do {
Line 11. $\Sigma_{I+1} \leftarrow \varnothing$; $\sigma_{o,I} \leftarrow \sigma_{min}$;
Line 12. for ($\sigma_{min} \leq \sigma_I \leq \sigma_{max}$) do {
Line 13. Execute AI model X with the hyperparameter value set $\{\sigma_I\} \cup \Sigma_I - \{\sigma_{o,I-1}\}$;
Line 14. If (the prediction result with σ_I is better than the prediction result with $\sigma_{o,I}$) then
Line 15. $\sigma_{o,I} \leftarrow \sigma_I$;
}
}
Line 16. $\Sigma_{I+1} \leftarrow \Sigma_{I+1} \cup \{\sigma_{o,I}\}$; $I \leftarrow I + 1$; }

In Part 1 (Lines 1–8), for every hyperparameter σ, we execute the AI model X by varying the σ values in the range $[\sigma_{min}, \sigma_{max}]$. For $\sigma^* \in \Sigma - \{\sigma\}$, σ_d^* is used in model X (Line 5). The σ_d values for XGBOOST are 2 for w, 100 for n_estimators, 6 for max_depth, 0.3 for learning_rate, and 0 for gamma. The σ_d values for Random Forest are 2 for w, 100 for n_estimators, None for max_depth, 2 for min_samples_split, and 1.0 for max_sample. If max_depth = None, then the nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples. Figure 4 shows the MAPE against w. This Figure indicates that the default value $w = 2$ turns out to be the optimal value.

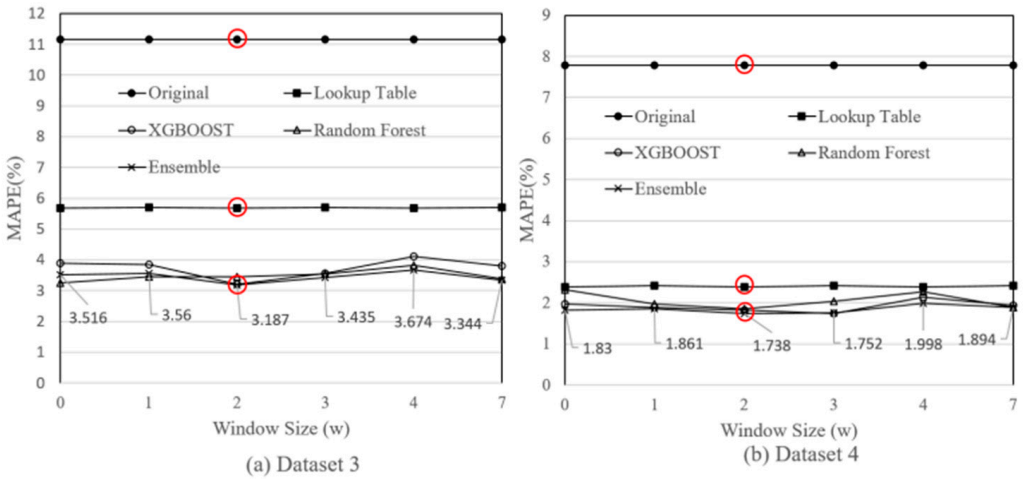


Figure 4. Selection of the change rate window size. (Red circles indicate the optimal value).

The red square curves in Figure 5 show the MAPE against σ for XGBOOST in Part 1. The σ_o values selected are 110 for n_estimators, 3 for max_depth, 0.08 for learning_rate, and 0 for gamma. Then, we perform Part 2 (Lines 9–16). Part 2 is the same as Part 1 except that σ_o^* is used instead of σ_d^*. The blue triangle curves in Figure 5 show the MAPE against

σ for XGBOOST in Part 2. The σ_o values selected are the same as those in Part 1, that is, $\Sigma_I = \Sigma_{I-1}$ in Line 9.

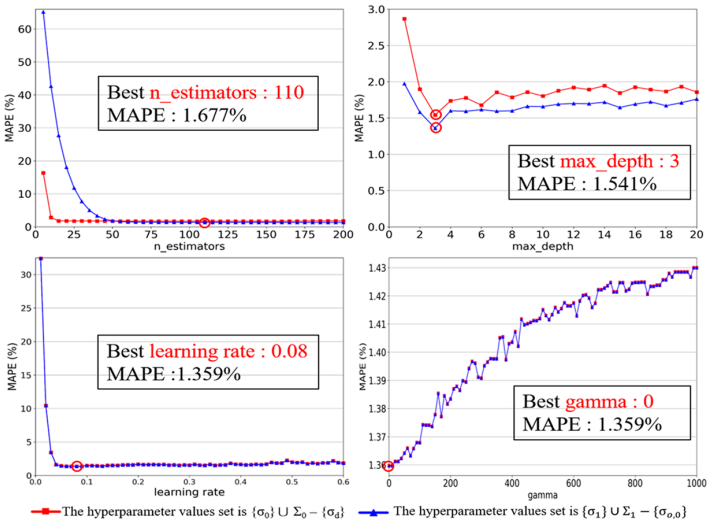


Figure 5. Hyperparameter tuning for XGBOOST. (Red circles indicate the optimal value).

Therefore, the selected values converge. If $\Sigma_I \neq \Sigma_{I-1}$, we repeat Part 2. Similarly, we select the hyperparameters for Random Forest following the same process, and Figure 6 indicates that σ_o values selected are 10 for n_estimators, 21 for max_depth, 20 for min_samples_split, and 0.06 for max_samples.

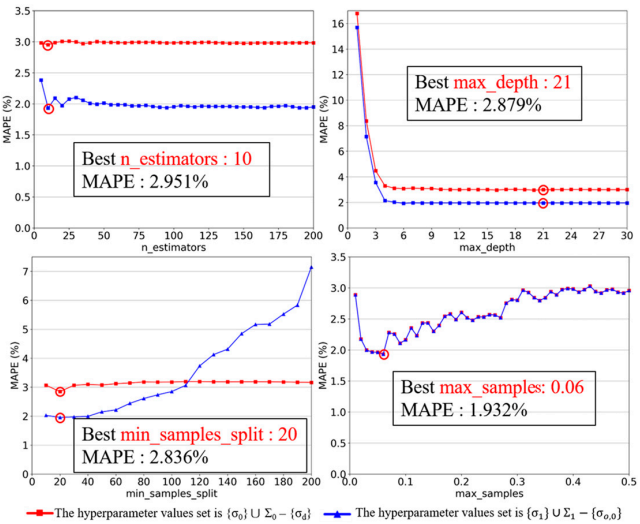


Figure 6. Hyperparameter tuning (tuning) for Random Forest. (Red circles indicate the optimal value).

SensorTalk3 also performs an ensemble of XGBOOST and Random Forest. The MAPEs of XGBOOST are smaller than Random Forest for most cases. However, we found that Random Forest outperforms XGBOOST on some data. Therefore, SensorTalk3 adopts the ensemble method [31] to yield the best results. Specifically, XGBOOST and Random Forests are used as the base learners. Then, we use linear regression as the meta-learner to combine the predictions of the two base learners, and the ensemble model is trained to improve the performance of the base learners.

5. Performance Evaluation

In this section, we first describe the SensorTalk3 performance in terms of the accuracy of calibration. Then, we describe how SensorTalk3 is implemented in Raspberry Pi4 with good time and space complexities.

5.1. Accuracy of Calibration

Based on the AI models described in Section 4, the MAPE performance of the original method, lookup table method [6,7], XGBOOST, Random Forest, and SensorTalk3 (Ensemble) are shown in Table 2. The table indicates that SensorTalk3 significantly reduces the MAPEs from 11.159% to 3.187% for Dataset 3, and from 7.792% to 1.738% for Dataset 4.

Table 2. (Ensemble) and other AI models.

Dataset.	MAPE				
	Original	Lookup Table	XGBOOST	Random Forest	Ensemble
Dataset3	11.159%	5.690%	3.203%	3.453%	3.187%
Dataset4	7.792%	2.393%	1.818%	1.861%	1.738%

In Figure 4, if the change rate features are not used (as most previous approaches did), then the MAPEs for Datasets 3 and 4 are 3.516% and 1.83%, respectively. When the change rate with the window size 2 is used, the MAPEs are reduced to 3.18% (for Dataset 3) and 1.738% (for Dataset 4), respectively.

An interesting question is how the accuracies of temperature and moisture affected the EC prediction. Figure 7 shows the MAPEs for the EC prediction using the STD moisture/temperature sensors and the DUT moisture/temperature sensors, where their error rates for Dataset 3 are 3.51% and 3.69% with w = 0, 3.18% and 3.42% with w = 2, respectively. For Dataset 4, the error rates of the EC prediction using the STD/DUT moisture/temperature sensors are 1.83% and 1.96% with w = 0, 1.73% and 1.90% with w = 2, respectively. Compared to STD moisture/temperature sensors, the DUT moisture/temperature sensors have an average error of 8.3%. The figure indicates that the effect of the accuracies of moisture/temperature is not significant in terms of EC calibration. That is, even if STD moisture/temperature sensors are not used, SensorTalk3 can still calibrate EC well if the errors of the usual temperature/humidity sensors are reasonably small.

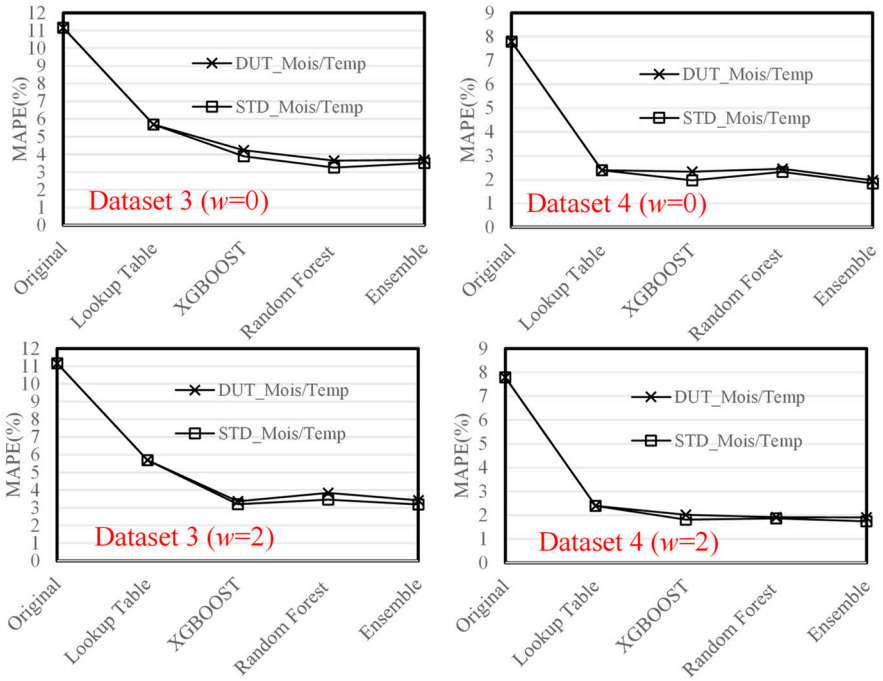


Figure 7. Effect of the accuracies of temperature and moisture.

5.2. Time and Space Complexities of the Edge-Based IoTtalk Engine

In SensorTalk3, the IoTtalk engine is installed in an industry version of Raspberry Pi4 (Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5 GHz and Memory: 4 GB LPDDR4), which provides the edge computing solution.

Most edge AI solutions (see [32] and the references therein) perform model training in the network and may further train the edge nodes by using federated learning or transfer learning. Unlike these restricted approaches, the edge-based SensorTalk3 performs real-time training at the edge.

In Figure 2, the data sent from (4)–(6) to the IoTtalk Engine periodically for every 20 s. From the measurement of edge-based SensorTalk3, the time complexity $C_{T,n}$ for processing n consecutive data items in the training phase is expressed as

$$C_{T,n} \approx 0.06n + 249.2 \text{ (ms)} \tag{8}$$

From Equation (8), the execution time for one input data is $C_{T,1} \approx 249.26$ (ms). In Section 4, SensorTalk3 is an ensemble method of XGBoost and Random Forest. Each of the AI models has 5 hyperparameters to tune. Therefore, when edge-based SensorTalk3 receives one data item, 10 models and the ensemble method are executed to process the data item. The processing time is $249.26 \times 11 = 2741.86$ ms. The sampling rate of SensorTalk3 is 1/20 s. That is, we have 20 s to process a data item. From the above calculation results, the sampling rate can accommodate the execution of model training. Each model consumes 158 MB of storage, and 4 GB memory suffices to accommodate $158 \times 11 = 1.74$ GB storage consumption. Therefore, SensorTalk3 is able to collect data and train AI models in parallel at the edge node.

From 10,000 measurements, the time complexity to process one data item in the inference phase is 0.01407 s (with a variance of $0.4007/(0.01407)^2$).

The analysis of the time and the space complexities in this subsection shows that we have not used up the time/space resources of the edge node, which proves the feasibility of transfer learning conducted in the edge-based SensorTalk3 solution. Such feasibility is successfully achieved after we have resolved a major challenge of SensorTalk3: the IoTtalk engine is implemented in Python3. When the engine is installed in Raspberry Pi4, the IoTtalk code can be easily broken into and illegally modified to cause malfunctions. To resolve this security issue, we collaborated with Winbond by using the W77Q TrustME® Secure Serial Flash memory. The details are given in Appendix A. We note that transfer learning in the cloud will take extra time before a sensor is calibrated, and during the calibration period, the sensor will produce inaccurate values. With SensortTalk3's transfer learning in the edge node, we can alleviate this problem.

5.3. Calibration Frequency

An important issue for automatic calibration is how to detect when an EC sensor drifts and to perform the AI training procedure in Section 4. One possibility is to periodically perform AI training. A major problem of this approach is that it is difficult to find the optimal frequency to re-train. If we re-train too frequently, then the AI training cost is too high. If we re-train too infrequently, then the EC sensor will report wrong values for a long time before it is calibrated again.

Thus, we propose the dual-sensor detection approach that uses two (non-standard) EC sensors for failure detection. Consider the timing diagram in Figure 8.

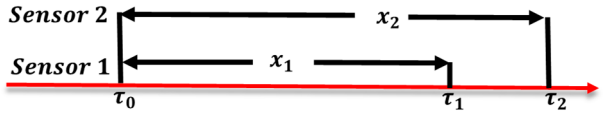


Figure 8. The timing diagram.

Initially, both EC Sensors 1 and 2 are calibrated at time τ_0 by a standard EC sensor using our AI training procedure. After τ_0, both sensors start producing the measured samples. When the SensorTalk server receives the redundant data from these two sensors, we check if they are consistent. If not, then at least one of them needs to be calibrated. Suppose that Sensor 1 drifts at τ_1 and Sensor 2 drifts at τ_2, respectively. Let $x_1 = \tau_1 - \tau_0$ and $x_2 = \tau_2 - \tau_0$. It is clearly that we receive consistent results in the time period $x_{min} = \min(x_1, x_2)$. After $\tau_0 + x_{min}$, we will detect inconsistent measures. In Figure 8, $x_{min} = x_1$, which means that Sensor 2 is still producing correct data after time τ_1. However, we need to calibrate both sensors again at τ_1. Therefore, the frequency for calibration of a sensor is $1/E[x_{min}]$. In the optimal case (which, unfortunately, cannot be achieved), we only need to calibrate a sensor when it actually drifts, and the frequencies for calibration of Sensors 1 and 2 are $1/E[x_1]$ and $1/E[x_2]$, respectively. Therefore, as compared with the "theorical optimal" overhead, the relative calibration overheads for Sensors 1 and 2 in our approach are

$$O_1 = \frac{E[x_1]}{E[x_{min}]} \text{ and } O_2 = \frac{E[x_2]}{E[x_{min}]} \tag{9}$$

We derive x_{min} and then O_1 and O_2 as follows. From Figure 8, it is clear that

$$E[x_{min}] = E[x_1]\Pr[x_2 > x_1] + E[x_2]\Pr[x_1 > x_2] \tag{10}$$

Suppose that x_1 and x_2 have Erlang distributions. The Erlang distribution is widely used in IoT network modeling [6]. With the shape parameter i and the scale parameter β, the Erlang density function is expressed as

$$f_{E,i}(\beta, x) = \frac{\beta^i x^{i-1} e^{-\beta x}}{(i-1)!} \tag{11}$$

Let the density functions of x_1 and x_2 be $f_{E,i}(\beta_1, x_1)$ and $f_{E,j}(\beta_2, x_2)$, respectively. Then

$$E[x_1] = \frac{i}{\beta_1}, \; E[x_2] = \frac{j}{\beta_2} \tag{12}$$

From Equation (11)

$$\begin{aligned} & E[x_2]\Pr[x_1 > x_2] \\ &= \int_{x_2=0}^{\infty} \int_{x_1=x_2}^{\infty} x_2 f_{E,i}(\beta_1, x_1) f_{E,j}(\beta_2, x_2) dx_1 dx_2 \\ &= \int_{x_2=0}^{\infty} x_2 \left[\frac{\beta_2^j x_2^{j-1} e^{-\beta_2 x_2}}{(j-1)!} \right] \sum_{k=0}^{i-1} \left(\frac{\beta_1^k x_2^k e^{-\beta_1 x_2}}{k!} \right) dx_2 \\ &= \sum_{k=0}^{i-1} \left[\frac{\beta_1^k \beta_2^j}{k!(j-1)!} \right] \left[\frac{(k+j)!}{(\beta_1+\beta_2)^{k+j+1}} \right] \\ &\times \int_{x_2=0}^{\infty} \left[\frac{(\beta_1+\beta_2)^{k+j+1} x_2^{k+j} e^{-(\beta_1+\beta_2)x_2}}{(k+j)!} \right] dx_2 \\ &= \sum_{k=0}^{i-1} j \binom{k+j}{k} \left[\frac{\beta_1^k \beta_2^j}{(\beta_1+\beta_2)^{k+j+1}} \right] \end{aligned} \tag{13}$$

Substitute Equation (13) into Equation (10) to yield

$$E[x_{min}] = \sum_{k=0}^{i-1} j \binom{k+j}{k} \left[\frac{\beta_1^k \beta_2^j}{(\beta_1+\beta_2)^{k+j+1}} \right] + \sum_{k=0}^{j-1} i \binom{k+i}{k} \left[\frac{\beta_2^k \beta_1^i}{(\beta_1+\beta_2)^{k+i+1}} \right] \tag{14}$$

Let $\beta_2 = \alpha\beta_1$, then Equation (14) is rewritten as

$$E[x_{min}] = \sum_{k=0}^{i-1} \left(\frac{j}{\beta_1} \right) \binom{k+j}{k} \left[\frac{\alpha^j}{(1+\alpha)^{k+j+1}} \right] + \sum_{k=0}^{j-1} \left(\frac{i}{\beta_1} \right) \binom{k+i}{k} \left[\frac{\alpha^k}{(1+\alpha)^{k+i+1}} \right] \tag{15}$$

From Equations (9), (12) and (15), we have

$$\frac{E[x_{min}]}{E[x_1]} = \frac{1}{O_1} = \sum_{k=0}^{i-1} \binom{j}{i} \binom{k+j}{k} \left[\frac{\alpha^j}{(1+\alpha)^{k+j+1}} \right] + \sum_{k=0}^{j-1} \binom{k+i}{k} \left[\frac{\alpha^k}{(1+\alpha)^{k+i+1}} \right] \quad (16)$$

From Equations (9) and (12), we have

$$O_2 = \left(\frac{j}{\alpha i} \right) O_1 \quad (17)$$

Based on Equations (16) and (17), Figure 9 plots O_1 and O_2 against α, i and j. The figure indicates that the higher the α value, the larger the overhead O_1. A larger value of α implies $\times 2$ is smaller than $\times 1$, which requires Sensor 1 to be frequently calibrated before it actually drifts. Suppose that both Sensors 1 and 2 are produced from the same batch of the manufacture, then we assume that $\alpha = 1$ and $i = j$. If $j = i$ then Equation (18) is simplified as

$$O_1 = \left\{ \sum_{k=0}^{i-1} \binom{k+i}{k} \left[\frac{\alpha^i + \alpha^k}{(1+\alpha)^{k+i+1}} \right] \right\}^{-1} \quad (18)$$

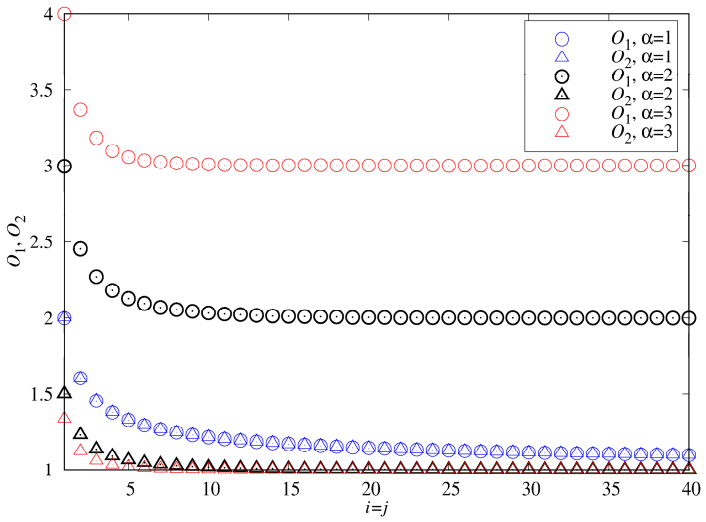


Figure 9. O_1 and O_2 against α, i and j.

If $\alpha = 1$ then x_1 and x_2 have identical Erlang distribution, and Equation (18) is simplified as

$$O_1 = O_2 = \left[\sum_{k=0}^{i-1} \binom{k+i}{k} \left(\frac{1}{2^{k+i}} \right) \right]^{-1} \quad (19)$$

The overheads O_1 and O_2 are the same when $\alpha = 1$. When $\alpha = 2$ and 3, O_1 is twice and triple of O_2, respectively. If $i = 1$ then x_1 and x_2 are exponential distributions with different means, and Equation (14) is rewritten as

$$O_1 = \left[\frac{\alpha^0 + \alpha^1}{(1+\alpha)^{1+1}} \right]^{-1} = 1 + \alpha \quad (20)$$

If $\alpha = 1$, then x_1 and x_2 have identical exponential distributions, and Equation (19) or Equation (20) are simplified as

$$O_1 = O_2 = 2 \quad (21)$$

which gives the mean value analysis [6] to provide the upper bound overheads of our approach. In Figure 9, if $\alpha = 1$, i = 2 then Equation (18) is re-written as

$$O_1 = O_2 = \left[\sum_{k=0}^{1}\binom{k+2}{k}\left(\frac{1}{2^{k+2}}\right)\right]^{-1} = \frac{8}{5} \tag{22}$$

If $\alpha = 1$, i = 3 then Equation (18) is re-written as

$$O_1 = O_2 = \left[\sum_{k=0}^{2}\binom{k+3}{k}\left(\frac{1}{2^{k+3}}\right)\right]^{-1} = \frac{16}{11} \tag{23}$$

Equations (21)–(23) indicate that as the sensor life is more stable (i.e., the variance is smaller or i is larger), the overheads O_1, O_2 are smaller. Indeed, Figure 9 shows that the overheads O_1 and O_2 decrease as $i = j$ increases. If non-standard Sensors 1 and 2 are homogeneous and stable ($\alpha = 1$ and $i = j \gg 1$), then $O_2 = O_1 \approx 1$, and the extra overhead is negligible.

6. Conclusions

SensorTalk3 is an innovative approach for EC sensor calibration, which can be conducted in the cloud or at the edge node. Our study indicated that the soil information (both temperature and moisture sensor data) plays an important role in SensorTalk3, which significantly outperforms the existing lookup table calibration approaches. The MAPE of SensorTalk3 can be as low as 1.738%.

SensorTalk3 is sensitive to the change rate of temperature, moisture and EC in 40 s (i.e., w = 2), and the changes larger or smaller than 40 s have a smaller impact on the calibration. Our study also indicated that the accuracies of moisture/temperature are not significant in terms of EC calibration. Specifically, if the error rates of uncalibrated moisture and temperature sensors are under 8.3%, SensorTalk3 can still calibrate EC well.

SensorTalk3 conducts EC sensor calibration by using XGBOOST and Random Forest. The time/space complexities of these two AI models are lower than complicated deep learning AI models. Therefore, we can perform model training in parallel during data collection at the edge node without GPU acceleration. When the training data are completely collected, AI training is also finished at the edge node. Such an AI training approach has not been found in most edge AI approaches. We also proposed the dual-sensor detection solution to determine when to conduct recalibration. The overhead of this solution is less than twice the optimal detection (which cannot be achieved practically). If the two non-standard sensors are homogeneous and stable ($\alpha = 1$ and $i = j \gg 1$), then $O_2 = O_1 \approx 1$, then the extra overhead is negligible. Traditional approaches necessitate the training of calibration AI models in the cloud. A major contribution of SensorTalk3 is that it revolutionizes this process by introducing a groundbreaking feature: transfer learning can now be conducted directly on the edge node. This cutting-edge capability becomes especially advantageous in the context of agricultural settings, where numerous farming sensors are deployed throughout vast fields. By enabling local transfer learning on inexpensive edge nodes, SensorTalk3 offers farmers a significantly more cost-effective solution for optimizing their systems. Additionally, there is a delay in achieving calibration and obtaining accurate readings when implementing transfer learning in the cloud. This can lead to inaccurate values being generated by the sensor during the calibration period. However, by utilizing transfer learning in the edge node with SensortTalk3, we can mitigate this issue.

SensorTalk3 has been deployed in commercial farm fields (refer to Figure 1b). Similar to other AI applications, when SensorTalk3 is utilized in new farm fields with varying conditions such as temperature range, soil type, relief, etc., standard transfer learning is necessary. This learning process can be conducted seamlessly and automatically through the IoT configuration of SensorTalk3 (refer to Figure 2).

Author Contributions: Conceptualization, Y.-W.L. and Y.-B.L.; methodology, Y.-W.L. and Y.-B.L.; software, B.-X.L.; validation, Y.-B.L.; data curation, B.-X.L.; writing—original draft preparation, Y.-B.L.; writing—review and editing, Y.-W.L.; supervision, Y.-W.L.; project administration, Y.-B.L.; funding acquisition, T.C.-Y.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Science and Technology Council (NSTC) 112-2221-E-033-023, 110-2622-8-A49-022, 112-2221-E-A49-049, 112-2221-E-A49-047, NCKU Miin Wu School of Computing, Research Center for Information Technology Innovation.

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

Appendix A

The edge-based SensorTalk3 solution protected by W77Q TrustME® secure serial flash memory. With W77Q supported by Winbond, the IoTtalk Python code is well protected in an industrial Raspberry Pi4 to avoid illegal access. The edge-based SensorTalk3 server is illustrated in Figure A1.



Figure A1. Secured edge-based SensorTalk3 server.

W77Q TrustME® Secure Serial Flash memory (Figure A2 (1)) provides a secure storage solution for Pi4 (Figure A2 (2)) with limited space, pins and power, which meets Common Criteria EAL2 Security Certification requirements [33]. W77Q is a drop-in replacement for standard Serial NOR Flash Memory devices, offering security, flexibility, and performance well beyond ordinary NOR Flash Memory devices. It provides support for execution in place, cryptographic key distribution, and secure data storage. W77Q features sophisticated cryptographic encryption of the communication channel, personalization of each device with unique keys, cryptographic read and write locks, protection of data integrity, secure firmware update, root of trust functions, and secure read, write and erase operations. W77Q supports single, dual, and quad SPI as well as QPI modes of operation, running at up to 133 MHz. Dual Transfer Rate (DTR) is supported at rates up to 66 MHz.

In the proof of concept (POC) stage, we connected W77Q with Pi4 through individual development boards, and executed the startup procedure for the IoTtalk engine as illustrated in Figure A2 (3). The Python code of IoTtalk is protected in W77Q with the 256-bit SHA256 key. When the system is turned on, the startup procedure first check if the MAC address and the cryptographic key of the Pi4 are correct. If so, the protected IoTtalk code is loaded into the Pi4 storage for execution, and then the security flash is disconnected [34]. After the POC stage, W77Q and Pi4 were integrated into a printed circuit board (Figure A3) and become a commercial product, as illustrated in Figure A1.

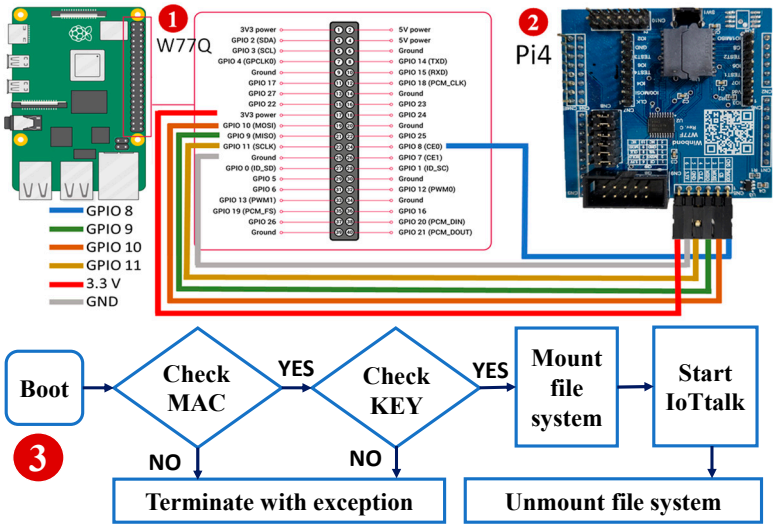


Figure A2. Connecting W77Q and Pi4 for IoTtalk engine.

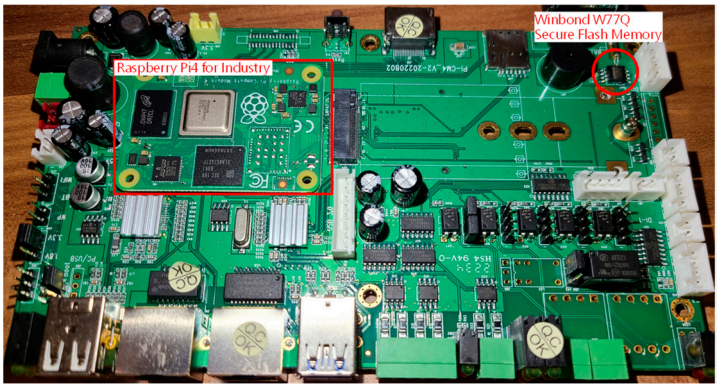


Figure A3. The edge-based IoTtalk circuit board.

References

1. Vuran, M.C.; Salam, A.; Wong, R.; Irmak, S. Internet of underground things in precision agriculture: Architecture and technology aspects. *Ad Hoc Netw.* **2018**, *81*, 160–173. [CrossRef]
2. Sridharani, J.; Chowdary, S.; Nikhil, K. Smart farming: The IoT based future agriculture. In Proceedings of the International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 20–22 January 2022; pp. 150–155.
3. Chen, W.-L.; Lin, Y.-B.; Lin, Y.-W.; Chen, R.; Liao, J.-K.; Ng, F.-L.; Chan, Y.-Y.; Liu, Y.-C.; Wang, C.-C.; Chiu, C.-H.; et al. AgriTalk: IoT for Precision Soil Farming of Turmeric Cultivation. *IEEE Internet Things J.* **2019**, *6*, 5209–5223. [CrossRef]
4. Yang, A.; Wang, P.; Yang, H. In Situ Blind Calibration of Sensor Networks for Infrastructure Monitoring. *IEEE Sens. J.* **2021**, *21*, 24274–24284. [CrossRef]
5. Joshi, V.R.; Srinivasan, K.; Manivannan, S.S. Intelligent agricultural farming system using internet of things. In Proceedings of the IEEE International Conference on Consumer Electronics—Taiwan (ICCE-TW), Yilan, Taiwan, 20–22 May 2019; pp. 1–2.
6. Lin, Y.-W.; Lin, Y.-B.; Hung, H.-N. CalibrationTalk: A Farming Sensor Failure Detection and Calibration Technique. *IEEE Internet Things J.* **2021**, *8*, 6893–6903. [CrossRef]
7. Lin, Y.-B.; Lin, Y.-W.; Lin, J.-Y.; Hung, H.-N. SensorTalk: An IoT Device Failure Detection and Calibration Mechanism for Smart Farming. *Sensors* **2019**, *19*, 4788–4807. [CrossRef] [PubMed]
8. Lin, Y.-B.; Lin, Y.-W. SensorTalk: Extending the Life for Redundant Electrical Conductivity Sensor. *IEEE Internet Things J.* **2022**, *9*, 16619–16630. [CrossRef]
9. Murata. Soil Sensor. 2023. Available online: https://www.murata.com/en-global/products/sensor/soil (accessed on 22 October 2023).
10. Yang, G.; Rezaee, H.; Parisini, T. Sensor Redundancy for Robustness in Nonlinear State Estimation. In Proceedings of the 2019 IEEE 58th Conference on Decision and Control (CDC), Nice, France, 11–13 December 2019; pp. 3865–3870.
11. Li, B.; Wang, H.; Mu, L.; Shi, Z.; Du, B. A configuration design method for a redundant inertial navigation system based on diagnosability analysis. *Meas. Sci. Technol.* **2020**, *32*, 25111. [CrossRef]
12. Murata. *Soil and Water Environment Sensor Specification Sheet*; Technical Report JEDS18S-0003; Murata: Kyoto, Japan, 2021.

13. Winkler, N.P.; Neumann, P.P.; Schaffernicht, E.; Lilienthal, A.J. Using redundancy in a sensor network to compensate sensor failures. In Proceedings of the IEEE Sensors, Sydney, Australia, 31 October–3 November 2021; pp. 1–4.
14. Hilhorst, M.A. A pore water conductivity sensor. *Soil Sci. Soc. Am. J.* **2000**, *64*, 1922–1925. [CrossRef]
15. Futagawa, M.; Ban, Y.; Kawashima, K.; Sawada, K. On-site monitoring of soil condition for precision agriculture by using multimodal microchip integrated with EC and temperature sensors. In Proceedings of the International Conference on Solid State Sensors and Actuators (TRANSDUCERS), Barcelona, Spain, 16–20 June 2013; pp. 112–115.
16. Ou, I.-C.; Tsai, K.-J.; Chu, Y.-H.; Liao, Y.-T. Self-Sustaining Soil Electrical Conductance Measurement Using a DC–DC Power Converter. *IEEE Sens. J.* **2021**, *19*, 10560–10567. [CrossRef]
17. Concas, F.; Mineraud, J.; Lagerspetz, E.; Varjonen, S.; Liu, X.; Puolamaki, K.; Nurmi, P.; Tarkoma, S. Low-Cost Outdoor Air Quality Monitoring and Sensor Calibration: A Survey and Critical Analysis. *ACM Trans. Sens. Netw.* **2021**, *17*, 1–44. [CrossRef]
18. Zhang, S.; Tian, F.; Covington, J.A.; Li, H.; Zhao, L.; Liu, R.; Qian, J.; Liu, B. A Universal Calibration Method for Electronic Nose Based on Projection on to Convex Sets. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 2516012. [CrossRef]
19. Ye, H.; Li, X.; Dong, K. Crowdsensing based barometer sensor calibration using smartphones. In Proceedings of the 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation, Guangzhou, China, 8–12 October 2018; pp. 1555–1562.
20. Thanh, O.V.; Puigt, M.; Yahaya, F.; Delmaire, G.; Roussel, G. In situ calibration of cross-sensitive sensors in mobile sensor arrays using fast informed non-negative matrix factorization. In Proceedings of the 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 3515–3519.
21. Kim, S.; Sung, H.; Kim, S.; Je, M.; Kim, J.-H. ML-based humidity and temperature calibration system for heterogeneous MOx sensor array in ppm-level BTEX monitoring. In Proceedings of the 2021 IEEE International Symposium on Circuits and Systems (ISCAS), Daegu, Republic of Korea, 22–28 May 2021; pp. 1–5.
22. Robin, Y.; Goodarzi, P.; Baur, T.; Schultealbert, C.; Schutze, A.; Schneider, T. Machine learning based calibration time reduction for gas sensors in temperature cycled operation. In Proceedings of the 2021 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), Glasgow, UK, 17–20 May 2021.
23. Veiga, T.; Ljunggren, E.; Bach, K.; Akselsen, S. Blind calibration of air quality wireless sensor networks using deep neural networks. In Proceedings of the 2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS), Barcelona, Spain, 23–25 August 2021; pp. 1–6.
24. Jha, S.K.; Kumar, M.; Arora, V.; Tripathi, S.N.; Motghare, V.M.; Shingare, A.A.; Rajput, K.A.; Kamble, S. Domain adaptation based deep calibration of low-cost PM2.5 sensors. *IEEE Sens. J.* **2021**, *21*, 25941–25949. [CrossRef]
25. Motie, J.B.; Aghkhani, M.H.; Rohani, A.; Lakzian, A. A soft-computing approach to estimate soil electrical conductivity. *Biosyst. Eng.* **2021**, *205*, 105–120. [CrossRef]
26. Lin, Y.-B.; Lin, Y.-W.; Huang, C.-M.; Chih, C.-Y.; Lin, P. IoTtalk: A Management Platform for Reconfigurable Sensor Devices. *IEEE Internet Things J.* **2017**, *4*, 1552–1562. [CrossRef]
27. Chen, T.; Guestrin, C. XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
28. Sheng, C.; Yu, H. An optimized prediction algorithm based on XGBoost. In Proceedings of the International Conference on Networking and Network Applications (NaNA), Urumqi, China, 3–5 December 2022; pp. 1–6.
29. Ho, T.K. Random decision forests. In Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, Canada, 14–16 August 1995; pp. 278–282.
30. More, A.S.; Rana, D.P. Review of random forest classification techniques to resolve data imbalance. In Proceedings of the International Conference on Intelligent Systems and Information Management (ICISIM), Aurangabad, India, 5–6 October 2017; pp. 72–78.
31. Rokach, L. Ensemble-based Classifiers. *Artif. Intell. Rev.* **2010**, *33*, 1–39. [CrossRef]
32. Chang, Z.; Liu, S.; Xiong, X.; Cai, Z.; Tu, G. A Survey of Recent Advances in Edge-Computing-Powered Artificial Intelligence of Things. *IEEE Internet Things J.* **2021**, *8*, 13849–13875. [CrossRef]
33. Winbond. W77Q32JW/W77Q16JW 1.8V 32M-BIT/16M-BIT Secure Serial NOR Flash Memory with Dual/Quad SPI, QPI & DTR. Technical Report W77QDS0100 Rev. B. Winbond, 2021.
34. Winbond. *Winbond's W77Q TrustME®for Raspberry Pi4 Setup Guide*; Winbond: Taichung City, Taiwan, 2021.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.