Adversarial Adaptation for Long-term Label-free RFID Sensor Recalibration

Anonymous for Review

Abstract—**Device-Free Wireless Sensing (DFWS) technologies such as Radio-Frequency Identification (RFID) and WiFi have been widely adopted in many indoor monitoring systems for localisation or human activity recognition. With the increase in long-term deployments of these systems, there emerges an outstanding challenge: how to recalibrate sensors automatically. This challenge is rooted in the fact that these sensors often produce readings that are temporally drifted, or gradually deviate from the true values, due to the deterioration of sensing units and environmental interference. To directly tackle this challenge, this paper presents an adversarial adaptation technique that automatically aligns current, shifted sensor readings with the originally collected readings without the need for collecting true sensor readings nor target labels such as locations or activities. We evaluate our technique over a one-year-long dataset collected in a realistic domestic environment and two third-party datasets. In comparison with the state-of-the-art techniques, our approach has significantly improved performance in predicting target labels by constantly recalibrating sensor data.**

Index Terms—**Sensor recalibration, RFID, adversarial adaptation, generative adversarial network, representation learning, Autoencoder, smart home, localisation, human activity recognition**

I. Introduction

In recent years, Device-Free Wireless Sensing (DFWS) technologies are increasingly employed for non-intrusive indoor monitoring [23]. The key advantage of DFWS approaches is that they do not require users to wear or carry devices. Users also find them more acceptable as they are not viewed directly, as they would be for instance when monitored using cameras [15]. Examples of DFWS include WiFi and Radio-Frequency Identification (RFID) systems. The radio frequency (RF) signals are used as a sensing medium, where the presence, the location and the activity of humans can be estimated by analysing the way their presence interferes with the wireless signal; i.e. the way they absorb and/or reflect it, thus influencing and distorting the measurement of Received Signal Strength (RSS) from a fixed point in space.

However, while RF signals can be perturbed by the presence of humans in the environment, they are also highly susceptible to multipath errors and other perturbations, due to physical obstructions, interference from metal presents in the environments, and even from other RF-based appliances. Without recalibration, we cannot reliably use RFID systems to predict the current situation of an environment or a user, which cannot trigger correct behaviours of a system and lead to adverse consequences. However, it can be expensive to recalibrate sensors; for example, constantly measuring true readings with a high-quality reference sensor, or re-annotating shifted sensor data with target labels (such as locations or activities of a user). Both practices can be costly in terms of time and technician effort and sometimes can be impractical or infeasible in certain environments. Therefore, ensuring robust and reliable performance over a long-term deployment remains an outstanding challenge [22].

To directly tackle this sensor error challenge over a long-term deployment, we propose an adversarial adaptation technique that can automatically and constantly matches the current sensor data to the sensor data collected at the initialisation stage. We first learn robust latent representations of sensor data via an Autoencoder. Then, we will train an aligner and a discriminator in an adversarial manner to minimise the data distribution between the current sensor data and the initial sensor data. That is, an aligner transforms the current data into a representation such that the reconstruction error on the original Autoencoder will be minimised. A discriminator as an adversary aims to maximise the reconstruction error difference between the current data and the original data. Through this minmax play, an aligner will learn the transformation function. To further confuse the discriminator, we adopt a recent mixup technique [25] to mix original and noisy or drifted sensor data, which has demonstrated the efficacy in improving the robustness of the aligner.

The main contributions of our work are summarised below.

- We have novelly applied an adversarial adaptation technique to sensor recalibration challenge. It enables continuously and automatically aligning the current, noisy sensor data to the clean sensor data that has been collected during the initialisation stage of a sensor system deployment. It does not require collecting any true sensor readings at the deployment stage nor any target labels (such as a location or human activity).
- We have deployed Ultra High Frequency (UHF) RFID [19] in a real-world environment and attempted to sporadically collect and annotate sensor data over a year-long period to allow for analysis of sensor error. Our technique has demonstrated the strength in recalibrating the sensors and leads to improved localisation accuracy.
- We have demonstrated the generalisation of our approach by evaluating on two long-term, real-world datasets that are collected on contact switch and pressure sensors.

II. Related Work

This section briefly introduces the existing dense RFID DFWS deployments and highlights the difference of our work from them. We also describe sensor recalibration techniques in general and unsupervised domain adaption techniques.

A. *Dense RFID DFWS*

A number of other project have proposed similar dense deployments of passive RFID tags in walls or floors. Torres et al. [18] have placed batteryless RFID tags into floor carpets to detect falls. The carpets were deployed in a controlled environment and participants are recruited to perform different activities and falls. The purpose is to learn the correlation between RSSI and body motion of falls. Yao et al. [13], [27] have attempted to deploy passive RFID tags on the wall in a room to detect low-level activities such as sitting, standing, and arm waving. Similarly Ruan et al. [15] have also fabricated a grid of passive RFID tags in the rubber cladding of the wall. The purpose is to detect activities such as walking, running, and falling. However, all of these examples are evaluated in tightly controlled/stable laboratory settings, over short periods of time [22]. On the contrary, we explicitly investigate the long-term behaviour of a dense RFID system, by analysing data we have collected sporadically, over a year, in order to highlight and address the concept drift problem in the context of a localisation task.

B. *Sensor Recalibration*

A wide range of techniques have been proposed to address the problem of sensor drift. Statistical dimensionality reduction methods such as PCA and PLS have long been used to remove sensor drift [1]. Exploiting spatial and temporal correlations, Kalman Filtering and Spatial Kriging have also been applied to detect and automatically calibrate sensor drift [10]. Other machine learning based techniques have also been tried. For example, Rao et al. [14] used Support Vector Machine (SVM) with Gaussian Kernels and Ensemble Of Trees (EOT) regression methods to address the sensor drift used in nuclear power plants, which shows satisfactory empirical test results.

Distributed drift management methods have also been investigated. Saeed et al. [16] used low-cost computing machines, namely, Raspberry Pi, to successfully detect sensor drift. More recently, deep learning based methods have also emerged. Wang et al. [24] used RNN (LSTM) to predict the drift of chemical sensors. The method has achieved better results than other methods like SVM. Yan et al. [26] used Autoencoders together with domain adaptation to correct sensor drift. Their approach achieves higher accuracy than other basic drift correction methods such as PCA. Inspired by the success of this work, we also turn sensor recalibration problem into a domain adaptation problem. Differently, we not only use an Autoencoder for feature learning but also employ an adversarial adaptation to align sensor distributions.

C. *Adversarial Domain Adaptation*

Recently, adversarial domain adaptation is emerging as a promising technique for domain adaptation in the context of transfer learning [21]. It performs feature space transformation by mapping the source distribution into the target distribution and the disparity between the two domains is measured by a domain discriminator [5]. Ganin et al. [6] have introduced the adversarial loss in neural networks to learn features that are discriminative for the task in the source domain and indiscriminative with respect to the data shift between the domains. Tzeng et al. introduced a domain confusion loss to optimise the feature representations such that the discrepancy between the source and target distributions is minimised [20]. Later, Adversarial Discriminative Domain Adaptation (ADDA) is proposed to improve this technique with an asymmetric mapping learned through a domain-adversarial loss to map the target data to the source representations [21]. Another promising technique is Deep Adaptation Network (DAN) [11], which is proposed to learn fine-grained feature mappings. It projects the hidden representations of the task-specific layers of a CNN onto RKHS and an optimal multi-kernel selection method is employed to reduce the discrepancy between domains.

III. Background and Motivation

This section will introduce our sensing technologies, sensor deployment in a simulated real-world house setting, data collection, and observed sensor error problem.

A. *Real-World Deployment*

We have configured our lab space into a $60m^2$ fully furnished test apartment, including a combined living/dining/kitchen area, bedroom, and bathroom. The system we use is a ceiling-to-floor designed for device-free monitoring. We attached RFID tags to Ethylene Vinyl Acetate (EVA) floor mats, in groups of four, to cover the whole lab space. We placed these mats across the bedroom and kitchen of the testbed, as depicted in Figure 1. The distribution of these tags in the test apartment is illustrated in Figure 2.

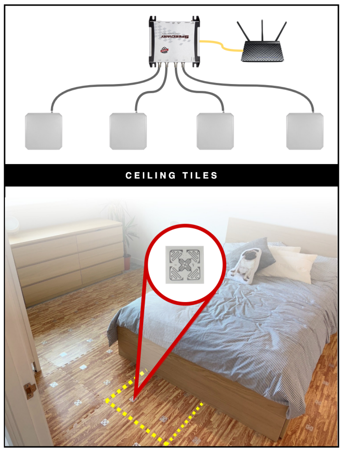

Fig. 1. Illustration of the experimental setup, showing reader hardware, and tags deployed in the bedroom of the laboratory. Dashed line highlights an EVA floor mat.

B. *RFID Signals*

The deployed RFID tags are continuously interrogated by multiple antennas concealed above the ceiling using the

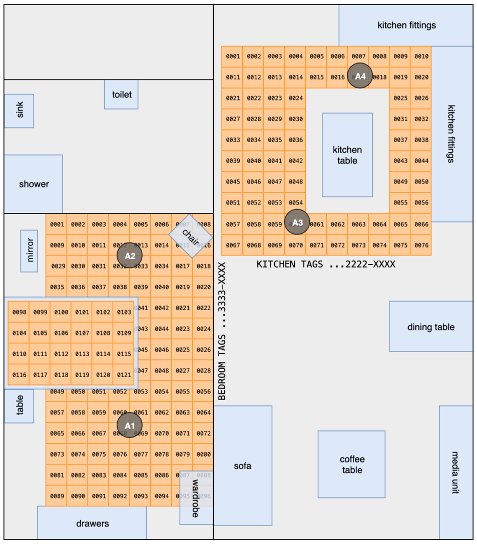

Fig. 2. Layout of RFID floor tags. The identifier numbers shown are the final byte of the unique tag identifier, while the preceding byte denotes the room code. 'A#' represents the four interrogating antennas.

backscatter coupling principle. That is, the tags remain in dormant stage and they are energised at regular intervals and exploit the energy of the interrogating signal. The energised tags will respond by emitting a signal of its own that encodes a unique identifier.

We utilised Commercial Off-the-Shelf (COTS) Ultra-High Frequency (UHF) RFID hardware, operating between 865-868 MHz:

- IMPINJ Speedway Revolution R420 UHF RFID Reader;
- 4 x Laird Far-Field RAIN RFID Antenna;
- 196 x IMPINJ RFID Monza 4QT Tags (on floor).

The reader transmits a signal of a power P_T. Individual RFID tags in the path of this signal are energised, and emit a signal of their own. If this signal is sufficiently powerful to reach the interrogating antenna, the reader measures the received power P_R. Using this information, the reader is then able to calculate the Received Signal Strength Indicator (RSSI). Given P_T and P_R, the path loss (in dB) can be calculated as $P_L = 10log_{10}\frac{P_T}{P_R}$. The RSSI, ignoring any antenna gain, can be measured as the difference between the transmitted power of a signal and the path loss: $RSSI = P_T - P_L$. According to the manufacturer, the range of RSSI values reported by the reader is -110 (min.) to 0 (max.) dB [7], although in reality the maximum dB is lower, with a zero value RSSI being used to indicate that no power was received back from a given tag.

In principle, these measurements should remain fairly constant without the presence of humans in the environment, assuming the environment remains static. In an indoor environment, we would expect movement of objects to influence measurements due to: (i) the effect of multipath propagation; and (ii) complete obstruction of a tag by an object or human. With humans in the environment, we would expect to see significant disruption of measurements, due in part to absorption [2], multipath effects, and occlusion. Figure 3 presents what happens to the RSS readings before and after people are in the lab during a week. There is clearly shift in the readings even under the same condition that the lab is completely empty.

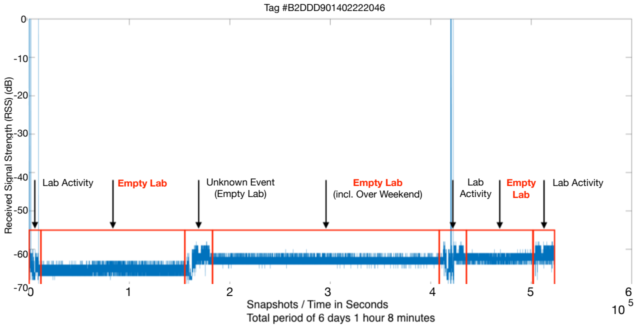

Fig. 3. An example of RFID noise over time

C. Data Collection

To collect the data, we recruited participants from across our research institute through open invitation via mailing lists and through word of mouth. Our study received ethical approval from the university for the collection and storage of data from human participants using our RFID system. We targeted only healthy adults, excluding members of the elderly population for participation in our study.

Participants were invited to a one-hour long session to perform naturalistic 'activity pathways': that is, to behave in a natural way, as if they were in their own home, as they moved throughout the testbed (between the two rooms) to perform activities of daily living. We recorded raw RFID data and video footage on how a participant performed. An annotator (one of the researchers in the project) manually inspected the video and provided location labels on regions where a participant stood, sat, or moved. The reason for location annotation is that we are interested in assessing the effectiveness of (UHF) RFID in detecting user locations when they perform daily activities.

The data was collected over the period from 30 July 2019 to 31 August 2020. However, for this work we have only annotated and used 5 non-consecutive days of data, with the aim to capture (UHF) RFID noise and error over a long-term deployment. This amounted to a total of 29,232 snapshots of data, where snapshots were recorded at a rate of one snapshot per second. A 'snapshot' represents peak RSSI measurements for each tag on the floor. The number of features in the dataset is equal to the number of target tags: 196.

D. Problem Statement

To demonstrate the sensor error problem, we perform the following experiment. We select a subject's data across different days and examine their sensor distributions on each location and/or activity. Figure 4 visualises of sensor traces on 3 sensors when the activity of reading is performed on 4 different days. Clearly we can observe the sensor readings fluctuate greatly, due to environmental factors, including temperature, humidity, or any other radio interference.

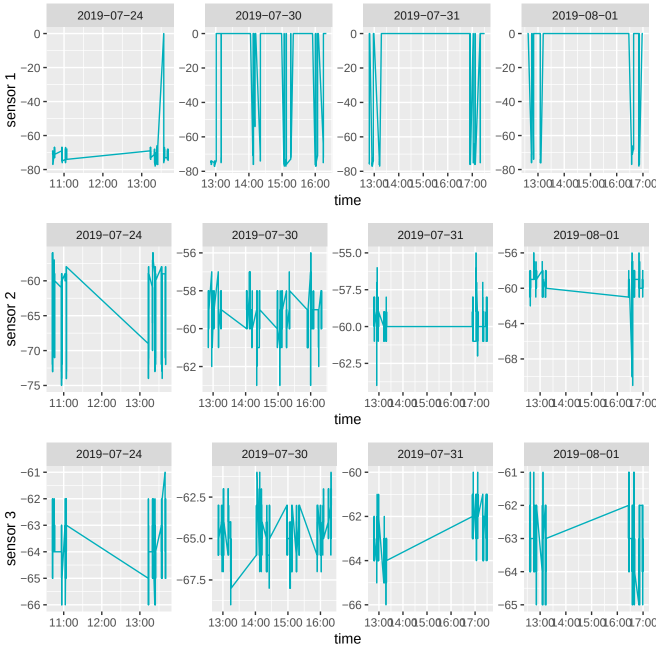

Fig. 4. Sensor feature distribution on different days

The variation in sensor readings can have significantly negative impact on the accuracy of localisation. Figure 5 presents the accuracy of predicting location labels on sensor readings on different days. Here, we train a classifier with 80% of data on D0. Then we use the classifier to predict locations on the remaining 20% of D0's data, on 100% of the other days' data such as D1, D2, and D3. As we can see, the accuracy on D0's test data is rather high, indicating the sensor readings on the same day are still consistent. But from D1, the accuracy drops significantly, which demonstrates that the later days' data dramatically deviates from Day 0.

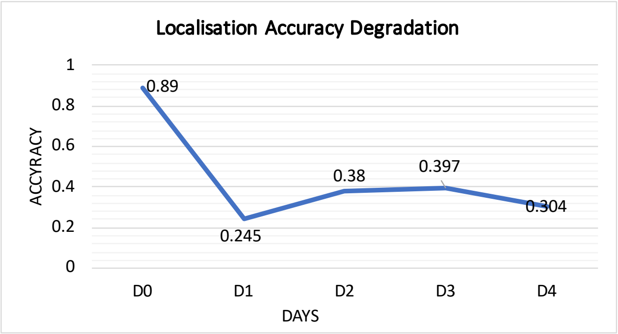

Fig. 5. Localisation accuracy degradation over days

This leads to the research question in our paper: *how to recalibrate sensor data automatically and continuously so that the system can maintain high accuracy over a long-term deployment*? A constraint in HAR is that labels on activities are often difficult to acquire in a real-world deployment. If we can collect sufficient labels, then we always re-train the classifier on each data and achieve high accuracy. However, it is not feasible to constantly query users about what they are doing. Ideally, an unsupervised learning approach during the deployment stage is more amenable. Therefore, the objective of this paper is to enable automatically and continuously sensor recalibration without the need of extra labelling.

IV. Proposed Approach

This paper proposes an unsupervised sensor alignment approach to alleviate the problem of sensor noise. That is, we only use the labelled-training data to build our first model and during the deployment stage, the system will automatically recalibrate sensor data without the need for collecting any new labels. In this section, we will first present an overview of the proposed approach and then describe the technical details of each component.

A. Overview

Let $D_{t_0} = \{(x_{t_0}^{(i)}, y_{t_0}^{(i)})\}_{i=1}^{n_{t_0}}$ be the sensor data collected at an initialisation stage t_0, where each data point x has a label y, and y is a prediction target; e.g., a location or activity. Similarly, $D_t = \{x_t^{(i)}\}_{i=1}^{n_t}$ denotes sensor data collected at a later stage t ($t > t_0$); e.g., the real-world deployment stage. Due to sensor drifting, the two population distributions of X_{t_0}, X_t are different. And our objective is to recalibrate the sensor data at t by aligning D_t to D_{t_0} such that the two distributions are similar. Therefore, a classifier trained on D_{t_0} then can be re-used to predict labels on D_t.

As sensor data are generally noisy [28], the readings can have large variations. Thus, our first step is to learn robust, inherent sensor features. To do so, we apply an Autoencoder which is an unsupervised technique to learn effective features [3], [17]. The second step is to learn an alignment function to from D_t to D_{t_0}.

B. Feature Learning

We use an Autoencoder to learn inherent features from sensor data. The Autoencoder consists of an encoder and a decoder, where the encoder compresses the input data X to a lower-dimension representation and the decoder reconstructs the input data by minimising some loss function between the input data and reconstructed data. More specifically, let $X \in \mathbb{R}^m$ be m-dimensional sensor feature space, an Autoencoder consists of 2 networks: an encoder f and a decoder g such that

$$f : X \to Z, g : Z \to X$$
$$f, g = argmin||X - g(f(X))||^2$$

where $Z \in \mathbb{R}^l, l < m$ represents the learnt latent features; that is, $z = f(X; \theta_f)$, where θ_f is the encoder's parameters such as weight and bias of the network f. The latent features are supposed to be more robust to sporadic sensor noise.

Once we have trained the Autoencoder, we will take the latent features and their corresponding labels $\{(z_{t_0}^{(i)}, y_{t_0}^{(i)})\}_{i=1}^{n_{t_0}}$ to train a classifier. After training on the initialisation data

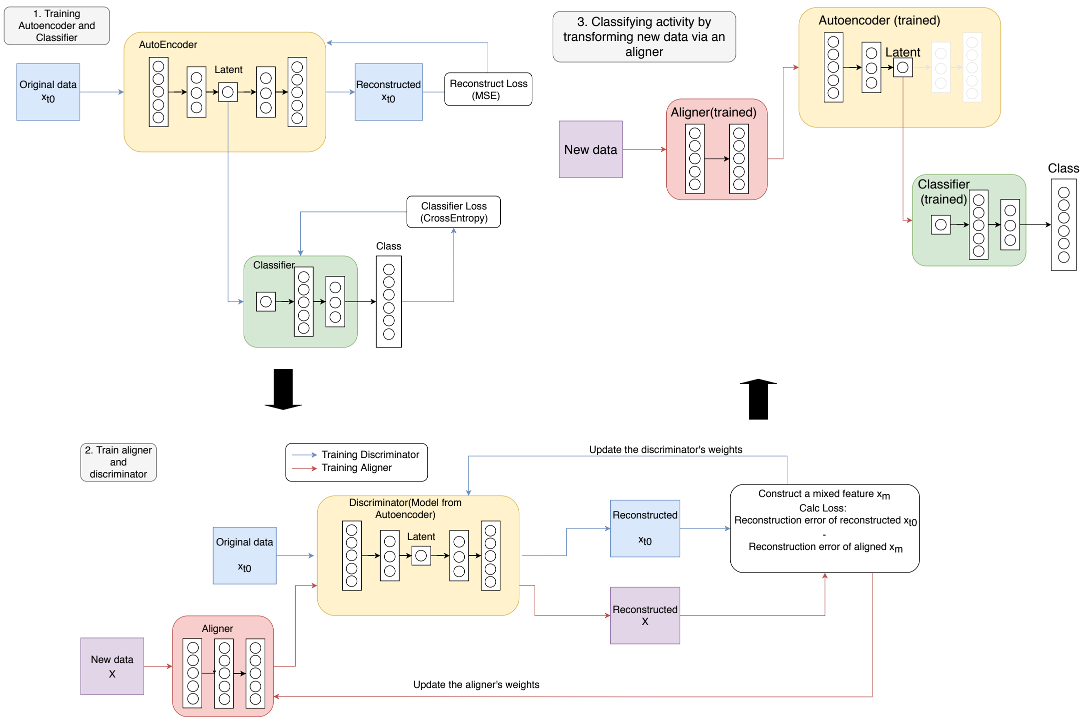
1. Training Autoencoder and Classifier
AutoEncoder
Original data x_{t0}
Latent
Reconstructed x_{t0}
Reconstruct Loss (MSE)
Classifier Loss (CrossEntropy)
Classifier
Class
3. Classifying activity by transforming new data via an aligner
Autoencoder (trained)
Latent
New data
Aligner(trained)
Classifier (trained)
Class
2. Train aligner and discriminator
Training Discriminator
Training Aligner
Update the discriminator's weights
Discriminator(Model from Autoencoder)
Latent
Original data x_{t0}
Reconstructed x_{t0}
Construct a mixed feature x_m
Calc Loss:
Reconstruction error of reconstructed x_{t0}
-
Reconstruction error of aligned x_m
Reconstructed X
New data X
Aligner
Update the aligner's weights

Fig. 6. Workflow of the proposed approach

D_{t_0}, both Autoencoder and classifier will be fixed. For all the new data D_t, we will align D_t to D_{t_0} and apply the same classifier for prediction, which reduces the need for collecting new labels and re-training or learning a separate classifier.

C. Feature Alignment

A recent research on aligning neural signals for brain-machine interaction has achieved promising results in automatically and constantly adapting the signals to predict muscle activities [4]. Here we adapt and extend this model for sensor data recalibration. The employed adversarial alignment model [4] is similar to generative adversarial network (GAN). It consists of an aligner and a discriminator, both of which are implemented as a deep neural network. The discriminator aims to maximise the separation of sensor features in D_t from those in D_{t_0} by maximising the difference of reconstruction errors between D_t and D_{t_0}. The aligner takes as input a sensor feature x from D_t and transforms into a vector $\tilde{x}$ such that the reconstruction error on the discriminator is minimised. This suggests that the transformed vector well conforms to the distribution of the original features in D_{t_0}.

The aligner is implemented as a neural network consisting of an input layer, a hidden layer, and an output layer, all of which have the same dimension; *i.e.*, the dimension of sensor feature m. The parameters on the aligner is weights W ($m \times m$) and biases B on the hidden layers' weights such that

$$\tilde{x} = \sigma(W * x + B) \tag{1}$$

where σ is the activation function at the hidden layer and and $\tilde{x}$ denotes the output of the transformed vector of x.

The loss function of the aligner is the reconstruction error on the discriminator;

$$l_a = ||\tilde{x} - g(f(\tilde{x}))||^2 \tag{2}$$

The discriminator is implemented as an Autoencoder that is initialised with the same weight as the original Autoencoder. It takes as input $x \in D_t$ and aims to maximises the reconstruction error difference from D_{t_0}. The loss function of the discriminator is defined as:

$$l_d = -(||\tilde{x} - g(f(\tilde{x}))||^2 - ||x_{t_0} - g(f(x_{t_0}))||^2) \tag{3}$$

where x_{t_0} is a sensor feature from D_{t_0}. The idea to maximise the difference of the reconstruction error from $\tilde{x}$ to x_{t_0} is equivalent to minimise the difference of the reconstruction error from x_{t_0} to $\tilde{x}$. The rationale is that as the discriminator is initialised as the original Autoencoder, the reconstruction error on the original sensor feature x_{t_0} is small and that on the transformed feature $\tilde{x}$ is large. Reversing the error difference from x_{t_0} to $\tilde{x}$ is to minimise the error on the transformed feature $\tilde{x}$.

To direct the discriminator to explore intrinsic structure to separate the original and transformed features, we will apply the *mixup* technique [25] that mixes original and target data to further confuse the discriminator. Here a mixed feature vector x_m is a linear combination of the original feature x_{t_0} and the transformed feature $\tilde{x}$:

$$x_m = \lambda x_{t_0} + (1 - \lambda)\tilde{x} \tag{4}$$

Then the loss of the discriminator is updated as

$$l_d^m = -(||x_m - g(f(x_m))||^2 - ||x_{t_0} - g(f(x_{t_0}))||^2) \quad (5)$$

Figure 6 presents the workflow of our approach. It starts with training an Autoencoder and a classifier with D_{t_0}, and then training an aligner in an adversarial manner to match the distribution of new data D_t to that of original data D_{t_0}. For prediction, the new data will be transformed via an aligner and an Autoencoder into latent representations, which will be sent to the classifier for prediction.

V. Evaluation and Results

The objective of our evaluation is to assess the accuracy of localisation and activity recognition with our sensor recalibration approach.

A. Implementation and Configuration

We implement our model in PyTorch and Table I lists the hyperparameter settings for the Autoencoder, classifier, and aligner, for each of which we use Leaky ReLu as the activation function with a smoothing factor $\alpha = 0.2$ and SDG as the optimiser with the momentum being set as 0.9. We performed grid search on these hyperparamters and selected the setting that results in the highest accuracy.

TABLE I
Hyperparameter settings

Hypterparameter	Autoencoder	Classifier	Aligner
Neurons at each layer	196-64-196	64-12	196-196-196
Learning rate	0.001	0.001	0.0002
Cost function	MSE	Cross-entropy	MSE

We have performed grid search on these hyperparamters and selected the setting that results in the highest accuracy.

B. Sensor Alignment

Goal. We aim to assess if our alignment approach can improve the accuracy of human activity recognition without any extra labelling effort.

Methodology. We train our model with a previous day's labelled data and 80% of the current day's unlabelled data, and test the model on the remaining 20% of the current day's data. The accuracy is measured in micro-F1. The prediction target is location, which is the main interest of our DFWS system. The input to our technique is the normalised peak RSSI values on each tag. To demonstrate the effectiveness of our approach, we will compare with the other unsupervised domain adaptation techniques that consist of one traditional technique JDA [12] and two recent deep learning based techniques DAN [11] and ADDA [21].

Besides the above three baselines, we also compare the performance of variations of our algorithms. We have designed two loss functions on the discriminator: with and without mixed features respectively. This comparison assesses which loss leads to better discriminative alignment. We run each experiment 10 times and report the averaged accuracy and training time as an indication of computation cost.

TABLE II
Comparison of localisation accuracy after aligning one day's data to another day's data. The best accuracy is marked in bold and underlined text.

Source	Target	No Alignment	JDA	ADDA	DAN	Aligner	Aligner mixup
D0	D1	0.25	0.37	0.34	**0.58**	0.56	0.57
	D2	0.38	0.30	0.38	0.55	0.58	**0.65**
	D3	0.40	0.33	0.33	0.53	0.63	**0.64**
	D4	0.30	0.26	0.32	0.45	**0.57**	0.52
D1	D2	0.43	0.31	0.32	0.41	0.53	**0.66**
	D3	0.41	0.22	0.41	0.44	0.54	**0.67**
	D4	0.40	0.24	0.44	0.51	0.53	**0.61**
D2	D3	0.43	0.26	0.44	0.61	0.64	**0.66**
	D4	0.39	0.26	0.56	0.60	0.53	**0.65**
D3	D4	0.42	0.37	0.61	0.65	0.45	**0.66**
Average		0.38	0.29	0.42	0.53	0.56	**0.63**

Results. Table II reports the localisation accuracy by aligning the target's data to the source's data. It compares our two aligners with a baseline ('No Alignment') and three state-of-the-art techniques. As we can see, without alignment, by training on one day's data, and testing on a different day's data, we only get very low accuracy: with 0.25 as min, 0.43 as max, and 0.38 as mean averaged on 10 experiments. Our aligner and aligner with mixup achieve the highest accuracy for 9 out of 10 times. The aligner with mixup has the averaged accuracy of 0.63, which is 0.25 over the baseline and 0.10 over the best comparison technique; i.e., DAN.

Among the state-of-the-art techniques, we have observed negative transfer on JDA, where the accuracy after adaptation is lower than no alignment. ADDA achieves the averaged accuracy of 0.42, which is slightly above the baseline. DAN improves over the baseline with 0.15 gain in averaged accuracy. Our aligner techniques and DAN work much better than ADDA, demonstrating that matching reconstruction error difference or transforming raw features into sub-space representations can be more robust than matching raw feature distributions.

In terms of the computation cost, Figure 7 compares the training time of our aligner algorithms and comparison techniques. For our proposed approaches, training Autoencoder and classifier on the initial stage's data is 12 and 2 mins. Aligning only takes 1 min, which is much cheaper than the other comparison techniques; i.e., 52 in ADDA, 123 in JDA, and 8 in DAN. This shows that our techniques are more amenable for real-world deployment, which allows for faster alignment on a daily basis.

C. Sensitivity to Initial Training Data

Goal. Here is to assess how much initial data needs to be collected to enable robust sensor alignment.

Methodology. We perform two experiments. Firstly, we train our model with previous N days' labelled data and 80% of the unlabelled $N+1$th day's data, and then test on the remaining 20% of $N+1$th day's data. We vary N from 2 to 4. Secondly, we iterate each day as target and select M days' data from the remaining days as source to train our algorithms. We set M as 2 and 3, and average the accuracy.

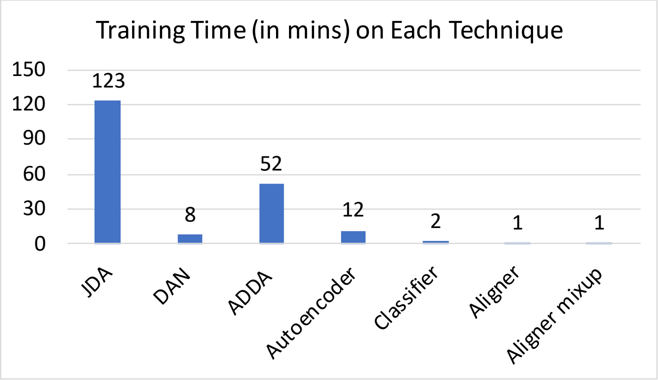

Fig. 7. Comparison of training time

Results. Table III compares the localisation accuracy between our aligners and baselines by training on multiple days' data. The results on the top panel are consistent with the above. When the source and target's data are similar, the aligner with mixup improves the recognition accuracy with more source training data.

TABLE III
LOCALISATION ACCURACY BY TRAINING ON MULTIPLE DAYS' DATA

Source	Target	No Alignment	JDA	DAN	ADDA	Aligner	Aligner mixup
D0+D1	D2	0.416	0.277	0.423	0.403	0.527	**0.599**
D0+D1+D2	D3	0.427	0.322	0.450	0.441	0.524	**0.613**
D0+D1+D2+D3	D4	0.523	0.428	0.550	0.461	0.530	**0.581**
2 DAYS	1 DAY	0.334	0.380	0.450	0.560	0.574	**0.665**
3 DAYS	1 DAY	0.477	0.370	0.430	0.610	0.590	**0.648**
Average		0.435	0.356	0.461	0.495	0.549	**0.621**

D. Sensitivity to Target Training data

Goal. In a real-world system, we might not be able to use 80% of current day's data for alignment. Here we evaluate the sensitivity of our algorithms to the percentage of training data on the target day.

Methodology. We reduce the unlabelled training data on the target day from 80% to 10% and compare the accuracy between the aligner mixup algorithm and the best comparison technique DAN.

Results. Figure 8 presents the accuracy of DAN and aligner mixup algorithm on different target training percentages, which is averaged on all the permutations in Table II. As we can see, from 80% to 10% both techniques drop the accuracy within the range of 10%. This demonstrates that our technique is less sensitive to the training data percentage than DAN.

E. Validation of Generality on Other Datasets

Goal. To validate our alignment approach on more general sensor noise problem, we also experiment the other state-of-the-art HAR datasets.

Methodology. We select the following 2 datasets that are collected in various lengths to demonstrate the effectiveness of our sensor alignment techniques. We use House A (HA) and B (HB) collected by University of Amsterdam [9]. These datasets collected binary sensor data on single-resident's daily activities including sleeping, leaving home, showering, and meal preparation over a couple of months. The sensors being used are RFM DM 1810, which has an digital and analog input. A sensor sends an event when the state of its digital input changes or when some threshold of its analog input is violated. The sensors are deployed on doors, cupboards, fridge, freezer, and microwave as a contact switch to detect whether these objects are open or close. As these sensors output binary readings; *i.e.*, when a sensor is activated, it produces 1, they do not produce continuous readings. There is no observable gradual change in readings, and sensor data are more susceptible to human interference; for example, dislodging a sensor from one position to another [8]. By experimenting on these datasets, we will demonstrate the generality of our approach in dealing with other types of sensors and errors. These two datasets have 14 and 22 sensor features with 505 and 497 instances collected over 5 and 4 weeks respectively. To do so, we split the data into weeks and perform weekly evaluation.

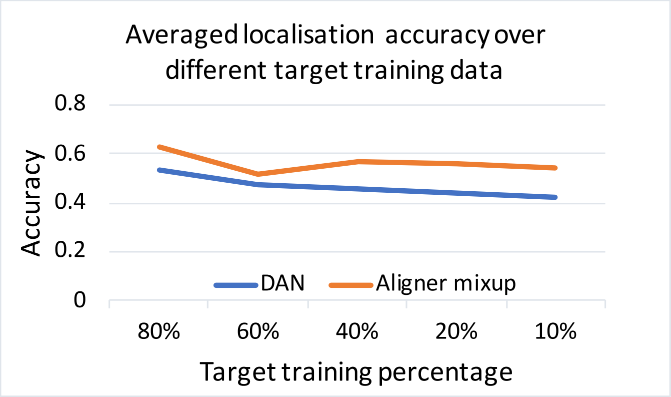

Fig. 8. Comparison of localisation accuracy when reducing the training percentage on target data

TABLE IV
ACCURACY OF ACTIVITY RECOGNITION ON HOUSE A AND B

Dataset	Source	Target	No Alignment	JDA	ADDA	DAN	Aligner	Aligner mixup
HA	W0	W1	0.5	0.16	0.42	**0.82**	0.75	0.63
		W2	0.51	0.16	0.1	**0.73**	0.64	0.69
		W3	0.5	0.26	0.35	**0.9**	0.81	**0.9**
	W1	W0	0.49	0.11	0.18	0.62	**0.84**	0.82
		W2	0.53	0.19	0.52	0.65	**0.72**	0.66
		W3	0.5	0.29	0.75	0.69	0.87	**0.9**
	W2	W0	0.35	0.11	0.18	0.51	0.83	**0.83**
		W1	0.36	0.19	0.46	0.76	**0.85**	0.76
		W3	0.36	0.29	0.75	0.64	0.85	**0.9**
	W3	W0	0.56	0.16	0.18	0.74	**0.85**	**0.85**
		W1	0.5	0.24	0.46	0.75	**0.83**	0.75
		W2	0.6	0.24	0.52	0.7	0.75	**0.86**
HB	W0	W1	0.41	0.13	0.58	0.21	0.32	**0.76**
		W2	0.68	0.83	0.9	0.75	**0.94**	**0.94**
	W1	W0	0.1	0.79	0.9	0.75	0.41	**0.94**
		W2	0.12	0.79	**0.87**	0.66	0.86	0.71
	W2	W0	0.1	0.13	0.58	0.05	0.49	**0.8**
		W1	0.24	0.74	**0.87**	0.48	0.86	0.71
Average			0.41	0.32	0.53	0.63	0.75	**0.80**

Results. Table IV compares the activity recognition accuracy before and after alignment with different techniques. For these two datasets, we have done all the permutations; i.e., each week will be treated as a source and any other week as a target. As shown in Table IV, our alignment techniques have achieved nearly twice accuracy as 'No Alignment'; that is, 0.80 vs. 0.41. Again this demonstrates the need for sensor alignment to improve activity recognition over the long-term deployment. Without constant alignment, sensor readings might deviate from true readings and significantly compromise the recognition accuracy. In comparison with other techniques, our alignment techniques achieve 0.17 gain over DAN.

Between our two aligners, the results are similar, the mixup representations can further improve the overall accuracy of 0.05 and significantly improve the accuracy where there is much larger discrepancy between source and target data. For example, on House B, taking week 1 or week 2 as source domain, when aligning week 0's data with them, the baseline only gets the accuracy of 0.1, which suggests high dissimilarity. The aligner technique that only considers reconstruction error difference has not been able to bridge the gap, but by mixing up with both source and target samples, the accuracy has significantly improved; i.e., 0.94 vs. 0.41 and 0.8 vs. 0.49.

VI. Conclusion and Future Work

This paper presents a label-free RFID sensor recalibration technique to support long-term deployment. Through extensive experiments, we have shown that our alignment techniques can robustly match the current data to the previous data and significantly improve the localisation accuracy with affordable computational cost. In the future, we will extend our approach with semi-supervised learning to improve the recognition accuracy by querying labels for selected samples from target data. Also in our current design, we fix the Autoencoder and classifier once being trained, in order to reduce the cost of updating both of them on a daily basis. However, over time, if the sensor data distribution has significantly deviated or the activity routine has changed, it'd be desirable to update both Autoencoder and classifiers to accommodate such changes. Therefore, the future direction will be how to update them in a cost-effective manner.

References

[1] T. Artursson, T. Eklöv, I. Lundström, P. Mårtensson, M. Sjöström, and M. Holmberg. Drift correction for gas sensors using multivariate methods. *Journal of chemometrics*, 14(5-6):711–723, 2000.

[2] D. Arumugam and D. Engels. Specific absorption rates in the human head and shoulder for passive UHF RFID systems. *International Journal of Radio Frequency Identification Technology and Applications*, 2(1-2):1–26, 2009.

[3] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives, 2014.

[4] A. Farshchian, J. A. Gallego, J. P. Cohen, Y. Bengio, L. E. Miller, and S. A. Solla. Adversarial domain adaptation for stable brain-machine interfaces. In *ICLR 2019*, 2019.

[5] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML' 15*, page 1180–1189. JMLR.org, 2015.

[6] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17(1):2096–2030, Jan. 2016.

[7] Impinj Inc. Speedway Revolution Reader—Low Level User Data Support. Technical report, Impinj Inc., 2013.

[8] K. Kapitanova, E. Hoque, J. A. Stankovic, K. Whitehouse, and S. H. Son. Being smart about failures: Assessing repairs in smart homes. Association for Computing Machinery, 2012.

[9] T. L. M. Kasteren, G. Englebienne, and B. J. A. Kröse. Human activity recognition from wireless sensor network data: Benchmark and software. In L. Chen, C. D. Nugent, J. Biswas, and J. Hoey, editors, *Activity Recognition in Pervasive Intelligent Environments*, volume 4 of *Atlantis Ambient and Pervasive Intelligence*, chapter 8, pages 165–186. Atlantis Press, Paris, France, 2011.

[10] D. Kumar, S. Rajasegarar, and M. Palaniswami. Automatic sensor drift detection and correction using spatial kriging and kalman filtering. In *2013 IEEE International Conference on Distributed Computing in Sensor Systems*, pages 183–190, 2013.

[11] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. In *ICML'15*, ICML'15, page 97–105. JMLR.org, 2015.

[12] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu. Transfer feature learning with joint distribution adaptation. In *ICCV 2013*, ICCV '13, page 2200–2207, USA, 2013. IEEE Computer Society.

[13] G. A. Oguntala, R. A. Abd-Alhameed, N. T. Ali, Y. Hu, J. M. Noras, N. N. Eya, I. Elfergani, and J. Rodriguez. Smartwall: Novel rfid-enabled ambient human activity recognition using machine learning for unobtrusive health monitoring. *IEEE Access*, 7:68022–68033, 2019.

[14] N. S. V. Rao, C. Greulich, P. Ramuhalli, S. M. Cetiner, and P. Devineni. Sensor drift estimation for reactor systems by fusing multiple sensor measurements. In *NSS/MIC 2019*, pages 1–2, 2019.

[15] W. Ruan, L. Yao, Q. Z. Sheng, N. Falkner, X. Li, and T. Gu. Tagfall: Towards unobstructive fine-grained fall detection based on uhf passive rfid tags. In *MOBIQUITOUS'15*, page 140–149, 2015.

[16] U. Saeed, S. Ullah Jan, Y. Lee, and I. Koo. Machine learning-based real-time sensor drift fault detection using raspberry pi. In *ICEIC' 20*, pages 1–7, 2020.

[17] A. R. Sanabria, T. W. Kelsey, and J. Ye. Representation learning for minority and subtle activities in a smart home environment. In *PerCom 2019*, pages 1–7, 2019.

[18] R. L. Shinmoto Torres, A. Wickramasinghe, V. N. Pham, and D. C. Ranasinghe. What if your floor could tell someone you fell? a device free fall detection method. In *Artificial Intelligence in Medicine*, 2015.

[19] R. Smith, Y. Ding, G. Goussetis, and M. Dragone. A COTS (UHF) RFID Floor for Device-Free Ambient Assisted Living Monitoring. In *ISAmI 2020*, Oct. 2020.

[20] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[21] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *CVPR 2017*, pages 2962–2971, 2017.

[22] J. Wang, L. Chang, O. Abari, and S. Keshav. Are rfid sensing systems ready for the real world? In *MobiSys '19*, pages 366–377, 2019.

[23] J. Wang, Q. Gao, M. Pan, and Y. Fang. Device-free wireless sensing: Challenges, opportunities, and applications. *IEEE Network*, 32(2):132–137, 2018.

[24] Q. Wang, H. Qi, and F. Liu. Time series prediction of e-nose sensor drift based on deep recurrent neural network. In *2019 Chinese Control Conference (CCC)*, pages 3479–3484, 2019.

[25] M. Xu, J. Zhang, B. Ni, T. Li, C. Wang, Q. Tian, and W. Zhang. Adversarial domain adaptation with domain mixup. In *AAAI 2020*, pages 6502–6509. AAAI Press, 2020.

[26] K. Yan and D. Zhang. Correcting instrumental variation and time-varying drift: A transfer learning approach with autoencoders. *IEEE Transactions on Instrumentation and Measurement*, 65(9):2012–2022, 2016.

[27] L. Yao, Q. Z. Sheng, X. Li, T. Gu, M. Tan, X. Wang, S. Wang, and W. Ruan. Compressive representation for device-free activity recognition with passive rfid signal strength. *IEEE Transactions on Mobile Computing*, 17(2):293–306, 2018.

[28] J. Ye, G. Stevenson, and S. Dobson. Detecting abnormal events on binary sensors in smart home environments. *Pervasive and Mobile Computing*, 33:32 – 49, 2016.