Received January 11, 2021, accepted January 19, 2021, date of publication January 22, 2021, date of current version February 2, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3053704

Unsupervised Domain Adaptation in Activity Recognition: A GAN-Based Approach

ANDREA ROSALES SANABRIA[1], FRANCO ZAMBONELLI[2], (Fellow, IEEE), AND JUAN YE[1]

[1]School of Computer Science, University of St Andrews, St Andrews KY16 9SX, U.K.
[2]Dipartimento di Scienze e Metodi dell'Ingegneria, Università di Modena e Reggio Emilia, 42122 Modena, Italy

Corresponding author: Juan Ye (juan.ye@st-andrews.ac.uk)

The work was supported by the Italian MIUR, PRIN 2017 Project "Fluidware," under Grant N. 2017KRC7KT.

ABSTRACT Sensor-based human activity recognition (HAR) is having a significant impact in a wide range of applications in smart city, smart home, and personal healthcare. Such wide deployment of HAR systems often faces the annotation-scarcity challenge; that is, most of the HAR techniques, especially the deep learning techniques, require a large number of training data while annotating sensor data is very time- and effort-consuming. Unsupervised domain adaptation has been successfully applied to tackle this challenge, where the activity knowledge from a well-annotated domain can be transferred to a new, unlabelled domain. However, these existing techniques do not perform well on highly heterogeneous domains. This article proposes *shift*-GAN that integrate bidirectional generative adversarial networks (Bi-GAN) and kernel mean matching (KMM) in an innovative way to learn intrinsic, robust feature transfer between two heterogeneous domains. Bi-GAN consists of two GANs that are bound by a cyclic constraint, which enables more effective feature transfer than a classic, single GAN model. KMM is a powerful non-parametric technique to correct covariate shift, which further improves feature space alignment. Through a series of comprehensive, empirical evaluations, *shift*-GAN has not only achieved its superior performance over 10 state-of-the-art domain adaptation techniques but also demonstrated its effectiveness in learning activity-independent, intrinsic feature mappings between two domains, robustness to sensor noise, and less sensitivity to training data.

INDEX TERMS Human activity recognition, domain adaptation, ensemble learning, generative adversarial networks, covariate shift, kernel mean matching.

I. INTRODUCTION

In recent years, the drastic increase in ageing population has increased the burden on the already over-stretched health and social care systems. Technology-based personal healthcare systems have demonstrated their potential in providing cost-effective support for the elderly people to enable them to lead an independent, high-quality life while avoiding institutionalised care. Sensor-based human activity recognition (HAR) is about inferring people's daily activities from a set of unobtrusive sensors, including motion sensors, switch sensors, RFID, and accelerometers and gyroscopes embedded in wearables. It is the key enabler for healthcare systems, providing a way of tracking people's health conditions, detecting early sign of disease, and providing personalised healthcare service. In addition, HAR is heavily adopted in many other applications, including smart homes [22], robotics [31], gaming [1], and urban computing [49].

Because of its potential, HAR has been extensively studied and numerous prototypes and testbeds have been built over the years [8]. With the advance in machine learning and especially in deep learning techniques [33], we can fairly accurately recognise activities from a collection of sensor data. However, the main challenge still resides in the lack of training data. One type of approaches to tackle annotation challenge is unsupervised domain adaptation, where knowledge learned from a source domain (with labelled data) can be transferred to a target domain (without labelled data) [30].

Domain adaptation techniques have been increasingly applied to HAR applications; for example on accelerometer data, much effort has been devoted to transferring activity knowledge learned on one sensor position (e.g., arm) to another position (e.g., leg) [4], [6], [20]. These techniques often work well as they assume that the source and target

The associate editor coordinating the review of this manuscript and approving it for publication was Zhiwei Gao.

domains share feature space and they only need to tackle the difference in distributions. However, this assumption does not hold for many other HAR applications where two domains have heterogeneous feature spaces. For example on binary sensor data, often sensor data from two different environments reside in disparate feature spaces, it is necessary to have an intermediate way to bridge source and target feature spaces via knowledge [47] or meta-features [10]. These techniques often incur extra engineering effort.

To tackle the feature space heterogeneity between source and target domains, we propose *shift*-GAN, where we re-purpose and extend a Bi-directional Generative Adversarial Network (Bi-GAN) [48], [52] from the area of image-to-image translation to enable robust domain adaptation.

Bi-GAN allows transforming from one heterogeneous feature space to another and vice versa through two GAN models. Compared to general adversarial adaptation model where a single GAN is employed and a generator aims to transform one feature space to another, Bi-GAN enables better translation with stronger, bi-directional, constraints. Given the effectiveness shown by Bi-GAN in image-to-image translation, we decide that it is worth evaluating an adaptation and extension of the Bi-GAN architecture, specifically tailored to the HAR problem to see if it is more effective than the exiting techniques.

Since sensor data are often imperfect and noisy, and heterogeneity between sensor datasets can be much larger than that between images, which makes it challenging to generate data that well matches to the target domain. To improve the matching, we employ Kernel Mean Matching (KMM) [28] to enable covariate shift correction between transformed source data and original target data so that they can be better aligned and thus lead to a more accurate predictive model on the target dataset.

The key novelty of *shift*-GAN is to integrate Bi-GAN and KMM to learn intrinsic, robust transfer between two domains, which are activity-invariant and thus can be generalised to new activities that are unobserved at the training time. That is, without re-training, *shift*-GAN will still be able to translate sensor features on the new activity to the target domain. This allows continuous activity transfer over time, which is a significant benefit to activity recognition. To the best of our knowledge, we are the first to adapt Bi-GAN with kernel mean matching to tackle heterogeneous feature space issue in unsupervised domain adaption. Different from the state-of-the-art adaptation techniques in HAR, our technique can work well on both accelerometer and binary sensor data. Furthermore, we have specifically looked into the impact of sensor noise on the effectiveness of adaptation and study the generalisation of feature transferring in Bi-GAN.

The main contributions of this article are summarised as follows.

- We propose *shift*-GAN as a general unsupervised domain adaptation technique to enable activity transfer across heterogeneous datasets, including accelerometer and binary sensors.
- We have extended Bi-GAN by not just performing one-to-one instance translation but one-to-many instance translation along with instance selection process to allow more robust domain adaptation.
- We have validated *shift*-GAN extensively on 12 transfer tasks across 5 datasets. All these datasets feature different sensor deployments, spatial layouts of environments, and different end users. Our results have demonstrated that *shift*-GAN has outperformed 10 classic and deep domain adaptation techniques.
- We also are the first to go beyond domain adaptation and design and perform other HAR specific experiments on sensor noise, sensitivity to training data, and generalisation to new activities. These experiments matter a great deal in HAR and many other real-world applications where noisy data are pervasive, training data is scarce, and new classes are constantly emerging. The results have demonstrated the superior performance of *shift*-GAN over the state-of-the-art techniques in these experiments.

II. RELATED WORK

This section reviews the existing domain adaptation techniques [7], [43] on accelerometer and binary event sensor data in HAR applications, and compare and contrast with *shift*-GAN to identify our contributions. To broaden our scope, we also look into the general field of domain adaptation, including supervised, unsupervised domain adaptation and domain generalisation.

A. DOMAIN ADAPTATION IN HUMAN ACTIVITY RECOGNITION

There have been quite a few of attempts of transfer learning on accelerometer data; e.g., from one user to another [51], from one body position (e.g., chest) to another (e.g., hips) [6], and from one device to another [18]. As accelerometer data share the same dimensions; i.e., timestamp and x-, y-, and z-dimension, generated feature spaces can be uniform, as long as they are using the same feature extraction technique. Thus, the focus here is to align the distributions, rather than transfer feature spaces.

STL (Stratified Transfer Learning) [6] has been proposed to perform within-activity transfer; that is, it learns an activity model on a labelled dataset and then uses the model to generate pseudo labels on an unlabelled dataset. The knowledge transfer is performed per activity class by using multiple transfer kernels to project the source and target domain's feature spaces to a common subspace. Qin *et al.* [32] propose Adaptive Spatial-Temporal Transfer Learning (ASTTL) to allow more accurate source selection to perform domain adaption. Chang *et al.* [4] have looked into feature matching and adversarial learning in adapting the activity model from one sensor position to another. These recent techniques are built on a similar assumption that both source and target domains share the same feature space so that they can share the same activity model [6] or feature extractor [4], [32].

Domain adaptation on binary sensor data is different from accelerometer data as the main challenge is to tackle heterogeneity of feature spaces. In general, most sensorised environments have a different spatial layout and a different sensor deployment; for example, a variety collection of sensors being deployed in areas and on objects, depending on the requirements and constraints of the environments and their residents. Therefore, it often requires some methods to match and align feature spaces from the source to the target domain. Ontologies have been a popular approach that leverages the semantics of sensor placement such as location, which enables meaningful sensor feature transfer [34], [40], [46]. For example, Rosales and Ye [36] propose to a 2-staged domain adaptation where semantics similarity is employed to perform linear transformation of sensor features from one domain to another domain and then a variational autoencoder (VAE) is used for fine alignment between transferred features and source features. Other than semantics, Feuz and Cook [10] map feature spaces via meta-features on each sensor; that is, the time a sensor being activated, and intervals and sequence between sensor activation. These approaches have achieved promising results in resolving heterogeneity between feature spaces but they require extra effort to craft the knowledge [34], [36], [40], [46] and learn meta-mapping [10]. Also the effectiveness of these approaches is significantly subject to the reliability of such knowledge [36].

Different from the above techniques, *shift*-GAN enables unsupervised domain adaptation on both accelerometer and binary sensor data. It does not assume homogeneity of feature space, and does not require extra engineering effort. *shift*-GAN takes advantages of image-to-image translation techniques and covariate shift in a staged domain adaptation process to best align feature spaces.

B. DOMAIN ADAPTATION IN MACHINE LEARNING

Domain adaptation has long been an important topic in the area of machine learning [38], [53], which can be categorised into *supervised* domain adaptation where labels on the target domain are available, *unsupervised* domain adaptation where labels on the target domains are not available, and *domain generalisation* where a domain agnostic model is generalised by learning from multiple domains. Our technique falls into the category of unsupervised domain adaption. In the following, we will briefly introduce representative approaches in the other categories and then focus on unsupervised domain adaption.

1) SUPERVISED DOMAIN ADAPTATION

As labels are available in supervised domain adaption, it is possible to perform within-class adaptation. For example, Xu *et al.* [45] propose d-SNE where samples from both source and target domains are transformed to common latent space; i.e., stochastic neighborhood embedding (SNE) space, and then a modified Hausdoff distance is employed to minimise the distance between samples from the same classes but maximise the distance between samples from different classes. Morsing *et al.* [26] propose to deal with covariate shift by connecting samples in a penalty graph structure.

2) DOMAIN GENERALISATION

A classic approach in domain generalisation is to combine training samples from different source domains to train a classifier and regulate the weights of the classifier for an unseen target domain. CCSA (Classification and Contrastive Semantic Alignment) is one of the first deep learning techniques that tackle both domain adaptation and generalisation. It uses contrastive loss to encourage samples with the same class labels from different domains to be close in the embedding space [27]. Li *et al.* [21] employ an adversarial autoencoder to align distributions from different domains where Maximum Mean Discrepancy (MMD) is used to minimise the difference in distributions.

3) UNSUPERVISED DOMAIN ADAPTATION

Feature transformation is a classic type in unsupervised domain adaptation, which maps the features of the source and target domain into a high-dimensional space. Previous work [29] has demonstrated that finding *good* feature representations can help reduce the difference in distributions between domains. Pan *et al.* [29] have proposed to find such representation through *transfer component analysis* (TCA). TCA learns *transfer components* across domains in a Reproducing Kernel Hilbert Space (RKHS) using MMD. Grauman [12] have proposed to minimise the distance between the source and target domains with a kernel-based method called *geodesic flow kernel* (GFK) that integrates an infinite number of subspaces to represent the geometric changes and statistical properties from the source to the target domain. The above feature transformation methods require that feature distributions of the source and target domain share a common support. However, this condition is rarely met in real world adaptation tasks.

Deep Adaptation Network (DAN) [24] embeds the hidden representations of the task-specific layers of a CNN in RKHS and explicitly matches the mean embeddings of source and target domain distributions. Similarly, Long *et al.* [23] propose to project deep features from all the task-specific layers into RKHS and use multi-kernel learning to match the embeddings.

Domain-Adversarial Neural Network (DANN) [11] is proposed to learn domain-invariant features by combining domain adaptation with feature learning. The distribution alignment between two domains is achieved through standard backpropagation training. Tzeng *et al.* [39] have proposed an unsupervised adversarial adaptation method called Adversarial Discriminative Domain Adaptation (ADDA) that learns a discriminative representation using the labels in the source domain and builds an asymmetric mapping learned through a domain-adversarial loss to map the target data to the source representations. Saito *et al.* [37] employ a task-specific classifier as a discriminator to consider the relationship between target samples and class decision boundaries when aligning distributions. Zhao *et al.* [50] have proposed multi-source distillation domain adaptation that first adversarially maps the target domain into each source domain and selects the source training samples that are close to the target domain to fine-tune the source classifier. Then the improved source classifiers will classify the mapped target samples, and the prediction results will be aggregated for a final prediction.

Chen *et al.* have designed a Re-weighted Adversarial Adaptation Network (RAAN) for unsupervised domain adaptation to reduce disparate domain discrepancies and adapt the classifier [5]. First, they train a domain discriminator network together with a deep convolutional neural network in an adversarial manner to minimise the optimal transformation based on EM distance. To adapt the classifier, the label distribution is matched by estimating a re-weighted source domain label distribution. Adversarial Domain Adaptation with Domain Mixup (ADADM) [44] advances adversarial learning by mixing transformed source and real target domain samples to train a more robust generator.

shift-GAN also employs adversarial adaptation and the difference is that we use bi-directional GANs from image-to-image translation: generating target features from source and vice versa. We hypothesise that by imposing the bi-directional constraints *shift*-GAN can enable more reliable, generalised feature transformation than a single GAN based adversarial adaptation.

III. SHIFTED UNSUPERVISED DOMAIN ADAPTATION

shift-GAN is proposed as an unsupervised domain adaptation technique for resolving heterogeneous feature space between source and target domain. It has two stages: (1) we first perform bi-directional domain transformation via GANs [48], and then (2) perform covariate shift correction to align transformed target features with original source features. We hypothesise that through these two steps we will achieve more robust, effective feature transformation and lead to improved activity recognition on the target domain. In the following, we will introduce the research problem and challenges *shift*-GAN tackles and then we detail each component of *shift*-GAN.

A. PROBLEM DEFINITION

Let $\mathcal{D}_s = \{(\vec{x}_s^{(i)}, y_s^{(i)})\}_{i=1}^{N_s}$ be the labelled source domain and $\mathcal{D}_t = \{\vec{x}_t^{(i)}\}_{i=1}^{N_t}$ be the unlabelled target domain, where $\vec{x}_s(\in \mathbb{R}^{M_s})$ and $\vec{x}_t(\in \mathbb{R}^{M_t})$ is a M_s and M_t-dimensional feature vector, and M_s can be different from M_t. Both domains share the same label space $\mathcal{Y}$. *shift*-GAN aims to perform adaptation between $\mathcal{D}_s$ and $\mathcal{D}_t$ with the objective to predict labels for all the instances in $\mathcal{D}_t$.

We will illustrate the above definition through an example in Figure 1. Assume that there are two sensorised house settings (i.e., source and target) having different spatial layouts and installed with different sensors (as marked in red dots) [17]. The sensor data collected on these two houses are different and so are the sensor features extracted. The activity set $\mathcal{Y}$ to predict can be the same; i.e., a common set of daily activities such as preparing breakfast and performing personal hygiene.

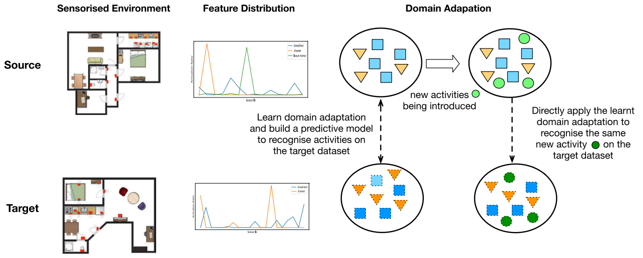


FIGURE 1. A use case [17] of generalised unsupervised domain adaptation.

shift-GAN aims to learn a feature space transformation function $g_{s\rightarrow t}$ that maps the source domain features into the target domain features; i.e., $g_{s\rightarrow t}(\vec{x}_s) = \tilde{\vec{x}}_t$. Then, we can build a classifier with transformed data $\{(\tilde{\vec{x}}_t^{(i)}, y_s^{(i)})\}_{i=1}^{N_s}$, with which we can predict labels on real target data D_t. For example in Figure 1, given the source domain's data are annotated with the activities of having breakfast and taking shower, *shift*-GAN will be able to recognise the same activities on the target domain data by learning the transformation functions between these two datasets.

We consider the transformation function $g_{s\rightarrow t}$ is *generalised* or activity-invariant, if it can be applied to sensor data on emerging, new activities that have not been observed in $\mathcal{D}_s$. Let $D'_s = \{(\vec{x}_s'^{(i)}, y'^{(i)})\}_{i=1}^{N'_s}$ be a new collection of labelled source domain's data, which has the same feature space as $\mathcal{D}_s$ but has a different label space; that is, $y_s'^{(i)} \in \mathcal{Y}'_s$ and $\mathcal{Y}'_s \cap \mathcal{Y}_s = \emptyset$. The transformation function is regarded *intrinsic* to features, independent of specific activity classes if $g_{s\rightarrow t}(\vec{x}'_s)$ still holds on the new data $\mathcal{D}'_s$ without the need of retraining; i.e., $g_{s\rightarrow t}(\vec{x}'_s) = \tilde{\vec{x}}'_t$. In Figure 1, if the source domain's data are annotated with a new activity 'leaving home', we can use the function to transform the source domain data on this new activity to the target domain, without the need of retraining the function.

B. OVERVIEW OF shift-GAN

shift-GAN consists of the following four steps:

1) *Feature space transformation* – perform unsupervised feature space transformation between source and target datasets with GAN; that is, we learn the mapping function $g_{s\rightarrow t}$ and obtain $\tilde{X}_t = g_{s\rightarrow t}(X_s)$.
2) *Feature distribution alignment* – shift the transformed features $\tilde{X}_t$ towards the real target data X_t; that is, $\bar{X}_t = \beta\tilde{X}_t$, where $\beta = [\beta_1, \beta_2, \ldots, \beta_N]$, N is the size of transformed samples $\tilde{X}_t$, and β_i is a weighting factor on each transformed sample, satisfying $P(X_t) = \beta P(\tilde{X}_t)$.
3) *Classifier training* – train a classifier on the aligned, transformed features $\bar{X}_t$ and their corresponding labels inherited from the source domain.
4) *Prediction* – use the trained classifier to predict labels on the data in the target domain.

C. FEATURE SPACE TRANSFORMATION VIA GAN

Now we introduce the basic GAN concepts and detail the characteristics of our Bi-GAN architecture, inspired by other works on bi-directional GANs [15], [52], and specifically inspired by Bi-GAN [48]. In the following, we will first present a short recap of various Bi-GAN architectures and illustrate why Bi-GAN in [48] is best suited to our purposes.

1) OVERVIEW OF BI-GAN ARCHITECTURES

GAN has been widely applied in domain adaptation, and a generator aims to takes target samples as input and generate source samples and a discriminator aims to tell whether the generated samples are from real source domain [39]. In recent years, the research in GAN has well advanced and several coupled GAN architectures have been proposed in domain adaptation and image-to-image translation [48], [52]. For example, DupGAN [15] learns domain-invariant representation via an encoder, a generator, and two discriminators. The encoder aims to encode samples from both domains into a latent space, a conditional generator decodes latent representations back into source and target domains conditioned on the domain code, and discriminators on each domain to tell whether a sample is from the specific domain or generated. However this approach assumes both source and target domain shares the same feature space, due to the design on the encoder. Bi-directional GAN [48], [52], originated in image-to-image translation, unpairs two GANs to enforce cycle (or bi-directional) consistency between source and target domains; that is, making sure each image can be recovered through two generators' operation. This approach has achieved promising performance and does not assume the same feature space between source and target domains. Therefore, we will base our approach on this architecture. In the following, we will present the preliminaries of GAN and Bi-GANs.

2) GENERATIVE ADVERSARIAL NETWORK (GAN)

The idea behind GAN is to train two models – a generator and a discriminator – in an adversarial process. The generator G takes as input a random noise vector z and uses a multilayer perceptron with $\theta^{(G)}$ as parameters such as weights and biases. The discriminator D estimates the probability of a given sample coming from a real dataset. It takes as an input x and uses another multilayer perceptron with $\theta^{(D)}$ parameters. The models are represented by two functions, each of which is differentiable both with respect to its inputs and parameters.

The two models compete against each other during the training process: the generator G is trained to capture the real data distribution so that its generative samples can be as real as possible. While, the discriminator D is trained to maximise the probability of assigning the correct label to both training examples and generated samples from G. In other words, D and G are playing a **minimax game**. The discriminator wishes to minimise $J^D(\theta^{(D)}, \theta^{(G)})$ while controlling only $\theta^{(D)}$. The generator wishes to minimise $J^G(\theta^{(D)}, \theta^{(G)})$ while controlling only $\theta^{(G)}$. Their interaction can be summarised in the following loss function. Let P_d be the original data's distribution, P_g be the generator's distribution, and P_z be the noise variable z's distribution.

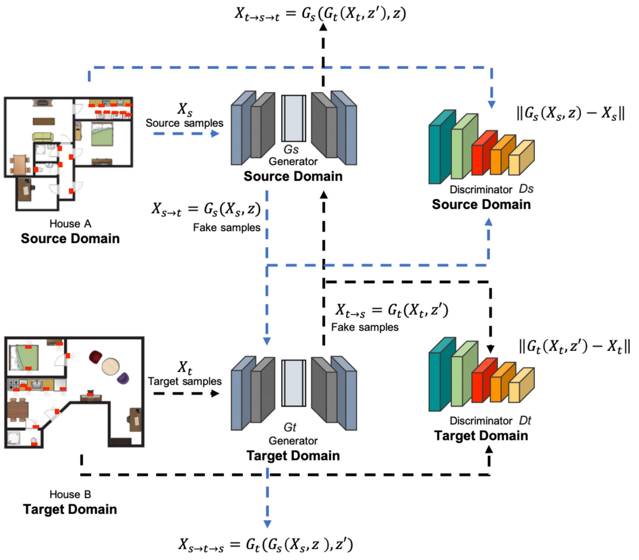


FIGURE 2. The workflow of Bi-GAN.

$$
\begin{aligned}
&\min_G \max_D \mathrm{L(D,G)} \\
&\quad = \mathbb{E}_{x\sim P_d}[\log D(x)] + \mathbb{E}_{z\sim P_z}[\log(1 - D(G(z)))] \\
&\quad = \mathbb{E}_{x\sim P_r}[\log D(x)] + \mathbb{E}_{x\sim P_g}[\log(1 - D(x))], \qquad (1)
\end{aligned}
$$

where $\mathbb{E}_{x\sim P_d}[\log D(x)]$ corresponds to the log-likelihood of maximising the probability of assigning the correct label, and $\mathbb{E}_{x\sim P_g}[\log(1 - D(x))]$ represents the log-likelihood of generating samples as real as possible.

3) BI-DIRECTIONAL GAN (BI-GAN)

Figure 2 describes the architecture of Bi-GAN. The Bi-GAN model consists of two GANs: $\{G_s, D_s\}$ and $\{G_t, D_t\}$, each composed of a generator and a discriminator on the source and target domain respectively. $G_s(\vec{x}_s) = \tilde{\vec{x}}_t$ takes a source instance $\vec{x}_s$ and generates a corresponding instance $\tilde{\vec{x}}_t$ in the target domain. $G_t(\vec{x}_t) = \tilde{\vec{x}}_s$ takes a target instance $\vec{x}_t$ and generates a corresponding instance $\tilde{\vec{x}}_s$ in the source domain. Both generators are trained to generate fake samples as close as to the real samples in the other domains and their objective function is to minimise the reconstruction losses:

$$
\begin{aligned}
\mathcal{L}_s^g &= ||G_t(G_s(\vec{x}_s, z), z') - \vec{x}_s||, \qquad (2) \\
\mathcal{L}_t^g &= ||G_s(G_t(\vec{x}_t, z'), z) - \vec{x}_t||, \qquad (3)
\end{aligned}
$$

where z and z' are random noise introduced in G_s and G_t.

The discriminator D_s is a binary classifier to detect whether an input is generated by G_s or a real sample from the target domain, and D_t is to detect whether an input is generated by G_t or a real sample from the source domain. Their loss functions are defined as:

$$
\begin{aligned}
\mathcal{L}_s^d &= D_s(G_s(\vec{x}_s, z)) - D_s(\vec{x}_t), \qquad (4) \\
\mathcal{L}_t^d &= D_t(G_t(\vec{x}_t, z')) - D_t(\vec{x}_s) \qquad (5)
\end{aligned}
$$

Algorithm 1 Bi-GAN Training [48]

```
Data: Unlabelled source domain $\mathcal{D}_s = \{\vec{x}_s^{(i)}\}_{i=1}^{N_s}$ and
      unlabelled target domain $\mathcal{D}_t = \{\vec{x}_t^{(i)}\}_{i=1}^{N_t}$
Build two generators $G_A$ and $G_B$ and two discriminators
$D_A$ and $D_B$
repeat
    foreach iteration do
        sample $L$-sized instances from both $\mathcal{D}_s$ and $\mathcal{D}_t$;
        $\{\vec{x}_s^{(j)}\}_{j=1}^{L} \subseteq \mathcal{D}_s$ and $\{\vec{x}_t^{(j)}\}_{j=1}^{L} \subseteq \mathcal{D}_t$
        update the parameters on $D_s$ to minimise
        $\frac{1}{L}\sum_{j=1}^{L} \mathcal{L}_s^d(\vec{x}_s, \vec{x}_t)$
        update the parameters on $D_t$ to minimise
        $\frac{1}{L}\sum_{j=1}^{L} \mathcal{L}_t^d(\vec{x}_t, \vec{x}_s)$
    end
    sample $L$-sized instances from both $\mathcal{D}_s$ and $\mathcal{D}_t$;
    $\{\vec{x}_s^{(j)}\}_{j=1}^{L} \subseteq \mathcal{D}_s$ and $\{\vec{x}_t^{(j)}\}_{j=1}^{L} \subseteq \mathcal{D}_t$
    update the weights on both generators to minimise
    $\frac{1}{L}\sum_{j=1}^{L} \mathcal{L}^g(\vec{x}_s, \vec{x}_t)$
until converge;
```

The combined loss function on both generators and discriminators is:

$$\begin{aligned}\mathcal{L}^g(\vec{x}_s, \vec{x}_t) = {} & \lambda_s||G_t(G_s(\vec{x}_s, z), z') - \vec{x}_s|| \\ & +\lambda_t||G_s(G_t(\vec{x}_t, z'), z) - \vec{x}_t|| \\ & -D_t(G_t(\vec{x}_t, z')) - D_s(G_s(\vec{x}_s, z)) \qquad (6)\end{aligned}$$

Algorithm 1 adopts the same training process of Bi-GAN [48], which does not need labels on neither source nor target domains. At the end of training, both generators act as the mapping functions. Bi-GAN enables to transform examples from the source domain to the target domain. In principle, the generator can generate many instances on each source example. The quality of each instance can vary due to the random variable z. In image-to-image application, human experts can visually inspect the images and perform the selection process. However, this practice is infeasible for sensor data generation, so we extend one-to-many instance generation and selection process.

For a given source example $\vec{x}_s$, we use G_s to generate N number of target samples, calculate their reconstruction loss using Eq (2) (3), and order them in an ascending order. Then we select the top-k ($1 \leq k \leq N$) samples that have the smallest reconstruction loss. The rationale is to choose the best transformed samples for the target domain while covering the diverse feature space by using k samples. In the end, we will have $\tilde{X}_t = \{\tilde{\vec{x}}_t^{(j)}\}_{j=1}^{N_s'}$ ($N_s' = k * N_s$), where $\tilde{\vec{x}}_t^{(j)} = G_s(\vec{x}_s^{(i)})$ ($1 \leq i \leq N_s$ and $1 \leq j \leq N_s'$).

D. COVARIATE SHIFT CORRECTION VIA KERNEL MEAN MATCHING

The transformed examples $\tilde{X}_t$ might still not reflect the true target data distribution. To better align the distribution, we are looking into Kernel Mean Matching (KMM), which is designed as a non-parametric distribution matching method between training and testing samples. KMM reweights the training examples such that the means of the training and testing examples when projected in a Reproducing Kernel Hilbert Space (RKHS) are close. In this way, the training data will be better aligned with testing data, leading to improved classification accuracy [28]. KMM has been successfully applied with GAN to control the image generation process [16]. Inspired by the promising results, we will apply KMM to shift feature distributions to improve classification accuracy. In the following, we will briefly introduce the theoretical background of KMM and illustrate how it is integrated in *shift*-GAN.

The idea of KMM is to assign each instance in generated target domain data $\{\tilde{\vec{x}}_t^{(i)}\}_{i=1}^{N_s'}$ with an importance weight β_i, which will be factored in a weighted loss function on a classifier f:

$$L_w(f) = \sum_{i=1}^{N_s'} \beta_i l(f(\tilde{\vec{x}}_t^{(i)}), y^{(i)}). \qquad (7)$$

The purpose of the importance weight is to shift the source domain data closer to the target domain data such that $P(X_t) = \beta P(\tilde{X}_t)$, where $\beta = [\beta_1, \beta_2, \ldots, \beta_{N_s'}]$. In order to find suitable values of $\beta \in \mathbb{R}^{N_s'}$, we need to minimise the discrepancy between means of $\tilde{X}_t$ and X_t subject to

$$\beta_i \in [0, 1] \text{ and } |\frac{1}{N_s'}\sum_{i}^{N_s'} \beta_i(x_s^{(i)}) - 1| \leq \epsilon, \qquad (8)$$

where ϵ is set as 0.01. The first part limits the scope of discrepancy between $P(\tilde{X}_t)$ and $P(X_t)$ and ensures the robustness by limiting the influence of individual instances. The second part ensures that $\beta P(\tilde{X}_t)$ is close to a probability distribution.

To find β, a feature space $\mathcal{F}$ is used, which is a RKHS with a universal kernel $k(x, x') = \langle\Phi(x), \Phi(x')\rangle$. With the feature map Φ: $X_t \rightarrow \mathcal{F}$, we define

$$K_{ij} := k(\tilde{\vec{x}}_t^{(i)}, \vec{x}_t^{(j)}) \qquad (9)$$

$$\kappa_i := \frac{N_s'}{N_t}\sum_{j=1}^{N_t} k(\tilde{\vec{x}}_t^{(i)}, \vec{x}_t^{(j)}). \qquad (10)$$

Then the discrepancy equation is defined as:

$$\begin{aligned}& ||\frac{1}{N_s'}\sum_{i=1}^{N_s'} \beta_i \Phi(\tilde{\vec{x}}_t^{(i)}) - \frac{1}{N_t}\sum_{j=1}^{N_t} \Phi(\vec{x}_t^{(j)})||^2 \\ & \qquad = \frac{1}{N_s'^2}\beta^T K \beta - \frac{2}{N_t^2}\kappa^T \beta + C \qquad (11)\end{aligned}$$

where C is a constant. Thus, finding suitable β can be formulated as a quadratic problem [13], such that

$$\begin{aligned}& \min_{\beta} \frac{1}{2}\beta^T K \beta - \kappa^T \beta \\ & \text{subject to } \beta_i \in [0, B] \text{ and } |\sum_{i=1}^{N_s'} \beta_i - N_s'| \leq N_s'\epsilon. \qquad (12)\end{aligned}$$

Algorithm 2 *shift*-GANTraining

```
Data: Labelled source domain $\mathcal{D}_s = \{(\vec{x}_s^{(i)}, y_s^{(i)})\}_{i=1}^{N_s}$ and
        unlabelled target domain $\mathcal{D}_t = \{\vec{x}_t^{(j)}\}_{j=1}^{N_t}$
Learn a generator $G_s$ by training a Bi-GAN with
$\{\vec{x}_s^{(i)}\}_{i=1}^{N_s}$ and $\{\vec{x}_t^{(j)})\}_{j=1}^{N_t}$ in Algorithm 1
Generate top-$k$ samples in the target domain on each
source sample $\vec{x}_s^{(i)}$: $\{\tilde{\vec{x}}_t^{(l)}\}_{l=1}^{N_s'}$, $N_s' = k * N_s$
Learn $\beta$ with $\{\tilde{\vec{x}}_t^{(l)}\}_{l=1}^{N_s'}$ and $\mathcal{D}_t$ using Eq (12)
Build a SVM classifier $f$ with $\beta$ and $\{\tilde{\vec{x}}_t^{(i)}\}_{i=1}^{N_s'}$
Predict labels for instances in $\mathcal{D}_t$
```

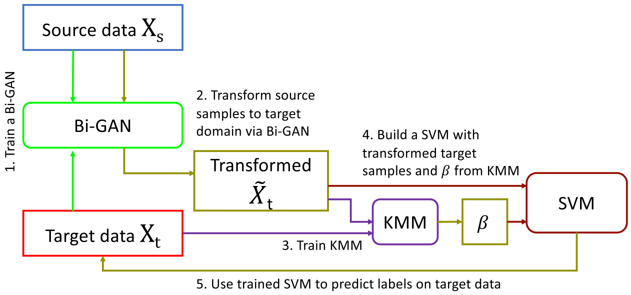

Source data X_s
1. Train a Bi-GAN
Bi-GAN
2. Transform source samples to target domain via Bi-GAN
Transformed $\tilde{X}_t$
4. Build a SVM with transformed target samples and β from KMM
KMM
β
SVM
Target data X_t
3. Train KMM
5. Use trained SVM to predict labels on target data

FIGURE 3. The overall workflow of *shift*-GAN.

E. PREDICTION ON TARGET DATASET

Algorithm 2 presents an overall algorithm of *shift*-GANand Figure 3 shows the workflow. It starts with training two generators G_s and G_t and then with G_s we can transform source dataset into target dataset. Then we align the transformed data with unlabelled target dataset to learn weighting factor β. After alignment, we build a SVM classifier with the transformed source dataset, with which we can predict labels on the transformed target dataset.

As shown in the next sections, *shift*-GAN can learn invariant transformation functions between source and target domain. When there are new activities introduced on the source dataset, then we just need to repeat the steps 2-5 in Algorithm 2 to retrain β and build a classifier for recognising these activities on the target dataset.

IV. EXPERIMENT SETUP

The main goal of the evaluation is to assess the effectiveness of *shift*-GAN in *generalised unsupervised* domain adaptation; that is, how accurately *shift*-GAN can recognise activities in the target domain without using any labelled data in the target domain, and to what extent the domain adaptation being learnt can be generalised to new, unseen activities. In the following, we will introduce the experiment setup and the implementation details.

A. DATASETS AND PREPROCESSING

To assess the generality and feasibility of *shift*-GAN, we consider two most common types of smart home datasets: ambient binary sensors and accelerometers. For ambient binary sensor datasets, we use three datasets collected and curated by the University of Amsterdam (named *HA*, *HB*, and *HC* respectively in the remainder of this article) [41]. They are collected on three different users in three different residential settings, each being deployed with binary sensors, including infra-red position sensors, switch sensors, and water flow sensors. All these sensors output binary readings (0 or 1), indicating whether or not a sensor fires. For binary sensor data, we employ the state-of-the-art technique to extract features [9]; that is, the activation ratio within a fixed interval (i.e., 60 seconds) as sensor features.

For accelerometer data, we use two widely-used datasets: PAMAP (PAMAP) [35], and UCI daily and sports (DSADS) [2], [3] datasets. The PAMAP dataset records 12 activities performed by 9 subjects, including sitting, lying, house cleaning, and ironing. Each subject wears 3 accelerometer units on their dominant arm, chest, and dominant side ankle. The DSADS consists of 19 daily activities performed by 8 subjects, including exercising on a stepper, rowing, and running on a treadmill. Each subject wears 5 accelerometer units on their torso, right arm, left arm, right leg, and left leg. We use the feature dataset [42] generated from these two datasets. That is, 27 features are extracted per sensor on each body part, including mean, standard deviation, and spectrum peak position. For the experiments, we use similar and different body parts as described in Table 1. Beyond the tasks within each dataset, we also consider the tasks between PAMAP and DSADS on their four common activities, including walking, standing, sitting, and lying.

All these 5 datasets are collected by third-parties and are publicly available. Table 1 records their main statistics and Figure 4 presents the activity distribution of 4 datasets. We skip DSADS because it has equal activity distribution. These 5 datasets exhibit a wide range of domain adaptation challenges, especially the varying feature complexity (from 14 to 405) and the large number of activities (from 6 to 19). Also there exists high similarity between activity classes and high diversity of patterns in one activity class. All these challenges have added extra complexity to unsupervised domain adaptation, which will be discussed later.

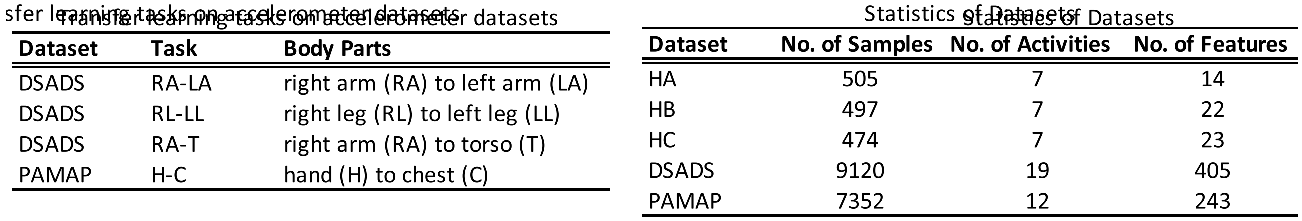
B. *shift*-GANCONFIGURATION

shift-GAN consists of two main components: Bi-GAN and SVM with KMM. For the Bi-GAN, both generators G_s and G_t have the identical network architecture. The leaky ReLU activation function is used in both generators with the exception of the output layer which uses `Tanh` function.

All models are implemented with PyTorch, the loss function of GAN is minimised using the Adam optimisation.[1] The optimiser is parameterised with a learning rate of 10^{-2} and the mini-batch size is set to 100. In order to choose the best setting, we have done the grid search on the number of

[1] The implementation of *shift*-GAN and grid search results can be openly accessed at https://github.com/An5r3a.

TABLE 1. Transfer learning tasks of accelerometer data and statistics of datasets.



Transfer learning tasks on accelerometer datasets

Dataset	Task	Body Parts
DSADS	RA-LA	right arm (RA) to left arm (LA)
DSADS	RL-LL	right leg (RL) to left leg (LL)
DSADS	RA-T	right arm (RA) to torso (T)
PAMAP	H-C	hand (H) to chest (C)

Statistics of Datasets

Dataset	No. of Samples	No. of Activities	No. of Features
HA	505	7	14
HB	497	7	22
HC	474	7	23
DSADS	9120	19	405
PAMAP	7352	12	243

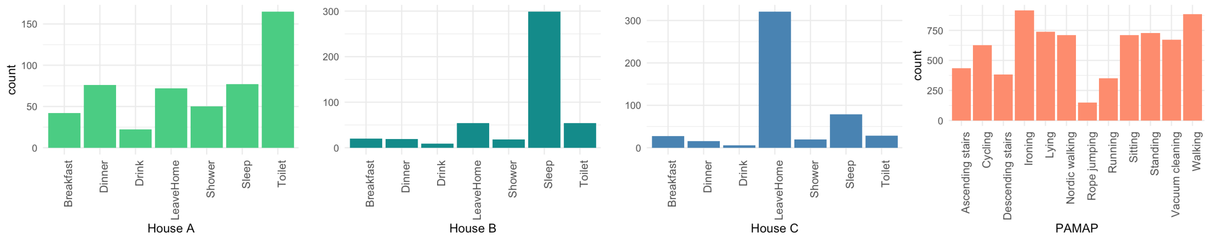


FIGURE 4. Activity distribution of the 4 datasets used in evaluation.

layers from 1 to 3 and the number of neurons from $S - S/2$ to $S + S/2$, where S is the number of sensor features and choose the setting that leads to the highest accuracy for each dataset. Similarly, we have run grid search for configuring the weights of source and target generators λ_s and λ_t in Eq (6) in the range of [100, 1000].

V. RESULTS AND DISCUSSION

In this section, we evaluate the proposed approach *shift*-GAN in unsupervised adaptation and compare with state-of-the-art domain adaptation techniques. More specifically, we seek to validate the following claimed contributions:

- *Effectiveness* – How accurately *shift*-GAN can recognise activities in the target dataset, compared with state-of-the-art domain adaptation techniques? We will also perform ablation analysis to assess the contribution of KMM, and as well as stability and convergence analysis on Bi-GAN working on sensor data.
- *Generality* – Can *shift*-GAN be activity-independent; that is, the learnt domain adaptation can be applied to new, emerging activities that have not been observed during the training?
- *Robustness* – How robust *shift*-GAN can perform in the face of sensor noise?
- *Stability* – How well *shift*-GAN can maintain the accuracy when the amount of training data is reduced?

The performance is measured in *F1 score*, which is the harmonic mean of precision and recall. As the selected datasets have imbalanced activity distribution, we report both the *micro-F1 score* which aggregates the contributions of all instances, and the *macro-F1 score* which computes the metric independently for each class and then takes the average. For each experiment, we run 10 times and report the mean of macro and micro F1-scores as the final result.

A. EFFECTIVENESS OF DOMAIN ADAPTATION

1) GOAL

Our first experiment is to assess the effectiveness of domain adaptation of *shift*-GAN.

2) EVALUATION PROCESS

To demonstrate the effectiveness of *shift*-GAN, we compare it with 10 state-of-the-art domain adaptation techniques. First we select 5 classic techniques that have achieved best performance in transferring heterogeneous feature spaces [36] including Geodesic Flow Kernel (GFK) [12], Transfer Component Analysis (TCA) [29], Feature-Level Domain Adaptation (FLDA) [19], Joint Distribution Adaptation (JDA) [25], and Importance-weighting with logistic discrimination (IW) [14], along with a linear baseline technique Canonical Correlation Analysis (CCA). There exists many deep learning-based domain adaptation techniques [53]. Here we mainly focus on the recent adversarial techniques that have achieved high adaptation accuracy and demonstrated robustness in transferring heterogeneous feature spaces. Therefore, we select Adversarial Discriminative Domain Adaptation (ADDA) [39], Domain-Adversarial Neural Network (DANN) [11], Deep Adaptation Networks (DAN) [24], and Adversarial Domain Adaptation with Domain Mixup (ADADM) [44]. Also we compare with a recent domain adaptation technique that has been evaluated on the same datasets and tasks, which is Stratified Transfer Learning (STL) [6]. For each technique, we use all the source domain data and randomly split the target domain data into 80% for training and 20% for testing. The labels on the target dataset are not used during training.

3) RESULTS

Table 2 compares the micro and macro F1-scores of *shift*-GANand state-of-the-art techniques. *shift*-GAN achieves

TABLE 2. **Mean of micro- and macro-F1 scores between *shift*-GAN and comparison techniques.**

	shift-GAN		DAN		ADADM		DANN		ADDA		GFK		TCA		JDA		FLDA		IW		CCA	
Tasks	*micro*	*macro*	*micro*	*macro*	*micro*	*macro*	*micro*	*macro*	*micro*	*macro*	*micro*	*macro*	*micro*	*macro*	*micro*	*macro*	*micro*	*macro*	*micro*	*macro*	*micro*	*macro*
A-B	**0.80**	0.68	0.78	**0.72**	0.60	0.51	0.55	0.53	0.47	0.51	0.6	0.56	0.51	0.21	0.25	0.21	0.11	0.1	0.01	0.1	0.07	0.03
B-A	0.79	0.76	**0.88**	**0.86**	0.82	0.71	0.76	0.71	0.56	0.63	0.51	0.31	0.54	0.42	0.18	0.14	0.26	0.13	0.04	0.04	0.07	0.04
A-C	0.81	0.74	0.77	0.72	0.51	0.34	**0.86**	**0.86**	0.24	0.38	0.75	0.48	0.57	0.55	0.58	0.4	0.11	0.12	0.03	0.01	0.1	0.04
C-A	0.77	0.62	0.82	0.78	**0.87**	**0.83**	0.74	0.73	0.41	0.54	0.64	0.49	0.26	0.25	0.37	0.25	0.17	0.12	0.04	0.1	0.1	0.04
B-C	0.79	**0.76**	**0.81**	0.76	0.78	0.64	0.72	0.71	0.46	0.50	0.58	0.5	0.23	0.11	0.22	0.21	0.3	0.31	0.03	0.01	0.13	0.03
C-B	**0.80**	0.68	0.79	**0.73**	0.78	0.64	0.72	0.71	0.58	0.64	0.6	0.27	0.52	0.15	0.25	0.14	0.04	0.04	0.01	0.01	0.13	0.02
RA-LA	**0.91**	**0.89**	0.72	0.64	0.74	0.67	0.34	0.30	0.46	0.50	0.71	0.7	0.57	0.5	0.74	0.72	0.4	0.4	0.31	0.31	0.14	0.14
RL-LL	**0.95**	**0.94**	0.78	0.74	0.60	0.56	0.39	0.34	0.46	0.50	0.78	0.77	0.54	0.45	0.78	0.78	0.5	0.43	0.3	0.31	0.09	0.04
RA-T	**0.90**	**0.89**	0.72	0.52	0.60	0.55	0.61	0.60	0.46	0.50	0.43	0.42	0.49	0.42	0.48	0.42	0.52	0.52	0.25	0.1	0.05	0.06
H-C	0.59	0.55	0.72	0.49	**0.80**	**0.71**	0.40	0.37	0.23	0.13	0.35	0.32	0.55	0.43	0.22	0.18	0.25	0.22	0.25	0.1	0.07	0.07
DSADS-PAMAP	**0.67**	**0.62**	0.72	0.47	0.50	0.45	-	-	-	-	0.19	0.18	0.46	0.26	0.21	0.11	0.47	0.41	0.25	0.1	0.22	0.2
PAMAP-DSADS	**0.80**	**0.65**	0.53	0.39	0.60	0.55	-	-	-	-	0.31	0.12	0.35	0.22	0.2	0.1	0.56	0.44	0.29	0.11	0.3	0.22
Binary Average	0.79	0.71	**0.81**	**0.76**	0.73	0.61	0.73	0.71	0.46	0.53	0.61	0.44	0.44	0.28	0.31	0.23	0.17	0.14	0.03	0.05	0.1	0.03
Acc Average	**0.80**	**0.76**	0.70	0.54	0.64	0.58	0.43	0.40	0.40	0.41	0.46	0.42	0.49	0.38	0.44	0.39	0.45	0.4	0.28	0.17	0.15	0.12
Overall Average	**0.80**	**0.73**	0.75	0.65	0.68	0.60	0.61	0.59	0.43	0.48	0.54	0.42	0.47	0.33	0.37	0.31	0.31	0.27	0.15	0.11	0.12	0.08

higher accuracy averaged across 12 tasks, compared to the state-of-the-art techniques. More specifically, the performance improvement in micro F1 of *shift*-GAN over each technique is: 5% (DAN), 12% (ADADM), 19% (DANN), 37% (ADDA), 26% (GFK), 33% (TCA), 43% (JDA), 49% (FLDA), 65% (IW), and 68% (CCA). In a similar way, the improvement in macro-F1 of *shift*-GAN over each technique is: 8% (DAN), 13% (ADADM), 14% (DANN), 25% (ADDA), 31% (GFK), 40% (TCA), 42% (JDA), 46% (FLDA), 62% (IW), and 65% (CCA). This results in the averaged gain of 36% in micro F1 and 35% in macro F1.

***shift*-GAN outperforms the adversarial domain adaptation techniques DAN, ADADM, DANN, and ADDA**. ADADM allows for heterogeneous feature spaces and achieves much higher performance as the mixture of source and target samples allows for training more robust generator. This is also the reason that ADADM achieves the highest accuracy on the H-C task, where the feature spaces on hand and chest are significantly different. DANN and ADDA only work on the homogeneous feature spaces. We have applied semantics similarity matrix [36] to transform target to source samples to allow domain adaptation between House A, B, and C. However, *shift*-GAN significantly outperforms them. The results demonstrate the need for bi-directional GANs to enable heterogeneous feature spaces and the cyclic consistency constraints allow for more effective feature transformation.

DAN outperforms *shift*-GAN on the binary datasets. The reason is that it leverages convolutional layers to learn transferred features on combined source and target features and then minimises the discrepancy on embeddings on task-specific layers. Combining features together allows better feature contrast and alignment than *shift*-GAN where the focus is only to learn the mapping functions between domains. However, when the dimension and discrepancy of features increases (e.g., the dimensions of accelerometer and binary sensor data are 81 and 23 respectively), such fine-grained layer-wise alignment does not work and Bi-GAN that explores the holistic mapping functions between domains achieves much higher accuracy.

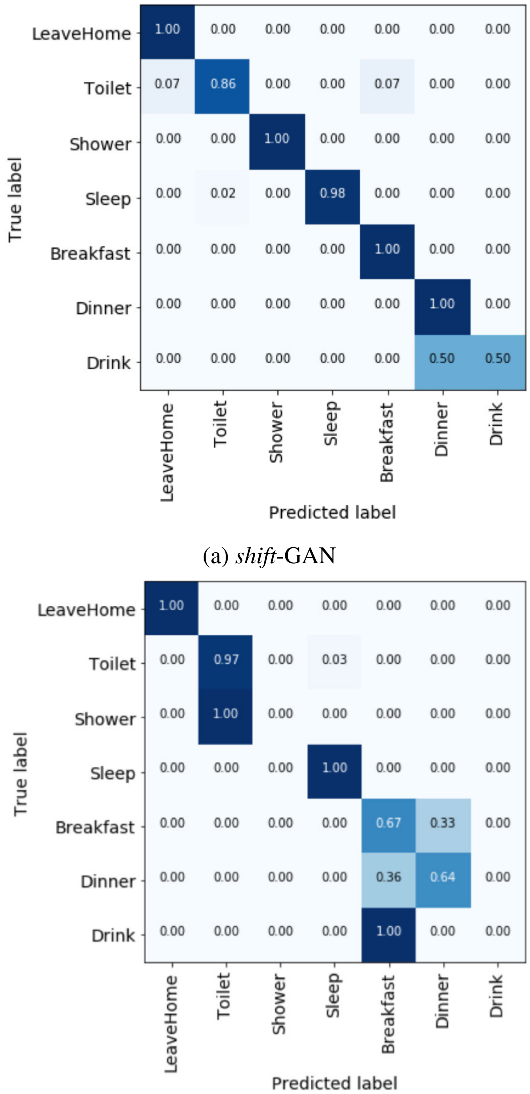
Among the classic domain adaptation techniques, GFK and TCA perform better than all the others, and these will be taken forward for the other comparison experiments. TCA learns *transfer components* across domains in a RKHS using Maximum Mean Discrepancy (MMD). This set of common transfer components underlie both domains such that the distance across domains is reduced in a RKHS. Its poor performance indicates that *shift*-GAN is better than TCA in capturing inherent common representations of the source and target domain.

GFK performs better than the other techniques, but still 25% less than *shift*-GAN in both micro and macro F1-scores. Figure 5 presents the confusion matrix of *shift*-GAN and GFK on the A-B task. We can see that GFK has good discriminative power, however, *shift*-GAN is more capable of recognising activities that have less distinctive patterns like 'Drink', and is better at finding discriminative features between activities that fire the same set of sensors; for example, 'Toilet' and 'Shower'. Therefore, we conclude that *shift*-GAN is more effective than GFK when dealing with imbalanced datasets and is better at recognising activities that have less frequent patterns than the other techniques.

CCA, IW and FLDA perform the worst. Compared to CCA and IW, *shift*-GAN is less affected by the sample size and it performs well when there is little overlap between the source and target domain [19]. FLDA focuses on a feature-level domain adaptation by describing the shift between the target and source domain for each feature individually. It assumes strong correlation between features on the corresponding activities in the source and target domain. This assumption is not valid on binary data because different activities can deploy the same set of sensors. This is why FLDA performs better on accelerometer data but much worse on binary sensor data. In comparison, *shift*-GAN does not assumes such strong correlations, but learns non-linear transformations between source and target feature spaces using GANs, leading to much more improvement in adaptation accuracy. JDA is unable to adapt the marginal distributions and conditional distributions when the source and target domains are very dissimilar, where *shift*-GAN has demonstrated superior performance.

TABLE 3. Comparison of training time (in minutes).

Training Time	*shift-GAN*	DAN	ADADM	DANN	ADDA	GFK	TCA	JDA	FLDA	IW	CCA
Binary	3.0	2.7	34.1	2.4	2.9	1.9	2.7	1.2	7.9	4.1	2.1
Acc	22.2	21.9	24.9	27.0	26.9	29.2	29.7	30.0	40.0	34.1	30.0



(a) *shift*-GAN

(b) GFK

FIGURE 5. Comparison of confusion matrices on task A-B between ***shift*-GAN and GFK.**

***shift*-GAN is computationally efficient compared to these state-of-the-art techniques**. Table 3 compares the training time between *shift*-GAN and the other techniques. The training time is averaged on binary and accelerometer tasks. On the binary tasks, *shift*-GAN has averaged 3 mins of training time per task, which is close to the best performing technique (i.e., DAN in 2.7 mins) and much lower than ADADM in 34.1 mins. On the accelerometer tasks, *shift*-GAN has the second shortest training time, just after DAN.

Table 4 compares the accuracy of *shift*-GAN and the accuracy of STL and other comparison techniques reported in [6], including Principal Component Analysis (PCA), Kernel Principal Component Analysis (KPCA), TCA, GFK, Transfer Kernel Learning (TKL). To be consistent, we report the instance accuracy on 4 tasks being reported on STL. *shift*-GAN has better or comparable performance against those state-of-the-art techniques. The improvement is more observable in tasks RA-T and H-C where the increase in accuracy is 43% and 25% respectively. The proposed method performs better than STL because in the pre-annotation step of STL, the classifiers are biased towards the majority class, and thus the instances belonging to the minority classes might be incorrectly labelled with the majority classes. This bias results in the failure in the later alignment process where feature alignment is performed per class between the source and target domain. In *shift*-GAN, Bi-GAN training on unlabelled source and target examples, so the labels do not play a role in transforming feature spaces across domains. Thus, it makes the transformation more robust and activity-independent.

B. ABLATION ANALYSIS

shift-GAN is composed of two components: Bi-GAN and KMM. We hypothesise that KMM is important to shift the Bi-GAN generated samples closer to true examples in the target domain and thus leads to higher accuracy in domain adaptation. To validate this, we run ablation analysis, where we compare the performance of *shift*-GAN with and without KMM.

***shift*-GAN has demonstrated its strength of performing a fine-grained feature space alignment**. Figure 6 compares the micro and macro F1-scores of *shift*-GAN and Bi-GAN (that is, *shift*-GAN without KMM) on 12 tasks. As we can see, introducing prior knowledge to SVM classifier has improved the performance. *shift*-GAN outperforms Bi-GAN on 8 out of 12 tasks with the averaged gain of 5% in micro-F1 and 9 out of 12 tasks with the averaged gain of 9% in macro-F1.

***shift*-GAN outperforms Bi-GAN when the dimension of sensor feature is higher and the number of activity class is larger**. For example, with 19 activity classes on DSADS dataset, *shift*-GAN can achieve the averaged gain of 15.3% in micro- and macro-F1. The main reason resides in instability of Bi-GAN in generating samples in high-dimensional spaces. As presented in Figure 6, Bi-GAN works well on low-dimensional datasets with a smaller number of classes, which has achieved slightly higher F1-scores than *shift*-GAN as the domain adaptation through Bi-GAN is effective and the samples generated are reliable. The performance on

TABLE 4. Accuracy comparison between *shift*-GAN and domain adaptation techniques in [6].

Tasks	*shift*- GAN	STL	GFK	TCA	PCA	KPCA	TKL
RA-LA	**0.90**	0.71	0.71	0.66	0.6	0.62	0.54
RL-LL	**0.94**	0.81	0.8	0.75	0.69	0.71	0.62
RA-T	**0.89**	0.46	0.44	0.4	0.39	0.3	0.33
H-C	**0.65**	0.43	0.36	0.35	0.35	0.24	0.36
Average	**0.85**	0.6	0.58	0.54	0.51	0.47	0.46

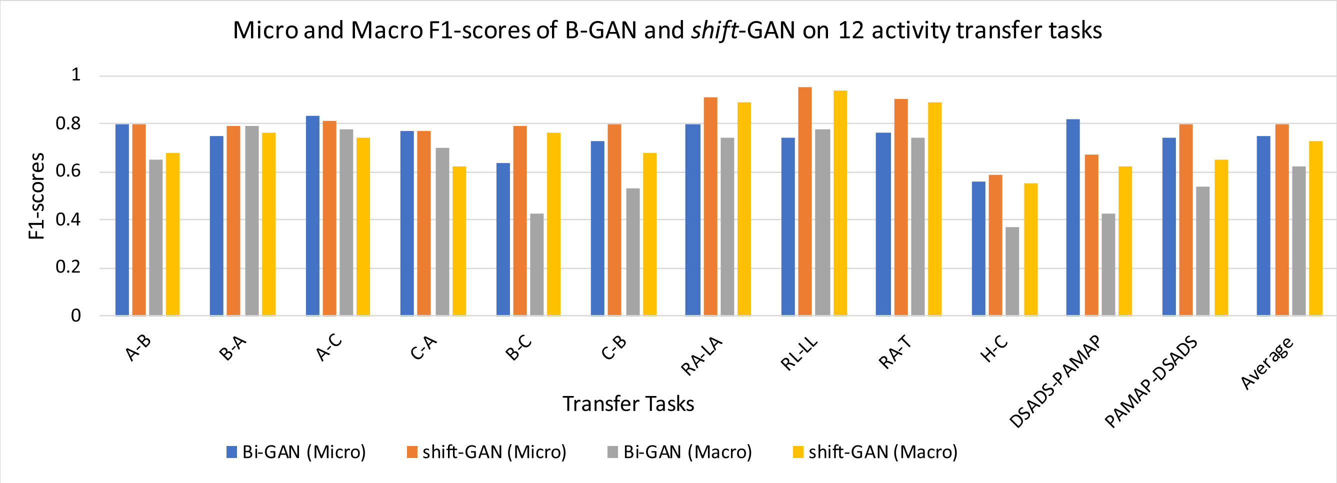
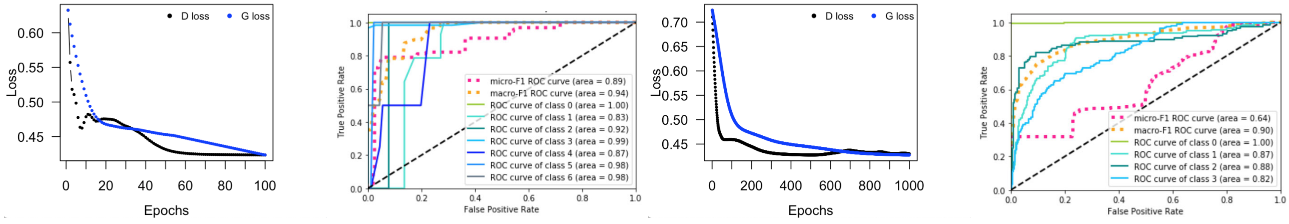


FIGURE 6. Ablation analysis of *shift*-GAN: with and without KMM (*i.e.*, Bi-GAN only).



(a) Loss on B-C

(b) ROC on B-C

(c) Loss on D-P

(d) ROC on D-P

FIGURE 7. Comparison of loss performance and ROC curves for tasks B-C and D-P (DSADS-PAMAP) during training.

Bi-GAN drops significantly on the tasks of H-C, DSADS-PAMAP, and PAMAP-DSADS. The H-C task on PAMAP dataset is complicated with a large number of classes (e.g., 12 vs. 7 on A-B) and imbalanced class distribution as shown in Figure 4.

For the tasks between DSADS and PAMAP, the challenge is the high-dimensional feature space matching. It is different from image-to-image translation [48], where the image size in both source and target domain is fixed and the feature spaces are more similar than sensor data; that is, the pixel positions such as contour of faces are matched between source and target domain. This also explains why the Bi-GAN achieves much higher F1-scores on the tasks of RA-LA, RL-LL, and RA-T, where the feature spaces are better matched; that is, the dimension and the semantics of feature space are the same. The feature space alignment is mainly on fine-grained, local adjustment. However, for the tasks between DSADS and PAMAP, the distance of their feature spaces are much larger: the dimension is different (i.e., 405 in DSADS and 243 in PAMAP), and the semantics of each feature are different. This leads to ineffectiveness of Bi-GAN. With the help of KMM, *shift*-GAN improves the performance over Bi-GAN with 19% in micro-F1 and 11% in macro-F1.

C. STABILITY AND CONVERGENCE OF GAN

Stability and difficulty to converge are two classic problems in GAN. These two can be a significant concern for unsupervised domain adaptation. Since we do not use any labels on the target domain, the performance of domain adaptation relies on stable feature space transformation between two domains. For this purpose, we have recorded the loss on both generators and discriminators over epochs and compare the performance of domain adaptation in ROC curves.

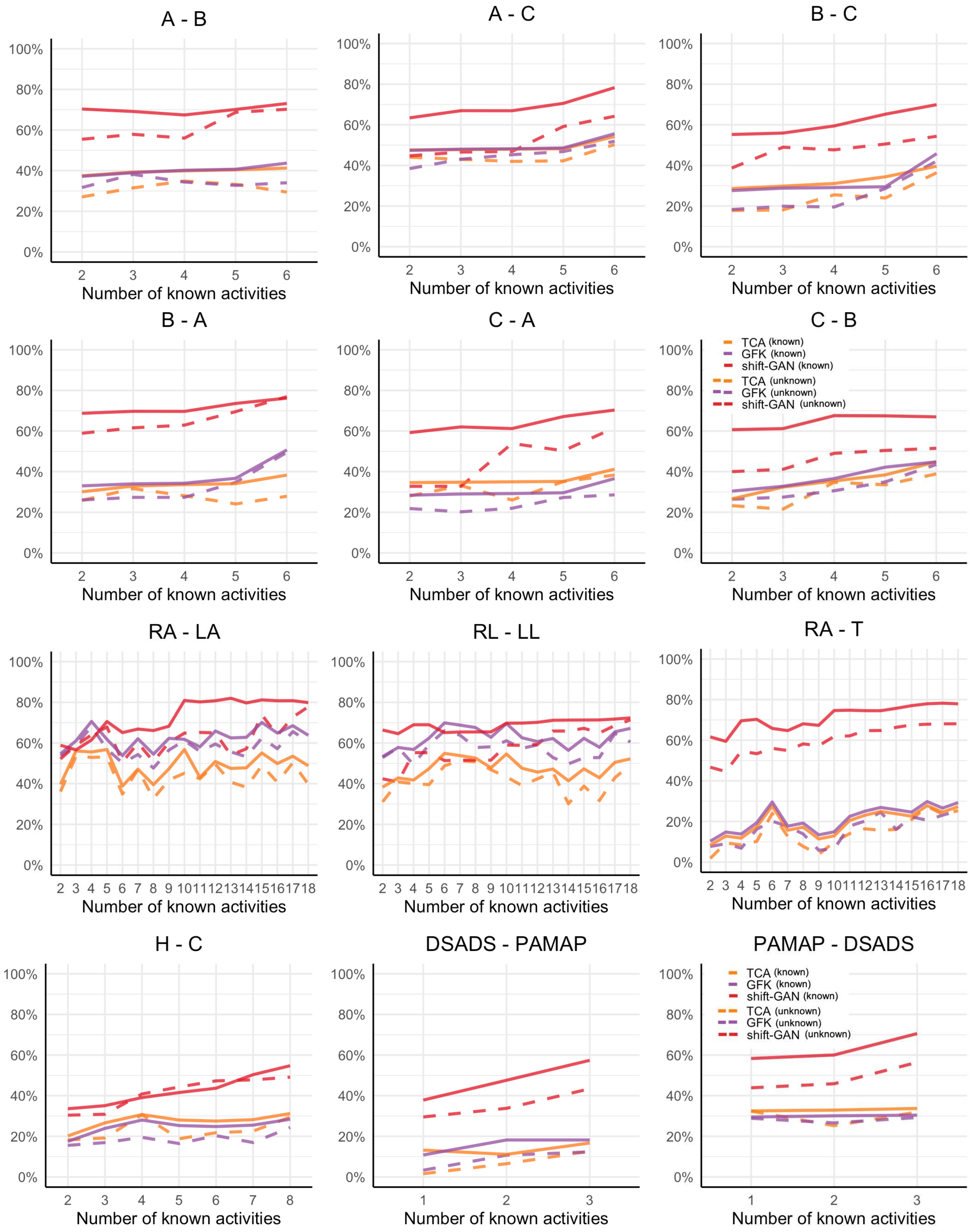


FIGURE 8. Micro and Macro F1-scores on activity discovery.

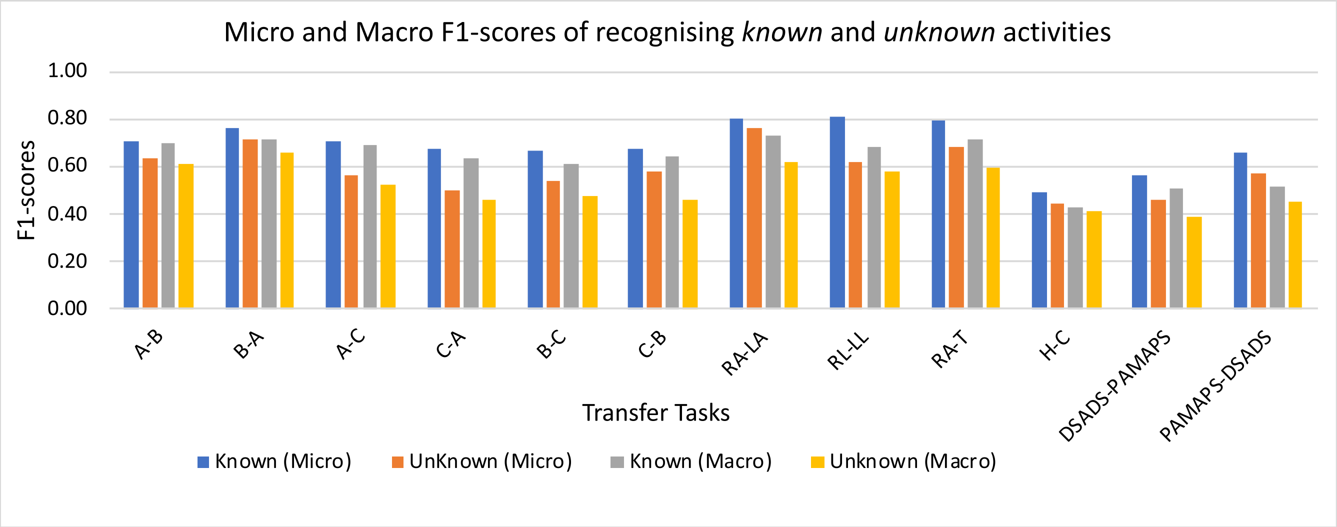


FIGURE 9. Performance of ***shift*-GAN in recognising *known* and *unknown* activities.**

Figure 7 contains four plots of the performance of the discriminator and generator during the training process for the tasks of B-C and DSADS-PAMAP. The discriminator and generator are on the primary GAN; i.e., from source to target domain. With both loss and accuracy, we can see that in our experiments GAN has converged well on sensor data, even on high-dimensional data.

D. GENERALISED DOMAIN ADAPTATION

1) GOAL

Our next experiment is to assess the generalisation of *shift*-GAN in transferring the activities that have not been observed in the training data. That is, after training both generators G_s and G_t, if the source domain has a new set of data that is labelled with a new set of activities $\mathcal{Y}'_s$ different from the original activity set $\mathcal{Y}_s$, can we transfer the knowledge on these new source data to predict the labels in $\mathcal{Y}'_s$ on the target domain? This experiment tries to demonstrate the learnt mappings are intrinsic to both domains, independent of the activity classes.

2) EVALUATION PROCESS

The evaluation process works as follows. Given $|A|$ as the total number of the common activities between both source and target domain, we start with n $(\leq |A| - 1)$ activities, and take all the source domain data on these n activities and 80% unlabelled training data on the target domain for training *shift*-GAN. For the rest of $|A| - n$ number of activities, we use the trained Bi-GAN models to generate target domain data, apply KMM, and build a classifier on the generated and corrected target domain data. We assess the accuracy of recognising both n *known* and $|A| - n$ *unknown* activities on the target domain. We set n to be $|A| - 1$ to 2. For each setting, we run 10 iterations and report the F1-scores on both known and unknown activities. We will compare *shift*-GAN with the best performing classic techniques – GFK and TCA, but not with deep learning techniques, as they learn feature transferring and classification altogether, which make it impossible to introduce new classes without retraining the whole technique.

3) RESULTS

The experiment results over the 12 transfer learning tasks are listed in Figure 8, where *shift*-GAN significantly outperforms TCA and GFK. As expected, the classification performance decreases when less activities are available for training; that is, when the number of known activity is small. However, *shift*-GAN is able to recreate instances for the unknown activities. This can be seen from the average F1-scores summarised in Figure 9 where the difference in accuracy between the known and unknown activities is not too big. It is interesting to observe a very stable trend in F1-scores for the known activities. This shows that *shift*-GAN is effective in learning feature representations for each activity individually and is capable to use the knowledge learnt from the source domain to predict a class label for a new activity from the target domain.

E. ROBUSTNESS TO SENSOR NOISE

1) GOAL

Our third experiment is to assess the robustness of learnt domain adaptation against to sensor noise. Noise is pervasive in all the sensor data and over time, sensors' performance can degrade and generate abnormal data. It is desirable that the domain adaptation can learn intrinsic feature mapping, independent sporadic, unexpected noise.

2) EVALUATION PROCESS

We use 80% of the target domain data for training and 20% for testing. In the testing phase, we inject Gaussian noise to

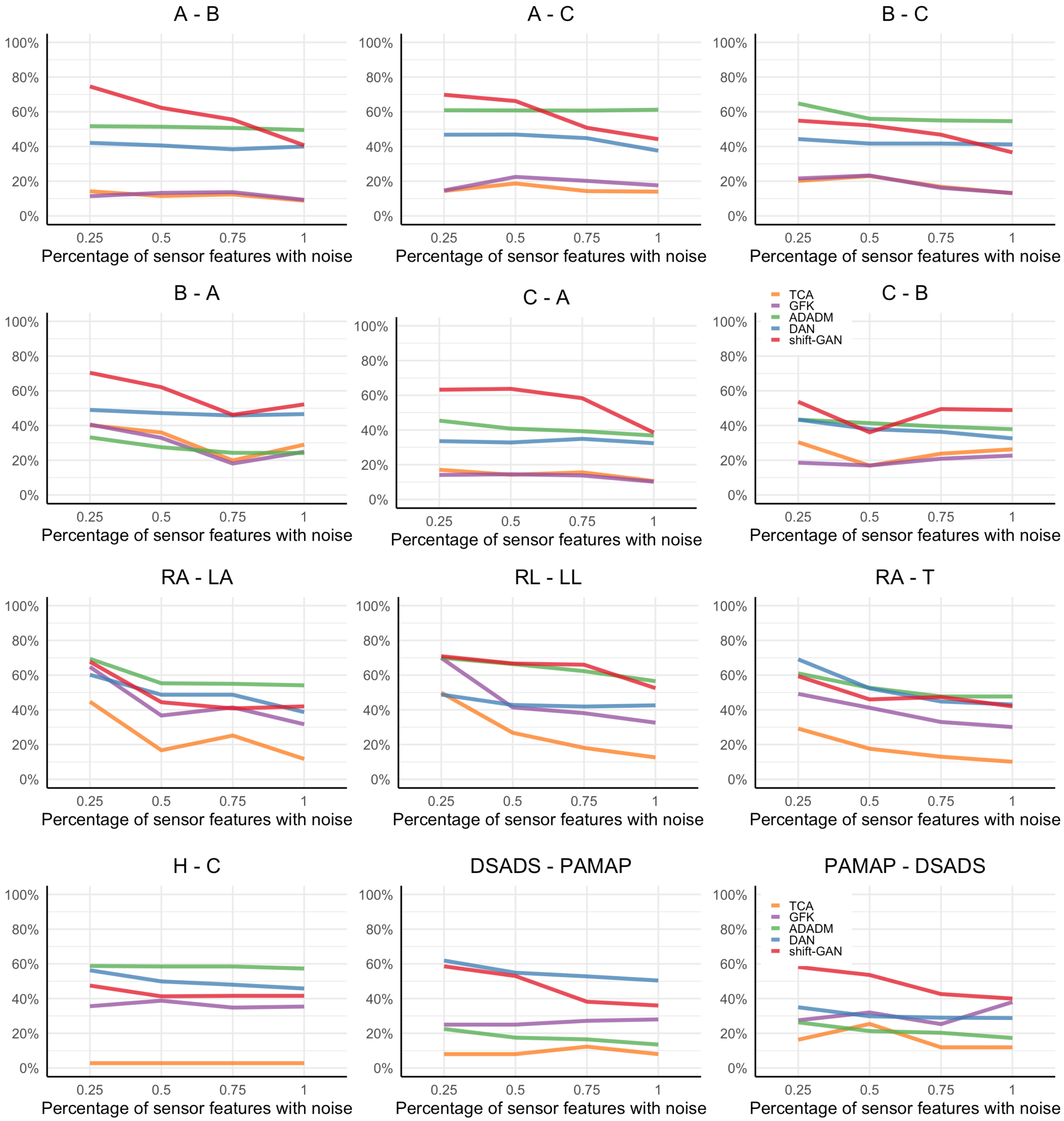


FIGURE 10. Comparison of macro F1-scores between *shift*-GAN and state-of-the-art-technique: DAN, ADADM, GFK and TCA, with Gaussian noise injected.

the testing data in the target domain. We randomly selected k percentage of sensor features and randomly sample Gaussian noise with mean and variance between 0 and 1. k varies from 25% to 100% with a step size of 25%. Note that a key outcome of this experiment is to measure the stability of *shift*-GAN by systematically increasing the percentage of noise sensor features until all sensor features are affected. A sensor feature can happen to not be selected or be selected more than once during the simulation process but in the final experiment ($k = 100\%$) all sensor features will be compromised. The reason behind this experiment is that in reality, the performance of the sensor can gradually or drastically vary over time. We try to simulate a real-world situation and its impact in our proposed method. We will compare *shift*-GAN with the best performing deep and classic techniques: DAN, ADADM, GFK and TCA.

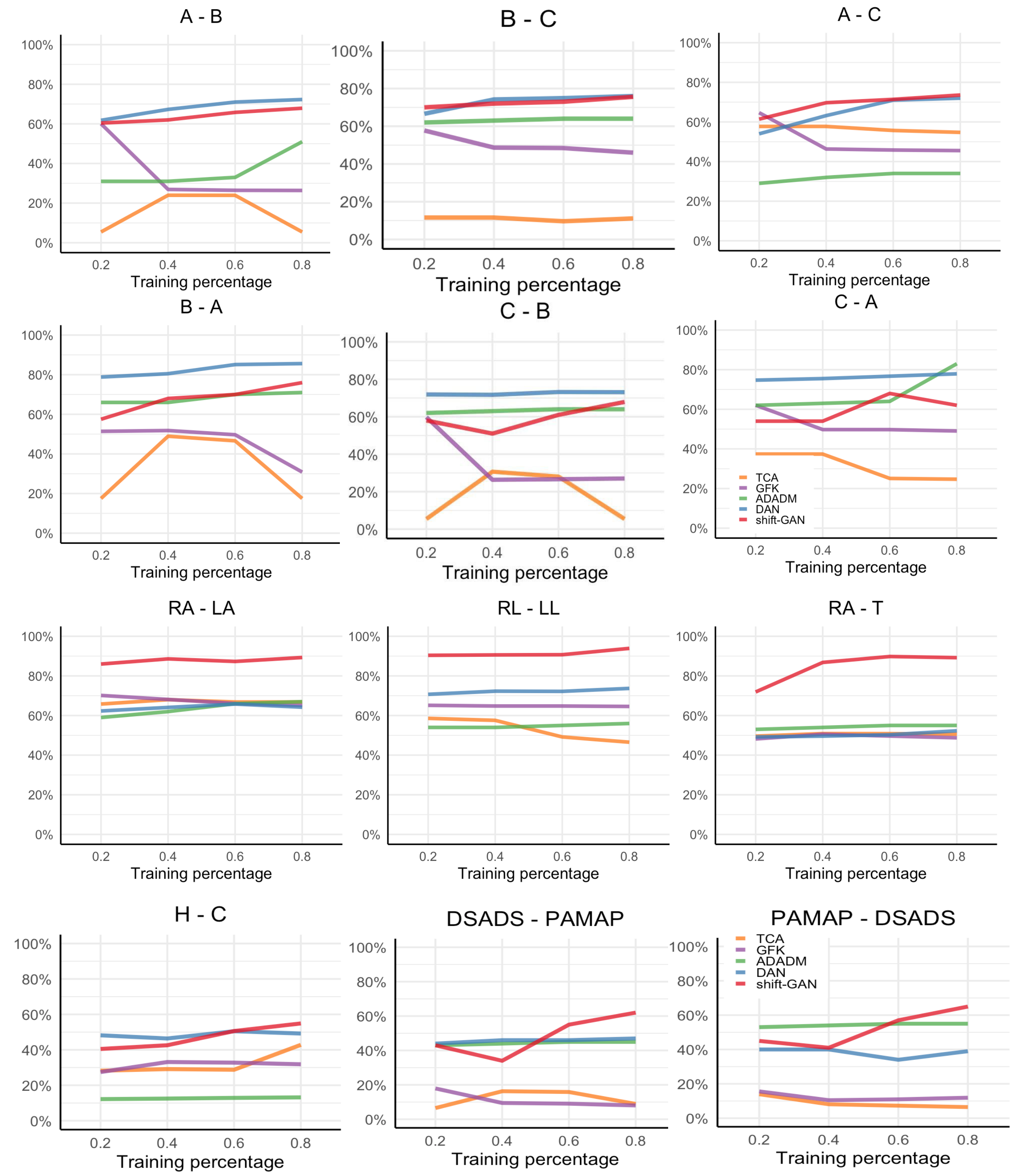


FIGURE 11. Macro F1-scores over different training percentages.

3) RESULTS

Figure 10 compares the accuracy of *shift*-GAN and the existing techniques on different levels of sensor noise. *shift*-GAN performs generally better than the other four methods across the different tasks and various percentage of noise sensor features. *shift*-GAN have achieved higher macro-F1 scores than DAN (11.7%), ADADM (1.8%), GFK (28.2%), and TCA (34.2%). This further demonstrates that using GAN for domain adaptation plus KMM for shift correction are stable and effective methods in domain adaptation tasks. ADADM, mixing up the samples, can be more robust in dealing with noise, as the accuracy on ADADM does not vary much with different noise effects. DAN achieves better performance on H-C and DSADS-PAMAP tasks, since concatenating source and target features will lead to more robust feature learning.

It is also evident that classic techniques are more sensitive to noise. For example, TCA degrades abruptly when noise is introduced in binary sensor data, and leads to low accuracy when a small amount of noise is introduced in accelerometer data. Also, we can notice that the performance of TCA does not varies in experiments H-C, DSADS-PAMAP and PAMAP-DSADS regardless of the amount of noise. This is because TCA is unable to find a feature representation for each activity in both domains and its biased towards one class; that is, after the domain adaptation process the feature representation is not meaningful which makes the SVM classifier struggle in distinguishing between classes and will only predicts one. In DSADS-PAMAP and PAMAPS-DSADS experiment, TCA is not only affected by the noise but also by the dimension of the dataset. GFK is mainly affected by the noise and the size of the dataset.

F. IMPACT OF TRAINING DATA

1) GOAL

Our last experiment is to assess the impact of training data on the effectiveness of domain adaptation; that is, if we reduce the amount of training data, will *shift*-GAN still perform well? This is an important aspect in human activity recognition as annotating sensor data with activity labels is a time- and effort-consuming task and often incurs privacy risk.

2) EVALUATION PROCESS

We assess the stability of *shift*-GAN by varying the size of training and testing sets and observe the effects on the classification accuracy. For each experiment, we randomly select p percentage of target data for training and the remaining data for testing. We vary p from 20% to 80%, with a step 20%. We compute ten times each experiment and we report the averaged macro F1-scores for each task.

3) RESULTS

Figure 11 presents macro-F1 scores over different training percentages on 12 transfer learning tasks. From the results, we observe that *shift*-GAN achieves better macro-F1 across various learning tasks; more specifically, 3% over DAN, 16% over ADADM, 25% over GFK and 35% over TCA, by averaging the accuracy across all tasks and all the sensor percentages. When training data is small; i.e., 20%, *shift*-GAN only outperforms on 3 tasks: RA-LA, RL-LL, and RA-T; achieves similar accuracy to the best performing technique on 3 tasks: A-B, B-C, and DSADS-PAMAP; and gets lower accuracy on the other 6 tasks. The reason that DAN and ADADM perform better still resides in the fact that they have integrated feature transfer learning with classification.

All techniques are more stable in the accelerometer experiments independently of the percentage of training data compared to the binary ones. This is probably due to the feature representations of the datasets. The binary sensor data collected in the in-the-wild real-world environments is more sparse, noisy, and imbalanced across classes while the acceleration sensor data curated in the controlled environments is balanced in class distribution.

VI. CONCLUSION AND FUTURE WORK

This article proposes *shift*-GAN as a generalised unsupervised domain adaptation algorithm to enable transferring activity recognition systems across heterogeneous domains without the need of labels in the target domain. We have conducted extensive empirical experiments on a dozen transfer tasks in both binary sensor and accelerometer sensor data. Our results have shown that *shift*-GAN outperforms the state-of-the-art transfer learning techniques on most of the tasks. Our technique leverages the generative capability of GAN in mapping features between the source and target domain, even on high dimensional and high heterogeneous spaces. The promising results pave the future for scaling activity transfer to a large number of environments. That is, we will only need a well-annotated source domain data and *shift*-GAN can enable the activity transfer to the other environments without extra effort on labelling.

shift-GAN has demonstrates its generalisation capability as the feature mapping function learnt can be activity-independent. We can generate target sensor data from new, emerging activities on the source domain, without the need of retraining the bidirectional GAN. This generalisation capability will enable continuous domain adaptation; that is, when there is a new activity from the source domain, we can transfer it to the target domain.

shift-GAN has exhibited robustness to sensor noise, and outperformed the existing techniques on 7 out of 12 tasks. It has achieved higher accuracy on the sensitivity to training data experiments, however, it does not outperform when the training data is small and two domains have feature spaces of high dimension and heterogeneity. For the latter conditions, none of the existing techniques performs well; i.e., the accuracy is between 40% and 60%. One possible direction for improvement is to use feature extractor to project both source and target domains into a lower-dimensional reference space, based on which to perform feature transfer.

In the future, we will further demonstrate the generality of our approach by testing *shift*-GAN on other types of data including image and text data. Also, we will look into a more unified network to bypass the need for the SVM classifier.

REFERENCES

[1] A. Almeida and A. Alves, “Activity recognition for movement-based interaction in mobile games,” in *Proc. 19th Int. Conf. Hum.-Comput. Interact. Mobile Devices Services (MobileHCI)*. New York, NY, USA: Association for Computing Machinery, Sep. 2017, pp. 1–8.

[2] K. Altun and B. Barshan, "Human activity recognition using inertial/magnetic sensor units," in *Proc. 1st Int. Conf. Hum. Behav. Understand. (HBU)*. Berlin, Germany: Springer-Verlag, 2010, pp. 38–51.

[3] K. Altun, B. Barshan, and O. Tunçel, "Comparative study on classifying human activities with miniature inertial and magnetic sensors," *Pattern Recognit.*, vol. 43, no. 10, pp. 3605–3620, Oct. 2010.

[4] Y. Chang, A. Mathur, A. Isopoussu, J. Song, and F. Kawsar, "A systematic study of unsupervised domain adaptation for robust human-activity recognition," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 4, no. 1, pp. 1–30, Mar. 2020.

[5] Q. Chen, Y. Liu, Z. Wang, I. Wassell, and K. Chetty, "Re-weighted adversarial adaptation network for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7976–7985.

[6] Y. Chen, J. Wang, M. Huang, and H. Yu, "Cross-position activity recognition with stratified transfer learning," *Pervas. Mobile Comput.*, vol. 57, pp. 1–13, Jul. 2019.

[7] D. Cook, K. D. Feuz, and N. C. Krishnan, "Transfer learning for activity recognition: A survey," *Knowl. Inf. Syst.*, vol. 36, no. 3, pp. 537–556, Sep. 2013.

[8] J. D. Cook, "How smart is your home?" *Science*, vol. 335, no. 6076, pp. 1579–1581, 2012.

[9] L. Fang, J. Ye, and S. Dobson, "Discovery and recognition of emerging human activities using a hierarchical mixture of directional statistical models," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 7, pp. 1304–1316, Jul. 2020.

[10] K. D. Feuz and D. J. Cook, "Transfer learning across feature-rich heterogeneous feature spaces via feature-space remapping (FSR)," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 1, pp. 1–27, Mar. 2015.

[11] 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.*, vol. 17, no. 1, pp. 2030–2096, Jan. 2016.

[12] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* Washington, DC, USA: IEEE Computer Society, Jun. 2012, pp. 2066–2073.

[13] A. Gretton, A. J. Smola, J. Huang, M. Schmittfull, K. M. Borgwardt, B. Schölkopf, J. Candela, M. Sugiyama, A. Schwaighofer, and N. Lawrence, "Covariate shift by kernel mean matching," *Dataset Shift Mach. Learn.*, vol. 3, pp. 131–160, Jan. 2009.

[14] H. Hachiya, M. Sugiyama, and N. Ueda, "Importance-weighted least-squares probabilistic classifier for covariate shift adaptation with application to human activity recognition," *Neurocomputing*, vol. 80, pp. 93–101, Mar. 2012.

[15] L. Hu, M. Kan, S. Shan, and X. Chen, "Duplex generative adversarial network for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1498–1507.

[16] W. Jitkrittum, P. Sangkloy, M. W. Gondal, A. Raj, J. Hays, and B. Schölkopf, "Kernel mean matching for content addressability of GANs," in *Proc. 36th Int. Conf. Mach. Learn.*, vol. 97, K. Chaudhuri and R. Salakhutdinov, Eds. Long Beach, CA, USA: PMLR, Jun. 2019, pp. 3140–3151.

[17] T. L. M. van Kasteren, G. Englebienne, and B. J. A. Kröse, "Human activity recognition from wireless sensor network data: Benchmark and software," in *Activity Recognition in Pervasive Intelligent Environments* (Atlantis Ambient and Pervasive Intelligence), vol. 4, L. Chen, C. Nugent, J. Biswas, and J. Hoey, Eds. Amsterdam, The Netherlands: Atlantis Press, 2011, doi: 10.2991/978-94-91216-05-3_8.

[18] M. A. A. H. Khan and N. Roy, "TransAct: Transfer learning enabled activity recognition," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2017, pp. 545–550.

[19] M. W. Kouw, J. P. Laurens van der Maaten, H. J. Krijthe, and M. Loog, "Feature-level domain adaptation," *J. Mach. Learn. Res.*, vol. 17, no. 171, pp. 1–32, 2016.

[20] C. Krupitzer, T. Sztyler, J. Edinger, M. Breitbach, H. Stuckenschmidt, and C. Becker, "Hips do lie! A position-aware mobile fall detection system," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. (PerCom)*, Mar. 2018, pp. 1–10.

[21] H. Li, S. J. Pan, S. Wang, and A. C. Kot, "Domain generalization with adversarial feature learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5400–5409.

[22] J. Liao, L. Stankovic, and V. Stankovic, "Detecting household activity patterns from smart meter data," in *Proc. Int. Conf. Intell. Environ.*, Jun. 2014, pp. 71–78.

[23] M. Long, Y. Cao, Z. Cao, J. Wang, and M. I. Jordan, "Transferable representation learning with deep adaptation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 12, pp. 3071–3085, Dec. 2019.

[24] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. 32nd Int. Conf. Int. Conf. Mach. Learn. (ICML)*, vol. 37, 2015, p. 97–105.

[25] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2200–2207.

[26] L. H. Morsing, O. A. Sheikh-Omar, and A. Iosifidis, "Supervised domain adaptation: A graph embedding perspective and a rectified experimental protocol," 2020, *arXiv:2004.11262*. [Online]. Available: https://arxiv.org/abs/2004.11262

[27] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto, "Unified deep supervised domain adaptation and generalization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5716–5726.

[28] K. Muandet, B. Sriperumbudur, K. Fukumizu, A. Gretton, and B. Schölkopf, "Kernel mean shrinkage estimators," *J. Mach. Learn. Res.*, vol. 17, no. 48, pp. 1–41, 2016.

[29] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.

[30] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[31] C. Pereyda, N. Raghunath, B. Minor, G. Wilson, M. Schmitter-Edgecombe, and D. J. Cook, "Cyber-physical support of daily activities: A robot/smart home partnership," *ACM Trans. Cyber-Phys. Syst.*, vol. 4, no. 2, pp. 1–24, Feb. 2020.

[32] X. Qin, Y. Chen, J. Wang, and C. Yu, "Cross-dataset activity recognition via adaptive spatial-temporal transfer learning," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 3, no. 4, pp. 1–25, Dec. 2019.

[33] V. Radu, C. Tong, S. Bhattacharya, N. Lane, C. Mascolo, M. Marina, and F. Kawsar, "Multimodal deep learning for activity and context recognition," in *Proc. Ubicomp*, 2018, pp. 1–27.

[34] P. Rashidi and D. J. Cook, "Multi home transfer learning for resident activity discovery and recognition," in *Proc. KDD Knowl. Discovery Sensor Data*, 2010, pp. 56–63.

[35] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *Proc. 16th Int. Symp. Wearable Comput.*, Jun. 2012, pp. 108–109.

[36] A. R. Sanabria and J. Ye, "Unsupervised domain adaptation for activity recognition across heterogeneous datasets," *Pervas. Mobile Comput.*, vol. 64, Apr. 2020, Art. no. 101147.

[37] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA. Washington, DC, USA: IEEE Computer Society, Jun. 2018, pp. 3723–3732.

[38] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *Artificial Neural Networks and Machine Learning—ICANN*, V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, and I. Maglogiannis, Eds. Cham, Switzerland: Springer, 2018, pp. 270–279.

[39] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7167–7176.

[40] T. L. M. van Kasteren, G. Englebienne, and B. J. A. Kröse, "Transferring knowledge of activity recognition across sensor networks," in *Proc. 8th Int. Conf. Pervas. Comput.*, P. Floréen, A. Krüger, and M. Spasojevic, Eds. Berlin, Germany: Springer, 2010, pp. 283–300.

[41] T. L. M. van Kasteren, G. Englebienne, and B. J. A. Kröse, *Human Activity Recognition From Wireless Sensor Network Data: Benchmark and Software*. Paris, France: Atlantis Press, 2011, pp. 165–186.

[42] J. Wang, Y. Chen, L. Hu, X. Peng, and P. S. Yu, "Stratified transfer learning for cross-domain activity recognition," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. (PerCom)*, Mar. 2018, pp. 1–10.

[43] G. Wilson and D. J. Cook, "A survey of unsupervised deep domain adaptation," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 5, Sep. 2020, Art. no. 51, doi: 10.1145/3400066.

[44] M. Xu, J. Zhang, B. Ni, T. Li, C. Wang, Q. Tian, and A. Zhang, "Adversarial domain adaptation with domain mixup," in *Proc. 34th AAAI Conf. Artif. Intell.* Palo Alto, CA, USA: AAAI Press, 2020, pp. 6502–6509.

[45] X. Xu, X. Zhou, R. Venkatesan, G. Swaminathan, and O. Majumder, "D-SNE: Domain adaptation using stochastic neighborhood embedding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2492–2501.

[46] J. Ye, "SLearn: Shared learning human activity labels across multiple datasets," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. (PerCom)*, Mar. 2018, pp. 1–10.

[47] J. Ye, S. Dobson, and F. Zambonelli, "XLearn: Learning activity labels across heterogeneous datasets," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 2, pp. 1–28, Mar. 2020.

[48] Z. Yi, H. Zhang, P. Tan, and M. Gong, "DualGAN: Unsupervised dual learning for image-to-image translation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2849–2857.

[49] F. Zambonelli, "Pervasive urban crowdsourcing: Visions and challenges," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. Workshops (PERCOM Workshops)*, Mar. 2011, pp. 578–583.

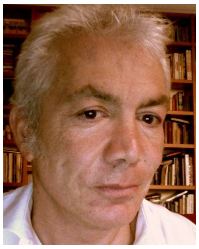
[50] S. Zhao, G. Wang, S. Zhang, Y. Gu, Y. Li, Z. Song, P. Xu, R. Hu, H. Chai, and K. Keutzer, "Multi-source distilling domain adaptation," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 7, pp. 12975–12983.

[51] Z. Zhao, Y. Chen, J. Liu, Z. Shen, and M. Liu, "Cross-people mobile-phone based activity recognition," in *Proc. 22nd Int. Joint Conf. Artif. Intell. (IJCAI)*. Palo Alto, CA, USA: AAAI Press, 2011, pp. 2545–2550.

[52] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2223–2232.

[53] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, Jan. 2021.

ANDREA ROSALES SANABRIA is currently pursuing the Ph.D. degree with the School of Computer Science, University of St Andrews. Her research interest includes unsupervised domain adaptation in sensor-based human activity recognition. She specialized in generative adversarial networks and contrastive learning.



FRANCO ZAMBONELLI (Fellow, IEEE) received the Ph.D. degree in computer science and engineering from the University of Bologna, in 1997. He is currently a Professor of computer science with the University of Modena and Reggio Emilia. His research interests include pervasive computing, multi-agent systems, and self-adaptive and self-organizing systems. He is also the ACM Distinguished Scientist and a member of the Academia Europaea. He is on the editorial board of the *ACM Transactions on Autonomous and Adaptive Systems*, *Pervasive and Mobile Computing* journal (Elsevier), *IEEE Technology and Society Magazine*, and *The Computer Journal* (BCS).

JUAN YE received the Ph.D. degree in computer science from University College Dublin. She is currently a Senior Lecturer with the School of Computer Science, University of St Andrews. Her research interests include adaptive pervasive systems, specializing in sensor-based human activity recognition, sensor fusion, context awareness, domain adaptation, and continuous learning.