TransNet: Minimally-Supervised Deep Transfer Learning for Dynamic Adaptation of Wearable Systems

SEYED ALI ROKNI, Washington State University, USA
MARJAN NOUROLLAHI, Washington State University, USA
PARASTOO ALINIA, Washington State University, USA
SEYED IMAN MIRZADEH, Washington State University, USA
MAHDI PEDRAM, Washington State University, USA
HASSAN GHASEMZADEH, Washington State University, USA

Wearables are poised to transform health and wellness through automation of cost-effective, objective, and real-time health monitoring. However, machine learning models for these systems are designed based on labeled data collected, and feature representations engineered, in controlled environments. This approach has limited scalability of wearables because (i) collecting and labeling sufficiently large amounts of sensor data is a labor-intensive and expensive process; and (ii) wearables are deployed in highly dynamic environments of the end-users whose context undergoes consistent changes. We introduce *TransNet*, a deep learning framework that minimizes the costly process of data labeling, feature engineering, and algorithm retraining by constructing a scalable computational approach. TransNet learns general and reusable features in lower layers of the framework and quickly reconfigures the underlying models from a small number of labeled instances in a new domain, such as when the system is adopted by a new user or when a previously unseen event is to be added to event vocabulary of the system. Utilizing TransNet on four activity datasets, TransNet achieves an average accuracy of 88.1% in cross-subject learning scenarios using only one labeled instance for each activity class. This performance improves to an accuracy of 92.7% with five labeled instances.

CCS Concepts: • **Human-centered computing** → *Ubiquitous and mobile computing*; • **Computing methodologies** → *Machine learning*; • **Computer systems organization** → *Sensor networks.*

Additional Key Words and Phrases: Wearable computing, machine learning, deep learning, transfer learning, adaptation, reconfiguration, reliability

ACM Reference Format:
Seyed Ali Rokni, Marjan Nourollahi, Parastoo Alinia, Seyed Iman Mirzadeh, Mahdi Pedram, and Hassan Ghasemzadeh. 2021. TransNet: Minimally-Supervised Deep Transfer Learning for Dynamic Adaptation of Wearable Systems. *ACM Trans. Des. Autom. Electron. Syst.* 1, 1 (April 2021), 31 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Authors' addresses: Seyed Ali Rokni, Washington State University, School of Electrical Engineering and Computer Science, Pullman, WA, 99164–2752, USA, alirokni@eecs.wsu.edu; Marjan Nourollahi, Washington State University, School of Electrical Engineering and Computer Science, Pullman, WA, 99164–2752, USA, m.nourollahidarabad@wsu.edu; Parastoo Alinia, Washington State University, School of Electrical Engineering and Computer Science, Pullman, WA, 99164–2752, USA, parastoo.alinia@wsu.edu; Seyed Iman Mirzadeh, Washington State University, School of Electrical Engineering and Computer Science, Pullman, WA, 99164–2752, USA, seyediman.mirzadeh@wsu.edu; Mahdi Pedram, Washington State University, School of Electrical Engineering and Computer Science, Pullman, WA, 99164–2752, USA, mahdi.pedram@wsu.edu; Hassan Ghasemzadeh, Washington State University, School of Electrical Engineering and Computer Science, Pullman, WA, 99164–2752, USA, hassan@eecs.wsu.edu.

1084-4309/2021/4-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The increasing ubiquity of wearable devices with embedded sensors has created a unique opportunity to transform health and wellness through automation of cost-effective, objective, continuous, and real-time health monitoring and interventions. Currently, however, computational models (e.g., machine learning and signal processing) for these systems are designed based on labeled training data collected, and representative features devised, in controlled environments. This approach for training computational models has created several real impediments to the scalability of wearable technologies. In particular, we note that collecting sufficiently large amounts of labeled/annotated sensor data is a time consuming, labor-intensive, and expensive process that has also been identified as a major barrier to personalized and precision medicine [1–4]. Furthermore, wearables are deployed in highly dynamic and uncontrolled environments of the end-users whose physical, behavioral, social, and environmental context undergoes consistent changes. Such changes result in drastic performance degradation of the computational models that are trained in laboratory settings or traditional confines of healthcare environments [5–7]. Therefore, it is imperative that we develop reconfigurable computational models with minimal human supervision as wearable sensor systems, settings in which they are utilized, and their configuration changes.

The configuration of a wearable sensor system can change due to changes in physical sensors (e.g., sensor addition, removal, displacement, misplacement, misorientation, upgrade), users (e.g., adoption of the system by a new user), context behavior (e.g., aging, changes in health conditions), or application scope (e.g., new events to be monitored by the system). These evolutions of the system result in a shift in the distribution of the sensor data and therefore introduce a new domain. For the purpose of this article, we refer to *domain* as a configuration of the system where variations in the distribution of the sensor data over time are minimal. However, a domain shift occurs when the configuration of the wearable system changes. Feature engineering is an important preprocessing task with the goal of designing discriminative attributes of the signals to transform raw sensory data to a format that can be effectively used by an event classifier (e.g., activity recognition algorithm) in a particular domain [8]. However, the performance of the classifier is highly dependent on the expert-driven features [9]. As the system begins to transition from one domain to another, we will need to secure additional supervision from a human expert by redoing the costly process of data collection, labeling, and feature engineering.

Addressing the problem of expanding the computational capabilities of a networked wearable system from one domain with a pre-trained classifier to a new domain with a different configuration is quite challenging. We aim to address this problem through cross-domain knowledge transfer to improve the learning performance of the model while avoiding the expensive data collection and labeling, feature engineering, and algorithm retraining efforts. Our general approach relies on learning the underlying representation of the sensory data in one domain, called *source*, and quickly adapting the learned representation in a new domain, called *target*, with minimal supervision/ training data. The proposed approach is motivated by recent advancements in deep learning research focused on representation learning methodologies to avoid application-specific feature engineering.

In this article, we present development and validation of TransNet[1], a deep learning framework aimed at facilitating cross-domain knowledge transfer by learning effective data representation/ features from sensor data and constructing a computational model based on the learned features. TransNet is composed of three main modules including a preprocessing module, a convolutional stack, and a recognition module each of which consists of one or more processing layers in our deep learning architecture. The features learned in the first two modules (i.e., preprocessing module

[1]TransNet source code and experimental data collected in this project are available at https://github.com/ali-rokni/TransNet

and convolutional stack) are domain-agnostic and therefore the lower layers are reused when the system is utilized in a new domain. When new labeled instances are acquired in the new domain, these labeled instances are used to retrain the recognition module only, resulting in TransNet to quickly master event classification in the new domain. This design strategy results in temporal and spatial improvements of TranNet as more labeled instances are obtained in the new domain. In addition to the knowledge transfer capabilities, TransNet offers a computationally-simple structure through a small number of layers, automatically tunable parameters, and a small number of training epochs. TransNet introduces a preprocessing layer including a combination of quantization and an embedding layers that compress the input data into a lower dimension, therefore result in a computationally-efficient design having about one-order-of-magnitude less number of parameters compared to the state-of-the-art deep learning networks. In addition, TransNet proposes a computationally efficient design for the application of physical activity recognition using wearable sensors with limited resource, including power and training data that beats the state-of-the-art activity recognition techniques.

2 PRELIMINARIES

2.1 Motivation

When used in uncontrolled environments, wearables face many forms of uncertainty that dramatically impact their performance. For instance, activity recognition accuracy declines 66.3% due to sensor displacement [5, 10], energy expenditure estimation error increases by a factor of 2.3 due to sensor displacement [11], and activity recognition accuracy drops 61.2% when the system is utilized by new users [6]. In the current practice, a human expert needs to repeat the costly process of data collection and feature engineering upon a domain change, mainly due to the inherent shift in the distribution of the sensor data. As a result, unless novel learning frameworks are designed, recent findings [12–19] suggest that compensating for the uncertainties requires collecting and labeling large amounts of sensor data for various configurations (i.e., new user, sensor platform, context, event, and system configuration). We aim to address this challenge by developing a scalable, reconfigurable, and computationally-simple deep-learning-based semi-supervised domain adaptation framework, TransNet, in this article.

Fig. 1 shows how TransNet can be utilized in three transfer scenarios, including transfer across different body locations, across different users, and across different vocabulary/activities. First, we train a deep convolution neural network on one domain (e.g., a user wearing a sensor on the wrist), then transfer the trained network to a new domain (e.g., the user wearing the sensor on the hip), then freeze all the layers except the classification layers including the fully connected and softmax layers and train the network only updating the weights from the unfrozen layers using few labeled samples in the new domain.

2.2 Pilot Application and Assumptions

Our pilot application in this paper is activity recognition where wearable inertial sensors are used to detect human movements such as ‘walking’, ‘running’, or ‘sitting’ [20]. A sensing device typically consists of three categories of components: (i) embedded sensors such as accelerometer, gyroscope, and magnetometer to capture various human body attributes including 3D acceleration, 3D angular velocity, and direction/magnitude of the magnetic field; (ii) embedded software to perform signal processing and machine learning; and (iii) a radio module to communicate locally-processed results to the outside world.

Each embedded sensor produces several sensor streams. For example, a 3D accelerometer measures body acceleration in x, y, and z directions. We assume that each sensor stream is sampled at

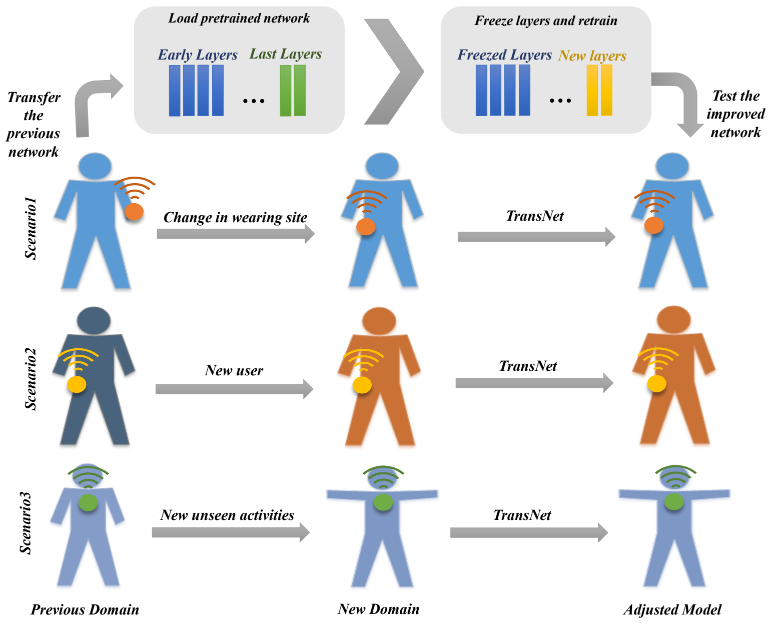


Fig. 1 Several transfer learning scenarios in which TransNet can be utilized.

a fixed sampling frequency. Let $S = \{s_1, \dots, s_n\}$ be the set of all sensor streams forming a wearable sensor network. Assume that each s_i is sampled at a frequency of f Hz. Thus, sensor stream s_i produces one measurement $v_{it} = v^*_{it} + \epsilon_{it}$ at time t where v^*_{it} denotes the true body attribute (e.g., acceleration in x-axis) pronounced by s_i, and ϵ_{it} represents measurement error and/or noise. The collective sensor streams in S produce a vector of the form:

$$\mathcal{V}_{it} = [v_{1t}; v_{2t}; \dots; v_{nt}] \tag{1}$$

Typically, each signal stream undergoes a smoothing process (e.g., low-pass *filtering*) to reduce high frequency noise. Because the sensor continuously captures body motions in free-living settings, 'start' and 'end' of each activity is unknown *a priori*. Therefore, the next phase in an activity recognition pipeline is *segmentation*, which identifies 'start' and 'end' points of the movements. Because the underlying signals, due to body motions, are quasi-periodic, the common segmentation approach is a sliding time window with a small overlap between adjacent windows and a window-size that is large enough to include at least one instance of human activity. The segmentation window of size w on sensor stream s_i produces an input array $C_t = [C_{1t}; C_{2t}; \dots; C_{nt}]$ where

$$C_{it} = [v_{it}, \dots, v_{i(t+w-1)}] \tag{2}$$

The array C_t is fed into TransNet as a collection of streaming time-series signals. We refer to this array of filtered and segmented signal as an *instance* acquired at time t.

Let $\mathcal{A}=\{a_1, a_2, \dots, a_m\}$ be an *activity vocabulary*, also called *label space*, composed of m activities of interest for a given domain. The activity recognition task consists of the vocabulary $\mathcal{A}$ and a conditional probability distribution $P(a|C_t)$, which is the probability of inferring label $a \in \mathcal{A}$ given an observed instance C_t. Fig. 2 shows the process of activity recognition. Using traditional classification methods, a set of expert-engineered features are extracted from the segmented signals. A feature vector is then fed to the classifier. Although these features may work well for a specific

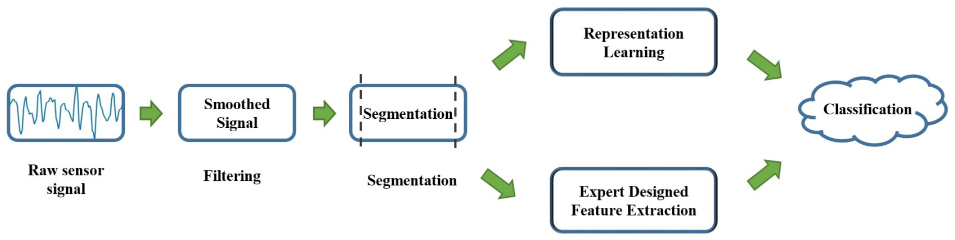


Fig. 2 Conventional activity recognition (based on expert-engineered features) versus TransNet (based on deep representation learning).

domain, their effectiveness declines when the target task changes and therefore an expert needs to re-engineer the features for the new domain. In contrast, deep representation learning approach, which is our general approach in designing TransNet, constructs several layers of a neural network to extract the underlying representation of the input signal such that the obtained representation is less domain-specific and more robust to changes in the target task/domain [21–23].

3 TRANSNET ARCHITECTURE

Fig. 3 shows our proposed neural-network-based architecture for TransNet. TransNet constructs a deep learning structure to learn a useful representation of sensory data from human activities. The representations/ features in the lower layers are learned in the source domain and are not domain-specific; hence, they are reused in a target domain [21, 24]. As a result, TransNet can quickly reconfigure its recognition module with minimal training data to reliably represent the target task.

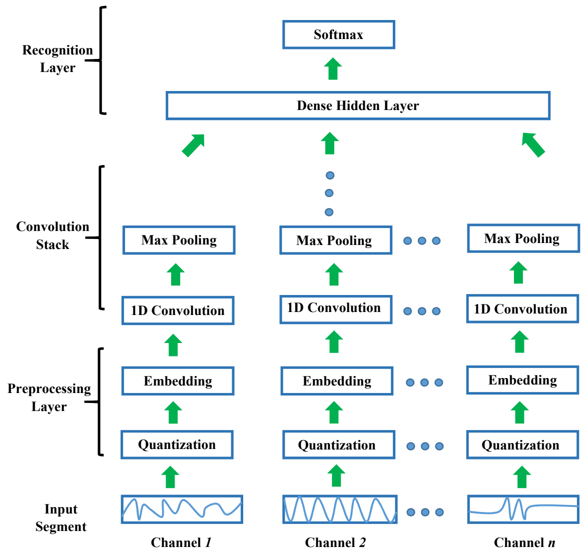


Fig. 3 Overall structure of TransNet.

3.1 Preprocessing Module

The input layer in TransNet consists of n channels associated with the filtered and segmented signal streams. The input segments on each channel are passed to a *quantization* layer. The quantization process aims to reduce the sensitivity of the model to small changes in sensor readings due to domain change, and therefore makes the model more robust when transitioning from one domain to another.

In general, a quantizer has a countable set of possible output values that have fewer members than the set of possible input values. Therefore, we can define the quantization process, Q, as a mapping of the input values from a larger possible set $\mathcal{L}$ (e.g., sensor readings) to output values in a countable smaller set $\mathcal{S}$ (e.g., a fixed set of integer values).

$$Q : \mathcal{L} \rightarrow \mathcal{S} \tag{3}$$

where $|\mathcal{L}| \gg |\mathcal{S}|$. For example, a uniform rounding quantizer with a quantization step size equal to some value Δ for an input value x is defined as

$$Q(x) = \Delta \cdot \left\lfloor \frac{x}{\Delta} + \frac{1}{2} \right\rfloor \tag{4}$$

Because the output of the quantizer has fewer members than the possible input values, quantization leads to smaller input space, and consequently more memory-efficient representation of the input.

By applying a rounding quantization on sensor streams on each channel, we obtain a sequence of integer values $S_t = \{s_t, \ldots, s_{t+w-1}\}$ such that each s_i has a range $\pm\lfloor|B|\rfloor + 1$ where $\pm|B|$ denotes the dynamic range of the sensor associated with s_i. These integer values can be viewed as primitives of a human movement language as perceived by a particular sensor [25]. Although each s_i can potentially have any values within the range $[-\lfloor|B|\rfloor + 1, +\lfloor|B|\rfloor + 1]$, only a small portion of this range is used by an activity instance and this portion varies from one activity to another.

The sequence of primitives is analogous to the sequence of words in spoken language, which is the focus of research in Natural Language Processing (NLP). In the NLP research, a common practice is to use an embedding layer as the first layer of the network. An embedding layer converts each input data (word) to a dense vector of a fixed size. The training objective of the embedding layer is to learn word vector representations that are good at predicting the nearby words [26]. The network first initializes its weights by random values. It then iteratively adjusts the weights during training to minimize the error that it makes when using words to predict their contexts. As a result of this embedding layer, we are able to capture semantic relationships in language that are very difficult to capture otherwise. Although the inclusion of an embedding layer was initially incepted in the NLP research, its application has been extended to other areas. For example, the use of embedding layers to encode student behavior in massive open online courses (MOOCs) has been explored by prior research [27]. Because the additive representation of human movements is similar to the representation of human speech [28], we expect an embedding layer to be an effective approach in abstracting details of sensor readings for human activity recognition. To this end, we feed the sequence S_t to an embedding layer to generate a distributed representation of the input data [29, 30].

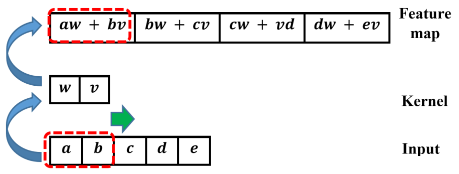
Considering the analogy between a sequence of quantized sensor readings for a particular activity with a sequence of words in a particular sentence, we aim to encode more useful information with each sampled data point by analyzing its context (i.e., neighboring sensor reading). This approach is also consistent with the physical properties of the human movements. For example, human *gait* (i.e., walking) is composed of two phases including *stance* and *swing*. The stance phase is the time when the foot is on the ground, which comprises more than half of the walking cycle. In a swing phase, however, one foot is on the ground and the other in the air. For part of the stance phase, both feet will be on the ground for a period of time. While in the stance phase, acceleration samples of an ankle-worn sensor node are very similar to that of *standing* activity. However, using an embedding layer and analyzing the context of the samples, these acceleration readings of *walking* are interpreted differently than those in *standing* because the neighboring samples are utilized for a more accurate explanation of each sensor reading.

Furthermore, the addition of an embedding layer is useful in reducing the effect of instrumental noise because similar activities captured by different sensors or performed by different users can exhibit close representation in this space. In particular, while the neural network is being trained, the embedding vectors are updated, which leads to the similarities between observations that can be found in a multi-dimensional embedding space. To the best our knowledge, our work is the first study that introduces an embedding layer to model inertial sensor readings in deep learning architectures.

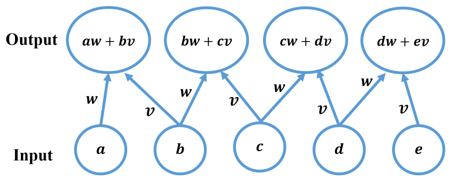
3.2 Convolutional Stack

The convolutional stack is composed of convolutional and max-pooling layers, as discussed in this section.

3.2.1 Convolutional Layers. Right after the embedding layer, several 1D convolutional layers are stacked to capture local dependencies and scale-invariant characteristics of the input. Suppose we want to perform a 1D convolution on an input channel of length N using a kernel of length $L \ll N$. As shown in Fig. 4a, the convolution process slides the kernel over the input channel and computes the dot product of the kernel and the corresponding chunk of the temporal input. Applying this kernel on the entire input segment, we compute the dot products in a vector, referred to as *activation map* or *feature map*. To create an analogy to the traditional feature extraction process, a kernel can be viewed as a *feature extractor* and the feature map can be considered as the corresponding *feature value*. While the toy example in Fig. 4a shows that the kernel slides over the input data with a *stride length* of 1, the kernel can have a longer stride length (i.e., >1) in general.



(a) A simple example of 1D convolution



(b) Sparse connectivity and parameter sharing:

Fig. 4 An example of 1D convolution with a kernel of length 2 and input of length 5. (a) As kernel slides over input, the dot product of kernel and the partial input is computed as one entry of feature map.(b) The same convolution example represented in a feed-forward neural network.

At each convolutional layer, the network uses convolution instead of general matrix multiplication. We can consider a segment of an input channel as a discrete input function $I \in [1, N] \rightarrow \mathcal{R}$, and the kernel as a discrete function $K \in [1, L] \rightarrow \mathcal{R}$. The feature map $F \in [1, \lfloor (N-L)/d \rfloor + 1] \rightarrow \mathcal{R}$ resulted from the convolution between I and K with a stride length of d is defined as:

$$F(i) = \sum_{t=1}^{L} K(t) \cdot I(t + (i-1)d) \tag{5}$$

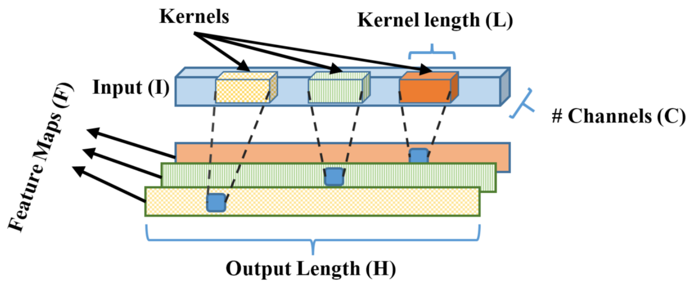


Fig. 5 Three different kernels operate on a multi-channel input and produce three different feature maps.

We note that increasing the stride length will reduce the length of the output.

As previously discussed, kernels act as feature extractors. To extract more features, at each convolutional layer, we use multiple kernels where each kernel looks for a specific type of template or concept in the input. Therefore, the output of each convolutional layer is multiple feature maps corresponding to different kernels. Fig. 5 shows an input with C channels. To produce the feature map by applying kernels on all channels of this input, kernels should also have a depth of C. We compute multiple feature maps, each of which corresponds to one kernel. To extract more complex features, multiple convolutional layers are stacked on top of each other. The convolutional stack enables the network to learn the hierarchy of features. The kernels at the earlier layers usually represent low-level features and kernels in higher layers correspond to more complex features (i.e., features of features) [23, 31].

As illustrated in Fig. 4b, compared to a typical feed-forward layer in neural networks, a convolutional layer carries two important features: *sparse connectivity* and *parameter sharing*. Since the length of kernels are significantly smaller than the input size, the number of connections from the input layer to an output unit/ node in the feature map is limited to a few input nodes which improve the memory and computational efficiency.

To model the convolutional layer as a normal feed forward layer, we replicate each small kernel across every position of the input (except for boundaries). This phenomenon is referred to as *parameter sharing*, which allows kernels to capture useful features regardless of their temporal position. Since we use a sliding window for segmentation, the sensor readings associated with an activity may appear shifted from one segment to another. This temporal variation of the events across various instances leads to similar patterns occurring at different time positions. To be able to capture the same set of patterns on the shifted signals, the convolution function should be translate-invariant. Particularly, *parameter-sharing* helps the convolutional layers to become invariant to local translations [32].

3.2.2 Max Pooling. After applying the non-linear activation function on the output of the convolutional layer, a pooling function is commonly used to create a more compact and manageable representation of the data. In general, a pooling function alters the output of a unit with a summary statistics of nearby outputs [32]. For TransNet, we use max-pooling [33], which replaces the output of a unit with a maximum output of the nearby units.

3.3 Recognition Module

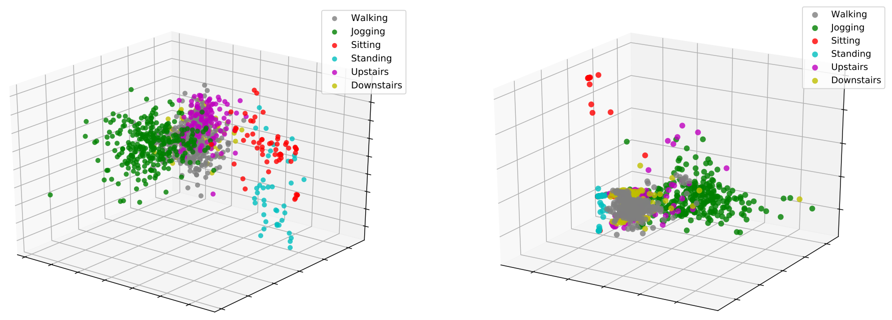
On top of the convolutional stack, TransNet has a *recognition module*, which starts with a densely connected layer. Contrary to convolutional layers, instead of preserving the temporal structure of the data, this module aggregates outputs of all channels and builds a scoring function. This combined representation forms an efficient feature space to classify instances of different activities. In Fig. 6, we compare the separability power of a set of expertly engineered features with the learned representation by TranNet. Particularly, we perform a Linear Discriminant Analysis (LDA) on 1000 instances of different activities collected by a 3D accelerometer on a smartphone [8]. As shown in Fig. 6a, different activities are much more distinguishable in the learned representation by TransNet compared to the hand-crafted feature space illustrated in Fig. 6b.

Note that the notion of hand-crafted features refers to a set of features that are defined by the system designer. In general, the choice of hand-crafted features can be informed by domain knowledge. However, in many applications, the decision of what features are most effective for machine learning algorithm design is hard to make by only incorporating prior knowledge. This situation is, in particular, true for time-series signals where a human interpretation and understanding of the raw sensor signals are difficult to develop. In activity recognition tasks, it is common to compute a large set of features from inertial sensors first. The large feature set is then fed into a feature selection algorithm to choose an effective subset of features for inclusion in the classification process. For example, in our experiments for alternative classifier design, discussed in details in Section 6.4, we first extract 18 features from each signal segment associated with an inertial sensor stream. We then use χ^2 method to eliminate irrelevant features from the feature vector. The remaining features are referred to as final hand-crafted features that are also used to plot the data in Fig. 6b.

The recognition process finishes with a softmax function which normalizes the scores and the activity with maximum score is returned as the predicted class. Particularly, the softmax function, which is defined as

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{m} e^{z_k}} \tag{6}$$

where the z_j is un-normalized score of activity j. Particularly, softmax function is a generalization of the logistic function that squashes the real value scores to the range $[0, 1]$.



(a) Learned representation

(b) Engineered feature space

Fig. 6 Linear Discriminate Analysis on (a) learned features by TransNet (b) hand-crafted features.

4 CROSS-DOMAIN ADAPTATION

As discussed previously, wearable technologies are quite dynamic in nature and due to their exposure to end-users their configuration and utilization change over time. Predicting all possible system configurations and training a specific model for such dynamically evolving systems is an

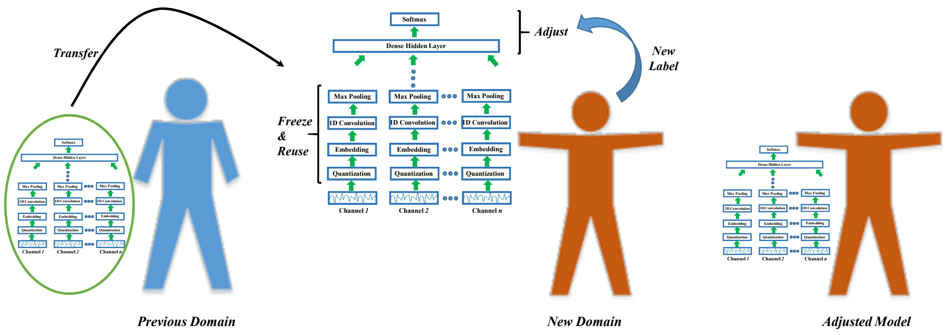


Fig. 7 A scenario of adapting TransNet weights in a new context.

unrealistic solution. Therefore, it is desirable to develop an adaptive solution capable of adapting easily as new changes in the system occur. Representation learning via deep neural networks enables us to learn simple-to-complex features that are not task-specific and could be conveniently reused in other related problems and machine learning tasks. In this section, we describe our transfer learning method that allows us to adapt the model weights for a new task with minimal supervision. In particular, having a previously trained model in a specific domain (e.g., trained with data from a particular user), we aim to adjust the model to perform reliably in a new domain (e.g., when the system is adopted by a new user with unseen sensor data) by acquiring only few labeled instances in the new domain.

Algorithm 1 describes our basic process for conducting transfer learning using TransNet. As illustrated in Fig. 7, in our transfer learning method, we first freeze all layers of previously trained network except for the top layer (i.e., recognition layer). This helps us reuse the learned weights of the lower layers in the new domain and converge faster by reducing the learnable parameters to only the top layer. We then feed the network with one instance of each class from the new domain. That is, the system obtains a training dataset $< X_{tr}, Y_{tr} >$ during each data acquisition session in the new domain. The network starts to adjust the weights of the top layer by performing multiple training epochs on the fed instances. As the training process continues, we measure the performance of the adjusted network on the new domain as well as the fed instances. If the performance of the network is not acceptable yet, it indicates that further training of the network is needed. However, to prevent overfitting, we stop training with more epochs if we observe a large gap between the performance of TransNet on $< X_{tr}, Y_{tr} >$ and that of the validation data. At this time, if TransNet does not have the acceptable performance, we repeat the training process by acquiring more labeled instances in the new domain.

ALGORITHM 1: Cross-domain adaptation in TransNe.

```
Input: Acceptable classifier error $\epsilon$; previously trained network $h$; threshold $\tau$
Freeze all layers except the top layer;
repeat
    $< X_{tr}, Y_{tr} > \leftarrow$ acquire a batch of labeled training instances in the new domain;
    while (train error - validation error) $\leq \tau$ do
        Update the top layer of $h$ by training for more number of epochs on $< X_{tr}, Y_{tr} >$;
    end
until classifier error $\leq \epsilon$;
```

Contrary to many traditional classifiers, which need to be trained on the entire training data, TransNet can be used to accommodate incremental training. This online learning feature helps TransNet to incrementally improve upon obtaining more labeled instances in the new domain without the need for retraining from scratch. In the next subsection, we will discuss details on how TransNet can be deployed in real-world settings for incremental learning.

5 DEPLOYMENT IN REAL-WORLD SETTINGS

To deploy TransNet in real-world situations and utilize the approach presented in Algorithm 1 for transfer learning in end-user settings, few deployment details need to be taken into consideration. These considerations are associated with two tasks, including the acquisition of labeled training data and performance assessment of the classification model as TransNet is being utilized in a new domain. Furthermore, one needs to have a reasonable understanding of the hardware capacity of existing mobile and wearable devices. In this section, we briefly discuss these important deployment issues.

5.1 Gathering Labeled Sensor Data in New Domains

During each data acquisition iteration in Algorithm 1, TransNet acquires labeled sensor data to form a training dataset $< X_{tr}, Y_{tr} >$. This can be reliably achieved by incorporating an active learning algorithm that examines unlabeled sensor data in the new domain and selects those instances that are most informative with respect to the existing model (i.e., evolving neural network model).

Active learning can be used to iteratively interact with a human user to retrieve important information that can help with the improved performance of a machine learning model [34–40]. The idea of using active learning for activity recognition has been also studied in the past [41–45]. Therefore, to deploy our domain adaptation approach described in Algorithm 1, an active learning algorithm can be utilized to acquire the labeled training data in the new domain.

ALGORITHM 2: Active learning to acquire training data for cross-domain adaptation.

```
Input: New domain unlabeled data $\mathcal{X}$, query budget $B$
Output: Labeled training dataset $< X_{tr}, Y_{tr} >$
$< X_{tr}, Y_{tr} > \leftarrow \emptyset$;
$C \leftarrow$ Partition $\mathcal{X}$ into $B$ disjoint clusters $\{C_1, \ldots, C_B\}$;
foreach $C_b \in C$ do
    Compute $\mathcal{E}(I_i)$ for all $X_i \in C_b$;
    Find $X_i \in C_b$ with highest value of $\mathcal{E}(I_i)$;
    Query human expert to annotate $X_i$;
    Add labeled instance $(X_i, Y_i)$ to the $< X_{tr}, Y_{tr} >$;
end
```

In Algorithm 2, we show a general active learning approach that can be incorporated into the data acquisition section of Algorithm 1. This greedy algorithm iteratively chooses the best candidate sensor data (i.e., a signal segment) from a set of unlabeled data items in the new domain for data annotation by a human expert and for inclusion in the training dataset. Assuming that at each iteration of the domain adaptation process, we aim to acquire B labeled instances, Algorithm 2 first clusters the unlabeled data into B groups. From each cluster C_b, it then selects the most informative instance according to the expected gain criterion $\mathcal{E}(I_i)$, a measure of informativeness of the selected instance. A commonly used criterion for the informativeness of the data is the uncertainty of the evolving model with respect to the given sensor data. To this end, one can use entropy to measure how certain the model is about its predicted label for a given instance X_i. Therefore, $\mathcal{E}(I_i)$ can be written as

$$\mathcal{E}(I_i) = -\sum_{j=1}^{n} P_{ij} \log P_{ij} \tag{7}$$

where P_{ij} denotes the probability of instance X_i being classified as activity $a_j \in \mathcal{A}$. Because the evolving neural network model is less certain to classify instances that carry a higher entropy, such instances will naturally be more informative if labeled and used for model training [42].

5.2 Continuous Assessment of the Evolving Target Model

When deploying TransNet in real-world settings, we also need a validation dataset based on which the performance of the classifier is evaluated as newly acquired labeled data are being fed to the network in the new domain. We note that the sensor data of the new domain are used to query a human expert (e.g., the end-user) for activity labels. As mentioned previously, the sensor data are queried in batches. Each labeled batch of the sensor data can be used as a validation dataset prior to being fed into the neural network for model adaptation. This analysis of the performance of the evolving network can be accomplished through a k-fold cross validation or even one-leave-out validation strategy because the training process is computational simple as only the top layer of the network is updated during model adaptation.

5.3 Resource Requirements of TransNet

As TransNet is implemented for cross-domain adaption in mobile devices, another consideration when deploying it in a real-world setting is the compute and storage requirements. The main components of the pipeline of TransNet are training data collection and pre-processing, training the network, and the network forward-pass on the new data. Recent research has proposed promising techniques to efficiently deploy deep neural networks on mobile devices. Examples of such techniques include distributing the training process, and network compression [46–51]. Inspired by the previous research, to efficiently execute TransNet on mobile devices such as smartphones, the data pre-processing and network training can be performed offline using high-performance computer nodes or on cloud centers. Moreover, the forward-pass, which includes primarily matrix multiplication and convolution, can be efficiently executed directly on mobile devices. Table 1 show hardware specifications of a number of the new generation of smartphone models. We computed the computational power and memory capacity of TransNet using one of the datasets (IRH). A forward pass of TransNet acquires 1.025 MFLOPS (floating point operations per second) and 1.34 MB RAM on the processor which is about 1000 times less than the performance power and storage capacity in new generations of smartphones.

6 VALIDATION APPROACH

In this section, we discuss our evaluation strategy to demonstrate the effectiveness of TransNet. Our evaluation uses data collected in experiments involving human subjects performing various daily physical activities while wearing inertial sensor nodes. In addition to our collected experimental data, we utilize three publicly available datasets.

6.1 Data Collection

We designed an experiment to collect wearable sensor data for daily and fitness activities. Approval from the appropriate Institutional Review Board (IRB) was obtained prior to participant screening and data collection. Twelve healthy subjects aged between 20 and 29 with a height range of 152.4

Table 1 Smartphone specifications

Model	Processor	Performance (GFLOPS)	Storage Capacity (GB)	RAM (GB)
Huawei nova 7 Pro 5G	HiSilicon Kirin 810	67	128/256	8
LG V60 ThinQ	Qualcomm SM8250 Snapdragon 865	122.38	128	8
Samsung Galaxy S20	Qualcomm Adreno 640	122.38	128	8/12
Apple iPhone 11	Apple A13 Bionic	154.9	64/128/256	4
Sony Xperia L4	Qualcomm SM6125 Snapdragon 665	115.2	64/128	6
Moto G8 Power Lite L4	Qualcomm SM6125 Snapdragon 665	115.2	64	4
Nokia 5.3	MediaTek MT6771 Helio P60	38.3	64	3/4/6

cm to 198.1 cm and diverse range of height, weight, and gender were recruited. Each participant was asked to perform a set of 24 physical activities ranging from high intensity activities such as *running* and *jumping* to medium and low intensity activities such as *normal walking* and *sitting while typing*. All non-strenuous activities were performed for 1 minute while other activities were performed for a set number of repetitions. Table 2 shows a description of these activities. Each participant wore 5 Shimmer [52] inertial sensor nodes on their ‘chest’, ‘back’, ’right-arm’, ‘left-thigh’, and ‘head’. The relative orientation of all the sensors with respect to the body coordinates remained consistent across all participants and throughout the data collection. The inertial sensors were sampled at 50Hz during each experimental activity. We refer to this obtained dataset as IRH in the rest of this paper.

6.2 Publicly Available Datasets

To assess generalizability of our algorithms in settings different than our governed experiment, we use three additional, publicly available, datasets.

6.2.1 WISDM Dataset. The first publicly available dataset we used in this study is WISDM [8]. This dataset contains $1,048,576$ samples of one 3D accelerometer sensor collected using mobile phones on an Android operating system. The sensors were sampled at 20Hz. The data samples belong to 35 subjects and 6 distinctive human activities including *walking, jogging, sitting, standing*, and *climbing stairs.*

6.2.2 OPPORTOUNITY Dataset. We used this dataset, referred to as *opportunity* (OPP) [53], to evaluate our approach on locomotion activities. The dataset contains inertial sensor data collected in an experiment with four subjects operating in a room. Each subject wore seven inertial measurement units (IMUs) on the ‘back’, ‘right arm’, ‘left arm’, ‘upper arm’, ‘lower arm’, ‘left shoe’ and ‘right shoe’. Because sensor nodes placed on the shoes carried a different sensor modality, we eliminated those sensors from our analysis in this paper. Each subject performed 5 runs of activities of daily living (ADL) following a given scenario involving *Grooming, Relaxing, Preparing Coffee, Drinking Coffee, Preparing Sandwich, Eating Sandwich, Cleaning up, wrapped in a Starting and Breaking.*

6.2.3 Sport and Daily Activities. The third publicly available dataset that we used is referred to as *Sport and Daily Activities* (SDA) [54, 55]. A total of 8 subjects performed 19 daily and sport activities for 5 minutes while wearing 5 inertial sensor nodes on their ‘torso‘, ‘left arm’, ‘left leg’, ‘right arm’ and ‘right leg’. Each sensor node consisted of a 3-axis accelerometer, a 3-axis gyroscope, and a 3-axis magnetometer. The sampling frequency was set to 25Hz. In this dataset, the subjects were

Table 2 Physical activities performed by each subject.

Number	Description	Duration/Repetition
1	Sittings with hands on lap	1 min
2	Sittings while writing	1 min
3	Sittings while typing	1 min
4	Lying down on back	1 min
5	Lying up on right side	1 min
6	Jumping Jacks	20 times
7	Butt Kickers	20 times
8	High Knees	20 times
9	Flutter Kicks	15 times
10	Lunges	20 times
11	Crunches	20 times
12	Squats	20 times
13	Pushups	10 times
14	Walking up stairs	5th levels
15	Walk down stairs	5th levels
16	Biking at 50 rpm	1 min
17	Dips	10 times
18	Standing with arms at sides	1 min
19	Stand with arms crossed	1 min
20	Slow walk at 1.0 mph	1 min
21	Normal walk at 2.5 mph	1 min
22	Fast walk at 4.0 mph	1 min
23	Jogging at 5.0 mph	1 min
24	Running at 7.0 mph	1 min

allowed to perform the activities in their own style and were not restricted on how the activities must be performed.

6.3 Implementation of TransNet

There are many hyper parameters and design choices while implementing TransNet. Our goal was to design an efficient network that can be easily utilized in cross-domain knowledge transfer scenarios. Prior research suggests that deeper networks with more parameters offer a higher learning capacity and can potentially achieve a higher prediction performance in the source domain at the expense of higher computation and training time [56, 57]. However, increasing complexity of the network increases the risk of overfitting to the source domain. This in turn leads to a performance decline in cross-domain knowledge transfer scenarios where data distribution in the target domain is different than that of the source domain. Furthermore, computing complexity of a highly dense and over-parameterized network limits potential implementation of a deep learning model on wearable devices with stringent constrained resources. This motivates us to design TransNet as a simple network with a small number of parameters and low computational complexity.

We experimentally optimized TransNet to have only two convolutional layers. Each convolutional layer includes 20 kernels with a kernel length of 3 and an ReLU Rectified Linear Unit (ReLU) [58] activation function given as $g(z) = max\{0, z\}$. Particularly, ReLU does not have gradient saturation problem in positive region, is very computationally efficient, and converges faster than alternative functions such as sigmoid/tanh [32, 59]. Both covolutional layers are followed by a max pooling layer of size 3. On top of these layers, we introduce a densely connected *recognition module*. TransNet was implemented in Keras [1] with a TensorFlow [2] back-end. All the experiments are reproducible and the source code is made publicly available.

An important tunable hyper-parameter is the learning rate which directly affects the speed of convergence. To adaptively adjust the learning rate during optimization, we use Adaptive Momentum (Adam) algorithm [60], which incorporates first and second order momentums to reconfigure the learning rate. Additionally, we use Dropout [61] which provides a powerful regularization with a computationally-efficient method to prevent over-fitting.

The number of training epochs is another hyper-parameter that needs to be tuned using a cross-validation approach. Without specific time or computational constraints, the common practice is to increase the training epochs until the validation accuracy exhibits a declining trend. In contrast to this common practice, we stop the training in early stages of the training while the test accuracy is still increasing. The early termination of the training process allows the learned features of TransNet to be generic enough to be used in a target domain. Our experimental results demonstrate that this compromise between computational complexity and prediction accuracy achieves significant generalizability performance in cross-domain knowledge transfer. Fig. 8 shows the test accuracy of TransNet during the first 10 epochs. In this scenario, 80% of the data were randomly selected for training and the remaining 20% were used for testing.

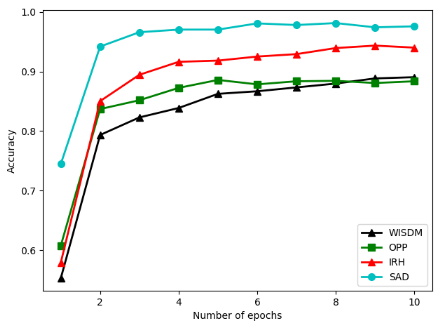


Fig. 8 Accuracy increases as the number training epochs grows.

6.4 Alternative Classifiers

To have a comprehensive evaluation of TransNet on various datasets, we compare the performance of TransNet against several machine learning algorithms with available implementations. In particular, we present the performance of several standard machine learning algorithms including Decision Tree (DT), Logistic Regression (LR), Support Vector Machine with both Linear (LSVM) and Gaussian (SVM) kernels, as well as two ensemble classifiers including Random Forest (RF) and Gradient Boosting (GB) in cross-domain scenarios. As described in Section 2.2, these classifiers do not learn their features automatically; that is, the feature extraction is incorporated as a preprocessing phase for these alternative classifiers. In the remainder of this section, we discuss our approach for designing efficient features for training the alternative classifier in an effort to provide a fair comparison of these algorithms with TransNet.

[1]https://keras.io
[2]https://www.tensorflow.org

6.4.1 Feature Extraction. For the alternative classifiers, we extracted 18 features from each signal segment associated with an inertial sensor stream. Potentially, there are many different features that can be extracted from human activity sensor signals. Inspired by prior research [8], as shown in Table 3, we extract both morphological and statistical features from each sensor stream. For example, while features such as *median* and *mean* capture intensity of the signal, *variance* and *histogram* intend to capture morphology of the signal.

Table 3 Features extracted from each signal segment.

Feature Name	Description
MED	Median of signal segment
MNVALUE	Mean of signal segment
MAX	Maximum of signal segment
MIN	Minimum of signal segment
P2P	Peak-to-peak amplitude of signal segment
VAR	Variance of signal segment
HIST (9 bins)	Histogram of signal segment
PRCNT (25%, 50%, and 75%)	Start-to-X% difference in amplitude

6.4.2 Feature Selection. To maximize generalizability of the alternative classifiers and reduce the risk of overfitting, we need to control the complexity of the model. To this end, we perform *feature selection* to identify the best feature set for each alternative classifier and use those features in subsequence analysis. This allows us to demonstrate the full potential of the competing alternative classifiers when comparing them against TransNet in cross-domain scenarios. For this study, we use χ^2 method to eliminate irrelevant features from the feature vector. The remaining features are then maintained in the feature vector to train/test the alternative classifiers.

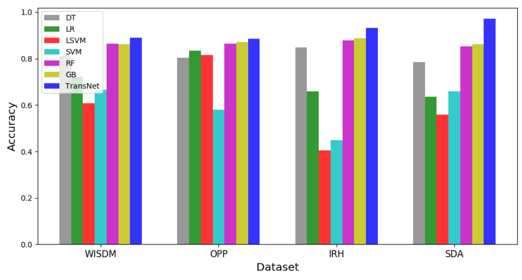
7 RESULTS

7.1 Standard Performance

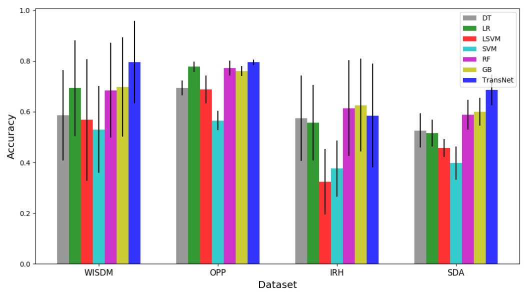
Our first analysis focused on comparing the performance of TransNet with that of the alternative classifiers when the two domains are similar. This can be viewed as an upper bound on the accuracy of each approach when there is no adaptation process. For the purpose of this paper, we assume that the two domains are similar if (i) the activity vocabulary of the two domains are identical; and (ii) the test subject is not entirely new to the system; that is, the data used to train the classification model was trained using some labeled instances of the test subject. We governed an experiment where 80% of the collected data were randomly selected for training and the remaining 20% were used for testing. As stated previously, we limit the number of training epochs to 10 for TransNet.

As shown in Fig. 9a, TransNet outperformed the alternative classifiers on all the four datasets. The accuracy of TransNet ranged from 88.7% on the OPP dataset to 97.1% on the SAD dataset. The ensemble classifiers, Random Forest and Gradient Boosting, performed the best among the alternative classifiers by a range of 86.4% to 87.2% on OPP and IRH datasets, respectively. Particularly, averaged over four datasets, TransNet outperforms the best alternative approach -Gradient Boosting- by 5.1%.

While the accuracy results shown in Fig. 9a are acceptable, in particular for TransNet and the ensemble classifiers, we are more interested in examining the performance of these algorithms across domains. In the following sections, we conduct comparative analyses to demonstrate superiority of TransNet in cross-subject and cross-vocabulary settings.



(a) Upper-bound performance



(b) Cross-subject performance

Fig. 9 Upper-bound and cross-subject accuracy of TransNet and alternative classifiers on four experimental datasets: (a) upper-bound accuracy in similar domains obtained by splitting the data into 80% training and 20% testing sets; and (b) cross-subject accuracy obtained based on leave-one-subject-out analysis.

7.2 Cross-Subject Performance

Prior research suggests that the activity recognition performance drops significantly when the system is utilized by a new (i.e., previously unseen) user [62]. Motivated by prior research, we were interested in investigating the accuracy of TransNet and the alternative classifiers in cross-subject scenarios. Thus, we performed a leave-one-subject-out evaluation of these algorithms on each subject and dataset. Fig. 9b shows the overall accuracy of different classifiers as a result of this analysis. An example of how this analysis was performed is as follows. The WISDM dataset contains labeled sensor data collected with 35 subjects. The accuracy numbers shown in Fig. 9b represent the accuracy of each classifier averaged over all 35 cases where in each test case data from 34 subjects were used for training and data from the remaining one subject was used for accuracy estimation (i.e., testing).

As shown in Fig. 9b, TransNet achieved the highest overall accuracy on all datasets except IRH. The accuracy of TransNet was 79% on both WISDM and OPP datasets, and it was 68.9% and 58.5% on SDA and IRH datasets, respectively. These accuracy values are 10%−38% less than the accuracy numbers reported in Section 6.3. Among the alternative classifiers, the ensemble classifiers based on bagging (RF) and boosting (GB) outperformed the other two classifiers. These two ensemble classifiers achieved a similar accuracy performance. An interesting observation was that the variation in the accuracy was higher on WISDM and IRH datasets where there existed a larger cohort of the users. Moreover, the OPP dataset exhibited the minimum variation in the accuracy potentially because this dataset has a small subject pool compared to other datasets. Another observation was that the accuracy was higher on OPP and WISDM datasets where the number of activities was less than that of the two other datasets.

The above-discussed analysis and Fig. 9 suggest that a classifier exhibits an obvious performance decline when the system is utilized by a completely new subject for whom no labeled training data exist. This performance downgrade motivates us to develop a personalized model for activity recognition. However, training a new classifier for each subject requires collecting sufficiently large amounts of training instances, which is time-consuming and expensive. In the following section, we will discuss our results on domain adaptation using TransNet.

7.3 Cross-Subject Adaptation Performance

As discussed previously, there is an obvious performance decline when the wearable system is utilized by new users. We used our cross-domain adaptation approach discussed in Section 4 to update the already trained TransNet network in new subjects using only few activity instances acquired from each new subject. The experiment governed for this purpose was similar to a leave-one-subject-out analysis except that we incrementally fed TransNet with few labeled instances of the new subject (i.e., test subject). From all instances of a new/test subject, we randomly chose i instances of each activity where i ranged from 1 to 5. Our experimental results showed that the amount of performance gain beyond i=5 was negligible. We refer to these labeled instances as *transfer set*. The remaining instances of the new subject were kept for testing. We refer to such instances as *test set*. Our experiment began by initially training TransNet using training instances from all subjects other than the test/new subject. We then froze TransNet weights except for the top layers (i.e., the *recognition module*). At step $1 \leq i \leq 5$, TransNet updated its wights by training on i instances of each activity from *transfer set*. We reseted the TransNet weights to their original trained network on other subjects prior to moving on with choosing a new test subject. By performing few training epochs using instances from *transfer set* and updating only the top layers, we observed a significant boost in the accuracy performance of TransNet on *test set*. We also repeated our experiments with alternative classifiers to provide a fair comparative analysis. Because the alternative classifiers are not online learners in their nature, we retrained those classifiers from scratch using a union set of instances of other subjects and selected instances from *transfer set*.

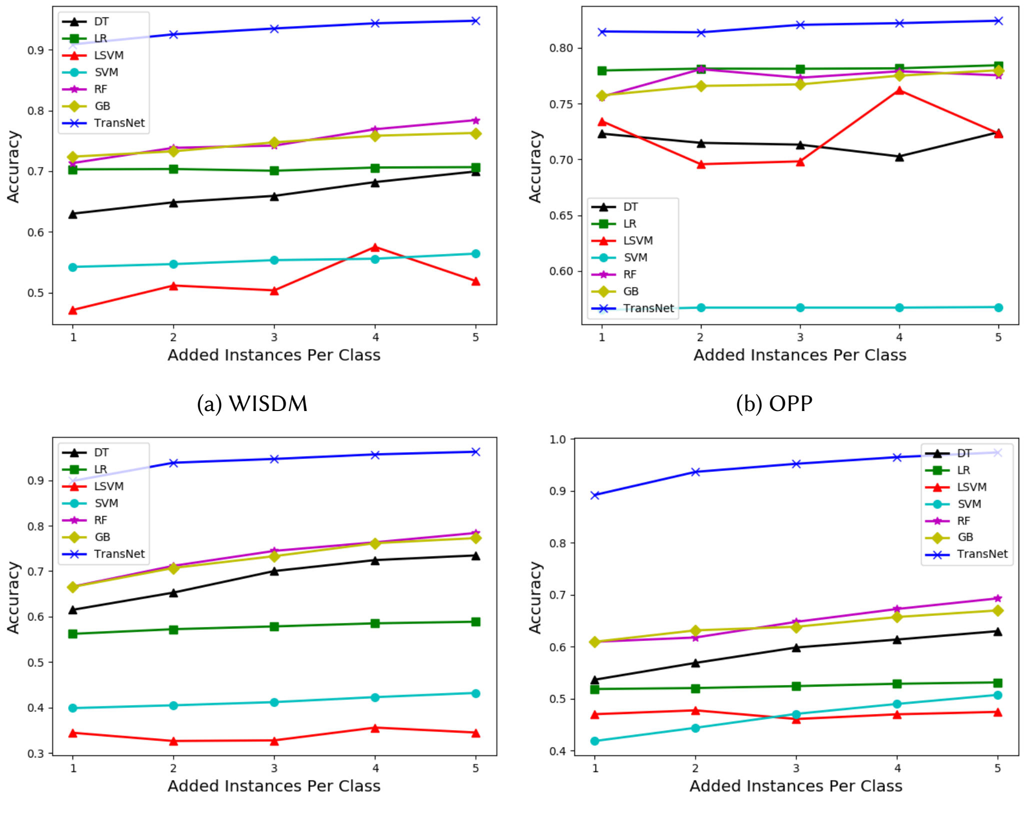


Fig. 10 Accuracy of different activity recognition models on the four datasets using a leave-one-subject-out approach while incrementally adding labeled instances of the new subject to the classifiers.

As shown in Fig. 10a, the accuracy of TransNet after the first iteration was 90.9%, which was about 25% higher than RF and GB, of the WISDM dataset. The accuracy continued to improve as more labeled instances were added to the network. After five iterations, TransNet achieved an accuracy of 94.8%. Additionally, the $F1$ score for TransNet was 30% higher than the ensembled classifiers with one labeled instance added to the system (Fig. 11a). The $F1$ score for TransNet ranged from 87.3% with one added instance to 91.7% for five added training instances per activity. The Logistic Regression method exhibited interesting performance. While the accuracy of Logistic Regression was close to that of the ensembled classifiers, the $F1$ measure was significantly low for Logistic Regression. This observation can be explained as follows. The activity instances of WISDM are not balances. Therefore, the Logistic Regression prediction is higher for activities with more instances than those with fewer examples. Furthermore, except for Linear SVM, all the algorithms showed a trend of improved performance as the number of added instances increased.

Considering the result on the OPP dataset in Fig. 10b, TransNet achieved 82% accuracy, which was 10% higher performance than the best alternative classifier (i.e., Logistic Regression). As shown in Fig. 11b, the $F1$ score for TransNet showed a nominal improvement when more training instances were added to the system. In addition to TransNet, Gradient Boosting exhibited a trend of both accuracy and $F1$ improvement, while for other classifiers the performance either did not improve or fluctuated as new labeled instanced were supplied.

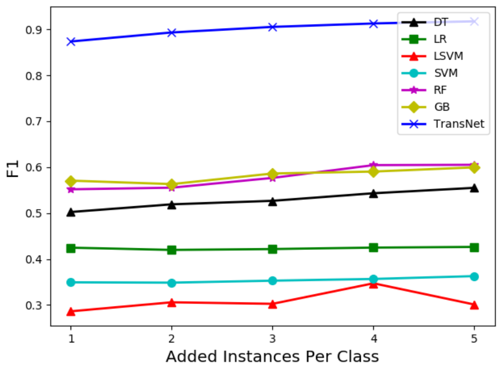
Fig. 10c and Fig. 11c show the accuracy and $F1$ measure of different classifiers for cross-subject domain-shift on IRH dataset. The accuracy of TransNet was 20% higher than that of the ensemble classifiers. The accuracy was initially 89.8% after adding one labeled instance of each activity. The accuracy increased to 96.2% after 5 activity instances were supplied. The $F1$ measure for TransNet started from 88.9% with one personalized activity instance and grew to 96.1% after five instances were added to the system. The difference between $F1$ measure of TransNet and that of other approaches ranged from 26.2% for Random Forest to 60% for SVM. TransNet, the ensemble classifiers and Decision Tree showed an obvious trend of improvement as a result of incrementally adding more instances from the new subject. For Logistic Regression and SVM with Gaussian kernel, the improvement growth was very slow; however, for SVM with linear kernel, the new instances seemed insufficient to adjust the decision boundary.

As shown in Fig. 10d and Fig. 11d, similar patterns were observed on the SDA dataset.The performance of TransNet in terms of both accuracy and $F1$ measure was 30% higher than those of the ensemble classifiers. Both measures started at 89.2% and grew to 97% after five iterations. Except for Logistic Regression and Linear SVM, all other classifiers showed a clear trend of improvement when adding more labeled instances to the training phase.

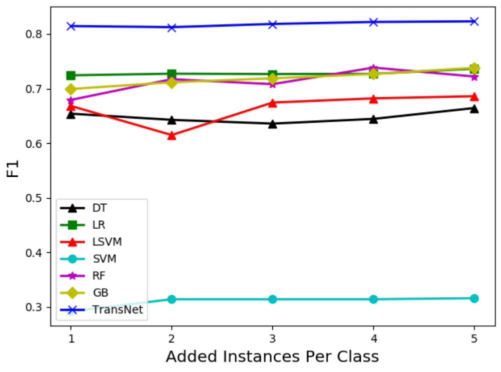
Overall, our cross-subject adaptation analysis shows that TransNet achieves an average accuracy of 88.1% in cross-subject learning scenarios using only one labeled instance for each activity class. This performance improves to an accuracy of 92.7% with five labeled instances. The analysis in this section shows that the features learned in the lower layers in TransNet are subject-agnostic, and therefore they can be reused in the new subjects. This observation is consistent with the prior deep learning research in other application areas [31].

7.3.1 Statistical Analysis. There are different statistical tests to validate the statistical significance of the results. Dietterich has reviewed five approximate statistical tests for determining whether one learning algorithm outperforms another on a particular learning task [63].

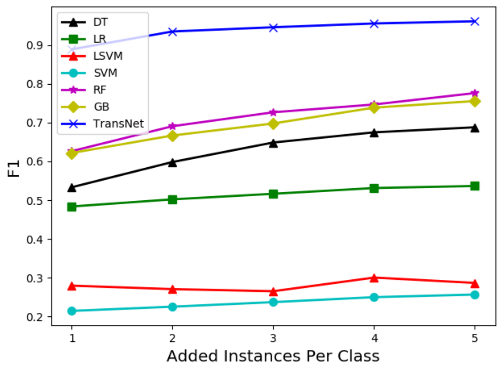
He showed that some widely used statistical tests have high probability of type I error (False Positive) in certain situations. Particularly, Dietterich suggested that a t-test based on taking several random train-test splits should never be used for comparing the performance of two learning algorithms.



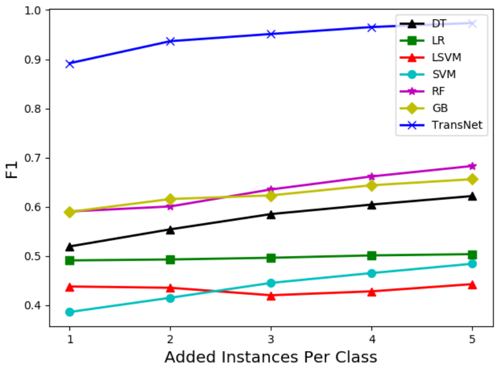
(a) WISDM



(b) OPP



(c) IRH



(d) SDA

Fig. 11 F1 score for different activity recognition methods on the four datasets using a leave-one-subject-out approach while incrementally adding labeled instances of the new/test subject to the classifiers.

However, it is shown that McNemar test has low type I error and is the only acceptable test for comparing algorithm that can be executed only once. Particularly, this is an important consideration in the field of deep learning.

To validate the statistical significance of our results, we compared the results of TransNet with the best two alternative classifiers i.e. Gradient Boosting and Random Forest and the test showed, for all of four classifiers, the results presented in previous chapter are statistical significant (with $p\text{-}value < 0.01$).

7.3.2 Freezing Different Layers of the Network. To demonstrate that TransNet achieves an acceptable performance, we compared the activity recognition accuracy and F1-Score of TransNet with those of two other networking training approaches. The first approach, denoted as 'No Transfer' in Table 4, refers to the case where the neural network is trained from scratch using the labeled data provided in the target domain. The second approach, denoted as 'Partial Transfer' in Table 4, represents the case where all the layers before the second/last convolution layer remain frozen during adaptation in the target domain. In this case, the remaining layers including the second convolution layer and dense layers are adjusted using the target training samples. We performed these analyses for the case where five labels of each class were available in the target setting. We limit the analyses to cross-subject transfer learning.

Table 4 shows the accuracy and F1-Score of the TransNet, those of the network trained from scratch (i.e., without transfer learning), and those of the network that was partially trained. As shown in these results, TransNet and partial transfer methods achieve higher performance values than the network without transfer learning. On the OPP dataset, TranNet obtains 80.4% accuracy and 80.5% F1-Score, and partial transfer method achieves an accuracy of 80.4%, and F1-score of 77.7%. The accuracy and F1-Score drops to 61.2%, and 57.3%, respectively, when training the whole network from scratch without knowledge transfer. Using the IRH dataset, TransNet and 'No Transfer' approaches achieve similar ranges of accuracy and F1-Score values. Specifically, TransNet achieves 86.4% accuracy, and 86.2% F1-Score, while 'No Transfer' achieves an accuracy of 87.5%, and an F1-Score of 84.3%. However, partially transferring the network slightly improves the F1-Score to 90.5%, and accuracy to 90.6% comparing to the other methods. On the WISDM dataset, TransNet achieves 91.3% and 91.4% accuracy and F1-Score, and 'Partial Transfer' shows 90.2%, and 90.1% accuracy and F1-Score, respectively. Note that accuracy and F1-Score values decline to 76.8%, and 73.2% when we retrain the network from scratch without employing a transfer learning approach. With the SDA dataset, TransNet obtains accuracy and F1-Score of 82.0% and 81.9%, respectively, while the F1-Score and accuracy for the case of training from scratch drop to 82.6% and 82.2%, respectively. On the other hand, partially transferring the network shows a slightly higher performance than TransNet in this case.

The above-mentioned results demonstrate that training the whole network with many parameters on the limited amount of labeled data in the target setting might result in overfitting of the obtained network, resulting in a decline in the validation performance. Moreover, training the ending layers such as the last convolution layer with the new data adapts only the high-level features extracted by this layer to the target dataset, therefore, is less likely to result in overfitting. Additionally, we performed a single-factor ANOVA statistical test on the F1-Score values of TransNet, 'No Transfer', and 'Partial Transfer' networks to determine any significance in their validation results. A *p-value* of 0.12 suggested no significant difference among the F1-Score values.

Finally, we compare the number of the trainable parameters in

This may suggest that TransNet, which includes the fewest learnable parameters, is a more promising network to implement on wearable devices with limited power and computational resources.

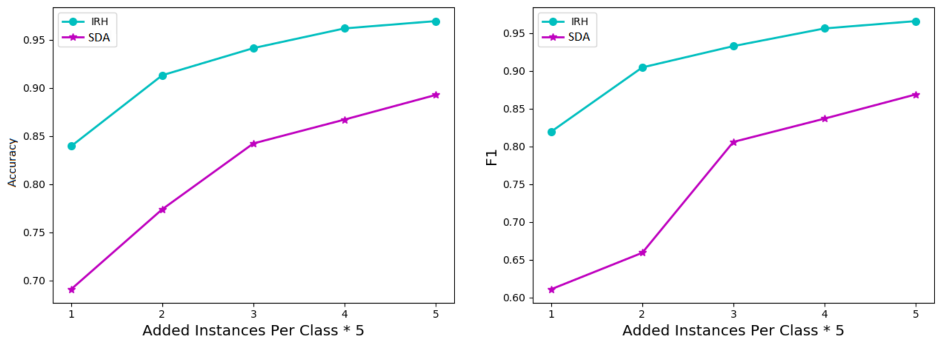
Table 4 Comparison of activity recognition accuracy and F1-Score of TransNet, network retrained from scratch (No Transfer), and network partially trained (Partial Transfer)

Dataset	TransNet		No Transfer		Partial Transfer	
	Accuracy	Fscore	Accuracy	Fscore	Accuracy	Fscore
OPP	80.4	80.5	61.2	57.3	80.4	77.7
IRH	86.4	86.2	87.5	84.3	90.6	90.5
WISDM	91.3	91.4	76.8	73.2	90.2	90.1
SDA	84.9	83.4	82.6	82.2	85.8	86.7

7.4 Cross-Location Adaptation

For a cross-location scenario, Table 6 has compared the performance TransNet with existing deep learning methods on OPP dataset. In this section further investigation on cross-location scenarios are performed on both IRH and SDA datasets. We excluded WISDM dataset because it only has one location. Particularly, we performed two analysis: *Model Transfer* and *Label Transfer.*

7.4.1 Model Transfer. In model transfer scenario, analogous to Section 4, a trained network for one location is transferred to be used in another location. Upon obtaining new labeled instances from the new location, the transferred network is customized for new context/location by adjusting weights of its recognition module. Fig. 13 shows how the performance of TransNet improves by acquiring more labeled data from the new location. For SAD dataset, the accuracy of the model starts around 68% when the model adjusted by 5 instances per class. The classifier achieves more than 88% accuracy upon obtaining 25 labeled instances per class. For IRH dataset, the accuracy starts around 84% and improves to 96% after adjusting with 25 instances per class. Similar pattern for F1-score of the classifier could be observed in Fig. 12b.



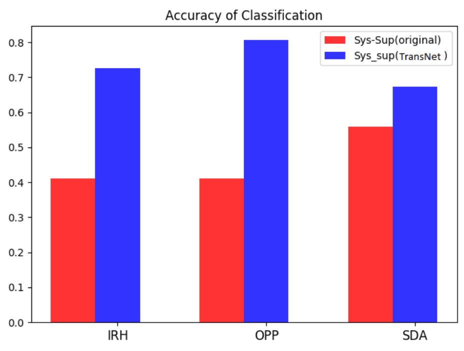
(a) Accuracy of cross-location adaptation (b) F1-score of cross-location adaptation

Fig. 12 Performance of TransNet in cross-location scenarios when a trained model for another location is utilized and the model is updated upon acquiring new labeled instances from the new location

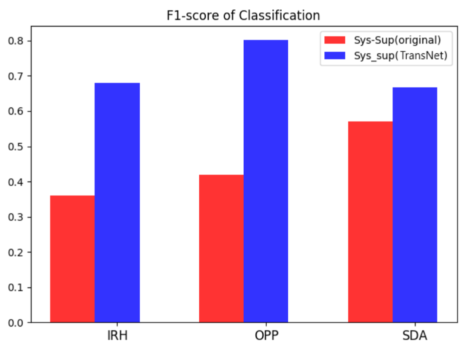
7.4.2 Label Transfer. In label transfer scenarios, the performance of transfer learning is sensitive to the representation of data. This section performs the system-supervised algorithm [64] using TransNet as a classifier for the target domain. In system-supervised method, to train a newly added sensor, the existing sensor sends its prediction on any new observation to the added sensor. After observing enough activities, a classifier is build on the target domain while the X is the target observations and Y is prediction of the source. As presented in Fig. 13a, in all three datasets, the performance of the final classifier is improved by using TransNet in the target. Particularly, the accuracy improvement range from about 13% for SDA dataset to 40% in OPP dataset. Similar improvement pattern for F1-score of the classifier is illustrated in Fig. 13b.

7.5 Cross-Vocabulary Adaptation Performance

In this section, we present our results on the performance of TransNet in adapting with new types of activities. Particularly, we present our results for expanding an activity vocabulary to recognize entirely new activities that have remained unseen by previously trained activity recognition models. The conventional approach for expanding an activity vocabulary learns an activity recognition model from scratch by collecting a large number of labeled instances of the new activity, adding those instances to the original training set, and retraining a classifier. In contrast, TransNet uses its already trained network and adjusts its parameters using a small number of labeled instances for the new activity class. To experiment this problem, let $\mathcal{A}$ be a set of activities of interest available in a particular dataset. We randomly split $\mathcal{A}$ into two disjoint sets $\mathcal{A}_1$ and $\mathcal{A}_2$ such that



(a) Accuracy of label transfer scenario



(b) F1-score of label transfer scenario

Fig. 13 Performance of TransNet in cross-location scenarios when a the source model transfers its prediction on new observation to the newly added sensor and a model is trained accordingly.

$$|\mathcal{A}_1| \simeq |\mathcal{A}_2| \tag{8}$$

We trained TransNet on instances of activities in $\mathcal{A}_1$. Similar to our approach explained in Section 7.3, having a trained TransNet network on $\mathcal{A}_1$, we first freeze all the layers except for the *recognition module*. We then randomly choose a set of 5 instances of each activities in $\mathcal{A}_2$ and call this set *transfer set*. The remaining instances of activities in $\mathcal{A}_2$ are used for test. We incrementally feed instances from the *transfer set* to TransNet such that after $1 \leq i \leq 5$ iterations weights of the top layers have been updated using i instances of each activity. By performing few training epochs on the instances of the *transfer set*, TransNet quickly learns to detect activities in $\mathcal{A}_2$.

We performed this experiment on the IRH and SDA datasets, which had sufficiently large number of activity classes. As shown in Fig. 14a and Fig. 14b, using only two instances of each activity in $\mathcal{A}_2$, TransNet was able to detect all instances in $\mathcal{A}_2$ with at least 90% accuracy on both datasets. Specifically, for the experiment involving the IRH dataset, we randomly chose and placed 12 class labels in $\mathcal{A}_1$ and inserted the remaining activity labels to $\mathcal{A}_2$. Similarly, for the SDA dataset, 10 class labels were selected for inclusion in $\mathcal{A}_1$ and the remaining 9 activity classes were placed in $\mathcal{A}_2$. Similar growing performance trends were observed for the $F1$ measure.

7.6 Comparing with Other Deep Learning Solutions

Although there exist several deep learning methods for activity recognition using inertial sensory data, very few have been evaluated for cross-subject problems. In this section, we compare TransNet with ConvLSTM [57, 65] and 3DConvLSTM [66], which have a reported evaluation in cross-subject scenarios. As shown in Table 5, an important advantage of TransNet compared to the existing deep learning methods is that TransNet has much less number of layers and learnable parameters. While other approaches reported to require model training using GPUs with thousands of cores, TransNet can be trained on a computer with one core-i7 CPU. Particularly, after the initial training, adjusting the weights requires significantly less number of learnable parameters and training epochs in TransNet. While two other methods reported to need hundreds of training epochs, using Adam as an adaptive learning rate, TransNet speeds up its learning rate and converges in less than 50 epochs. We believe that this significant performance improvements are critical in developing computational algorithms that can eventually reconfigure in light-weight and computational constrained embedded sensors that are commonly used in various IoT applications.

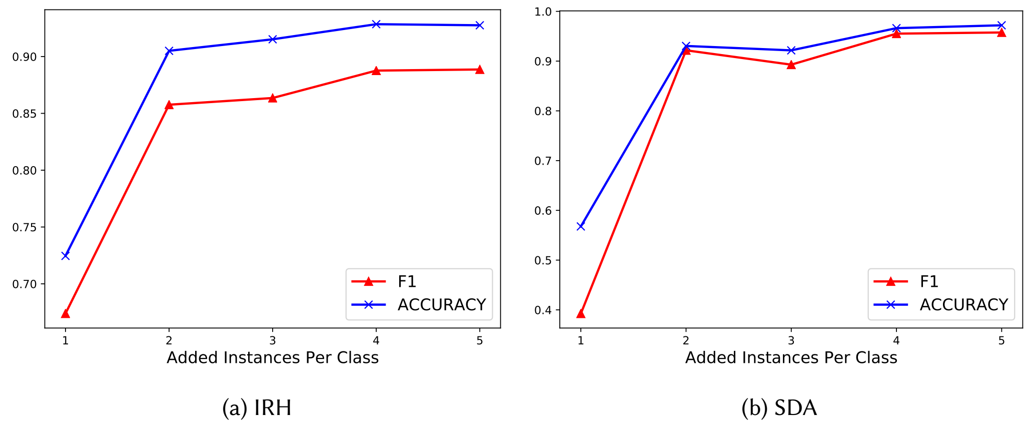


Fig. 14 TransNet accuracy and F1 measure for cross-vocabulary domain adaptation on two datasets (i.e., IRH and SDA) with large activity vocabularies.

Table 5 Comparison in network architecture between TransNet and two deep learning methods.

Method	# Convolution Layers	# of Kernels	kernel size	Max Pooling	Aggregator(s)	Aggregator size
ConvLSTM	4	64-64-64-64	5	No	2 x LSTM	128 - 128
Conv3DLSTM	5	128-96-64-48-32	3 x 5	Yes	2 x LSTM	128 - 128
TransNet	2	20 - 20	3	Yes	1 x Dense	30

The simple architecture devised in TransNet not only offers low computational complexity but also enables the model to be effective in cross-subject learning scenarios. In particular, a recent study evaluated the performance of ConvLSTM and 3DConvLSTM in cross-subject scenarios on the OPP dataset [66]. Table 6 summarizes the performance comparison of TransNet with ConvLSTM [57, 65] and 3DConvLSTM [66]. In addition to the complete cross-user scenarios, [66] performed an active learning approach to adapt the model upon receiving labeled data from the new user. As reported in Table 6, TransNet outperforms existing approaches by about 10%. Additionally, TransNet achieves the best performance among all algorithms in cross-subject with 5 additional instances per class from the new user.

Table 6 F1-score Comparison of cross-domain performance

Method	cross-subject	cross-sub + 5 instance/class	cross-location + 30%
ConvLSTM	≃ 67	≃ 68.2	36.5
3DConvLSTM	≃68.3	≃71.2	-
TransNet	78.8	82.1	83.8

8 RELATED WORK

Several studies based on the concept of transfer learning [67] have attempted to apply the machine learning knowledge obtained in a source domain to a new but related target problem. For example, authors in [64] utilized a teacher/ learner method to train a new sensor added to a wearable network. They demonstrated that by synchronizing the existing and new sensors, the existing sensor could provide labels of future activities. Authors in [9] proposed a synchronous dynamic view learning method to boost the accuracy of transferred labels even when the location of the target sensor changes during run-time. However, both these prior studies focused on cross-sensor knowledge transfer and required a reliable sensors node to provide labels to the target domain.

To avoid handcrafting features, the growing trend in representation learning from raw data with deep neural networks has demonstrated significant performance gain in image classification [59] and speech recognition [68]. In the case of wearable sensors, prior research showed that using Convolutional Neural Networks (ConvNets) generally improves the activity recognition performance [69–71]. Authors in [72] used ConvNets to detect activities of pedestrians for indoor localization. In another recent approach, the motion signal was converted into a spectral image sequence to input to independently train two ConvNets [73]. Moreover, in [74] authors showed that ConvNets are more robust when they are used in different domains. Authors in [75] showed that ConvNets could significantly improve the accuracy of detecting stereotypical movements in Autism patients. In addition, other studies such as those in [57, 76] aimed to capture temporal dependencies between high-level activities by applying Recurrent Neural Network and particularly LSTM [77]. In these studies, ConvNet is paired with LSTM to capture movement patterns while considering their temporal dependencies. Furthermore, [56] used Restricted Boltzman Machines (RBMs) to develop a deep generative model capable of utilizing unlabeled data samples of triaxial accelerometers. In another approach, Deep Belief Networks are used on smartphone sensor data to recognize human activities. While these studies showed the effectiveness of deep learning methods in detecting human activities, there are a limited number of studies that used them for transfer learning. Notably, similar to the transfer learning method explored in [31], authors in [65] introduced a feature transfer approach for activity recognition. They showed that using at least 50% of the training data in the target domain, their approach achieves a promising performance for transfer learning within a dataset. Then, authors in [66] extended their model and introduced a 3DConvLSTM which requires fewer data to adapt to the new domain. Finally, [78] propose a combinatorial transfer learning framework that learns structural similarities among the activities in an arbitrary domain and those of a different domain through graph modeling.

Comparing to the prior research, the work presented in this article introduces an embedding layer before the first convolutional layer. This layer allows TransNet to offer a computationally-simple structure through a small number of layers, automatically tunable parameters, and a small number of training epochs. The computationally-efficient design of TransNet combined with an embedding pre-processing layer results in obtaining almost one-order-of-magnitude less number of parameters compared to the state-of-the-art deep learning networks.

Additionally, many of the prior studies do not provide an approach for domain adaptation and knowledge transfer. In contrast, this article provides a comprehensive analysis of transfer learning capabilities on different datasets and in a variety of cross-domain scenarios. While other approaches that offer transfer learning capabilities require significantly larger training data in a new domain, TransNet quickly adapts to the new domain by acquiring a small number of training instances.

9 GUIDELINES

As our results suggest, neural networks are a promising approach to designing activity recognition systems that can quickly adapt to new settings. However, one drawback of neural networks is that they have to be carefully designed and engineered by human experts for each specific task and it often takes a large amount of time to find an optimal neural network architecture for a given machine learning task. The problem of identifying the optimal architecture for a neural network is referred to as *neural network search problem*. The neural network search is a difficult problem because not only the search space for the architecture of the network is quite large but also training and evaluating the performance of the neural network for all architectures in the search space is very time-consuming [79, 80]. In this section, we briefly discuss potential approaches that can be used to design neural networks for activity recognition systems.

One important factor in designing a neural network is the bias and variance trade-off as the model becomes more complicated with more layers and hidden states. More complex networks tend to be less biased to the input dataset but show more variance in general. Another factor is that over-parametrized models are less efficient in terms of training time. Therefore, the best model is a model with not too many or too few layers and hidden states. Nonetheless, as mentioned previously, finding such a network is a hard problem.

In general, a neural architecture search technique requires a search strategy and an approach for validating each candidate architecture. Common search strategies are based on reinforcement learning, evolutionary algorithms, Bayesian optimization, and gradient descent optimization. A common approach for validating a candidate architecture is to measure validation accuracy of the architecture on a held-out validation set. However, this approach requires training a neural network from scratch, which is time-consuming and power-hungry. To address this shortcoming, we can hire a function approximation method like neural network whose goal is to learn the accuracy of neural networks without training from scratch.

A reasonable approach for designing neural network based activity recognition systems is to start with a simple baseline model and improve the architecture through an iterative process.

Including discretization and embedding layers in TransNet helped in achieving robust models. The signal segments are discretized to reduce the sensitivity of the model to small changes and improve the transferability of the model across domains. We feed the discretized sequence to an embedding layer to generate a compact representation of the input data. In addition to dimensionality reduction, this embedding layer reduces the effect of different instrumental calibration. Therefore, similar activities captured by different accelerometers or performed by different users could have close representation in this space.

TransNet includes a stack of 1D convolutional layers. The sparse connectivity and parameter sharing features of ConvNets not only help in extracting useful local features on different body locations but also reduce the computational and storage complexity of the model. In TransNet, we chose two 1D convolutional layers because stacking too many convolutional layers increases the training time. The common practice is to start with 1 or 2 layers and increase if the desired accuracy was not achieved.

Another parameter is the batch size, which shows the number of inputs on which we train the network in each iteration before updating the weights. Batch size affects the generalization error since impacts the wideness of the final minima [81]. The smaller the batch size, the wider, the less the generalization error will be. However, having a small batch size increases the training time.

The number of epochs is another parameter to consider when designing the model. If the distribution of the training and validation datasets are different, training the model for too many epochs will increase the chance of over-fitting. The general recommendation is to early stop the training of the network based on a threshold value. We suggest stopping the training when the difference between the validation accuracy in two consecutive epochs is less than a threshold value (e.g., 0.01). Regularizing the model (e.g., L_1 or L_2 regularizers, or Dropout) is also recommended to prevent over-fitting.

10 DISCUSSION AND FUTURE WORK

Although we evaluated the performance of TransNet for two cross-domain adaptation scenarios, including cross-subject and cross-vocabulary, we have a suspicion that the transferability of the learned features is not limited to these analyzed scenarios. TransNet, or a minimally refined architecture of TransNet, may offer cross-platform, cross-modality, and hybrid knowledge transfer.

Our ongoing work involves investigating the effectiveness of TransNet for such cross-domain applications.

The transfer learning approach in TransNet freezes the convolution layers in the network. It only retrains the recognition layers using the target data rather than freezing some of the initial layers of the network and retraining the remaining layers. The reason behind this design choice is in two folds. First, freezing more layers will reduce the complexity, and therefore, the training time of the network in the target setting will decrease. Deployment of wearable computing systems in a real-world setting demands the design of low-power computational solutions. Second, given the limitations on the amount of labeled data in the end-user setting, training more layers of the network and freezing fewer layers may result in overfitting of the model with respect to the target data [31, 82].

As presented in Section 7, upon acquiring a few instances, TransNet can quickly master in the new context. However, accurately labeled instances are not always available. In cases where querying a human subject is not applicable, using predictions of other wearables or ambient sensors can potentially help in acquiring labels in the new domain. While these obtained labels are not always accurate, crowd-sourcing methods can be potentially utilized to increase the confidence of such noisy labels. Additionally, label propagation methods used in prior research can be exploited to improve the quality of the labels[9, 83].

An important aspect of TransNet is its ability to learn incrementally. In contrast to traditional classifiers, TransNet incrementally improves itself upon receiving more number of labeled instances without retraining from scratch. Leveraging this online learning feature, TransNet features the capability of adapting itself in continuously changing environments. Such a reliable adaptation approach becomes increasingly important when labeled instances from previous domains are not available in the new domain. One major concern computational models such as TransNet regarding incremental learning or in other words, lifelong learning from the previous domains is being prone to catastrophic forgetting/inference. Catastrophic forgetting happens when the performance declines significantly by replacing a part or in the worst case, all the previous knowledge with the new data. We are planning to investigate the performance of TransNet in lifelong learning [84–86].

Recent studies show the effectiveness of deep learning methods to devise classifiers with high accuracies. However, the number of learnable parameters are important considerations when designing wearable embedded systems with limited memory storage and computational power. TransNet includes a small number of layers and learnable parameters. Experimenting on different datasets, we showed that using only 10 training epochs, TransNet can achieve a high activity recognition performance. In the future, we would like to explore the effectiveness of our framework in end-user settings by implementing TransNet on embedded and mobile devices for real-time activity monitoring.

In Section 7.3, we split each experimental dataset into two disjoint activity sets ($\mathcal{A}_1$ and $\mathcal{A}_2$), at random, and computed cross-vocabulary adaptation performance. We repeated the process to obtain random splits with different seeds and observed a similar trend to those results shown in Fig. 14a and Fig. 14b. Intuitively, the reason for such a consistent set of results is that the first set, $\mathcal{A}_1$, is sufficiently large enough that it captures the underlying distribution of the data that is common across various human activities. Therefore, learning the transfer set, $\mathcal{A}_2$, requires a small number of labels for each activity class to adjust the parameters of the recognition layers. Nonetheless, we expect that if $\mathcal{A}_1$ is a very small set (e.g, in the extreme case if the system learns only one activity at a time), then the evolving network may not learn new activities with only a very small number of labels similar to what we observed in Fig. 14a and Fig. 14b. A detailed exploration of this phenomena is out of the scope of this work, and we plan to investigate it as part of our future work.

11 CONCLUSION

We introduced TransNet for transferring activity recognition capabilities across different domains and constructing a personalized model with minimal supervision. We showed that the learned features in lower layers of the network are not task-specific, and the lower layers could easily be reused in new domains. We showed that by using representation learning, we can reuse the general features learned from available training data and construct a model in a new domain with 1 to 5 labeled activity instances. Our analysis on four experimental datasets showed that TransNet achieves an average accuracy of 92.7% in activity recognition in cross-subject scenarios with only 5 labeled instances per activity. Furthermore, TransNet enables the user to expand the set of activities of interest by adjusting the classification layer after obtaining only 2 labeled instances of unseen activity.

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation, under grants CNS-1566359, CNS-1750679, and CNS-1932346, and the National Institute of Health, under grant 1R21NR015410-01. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding organizations.

REFERENCES

[1] D. Zakim and M. Schwab, "Data collection as a barrier to personalized medicine," *Trends in pharmacological sciences*, vol. 36, no. 2, pp. 68–71, 2015.

[2] I. Budin-Ljøsne and J. R. Harris, "Patient and interest organizations' views on personalized medicine: a qualitative study," *BMC medical ethics*, vol. 17, no. 1, p. 28, 2016.

[3] A. Alyass, M. Turcotte, and D. Meyre, "From big data analysis to personalized medicine for all: challenges and opportunities," *BMC medical genomics*, vol. 8, no. 1, p. 33, 2015.

[4] S. I. Mirzadeh, J. C. Ardo, R. Fallahzadeh, B. Minor, L. S. Evangelista, D. J. Cook, and H. Ghasemzadeh, "Labelmerger: Learning activities in uncontrolled environments," *2019 First International Conference on Transdisciplinary AI (TransAI)*, pp. 64–67, 2019.

[5] R. Saeedi, J. Purath, K. Venkatasubramanian, and H. Ghasemzadeh, "Toward seamless wearable sensing: Automatic on-body sensor localization for physical activity monitoring," in *The 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2014.

[6] R. Fallahzadeh and H. Ghasemzadeh, "Personalization without user interruption: Boosting activity recognition in new subjects using unlabeled data," in *Proceedings of the 8th International Conference on Cyber-Physical Systems*, ser. ICCPS '17. New York, NY, USA: ACM, 2017, pp. 293–302.

[7] A. Stisen, H. Blunck, S. Bhattacharya, T. S. Prentow, M. B. Kjærgaard, A. Dey, T. Sonne, and M. M. Jensen, "Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '15. New York, NY, USA: ACM, 2015, pp. 127–140.

[8] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.

[9] S. A. Rokni and H. Ghasemzadeh, "Synchronous dynamic view learning: a framework for autonomous training of activity recognition models using wearable sensors," in *Proceedings of the 16th ACM/IEEE International Conference on Information Processing in Sensor Networks*. ACM, 2017, pp. 79–90.

[10] R. Saeedi, B. Schimert, and H. Ghasemzadeh, "Cost-sensitive feature selection for on-body sensor localization," in *The 2014 ACM Conference on Ubiquitous Computing, UbiComp '14 Adjunct, Seattle, WA, USA - September 13 - 17, 2014*, 2014, pp. 833–842.

[11] P. Alinia, R. Saeedi, R. Fallahzadeh, A. Rokni, and H. Ghasemzadeh, "A reliable and reconfigurable signal processing framework for estimation of metabolic equivalent of task in wearable sensors," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 5, pp. 842–853, Aug 2016.

[12] R. Saeedi, N. Amini, and H. Ghasemzadeh, "Patient-centric on-body sensor localization in smart health systems," in *The Asilomar Conference on Signals, Systems, and Computers*, 2014.

[13] N. Kale, J. Lee, R. Lotfian, and R. Jafari, "Impact of sensor misplacement on dynamic time warping based human activity recognition using wearable computers," in *Proceedings of the Conference on Wireless Health*, ser. WH '12. New York,

NY, USA: ACM, 2012, pp. 7:1–7:8.

[14] Y. E. Ustev, O. Durmaz Incel, and C. Ersoy, "User, device and orientation independent human activity recognition on mobile phones: challenges and a proposal," in *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication.* ACM, 2013, pp. 1427–1436.

[15] P. Zappi, T. Stiefmeier, E. Farella, D. Roggen, L. Benini, and G. Troster, "Activity recognition from on-body sensors by classifier fusion: sensor scalability and robustness," in *Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007. 3rd International Conference on.* IEEE, 2007, pp. 281–286.

[16] H. Sagha, A. Calatroni, J. del R Millan, D. Roggen, G. Troster, and R. Chavarriaga, "Robust activity recognition combining anomaly detection and classifier retraining," in *Body Sensor Networks (BSN), 2013 IEEE International Conference on.* Ieee, 2013, pp. 1–6.

[17] K. Kunze and P. Lukowicz, "Dealing with sensor displacement in motion-based onbody activity recognition systems," in *Proceedings of the 10th international conference on Ubiquitous computing.* ACM, 2008, pp. 20–29.

[18] R. Uchida, H. Horino, and R. Ohmura, "Improving fault tolerance of wearable wearable sensor-based activity recognition techniques," in *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication*, ser. UbiComp '13 Adjunct. New York, NY, USA: ACM, 2013, pp. 633–644.

[19] K. Murao, T. Terada, Y. Takegawa, and S. Nishio, "A context-aware system that changes sensor combinations considering energy consumption," in *Pervasive Computing.* Springer, 2008, pp. 197–212.

[20] N. Ravi, N. Dandekar, P. Mysore, and M. Littman, "Activity recognition from accelerometer data," in *Proceedings of the National Conference on Artificial Intelligence*, vol. 20, no. 3. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005, p. 1541.

[21] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *International conference on machine learning*, 2014, pp. 647–655.

[22] G. M. Y. Dauphin, X. Glorot, S. Rifai, Y. Bengio, I. Goodfellow, E. Lavoie, X. Muller, G. Desjardins, D. Warde-Farley, P. Vincent *et al.*, "Unsupervised and transfer learning challenge: a deep learning approach," in *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, 2012, pp. 97–110.

[23] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," in *International Conference on Learning Representations (ICLR 2014).* CBLS, 2013, p. 16.

[24] Y. Bengio, A. Bergeron, N. Boulanger-Lewandowski, T. Breuel, Y. Chherawala, M. Cisse, D. Erhan, J. Eustache, X. Glorot, X. Muller *et al.*, "Deep learners benefit more from out-of-distribution examples," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011, pp. 164–172.

[25] G. Guerra-Filho, C. Fermuller, and Y. Aloimonos, "Discovering a language for human activity," in *Proceedings of the AAAI 2005 fall symposium on anticipatory cognitive embodied systems, Washington, DC*, 2005, p. 10.

[26] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[27] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein, "Deep knowledge tracing," in *Advances in Neural Information Processing Systems*, 2015, pp. 505–513.

[28] H. Ghasemzadeh, V. Loseu, and R. Jafari, "Structural action recognition in body sensor networks: Distributed classification based on string matching," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 2, pp. 425–435, 2010.

[29] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, no. Aug, pp. 2493–2537, 2011.

[30] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning.* ACM, 2008, pp. 160–167.

[31] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[32] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning.* MIT press, 2016.

[33] Y. Zhou and R. Chellappa, "Computation of optical flow using a neural network," in *IEEE International Conference on Neural Networks*, vol. 1998, 1988, pp. 71–78.

[34] S.-J. Huang, R. Jin, and Z.-H. Zhou, "Active learning by querying informative and representative examples," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 10, Oct. 2014.

[35] S. Sabato and T. Hess, "Interactive algorithms: from pool to stream," in *Proceedings of the 29th Annual Conference on Learning Theory (COLT), JMLR Workshop and Conference Proceedings*, New-York City, USA, Jun. 2016.

[36] Z. Xu, R. Akella, and Y. Zhang, "Incorporating diversity and density in active learning for relevance feedback," in *Proceedings of 29th European Conference on IR Research*, Rome, Italy, Apr. 2007.

[37] K.-S. Jun and R. Nowak, "Graph-based active learning: A new look at expected error minimization," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Washington, DC, USA, Dec. 2016.

[38] S. C. H. Hoi, R. Jin, J. Zhu, and M. R. Lyu, "Batch mode active learning and its application to medical image classification," in *23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006.

[39] X. Shen and C. Zhai, "Active feedback in ad hoc information retrieval," in *28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Salvador, Brazil, Aug. 2005.

[40] O. Sener and S. Savarese, "Active learning for convolutional neural networks: A core-set approach," in *ICLR 2018, The Sixth International Conference on Learning Representations*, Vancouver, Canada, May 2018.

[41] M. Nourollahi, S. A. Rokni, and H. Ghasemzadeh, "Proximity-based active learning on streaming data: A personalized eating moment recognition," *CoRR*, vol. abs/2003.13098, 2020. [Online]. Available: https://arxiv.org/abs/2003.13098

[42] Z. E. Ashari and H. Ghasemzadeh, "Mindful active learning," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, S. Kraus, Ed. ijcai.org, 2019, pp. 2265–2271. [Online]. Available: https://doi.org/10.24963/ijcai.2019/314

[43] M. Stikic and B. Schiele, "Activity recognition from sparsely labeled data using multi-instance learning," in *Proceedings of 4th International Symposium on Location- and Context-Awareness*, Tokyo, Japan, May 2009.

[44] T. Diethe, N. Twomey, and P. Flach, "Active transfer learning for activity recognition," in *Proceedings of proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, Bruges, Belgium, Apr. 2016.

[45] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *Pervasive computing.* Springer, 2004, pp. 1–17.

[46] J. Wang, B. Cao, P. Yu, L. Sun, W. Bao, and X. Zhu, "Deep learning towards mobile applications," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS).* IEEE, 2018, pp. 1385–1393.

[47] Y. Deng, "Deep learning on mobile devices: a review," in *Mobile Multimedia/Image Processing, Security, and Applications 2019*, vol. 10993. International Society for Optics and Photonics, 2019, p. 109930A.

[48] P. Zhang, E. Lo, and B. Lu, "High performance depthwise and pointwise convolutions on mobile devices." in *AAAI*, 2020, pp. 6795–6802.

[49] X. Ma, F.-M. Guo, W. Niu, X. Lin, J. Tang, K. Ma, B. Ren, and Y. Wang, "Pconv: The missing but desirable sparsity in dnn weight pruning for real-time execution on mobile devices." in *AAAI*, 2020, pp. 5117–5124.

[50] X. Zhang, C. Hao, Y. Li, Y. Chen, J. Xiong, W.-m. Hwu, and D. Chen, "A bi-directional co-design approach to enable deep learning on iot devices," *arXiv preprint arXiv:1905.08369*, 2019.

[51] B. Yang, L. Yang, X. Li, W. Zhang, H. Zhou, Y. Zhang, Y. Ren, and Y. Shi, "2-bit model compression of deep convolutional neural network on asic engine for image retrieval," *arXiv preprint arXiv:1905.03362*, 2019.

[52] A. Burns, B. R. Greene, M. J. McGrath, T. J. O'Shea, B. Kuris, S. M. Ayer, F. Stroiescu, and V. Cionca, "Shimmer™–a wireless sensor platform for noninvasive biomedical research," *Sensors Journal, IEEE*, vol. 10, no. 9, pp. 1527–1534, 2010.

[53] D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkl, A. Ferscha, J. Doppler, C. Holzmann, M. Kurz, G. Holl, R. Chavarriaga, H. Sagha, H. Bayati, M. Creatura, and J. d. R. Millán, "Collecting complex activity data sets in highly rich networked sensor environments," in *Seventh International Conference on Networked Sensing Systems*, 2010. [Online]. Available: http://www.inss-conf.org/2010/

[54] B. Barshan and M. C. Yüksek, "Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units," *The Computer Journal*, vol. 57, no. 11, pp. 1649–1667, 2013.

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

[56] M. A. Alsheikh, A. Selim, D. Niyato, L. Doyle, S. Lin, and H.-P. Tan, "Deep activity recognition models with triaxial accelerometers." in *AAAI Workshop: Artificial Intelligence Applied to Assistive Technologies and Smart Environments*, 2016.

[57] F. J. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[58] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011, pp. 315–323.

[59] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[60] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[61] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[62] R. Fallahzadeh and H. Ghasemzadeh, "Personalization without user interruption: boosting activity recognition in new subjects using unlabeled data," in *Proceedings of the 8th International Conference on Cyber-Physical Systems.* ACM, 2017, pp. 293–302.

[63] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural computation*, vol. 10, no. 7, pp. 1895–1923, 1998.

[64] A. Calatroni, D. Roggen, and G. Tröster, "Automatic transfer of activity recognition capabilities between body-worn motion sensors: Training newcomers to recognize locomotion," *Eighth international conference on networked sensing systems (INSS'11), Penghu, Taiwan*, vol. 6, 2011.

[65] F. J. O. Morales and D. Roggen, "Deep convolutional feature transfer across mobile activity recognition domains, sensor modalities and locations," in *Proceedings of the 2016 ACM International Symposium on Wearable Computers.* ACM, 2016, pp. 92–99.

[66] R. Saeedi, S. Norgaard, and A. H. Gebremedhin, "A closed-loop deep learning architecture for robust activity recognition using wearable sensors," in *Big Data (Big Data), 2017 IEEE International Conference on.* IEEE, 2017, pp. 473–479.

[67] S. J. Pan and Q. Yang, "A survey on transfer learning," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 22, no. 10, pp. 1345–1359, Oct 2010.

[68] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on.* IEEE, 2013, pp. 6645–6649.

[69] M. Zeng, L. T. Nguyen, B. Yu, O. J. Mengshoel, J. Zhu, P. Wu, and J. Zhang, "Convolutional neural networks for human activity recognition using mobile sensors," in *Mobile Computing, Applications and Services (MobiCASE), 2014 6th International Conference on.* IEEE, 2014, pp. 197–205.

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

[71] C. A. Ronao and S.-B. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," *Expert Systems with Applications*, vol. 59, pp. 235–244, 2016.

[72] B. Zhou, J. Yang, and Q. Li, "Smartphone-based activity recognition for indoor localization using a convolutional neural network," *Sensors*, vol. 19, no. 3, p. 621, 2019.

[73] I. A. Lawal and S. Bano, "Deep human activity recognition using wearable sensors," in *Proceedings of the 12th ACM International Conference on PErvasive Technologies Related to Assistive Environments-PETRA'19.* ACM, 2019, pp. 45–48.

[74] A. Ignatov, "Real-time human activity recognition from accelerometer data using convolutional neural networks," *Applied Soft Computing*, vol. 62, pp. 915–922, 2018.

[75] N. M. Rad, A. Bizzego, S. M. Kia, G. Jurman, P. Venuti, and C. Furlanello, "Convolutional neural network for stereotypical motor movement detection in autism," *arXiv preprint arXiv:1511.01865*, 2015.

[76] N. Y. Hammerla, S. Halloran, and T. Ploetz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," *arXiv preprint arXiv:1604.08880*, 2016.

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

[78] P. Alinia, S.-I. Mirzadeh, and H. Ghasemzadeh, "Actilabel: A combinatorial transfer learning framework for activity recognition," *ArXiv*, vol. abs/2003.07415, 2020.

[79] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey," *arXiv preprint arXiv:1808.05377*, 2018.

[80] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *arXiv preprint arXiv:1611.01578*, 2016.

[81] S. Jastrzebski, Z. Kenton, D. Arpit, N. Ballas, A. Fischer, Y. Bengio, and A. J. Storkey, "Three factors influencing minima in sgd," *ArXiv*, vol. abs/1711.04623, 2018.

[82] Y. Guo, H. Shi, A. Kumar, K. Grauman, T. Rosing, and R. Feris, "Spottune: transfer learning through adaptive fine-tuning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4805–4814.

[83] V. F. Rey and P. Lukowicz, "Label propagation: An unsupervised similarity based method for integrating new sensors in activity recognition systems," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 3, pp. 94:1–94:24, Sep. 2017.

[84] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, 2019.

[85] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[86] S.-I. Mirzadeh, M. Farajtabar, and H. Ghasemzadeh, "Dropout as an implicit gating mechanism for continual learning," *ArXiv*, vol. abs/2004.11545, 2020.