Personalizing Activity Recognition Models through Quantifying Different Types of Uncertainty using Wearable Sensors

Ali Akbari, *Student Member, IEEE,* Roozbeh Jafari, *Senior Member, IEEE,*

***Abstract*—Recognizing activities of daily living (ADL) provides vital contextual information that enhances the effectiveness of various mobile health and wellness applications. Development of wearable motion sensors along with machine learning algorithms offer a great opportunity for ADL recognition. However, the performance of the ADL recognition systems may significantly degrade when they are used by a new user due to inter-subject variability. This issue limits the usability of these systems. In this paper, we propose a deep learning assisted personalization framework for ADL recognition with the aim to maximize the personalization performance while minimizing solicitation of inputs or labels from the user to reduce user's burden. The proposed framework consists of unsupervised retraining of automatic feature extraction layers and supervised fine-tuning of classification layers through a novel active learning model based on a given model's uncertainty. We design a Bayesian deep convolutional neural network with stochastic latent variables that allows us to estimate both aleatoric (data-dependent) and epistemic (model-dependent) uncertainties in recognition task. In this study, for the first time, we show how distinguishing between the two aforementioned sources of uncertainty leads to more effective active learning. The experimental results show that our proposed method improves the accuracy of ADL recognition on a new user by 25% on average compared to the case of using a model for a new user with no personalization with an average final accuracy of 89.2%. Moreover, our method achieves higher personalization accuracy while significantly reducing user's burden in terms of soliciting inputs and labels compared to other methods.**

***Index Terms*—Activity recognition, Personalization, Wearable sensors, Deep learning, Active learning, Uncertainty quantification, Unsupervised learning**

I. Introduction

RECOGNIZING activities of daily living (ADL) is gaining bold traction as it provides vital information about people, their activities, and important contextual insight that enhances the effectiveness of mobile health and wellness delivery paradigms [1]–[4]. Many different applications such as physical fitness monitoring, diet monitoring, assisted living, and remote health monitoring benefit from the contextual information provided by ADL recognition systems [5]–[8]. Development of wearable sensing technologies has provided a great opportunity for understanding the data behaviors of ADLs by means of various devices such as smartwatches and smartphones. Data gathered by wearable sensors could be analyzed by rigorous machine learning models to build a system to recognize these ADLs [9], [10]. However, a given model trained on a certain user may not generalize well to new users due to variation in how people perform specific activities [11]. Therefore, it is necessary to personalize underlying machine learning models to new users.

Supervised and unsupervised learning approaches contain promising methods to design adaptable machine learning models with personalization capabilities [12]–[14]. Supervised learning requires the gathering of annotated data from a new user to retrain the machine learning models. However, this collection process is time consuming and burdensome. In fact, it is shown that user compliance to the wearable devices decay over time, particularly, when the collection system requires constant interaction [15]. Therefore, it is vital to personalize the machine learning models for new users with minimum labeled data available in order to minimize the burden on the users. Unsupervised retraining approaches, which relax the need for user annotation, attempt to assign pseudo labels to unlabeled data by leveraging cross-user similarities, and then use them for retraining the model parameters [11]. It has been shown that training and adapting the models in an unsupervised manner is often less accurate than supervised approaches [16], especially, in the presence of significant inter-subject variability. However, in the field of ADL recognition, there is typically an abundance of unlabeled data available; thus, leveraging these algorithms effectively could improve and accelerate the personalization process [17].

Another promising solution known as active learning may be oriented in a supervised fashion by identifying the most critical data samples and soliciting their labels from the users to retrain the machine learning models [18]. The uncertainty of the classification models is one of the mostly used metrics for identifying those critical samples. These methods, however, suffer from a few limitations: 1) They do not consider different sources of uncertainty in their models. For example, the model may experience high uncertainty due to temporary noise in sensor measurements. In this case, the model will make an unnecessary request for labels while it is not reasonable to use it for training [19]. 2) They also do not limit the number of interactions with the user and so do not consider their limited capacity in responding to external prompts for labeling, which introduces aforementioned burden. This becomes a prevalent issue when the model experiences noisy measurements for

A. Akbari is with the Department of Biomedical Engineering, Texas A&M University, College Station, TX, 77840 USA (e-mail: aliakbari@tamu.edu)

R. Jafari is with the Department of Biomedical Engineering, Electrical and Computer Engineering, Computer Science and Engineering, Texas A&M University, College Station, TX, 77840 USA (e-mail: rjafari@tamu.edu)

an extended period of time [18]. 3) Lastly, they often solely rely on the labels acquired from the user and ignore all other unlabeled data which could potentially be further leveraged in the retraining process [20].

Typically, there are two types of uncertainties surrounding wearable recognition systems. The first is referred to as aleatoric uncertainty, which is data dependent and is related to any noise present in sensor measurements such as sensor displacement and sensor movements with respect to the body. Such uncertainty cannot be mitigated by increasing the training data. The second type is known as epistemic uncertainty which is related to the inability of the model to recognize certain samples due to the lack of training. This increases as the model receives novel input samples with unfamiliar characteristics. This uncertainty can be mitigated by enhancing the training data, therefore, epistemic uncertainty should be strongly considered for active learning tasks.

In this study, we propose an ADL recognition system with a personalization capability. We leverage both active learning and unsupervised learning methods to facilitate the retraining process. For the active learning component, we leverage the uncertainty of the model on its decision to identify the critical samples where annotations should be requested and used for retraining. We propose a unified Bayesian deep learning framework to model and quantify the aforementioned types of uncertainties (aleatoric and epistemic) by considering stochasticity on both the parameters of the neural network and the latent variables served as features. Our proposed method extracts the features from the time series automatically through an unsupervised deep learning framework and learns their posterior distribution given the input data through a variational autoencoder (VAE) based model. To account for randomness of the model weights, we utilize the Dropout Bayesian network [21]. To the best of our knowledge, this is the first work that proposes to distinguish different types of uncertainty for active learning via wearable sensors. Moreover, the proposed framework has the ability to learn from unlabeled data through the autoencoder framework, which is leveraged to retrain the parameters of the feature extraction neural network. In summary, the contribution of this study is as follows:

- We design a unified framework for automatic feature extraction, classification, and estimation of uncertainty of the classifier to incorporate active learning in human activity recognition.
- We propose a method for quantifying both epistemic and aleatoric uncertainties and demonstrate how differentiating them is essential to achieve a more effective active learning.
- We design a new framework for deep learning for activity recognition that incorporates uncertainty quantification and unsupervised retraining.
- We propose an active learning technique leveraging quantified uncertainties with the ability to consider limited capacity of the users to respond to external prompts and solicitation of labels.
- We show how the proposed personalization framework can adapt itself to new users more effectively and quickly compared to the existing paradigms.

II. Related Works

ADL recognition has become an important component of context-aware systems in the last decade [22]. This contextual information provides valuable knowledge to better interpret biomedical signals, diagnose with more confidence, intervene and treat more efficiently, detect emergency situations, make proper decisions based on the context, assist people with chronic disabilities to perform their daily activities independently, and monitor patients more precisely [23], [24]. Accurate and robust recognition of human activities requires gathering massive amount of labeled data to train powerful machine learning models [25], [26]. Different machine learning algorithms have been utilized to perform human activity recognition with various types of wearable and environmental sensors [27]–[30]. All these investigations have focused on hand-crafted features that are useful for simple or low-level human activities such as sitting or walking. However, detecting complex ADL needs more complicated features and patterns which cannot be executed easily with manually handcrafted features.

To address the challenge of extracting informative features for complex activities, deep learning has been recently used [31]–[33]. Several prior investigations have used data from inertial measurement units (IMU) or motion sensors to automatically extract features by leveraging convolutional neural networks (CNN) for detecting human gestures and activities [34]–[37]. In all these efforts, the principal aim has been to create generalized models for all users. The performance of these systems, however, degrade significantly when they are used for a new user who performs the activities differently form the set of training subjects [38].

To address the problem of inter-subject variability in human activities, the researchers have designed personalization techniques [19], [39]. An unsupervised retraining technique based on an ensemble model was designed for personalizing activity recognition models [40]. In this study, the model trained on the data of old subjects was used to assign labels to the data of a new subject, and then these data were used to update the ensemble model. However, it has been shown that with using merely unsupervised data it is hard to achieve high accuracy in personalization [16]. Active learning is a widely used technique in this area that seeks to identify important samples and interacts with the user to acquire labels for those samples. Those labeled data is then used to fully or partially retrain the classification models [41], [42]. Active learning techniques based on the entropy and the random forest committee of classifiers were investigated in prior studies. The entropy-based methods typically select data samples to solicit a label on the basis of the highest information gain. In random forest based methods, a forest of 100 trees was trained and the disagreement between the output of those trees for each sample was used to select the samples to obtain label for [20].

Uncertainty of the classifier is a widely used criterion to identify vital samples that require label [18], [43]. An active learning technique based on the uncertainty of the SVM model was developed for detecting exercise activities [18]. A logistic regressor was cascaded to the SVM's output to estimate the

SVM classifier's uncertainty. This work, however, does not take into account user's burden associated with soliciting labels, does not consider different types of uncertainty, and it uses handcrafted features. A prior study provided an active learning method under constrained query budget [6]. All these works only rely on the labeled data solicited from the user and they do not take advantage of abundance of unlabeled data available in the ADL recognition platforms. A personalization method integrating both unsupervised and supervised retraining was developed using model uncertainty for active learning [19]. This study quantifies the uncertainty using the similarity between the new data and the training data in dense regions. However, neither this work nor other uncertainty-based active learning methods consider the source of uncertainty when designing the active learning technique.

There are mainly two types of uncertainties present in sensor data. Aleatoric or data-dependent uncertainty is related to noisy sensor measurements and cannot be mitigated by increasing the training data, whereas the epistemic or model-dependent uncertainty is due to lack of enough training data and it can be alleviated by enhancing the training set. Therefore, considering the source of uncertainty is critical to identify the vital samples that can enhance the performance of personalization. In deep learning, often Softmax functions is used at the end of the pipeline to measure model uncertainty; however, it has been shown that it does not always capture model uncertainty [21], [44]. A framework has been designed to learn mappings from input data to aleatoric uncertainty in image recognition [21]. The authors compose these together with epistemic uncertainty approximations. However, this model learns aleatoric uncertainty based on the assumption that in the training phase they have access to examples of the disturbed data (e.g., highly textured input images or far objects) to train the system, which is not always the case in ADL recognition. In ADL recognition, the disturbance in sensor measurements could be the result of electrical noises or sensor movements with respect to the body, where examples of such a noisy data may not be available during the training phase. Therefore, the system should be able to estimate the aleatoric uncertainty with no need to observe the examples in the training phase. To address the aforementioned issues and to maximize the effectiveness of personalization while minimizing the user interaction, we propose a deep learning assisted method for measuring different types of uncertainties.

III. Overall Design

In this study we propose a framework for personalization of machine learning models for the applications of ADL recognition using wearable sensors. This unified framework can quantify different types of uncertainty of wearable sensors in order to provide an effective active learning model. Moreover, it allows for leveraging unlabeled data samples within an unsupervised autoencoder-based model to boost the personalization process. Figure 1 illustrates the overall proposed framework. Figure 1-a shows the training phase of the model using the labeled training data available from certain users, while Figure 1-b shows the personalization procedure.

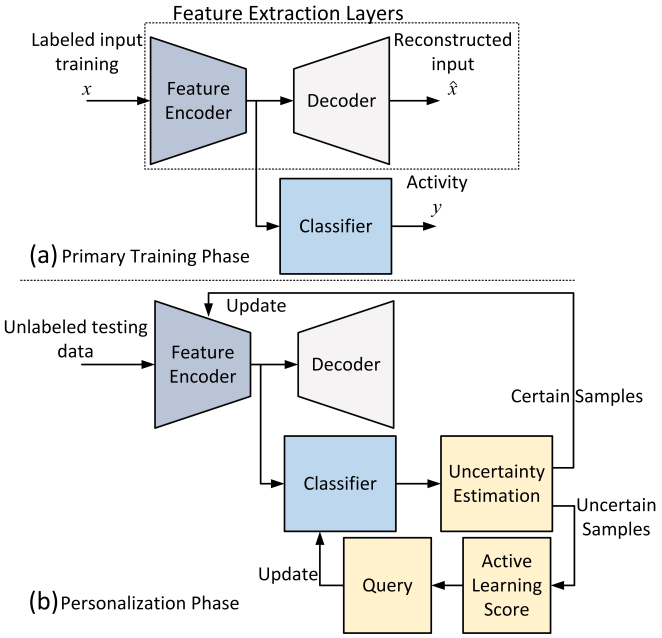

Fig. 1. The overall flow of the proposed model for personalizing ADL recognition models

In Figure 1-a, the encoder extracts features from the motion signals automatically through multiple convolutional layers. The decoder is then responsible to ensure that the features generated by the encoder learn the intrinsic structure of the input data regardless of the associated activity labels. These two components create a variational autoencoder (VAE) framework that is necessary for estimation of data-dependent uncertainty (*i.e.*, aleatoric) (see Section V-A). We add a classifier network in addition to this VAE framework to ensure that the extracted features remain discriminative for the ADL recognition task. It is important since the learned features should not only be task specific (discriminative), but also be able to retain the intrinsic structure of the data regardless of their task-specific labels [45]. In Figure 1-b, during the personalization phase, the model-dependent uncertainty (*i.e.*, epistemic) is used to identify the vital samples for which we should acquire labels from the user. This uncertainty is used in conjunction with a score function to limit the amount of user interaction while also considering other parameters into account (see Section V-C). Essentially, the critical samples for which the model is highly uncertain are identified by this module and labels are queried. However, the labels are not queried for the samples about which the model is confident. Those unlabeled samples are used to retrain the feature extraction layers in an unsupervised manner (see Section VI), while the labeled samples are used to fine-tune the classification layers.

IV. CNN for Automated Feature Extraction and Classification

Convolutional neural network (CNN) is a widely used model for various classification tasks due to its ability to extract features automatically. The network is fed with raw signal x and maps it to a latent variable z, which serve as extracted features. In a typical neural network f, we can write

$f = h \circ g$ where $g : x \rightarrow z$ maps raw inputs to a higher level feature space z and $h : z \rightarrow y$ is a discriminative function that maps the features to desired class labels. Multiple layers of CNN extract features from raw input signals by using kernels, which can be interpreted as filters applied to the signal via a convolution operation. The trainable weights of CNN kernels (W_g) and the weights of the classifier (W_h) are learnt through the training of the neural network.

In a classification setting, the network outputs a vector of unaries, where each unary corresponds to a class, and the vector obtained by the concatenation of all the unaries would be passed through a Softmax function to yield an estimation of a probability distribution over the classes. It has been shown that the Softmax does not provide a reliable and precise estimation of the actual model uncertainty [21], [44], [46]. Equation 1 shows the softmax where $f_i(x)$ is the network output for the i^{th} class before the softmax layer and N is the total number of classes.

$$softmax(f_i(x)) = \frac{e^{f_i(x)}}{\sum_{j=1}^{N} e^{f_j(x)}} \tag{1}$$

V. Supervised Active learning

Supervised retraining of classification models can end up with a higher accuracy than unsupervised retraining. However, it creates a huge burden on the user to provide lots of labeled training data for the system. On the other hand, a fully unsupervised retraining paradigm does not require such an extensive data collection and annotation but it does not have an ideal performance. To leverage both advantages, we propose a supervised active learning paradigm to select the most important samples and query the user only for those samples to minimize user's burden; in addition, our proposed system can make use of unsupervised retraining (Section VI). This can reduce the interaction of the system with the user drastically while the performance improvement after personalization would be still significant.

In order to select the most important samples to query from the user, the model needs to understand the uncertainty/confidence about its decision. To be more specific, samples of which the model is not confident need to be identified to fine-tune the classifier parameters [19]. However, it is also important that the model understands if it is uncertain due to lack of training or due to the noisy sensor data. The former, known as epistemic uncertainty, is essential for designing an effective active learning model while the latter is not. In fact, the model should query the label for the samples of which it is uncertain due to the lack of training. However, when the model is uncertain due to temporary noise in sensor data, it is not helpful to use those samples for retraining. Using those noisy samples for retraining could even degrade the performance of the classifier and lead to overfitting. In this section, we first explain our methodology for quantifying different types of uncertainty and then explain how the critical samples are identified to be labeled by the user for model retraining.

A. Uncertainty Quantification

In this section, we propose a unified framework to quantify both aleatoric and epistemic uncertainties in the classification model introduced in Section IV. To take into account the epistemic uncertainty, which is the measure of the model's uncertainty, we treat the weights of the discriminative function (W_h in Section IV) as random variables instead of deterministic ones. Moreover, to capture the aleatoric uncertainty, which is data-dependent, we consider the latent variables z as random variables instead of deterministic values. By considering the randomness on the weights and latent variables and by treating them as random values while considering their distribution, the final label inference can be written as Equation 2.

$$p_i = p(y_i|x) = \int p(y_i|W_h, x, z)p(W_h, z|x)dzdW_h \tag{2}$$

Where $p(y_i|W, x, z)$ is the likelihood function of the i^{th} class, which is calculated as the output of the neural network with a Softmax function in our classification task using Equation 1. $p(W_h, z|x)$ is the posterior distribution of weights and features given input data. The latent variable z in Equation 2 is independent of the weights W_h and also W_h is independent of the input data x, so Equation 2 can be written as Equation 3:

$$p_i = p(y_i|x) = \int p(y_i|W_h, x, z)p(W_h|D)p(z|x)dzdW_h \tag{3}$$

where D is the whole training dataset. Calculating the integral in Equation 3 is challenging but it can be approximated through Monte Carlo estimation as shown in Equation 4.

$$\begin{aligned} p_i = p(y_i|x) = \frac{1}{n}\sum p(y_i|\hat{W}_h, x, \hat{z}) \\ \hat{W}_h \sim p(W_h) \quad \hat{z} \sim p(z|x) \end{aligned} \tag{4}$$

Here, randomness on z is used to model the aleatoric uncertainty of the data. W_h distribution, on the other hand, is used to model the epistemic uncertainty.

In order to sample from the weight distribution, Dropout variational inference is a practical approach for approximation inference [21]. In this approach a Dropout layer is used after every dense layer, and the dropout is applied at the testing phase to sample from the approximate posterior (stochastic forward passes, referred to as Monte Carlo dropout [21]). It has been shown that dropping weights randomly during testing time is equivalent to sampling from the distribution of the weights [21]. To accomplish the sampling, each data sample is passed through the network multiple times (n times) and during every pass the weights of the network are dropped randomly with the probability of p_{drop}. The output of the network for all n passes are calculated and the average is interpreted as the Monte Carlo estimation for Bayesian inference.

Estimating the posterior distribution of latent variables/features $p(z|x)$ is then required to complete the calculation in Equation 3. Note that z is an unobserved latent variable while our observation is x. Bayesian analysis can be directly used to calculate $p(z|x)$ as $p(z|x) = \frac{p(x|z)p(z)}{p(x)}$; however, this leads to an intractable integral for calculating the denominator. In order to address this problem, we approximate $p(z|x)$ with a variational distribution $q(z|x)$ from a Gaussian distribution family and try to find the best set of parameters

for $q(z|x)$ such that it closely estimates the $p(z|x)$. To find the parameters of the variational estimation $q(z|x)$, we could minimize the Kullback-Leibler divergence (D_{KL}) between the two distributions [47]:

$$\min D_{KL}\{q(z|x)||p(z|x)\}$$

This minimization can then be written as follows:

$$\begin{aligned} D_{KL}\{q(z|x)||p(z|x)\} &= E_{z\sim q}[\log q(z|x) - \log p(z|x)] \\ &= E_{z\sim q}[\log q(z|x) - \log p(x|z) - \log p(z) + p(x)] \end{aligned} \tag{5}$$

where $p(x)$ is derived from the expectation as it does not depend on z. By rearranging Equation 5 we have:

$$\begin{aligned} &\log p(x) - D_{KL}\{q(z|x)||p(z|x)\} = \\ &E_{z\sim q}[\log p(x|z)] - D_{KL}\{q(z|x)||p(z)\} \end{aligned} \tag{6}$$

To minimize the KL divergence between $q(z|x)$ and $p(z|x)$, we can minimize the right hand side of Equation 6. This is exactly the objective function of a variational autoencoder [47]. The first term on the right hand side of Equation 6 is the loss of reconstructing input x from the latent variable z and the second term is the divergence between the variational approximation with the prior distribution of z. For this prior distribution, we use a standard Gaussian distribution with the mean of zero and variance of one. Therefore, by training a variational autoencoder and leveraging the latent variable z as the features that are provided to a classifier, we can approximate the posterior of the features to calculate the inference in Equation 4. In a typical VAE, the encoder estimates the variational approximation $q(z|x)$ and the decoder estimates the first term on the right hand side of Equation 6. Output of the encoder in a VAE is the mean and standard deviation that serve as the parameters of a Gaussian distribution that models the posterior distribution of z given x.

A typical VAE containing convolutional layers in the encoder can extract informative features from a signal in an unsupervised manner. In a typical VAE framework, the only concern is to extract features that can retain the structure of the input data. However, in a supervised classification problem the features should be discriminative with respect to the labels given in the training data. Based on this intuition, we propose a new framework of deep neural network as shown in Figure 2 by modifying the typical VAE objective function as follows:

$$\begin{aligned} \mathcal{L} = p(y|x) + \big(&E_{z\sim q}[\log p(x|z)] - \\ &D_{KL}\{q(z|x)||p(z)\}\big) \end{aligned} \tag{7}$$

In fact, maximizing the $p(y|x)$ is added to the typical VAE objective function in Equation 6 to produce the new objective function in Equation 7. This objective function guides the VAE to produce latent features that not only can reproduce the input data but also discriminate between different class labels. The encoder, which serves as a feature extractor, estimates the parameters of the posterior distribution of the features. The decoder ensures that the latent variable z is able to retain the structure of the input data, and is discarded after the training. The classifier samples from the distribution of the features, which is approximated by the encoder, and maps those samples to the class labels.

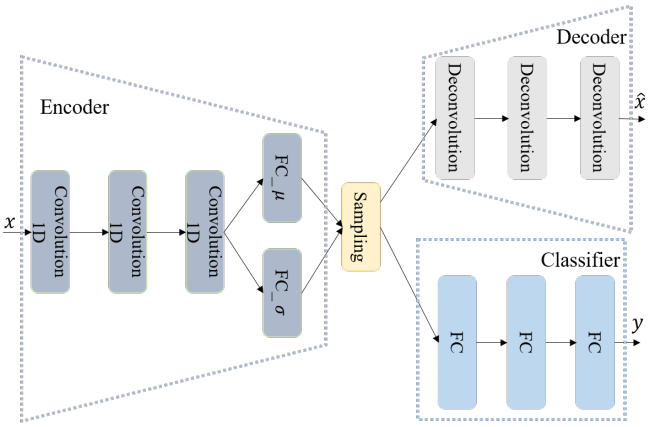

Fig. 2. The architecture of the proposed neural network for feature extraction and uncertainty estimation

It should be mentioned that this framework can be considered in line with the idea of pre-training an autoencoder in an unsupervised manner and replacing the decoder with a classifier to improve the performance [48]. The authors in [48] share this observation that features created by the autoencoder are a good representative for training datasets that support better generalization [49]. What distinguishes our work is that we embed the two processes of the classifier learning and the data-dependent feature extraction in a single framework, which improves the discriminative power of our features compared to the case of unsupervised pre-training.

The procedure for estimating the label and the confidence of the classifier is shown in Algorithm 1 for the trained neural network shown in Figure 2. In algorithm 2, $OneHotEncoding(.)$ is a function that returns the one hot encoding of a vector and $std(.)$ calculates standard deviation. To predict the class label for each input data, n samples are acquired from the distribution of the features, the weights of the classifier are dropped randomly, and labels are generated by the classifier. n is a hyperparameter of the model that is determined empirically through cross-validation in the training phase. In our experiments, n=100 ended up with the highest cross-validation accuracy. The final decision of the classifier is the average of the outputs. Moreover the standard deviation of the generated outputs is the measure of uncertainty. Intuitively, the classifier would generate more consistent labels for the samples that it is confident on, while for non-confident samples it would generate distinct labels that leads to higher standard deviations. The uncertainty calculated by Algorithm 1 is called the combined uncertainty as it contains both aleatoric and epistemic uncertainties. To only consider the epistemic uncertainty, we use the mean of z in step 4 of Algorithm 1, instead of sampling from the posterior distribution of the features. In fact, in this uncertainty we only consider the randomness of the weights of the classifier network and do not care about the randomness of z, which explains the aleatoric uncertainty. On the other hand, to only consider the aleatoric uncertainty, we do not drop weights during the testing time as shown in step 5 of Algorithm 1. Leveraging this approach, we establish a method where we discard the model uncertainty and

Algorithm 1 Label and uncertainty estimation

Input: test data x, encoder network g, classifier network h, p_{drop}, number of labels c Initialize $prediction = zeros(n, c)$
$q(z|x) = g(x)$
for $j = 1$ **to** n **do**
  Take a sample $z_j \sim q(z|x_i)$
  Drop weights with probability of p_{drop}
  $y_j^i = h(z_j)$ // the output of softmax function
  $prediction[j,:] = OneHotEncoding(y_j^i)$
end for
$\hat{y} = \frac{1}{n}\sum_{j=1}^{n} y_j^i$
$Uncertainty = std(prediction)$

TABLE I
CHARACTERISTICS OF THE PROPOSED DEEP NEURAL NET.

	Layer	# of kernels/ neurons	Activation function
Encoder	Conv2d_1	32	ReLU
	Conv2d_2	64	ReLU
	Conv2d_3	100	ReLU
	FC_mean	20	Sigmoid
	FC_std	20	Sigmoid
Classifier	FC_1	64	ReLU
	FC_2	128	ReLU
	FC_3	Same as the # of classes	Softmax

we only take into account the uncertainty over latent variables z which is data-dependent.

B. Deep Neural Network Implementation

In this section, we discuss the details of all neural networks for various components used in this study. The detail of all encoder and classification layers are presented in Table I. For the encoder network as shown in Figure 2, we use three layers of CNN followed by one fully connected (FC) layer for each of mean and standard deviation estimation. Based on our experiments using less number of layers did not offer an acceptable accuracy and using more layers does not offer significant improvement in the performance of the system while it increases model complexity. For the classifier network we use three fully connected layers. The decoder contains three deconvolution layers. Re-parametrization trick is used for handling the sampling from a Gaussian distribution when training the network with backpropagation algorithm [47].

In the preprocessing phase, the data is normalized to retain zero mean and unit variance and segmented prior to supplying it into the CNN. We utilize a fixed-size window with a length of one second and overlap of 50%.

C. Identifying Critical Samples

Our primary goal is to determine the most important samples that can contribute to improving the retraining process for model personalization. Here, we define such samples as the ones of which the model is uncertain due to the lack of training data. Therefore, we need to identify the source of uncertainty for test samples and pick the ones with maximum epistemic uncertainty. During the training phase, we calculate the average combined, aleatoric, and epistemic uncertainties over all the correctly classified samples shown as σ_{cor}^{comb}, σ_{cor}^{ale}, and σ_{cor}^{epi} respectively. For a test sample we also calculate the three uncertainties calling them as σ_{test}^{comb}, σ_{test}^{ale}, σ_{test}^{epi}. The process of identifying uncertain samples starts by picking samples that have combined uncertainty above a threshold. This threshold is determined empirically for each dataset. Afterwards, we calculate the ratio between the increase in epistemic to the aleatoric uncertainty compared to the correctly classified samples as shown in Equation 8.

$$\sigma_{ratio} = \frac{\sigma_{test}^{epi} - \sigma_{cor}^{epi}}{\sigma_{test}^{ale} - \sigma_{cor}^{ale}} \tag{8}$$

However, the uncertainty is not the only parameter to consider to determine when labels need to be solicited. Using merely the uncertainty might lead to querying the user too frequently. To further control this process, we define a parameter q_{limit} that indicates the total number of questions allowed to be asked within a certain period of time. The system keeps track of how many questions it has asked so far as $q_{inquired}$. The model becomes more strict in querying the user as the number of questions previously asked increases. In fact, the likelihood of asking more questions should decrease as the model asks more. We model this behavior as a linear function of the number of remaining queries. The final score function to select samples to be queried from the user is shown in Equation 9.

$$s = \sigma_{ratio} + \beta \frac{q_{limit} - q_{inquired}}{q_{limit}} \tag{9}$$

where s is the score assigned to each sample, and β is a tuning constant that is determined empirically during the training phase by varying it between 0 to 1. We set this value as 0.1 in our experiments. In addition to the aforementioned parameters, a good sample to query from the user is the one with higher occurrence. Note that for ADL recognition we need to segment the sensor data into windows of a fixed length. Therefore, a label queried from the user can be typically assigned to one segment of data. However, we argue that if there are multiple consecutive segments of the data of which the system is consistently uncertain, then the queried annotation can be used to label all those segments with a higher chance that the label applies to all segments. To incorporate this assumption and observation, at each time step, we calculate the proposed score s for all segments within the last five minutes and query the user only if the value is higher than the threshold for all of them. The threshold can be set empirically through cross validation in the training phase for online active learning.

Lastly, the labeled data queried from the user is used to fine tune the weights of the classifier. In fact, when using these labeled data, we freeze all the weights of the encoder and decoder layers and only retrain the weights of the classification layers. This is important because the amount of the labeled data is very small and is not enough to retrain all the feature extraction layers. Moreover, it is shown that usually the first layers, which are responsible for feature extraction, are more

generalizable and transferable, while the last layers are more task specific and less transferable between different users [50].

VI. Unsupervised Retraining

Scarce labeled data is not sufficient to retrain all feature extraction layers including the encoder and decoder networks since they require to be extensively trained to learn appropriate features of motion signals. However, due to our autoencoder structure, we can leverage the huge amount of unlabeled data to retrain the encoder component which produces the hidden state that represents the patterns and features of the new data. When this occurs, we keep the weights of the classification layers unchanged as the encoder/decoder weights are updated. This unsupervised retraining allows the feature extraction layers to adapt to the patterns and morphology of the motion signal of the new user.

When combining this unsupervised retraining of the feature extraction layers with the supervised fine-tuning of the classification layers through the active learning process, the whole network is adapted effectively to the data of the the new user. So, feature extraction layers are updated to capture the patterns of the new signal, while the classification layers are updated to learn how to map those features to the desired activity classes. Consequently, we must first accomplish the unsupervised retraining to update the encoder-decoder weights before we perform the supervised fine-tuning of the classifier's weights using the labels gathered by the active learning module.

VII. Results

We evaluate the effectiveness of our methods for personalization of deep learning ADL recognition system on the new users. In this section, we start by introducing the datasets used for the evaluation and then presenting the performance of our personalization method on detecting ADLs and compare it to baselines and state-of-the-art approaches. Afterwards, we deeply analyze our uncertainty modeling by investigating its behavior in response to different sources of uncertainty to understand how it can improve the effectiveness of active learning designs. Finally, we investigate how personalization accuracy is affected upon presence/absence of each component of the proposed framework, including the supervised fine tuning of classification layers through active learning and the unsupervised retraining of the feature extraction layers. The purpose of this investigation is to show the importance of each of those components in the personalization process.

To demonstrate the effectiveness of our proposed framework, we used two publicly available datasets including PAMAP2 [51], and MoST [52]. PAMAP2 comprises of 18 physical activities measured by three wearable inertial measurement units (IMUs) with the sampling frequency of 100 Hz performed by 9 different subjects. IMUs were worn on three different body parts: the wrist of the dominant hand, the chest, and the ankle of the dominant foot. In this study, we used eight out of 18 activities, which have most number of samples. We used 3D acceleration and gyroscope sensors that results in 18 axis of data. MoST dataset, collected by our own group, contains 23 daily activities captured by six IMUs working at the frequency of 200 Hz placed on the arm, wrist, chest, ankle, and both legs. The data was collected from 20 healthy subjects. Since, several activities in this dataset are similar, we grouped them as one activity and once again, removed the classes with small training data. Table II represents the list of activities used in this study.

TABLE II
Activities in different datasets

	PAMAP2	**MoST**
	Biking	Sit-to-stand
	Sitting	Sitting
	Standing	Standing
	Walking	Walking
	Stair climbing	Grasping floor
	Lying down	Lying down
	Running	Turning 90°
	Rope jumping	Jumping
# of samples	17000	7500

A. Personalization Results

We assume that there is a large amount of labeled training data available for certain subjects, called primary subjects, that can be used to train the initial ADL recognition model. We aim to personalize the model that is trained on the primary subjects to a new subject to achieve the highest performance possible. For each dataset, we exclude one subject as the new subject and use the remaining subjects as the primary subjects to train the initial model. For the excluded subject, i.e., the test subject, we used 50% of the data for personalization (i.e., retraining) and used the remaining 50% for testing the accuracy of the personalized model. We repeat the experiments by changing the excluded subject to cover all the subjects. In other words, in our experiments, each subject has been treated as the new subject once. All reported results in this section are the average over all repetitions.

Table III shows the results of comparing our model to other baseline and existing methods regarding personalization accuracy. The first row in Table III represents the accuracy of the ADL recognition when the model is trained on primary subjects and it is tested on the same subjects. This shows the upper bound of training, when the testing and training data come from the same subjects. The second row shows the performance of the system when it is trained on primary subjects but it is tested on a new subject with no retraining and/or personalization of the model. This shows how the performance drops when the model is used for a new subject and also emphasizes the necessity of personalization in such systems. The third row shows the results of using all the labeled data of the new user for personalization through retraining of the whole neural networks. This case represents another upper bound, which achieving that in real-world scenarios is impossible due to high burden on the users. The fourth row represents the performance of personalization when the labels are queried for random samples. This is a baseline to show why smart active learning is required. The fifth row represents another baseline for active learning by using the output of Softmax function as a measure of uncertainty. In this approach, a query for a label is submitted to the user when the

TABLE III
PERSONALIZATION ACCURACY [%]

	PAMAP2	MoST
Primary subjects	94.2	91.8
New subjects with no retraining	53.7	64.6
Use all labeled data for retraining	93.8	91.9
Query labels for random samples	76.2	70.0
Active learning with softmax	74.9	77.2
Entropy-based active learning [20]	79.3	82.3
RF-based active learning	80.0	81.4
Our personalization method	**89.6**	**88.7**

output of the Softmax is less than a threshold. Based on our experiments setting that threshold to 0.7 obtains the best results in our study. The sixth row of the table shows the results of a state-of-the-art active learning method using entropy. Based on this model, the samples with the highest entropy are the most informative samples for the classifier and their label are acquired from the user [20]. The seventh row shows another state-of-the-art method that uses the agreement between the labels generated by different trees in a random forest classifier as a measure of uncertainty to select most uncertain samples for label solicitation [20]. Finally, the last row shows the performance of our proposed method for personalization. It should be noted that in this experiment, the total number of questions allowed to be asked is set to 200 for all the algorithms, which means that we query the labels for 200 samples of new data.

As Table III shows, our method outperforms all the baseline methods by 14.3% and existing methods by 8%. This shows the superiority of our method that is due to more effectively quantifying the uncertainty and the unsupervised retraining of feature encoder weights. In fact, we distinguish between different types of uncertainty and try to ask questions when the system is uncertain due to lack of training rather than facing noisy sensor measurements. Figure 3 reveals the importance of differentiating between different types of uncertainties for more effective active learning. For this analysis we add a synthetic Gaussian noise to our sensor measurements to understand the importance of the uncertainty decomposition. We add different level of noise to different data samples using Equation 10.

$$x_{noisy} = x_{clean} + \alpha.\epsilon \qquad \epsilon \sim N(0,1) \tag{10}$$

where α is the amplitude of the noise that is different for each data sample and is chosen form a uniform distribution from 0 to 0.1, and ϵ is a white noise. As Figure 3 shows, querying the label for the samples that are chosen based off of epistemic (i.e., model dependent) uncertainty provides highest improvement compared to using the combined or aleatoric (i.e., data dependent) uncertainty. Moreover, the ratio proposed in our method in Equation 8 achieves the highest personalization performance. The reason for this is as follows: the two uncertainty metrics have a complementary nature to some extent meaning that in presence of novel data, both epistemic and aleatoric uncertainties increase. However, the increase in epistemic uncertainty in this case is much more significant than the aleatoric uncertainty as the novel data is associated with model dependent uncertainty (see Section VII-B). The uncertainty ratio proposed in Equation 8 takes this observation

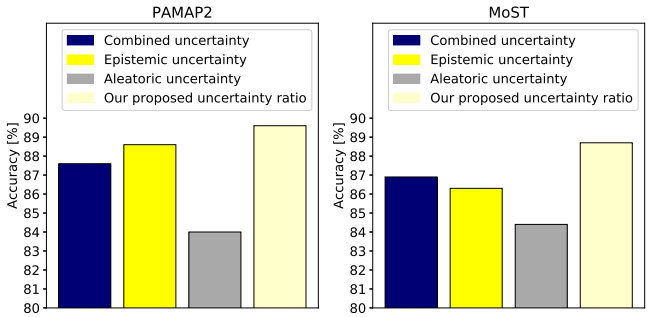

Fig. 3. Effect of using different uncertainty metrics on personalization performance

into account and helps the model to better understand if it is facing uncertainty due to lack of training rather than noisy measurements. In conclusion, Figure 3 provides the indication on the importance of distinguishing between the types of uncertainty for active learning tasks.

Figure 4 depicts the performance of personalization vs. the number of queries form the user averaged over all the subjects within each dataset. As the figure shows, our proposed method achieves the highest accuracy compared to the other methods, especially when the number of questions is very small. There are two reasons: first, our method attempts to choose the most important samples by taking into account the limitation on the number of questions; second, it uses unlabeled data along with the queried labels to adapt the feature extraction layers to the new data. Based on Figure 4, by querying at least 60 and 140 data points from the user, in MoST and PAMAP2 datasets, respectively, our method can achieve more than 20% improvement in the performance of the system compared to the case of using no personalization for the new users. Moreover, the acquired accuracy from the personalization by querying only 200 data points from the new user is only 3.7% less than the upper bound of using all labeled data from the new user (i.e., 1700 datapoints in PAMAP2 and 400 data points in MoST dataset on average).

B. Uncertainty metrics

To assess the quality of our uncertainty measurement technique and to better understand how they are associated with different sources of uncertainty in sensor data, we investigate three questions:

- How do epistemic and aleatoric uncertainties change when the system is dealing with novel data?
- How do these types of uncertainty behave when dealing with noisy sensor measurements? With last two questions, we investigate whether these two types of uncertainties are separable.
- Is there a relationship existing between these uncertainties and misclassifications?

First, we analyze the behavior of the quantified uncertainties when the system is presented with novel data. The novel data is the data drawn from a different distribution compared to the training data, and it could come from new activities for which the system has not been trained, or new users that perform activities differently. A good measure of uncertainty must

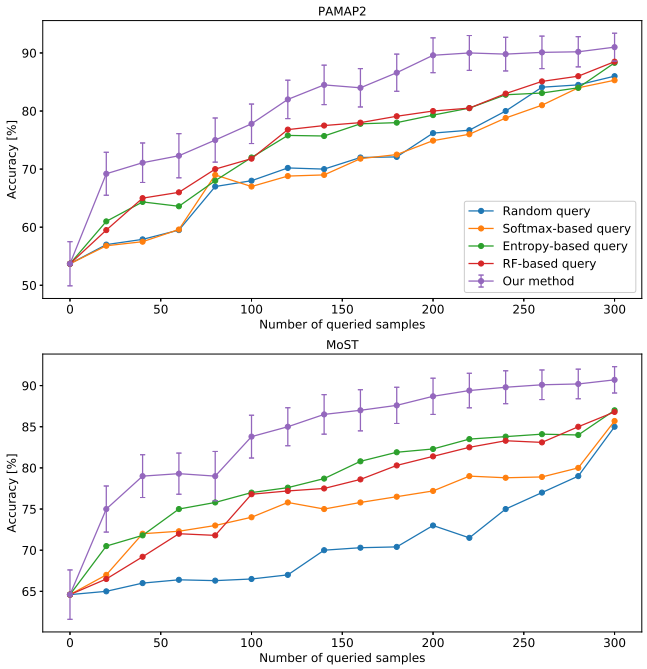

Fig. 4. Personalization performance vs. number of queried samples through active learning averaged over all the subjects

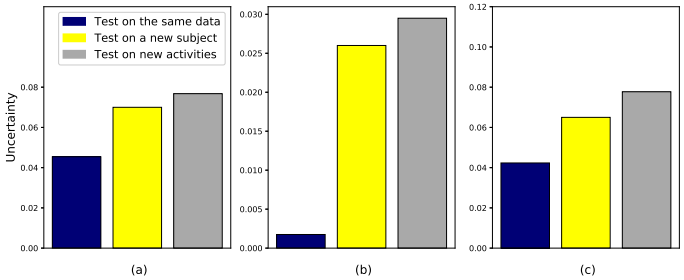

Fig. 5. The uncertainty when the model is tested on novel data (averaged over all the subjects). (a) combined uncertainty; (b) epistemic uncertainty; (c) aleatoric uncertainty

increase when the system is provided with novel unfamiliar data. Herein we compare the uncertainty in three different cases including testing on data same as the training data, data of new subjects, and data of new activities, for which the results are shown in Figure 5. In the first case (the navy bar in Figure 3), we test the model with data from the same distribution as the training (non-novel data). In other words, the model is tested on the data of the same subjects and with the same activities as in the training set. In this figure, it is seen that all combined (Figure 5-a), epistemic (Figure 5-b) and aleatoric (Figure 5-c) uncertainties are smaller compared to when the model is tested on the data of new subject or new activities (novel data). To test the model on novel data, in the second case (the yellow bar), the model is trained on all but one subject and is tested upon the data of the excluded (new) subject. In the last case (the gray bar), we test the model with the data of new activities that were not used in the training. Expectedly, as Figure 5 shows, all combined (Figure 5-a), epistemic (Figure 5-b) and aleatoric (Figure 5-c) uncertainties increase when the model faces novel data (yellow and gray bars in comparison to the navy bar). However, this increase is much more significant in the epistemic uncertainty (Figure 5-b) compared to the aleatoric uncertainty (Figure 5-c). This confirms the initial hypothesis about the type of uncertainties and it is in line with prior reports [53]. In fact, it shows that the epistemic uncertainty, which considers the model uncertainty, is much more sensitive to the lack of training data compared to the aleatoric uncertainty. By measuring this uncertainty, we can realize if the input data is not familiar for the model. In such cases, it is reasonable to solicit the user to get more information about the activity label for retraining the model.

Second, we expect the uncertainty to increase in response to noisy sensor measurements too. Thus, the second experiment is devoted to the analysis of the effect of sensor noise on each type of uncertainties quantified in this study as shown in Figure 6. We seek to understand whether the uncertainty sourced by the noisy data is distinguishable from the uncertainty of the model sourced by lack of sufficient training data. We add synthetic noise to data where varying level of noise is represented in the X axis of Figure 6. Each axis of raw sensor data is corrupted with a Gaussian noise, as described in Equation 10, with the only difference here being that we use various constant values for α as depicted on the X-axis of Figure 6. It should be noted that we chose the range [0,0.4] for α to better show how the uncertainty changes in response to different levels of noise in the data. We did not use larger values for α because the noise dominates the signal and the data becomes non-informative. As Figure 6 shows, the aleatoric uncertainty (dashed navy line), which corresponds to the data, increases consistently as the noise magnitude (and power) increases. Contrarily, the epistemic uncertainty (dotted red in Figure 6) does not show an increasing pattern as consistent as the aleatoric uncertainty. However, we naturally expect to see an increase in the uncertainty when the system is fed with noisy data, and that is what is observed with the aleatoric as well as the combined uncertainty, which is dominated by the aleatoric. Hence, the decreasing pattern in the epistemic uncertainty, as shown in Figure 6, indicates that this type of uncertainty is not measuring the true data-dependent uncertainty due to noisy signals. Combined uncertainty (solid navy line in Figure 6), similar to aleatoric uncertainty, shows a steady increase which is desired since the noisy data introduces challenges for the predictions. This shows that the uncertainty increases when the system is presented with noisy sensor measurements while it can still capture the fact that this uncertainty is not caused by the model not knowing the data but because the data is noisy.

Figures 5 and 6 show that the proposed measures of uncertainty are sensitive to different sources of uncertainty and this allows the system to understand which type of uncertainty is being observed. Now we investigate the change in the uncertainties when the output of the model is correct versus when the predictions are incorrect. We expect to observe an increase in the uncertainty when the classifier makes a mistake. If this appears to be the case, then the system will be able to detect potential errors and appropriately intervene. Figure 7 depicts the aleatoric, epistemic and combined uncertainties, averaged for both correctly and incorrectly classified data samples. According to the figure, the uncertainties in the correctly classified data samples (the navy bar) are steadily

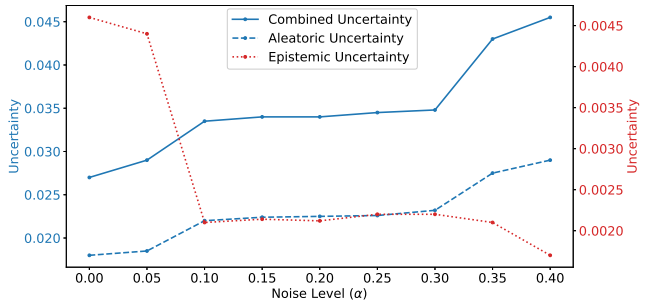

Fig. 6. The uncertainty of the model in presence of different noise levels (averaged over all the subjects). X axis corresponds to values of α in Equation 10.

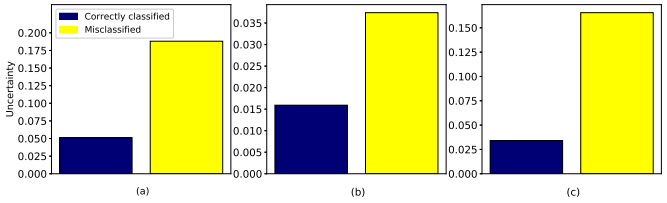

Fig. 7. Comparing the uncertainty for the samples that are correctly classified by the classifier and the samples that are misclassified (a) combined uncertainty; (b) epistemic uncertainty; (c) aleatoric uncertainty

lower than the ones that are misclassified (the yellow bar). This indicates that the model makes more mistakes on the data samples on which it is less certain. Therefore, the uncertainties developed here can serve their true purpose.

To further investigate the effectiveness of the proposed uncertainty metrics, we compare it to the Softmax output. To accomplish this comparison, we first determine 0.075 as a threshold on our combined uncertainty for distinguishing between certain vs. uncertain data samples based on Figure 7-a. Surprisingly, 45% of all misclassified samples that are labeled as uncertain by our system (i.e., their uncertainty is above the threshold) have a Softmax output of higher than 0.95. This shows that, even the misclassified samples that are not located close to the decision boundary of the neural network (i.e., Softmax is very certain about them) can be detected by the proposed uncertainty metric. In other words, the proposed uncertainty is capable of capturing the uncertainty even for the samples that are far from the decision boundaries of the classifier. Moreover, from all the misclassified samples, only 13% have the Softmax output of below 0.95, which shows the inability of Softmax in detecting uncertainty corresponding to misclassification, while 47% of them have been detected as uncertain by our model. This shows the superiority of our algorithm over Softmax regarding detecting the uncertainty to mark misclassified samples.

C. Evaluating Components of the Proposed Model

The proposed personalization framework for human ADL recognition consists of unsupervised retraining of the feature extraction layers and supervised fine-tuning of the classification layers using the labels that are acquired by the active learning. In this section, we aim to analyze the effect of each of those two components on the personalization performance. Figure 8 compares the personalization performance

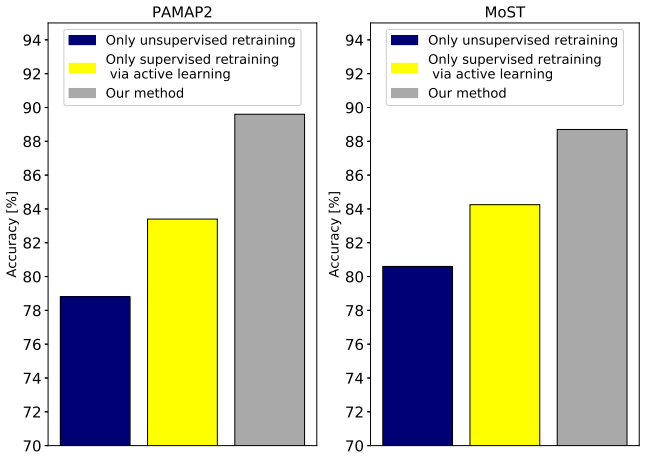

Fig. 8. The effect of different components of the proposed method on personalization performance

of our method when various components are disabled. As the figure shows, in one hand, the accuracy of the ADL recognition for new user (i.e., personalized model) drops by 5.4% when we do not retrain the feature extraction layers in an unsupervised manner, and we only rely on supervised fine-tuning of the classification layers. On the other hand, the performance drops by 9.5% when we ignore the supervised fine-tuning with the labels acquired by active learning, and we only rely on unsupervised retraining. This demonstrates the importance of supervised fine-tuning in personalization. Overall, Figure 8 shows that the unsupervised retraining of feature extraction layers and the supervised fine-tuning of the classification layers using the labels acquired by active learning are complementary, and both techniques are required to obtain effective personalization.

VIII. Conclusion

We proposed a personalization framework for ADL recognition using deep learning. The proposed method consists of supervised fine-tuning of classification layers as well as unsupervised retraining of feature extraction layers. For the supervised fine-tuning we proposed an active learning technique to acquire labels for most important samples. To achieve a more effective supervised active learning, we designed a unified deep Bayesian neural network to detect different types of uncertainties. Through experimental analysis, we demonstrated how data-dependent and model-dependent uncertainties could be distinguished and measured by the proposed method. We leveraged the model dependent uncertainty to identify the samples that are important to gain maximum accuracy through fine-tuning the classification model. Our experiments showed that in general, personalization is critical when an ADL recognition system is used for a new user. Moreover, using unlabeled data as well as labeled data acquired by active learning, understanding the source of uncertainty, and limiting the amount of interaction with the user to solicit labels while designing the active learning method are vital components to achieve the maximum personalization performance while minimizing the burden on the users. The proposed method is important to improve usability of ADL recognition systems

that provide important contextual information for many mobile health and wellness service applications such as patient monitoring, assisted living, dietary and fitness monitoring.

ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation, under grants CNS-1734039 and ECCS-1509063, and the National Institutes of Health under grant 1R01EB028106-01. 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] A. Pantelopoulos and N. G. Bourbakis, "A survey on wearable sensor-based systems for health monitoring and prognosis," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 1, pp. 1–12, 2010.

[2] D. De, P. Bharti, S. K. Das, and S. Chellappan, "Multimodal wearable sensing for fine-grained activity recognition in healthcare," *IEEE Internet Computing*, vol. 19, no. 5, pp. 26–35, 2015.

[3] X. Liu, L. Liu, S. J. Simske, and J. Liu, "Human daily activity recognition for healthcare using wearable and visual sensing data," in *2016 IEEE International Conference on Healthcare Informatics (ICHI)*. IEEE, 2016, pp. 24–31.

[4] E. V. Añazco, P. R. Lopez, S. Lee, K. Byun, and T.-S. Kim, "Smoking activity recognition using a single wrist imu and deep learning light," in *Proceedings of the 2nd International Conference on Digital Signal Processing*. ACM, 2018, pp. 48–51.

[5] C. Crema, A. Depari, A. Flammini, E. Sisinni, T. Haslwanter, and S. Salzmann, "Imu-based solution for automatic detection and classification of exercises in the fitness scenario," in *2017 IEEE Sensors Applications Symposium (SAS)*. IEEE, 2017, pp. 1–6.

[6] R. Solis, A. Pakbin, A. Akbari, B. J. Mortazavi, and R. Jafari, "A human-centered wearable sensing platform with intelligent automated data annotation capabilities," in *Proceedings of the International Conference on Internet of Things Design and Implementation*. ACM, 2019, pp. 255–260.

[7] S. Chernbumroong, S. Cang, A. Atkins, and H. Yu, "Elderly activities recognition and classification for applications in assisted living," *Expert Systems with Applications*, vol. 40, no. 5, pp. 1662–1674, 2013.

[8] D. J. Cook and M. Schmitter-Edgecombe, "Assessing the quality of activities in a smart environment," *Methods of information in medicine*, vol. 48, no. 05, pp. 480–485, 2009.

[9] E. Guenterberg, H. Ghasemzadeh, V. Loseu, and R. Jafari, "Distributed continuous action recognition using a hidden markov model in body sensor networks," in *International Conference on Distributed Computing in Sensor Systems*. Springer, 2009, pp. 145–158.

[10] H. Ghasemzadeh, E. Guenterberg, K. Gilani, and R. Jafari, "Action coverage formulation for power optimization in body sensor networks," in *Proceedings of the 2008 Asia and South Pacific Design Automation Conference*. IEEE Computer Society Press, 2008, pp. 446–451.

[11] 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.

[12] E. Hoque and J. Stankovic, "Aalo: Activity recognition in smart homes using active learning in the presence of overlapped activities," in *2012 6th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth) and Workshops*. IEEE, 2012, pp. 139–146.

[13] D. Riboni, T. Sztyler, G. Civitarese, and H. Stuckenschmidt, "Unsupervised recognition of interleaved activities of daily living through ontological and probabilistic reasoning," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2016, pp. 1–12.

[14] L. Yao, F. Nie, Q. Z. Sheng, T. Gu, X. Li, and S. Wang, "Learning from less for better: semi-supervised activity recognition via shared structure discovery," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2016, pp. 13–24.

[15] R. Wang, F. Chen, Z. Chen, T. Li, G. Harari, S. Tignor, X. Zhou, D. Ben-Zeev, and A. T. Campbell, "Studentlife: assessing mental health, academic performance and behavioral trends of college students using smartphones," in *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing*. ACM, 2014, pp. 3–14.

[16] P. Siirtola, H. Koskimäki, and J. Röning, "Importance of user inputs while using incremental learning to personalize human activity recognition models," *arXiv preprint arXiv:1905.11775*, 2019.

[17] M. A. Alsheikh, D. Niyato, S. Lin, H.-P. Tan, and Z. Han, "Mobile big data analytics using deep learning and apache spark," *IEEE network*, vol. 30, no. 3, pp. 22–29, 2016.

[18] A. Mannini and S. Intille, "Classifier personalization for activity recognition using wrist accelerometers," *IEEE journal of biomedical and health informatics*, 2018.

[19] H. Hossain, M. Al Haiz Khan, and N. Roy, "Deactive: scaling activity recognition with active deep learning," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 2, p. 66, 2018.

[20] S. Bagaveyev and D. J. Cook, "Designing and evaluating active learning methods for activity recognition," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. ACM, 2014, pp. 469–478.

[21] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, 2016, pp. 1050–1059.

[22] A. Avci, S. Bosch, M. Marin-Perianu, R. Marin-Perianu, and P. Havinga, "Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey," in *23th International conference on architecture of computing systems 2010*. VDE, 2010, pp. 1–10.

[23] S. L. Lau, I. König, K. David, B. Parandian, C. Carius-Düssel, and M. Schultz, "Supporting patient monitoring using activity recognition with a smartphone," in *2010 7th International Symposium on Wireless Communication Systems*. IEEE, 2010, pp. 810–814.

[24] J. Pansiot, D. Stoyanov, D. McIlwraith, B. P. Lo, and G.-Z. Yang, "Ambient and wearable sensor fusion for activity recognition in healthcare monitoring systems," in *4th international workshop on wearable and implantable body sensor networks (BSN 2007)*. Springer, 2007, pp. 208–212.

[25] H. Guo, L. Chen, L. Peng, and G. Chen, "Wearable sensor based multimodal human activity recognition exploiting the diversity of classifier ensemble," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2016, pp. 1112–1123.

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

[27] H. Zhao and Z. Liu, "Human action recognition based on non-linear svm decision tree," *Journal of Computational Information Systems*, vol. 7, no. 7, pp. 2461–2468, 2011.

[28] Z. Zhao, Y. Chen, J. Liu, Z. Shen, and M. Liu, "Cross-people mobile-phone based activity recognition," in *Twenty-second international joint conference on artificial intelligence*, 2011.

[29] J. Wu, A. Akbari, R. Grimsley, and R. Jafari, "A decision level fusion and signal analysis technique for activity segmentation and recognition on smart phones," in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*. ACM, 2018, pp. 1571–1578.

[30] J. Wu and R. Jafari, "Orientation independent activity/gesture recognition using wearable motion sensors," *IEEE Internet of Things Journal*, 2018.

[31] A. Akbari and R. Jafari, "Transferring activity recognition models for new wearable sensors with deep generative domain adaptation," in *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*, ser. IPSN '19. New York, NY, USA: ACM, 2019, pp. 85–96. [Online]. Available: http://doi.acm.org/10.1145/3302506.3310391

[32] X. Li, Y. Zhang, J. Zhang, S. Chen, I. Marsic, R. A. Farneth, and R. S. Burd, "Concurrent activity recognition with multimodal cnn-lstm structure," *arXiv preprint arXiv:1702.01638*, 2017.

[33] A. Akbari, J. Wu, R. Grimsley, and R. Jafari, "Hierarchical signal segmentation and classification for accurate activity recognition," in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, ser. UbiComp '18. New

York, NY, USA: ACM, 2018, pp. 1596–1605. [Online]. Available: http://doi.acm.org/10.1145/3267305.3267528

[34] J. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[35] 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 *6th International Conference on Mobile Computing, Applications and Services*. IEEE, 2014, pp. 197–205.

[36] T. Zebin, P. J. Scully, and K. B. Ozanyan, "Human activity recognition with inertial sensors using a deep learning approach," in *2016 IEEE SENSORS*. IEEE, 2016, pp. 1–3.

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

[38] G. M. Weiss and J. Lockhart, "The impact of personalization on smartphone-based activity recognition," in *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[39] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE Communications Surveys Tutorials*, vol. 15, no. 3, pp. 1192–1209, Third 2013.

[40] P. Siirtola, H. Koskimäki, and J. Röning, "Personalizing human activity recognition models using incremental learning," *arXiv preprint arXiv:1905.12628*, 2019.

[41] S. Sabato and T. Hess, "Interactive algorithms: from pool to stream," in *29th Annual Conference on Learning Theory*, ser. Proceedings of Machine Learning Research, V. Feldman, A. Rakhlin, and O. Shamir, Eds., vol. 49. Columbia University, New York, New York, USA: PMLR, 23–26 Jun 2016, pp. 1419–1439. [Online]. Available: http://proceedings.mlr.press/v49/sabato16.html

[42] T. Diethe, N. Twomey, and P. Flach, "Active transfer learning for activity recognition," 2016, pp. 429–434, cited By 7. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84994153101&partnerID=40&md5=d8ab0ba12489b69d8ff3871b9b2cf164

[43] T. Sztyler and H. Stuckenschmidt, "Online personalization of cross-subjects based activity recognition models on wearable devices," in *2017 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, March 2017, pp. 180–189.

[44] A. Subramanya, S. Srinivas, and R. V. Babu, "Confidence estimation in deep neural networks via density modelling," *CoRR*, vol. abs/1707.07013, 2017. [Online]. Available: http://arxiv.org/abs/1707.07013

[45] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li, "Deep reconstruction-classification networks for unsupervised domain adaptation," in *European Conference on Computer Vision*. Springer, 2016, pp. 597–613.

[46] A. Akbari and R. Jafari, "A deep learning assisted method for measuring uncertainty in activity recognition with wearable sensors," in *2019 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*. IEEE, 2019, pp. 1–5.

[47] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[48] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of machine learning research*, vol. 11, no. Dec, pp. 3371–3408, 2010.

[49] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?" *Journal of Machine Learning Research*, vol. 11, no. Feb, pp. 625–660, 2010.

[50] 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.

[51] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *Wearable Computers (ISWC), 2012 16th International Symposium on*. IEEE, 2012, pp. 108–109.

[52] T. R. Bennett, H. C. Massey, J. Wu, S. A. Hasnain, and R. Jafari, "Motionsynthesis toolset (most): An open source tool and data set for human motion data synthesis and validation," *IEEE Sensors Journal*, vol. 16, no. 13, pp. 5365–5375, 2016.

[53] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5574–5584. [Online]. Available: http://papers.nips.cc/paper/7141-what-uncertainties-do-we-need-in-bayesian-deep-learning-for-computer-vision.pdf