Federated Multi-task Learning with Hierarchical Attention for Sensor Data Analytics

Yujing Chen,[1] Yue Ning,[2] Zheng Chai,[1] Huzefa Rangwala[1]
[1]Department of Computer Science, George Mason University, Virginia, USA
[2]Department of Computer Science, Stevens Institute of Technology, New Jersey, USA
[1]{ychen37, zchai2, rangwala}@gmu.edu, [2]{yue.ning}@stevens.edu

***Abstract*—The past decade has been marked by the rapid emergence and proliferation of a myriad of small devices, such as smartphones and wearables. There is a critical need for analysis of multivariate temporal data obtained from sensors on these devices. Given the heterogeneity of sensor data, individual devices may not have sufficient quality data to learn an effective model. Factors such as skewed/varied data distributions bring more difficulties to the sensor data analytics. In this paper, we propose to leverage multi-task learning with attention mechanism to perform inductive knowledge transfer among related devices and improve generalization performance. We design a novel federated multi-task hierarchical attention model (FATHOM) that jointly trains classification/regression models from multiple distributed devices. The attention mechanism in the proposed model seeks to extract feature representations from inputs and to learn a shared representation across multiple devices to identify key features at each time step. The underlying temporal and nonlinear relationships are modeled using a combination of attention mechanism and long short-term memory (LSTM) networks. The proposed method outperforms a wide range of competitive baselines in both classification and regression settings on three unbalanced real-world datasets. It also allows for the visual characterization of key features learned at the input task level and the global temporal level.**

***Index Terms*—Sensor analytics, Attention mechanism, Multi-task learning.**

I. Introduction

Ubiquitous sensors seek to improve the quality of everyday life through pervasively interconnected objects. A wide array of sensors in the form of wearable devices (e.g., clothing and wrist-worn devices), smartphones, and infrastructure components (e.g., cameras, WiFi) are the chief enablers. These solutions, commonly referred to as the Internet of Things (IoT), allow for fine-grained sensing and inference of users' context, physiological signals, and even mental health states. The sensing and detection capabilities coupled with advanced data analytics provide an appealing end-to-end solution for various domains, e.g., environment monitoring, healthcare, education, and workplace management.

In many applications, sensor data is captured from multiple devices; and personalized predictive models are expected for individual devices. However, individual devices may not have sufficient high quality data to learn an effective model. This may be caused by many factors such as capacity, power, communication bandwidth, and skewed data distribution. Therefore, we leverage a multitask learning (MTL) framework to learn individual models (one per device) jointly. The rationale behind the MTL paradigm is that when there is not enough data to learn a high-quality model, transferring and leveraging predictive information from other related tasks can improve the generalization performance. [1] We use the word device and task interchangeably in this paper.

We consider several main challenges when applying MTL to sensor data analytics. First, the temporal signal data collected from sensors are usually high dimensional. This data can be noisy with complex feature correlations. Second, the distribution of the data points in the training set can be highly skewed leading to bias in the learning algorithms. Third, powerful models such as deep recurrent neural networks often have high model complexities or degrees-of-freedom, and require large volume of training data for fitting. Additionally, it may not be feasible to upload this data to a data center due to communication costs, and hardware constraints (e.g., power, processor speed, memory). Thus it is necessary to keep data on devices.

To tackle the above mentioned challenges, we propose a **f**eder**a**ted multi-**t**ask **h**ierarchical attenti**o**n **m**odel (FATHOM) that seeks to learn an individual model per device (aka task). Attention mechanisms have shown promising results on learning powerful feature representations [2], [3]. For the input data series, we design a task-specific attention layer to capture the inner-feature correlations of each device. Meanwhile, the shared temporal correlations of all devices are captured by a global temporal attention. Our proposed model in a federated learning framework is similar to prior work by *Smith et al.* [12]; however, the scope of this paper is focused on improving the overall prediction performance and not the challenges (e.g., communication cost, stragglers) in other federated learning frameworks.

We evaluate the performance of FATHOM on both classification and regression tasks. The key contributions of this paper can be summarized as follows:

- We propose a novel federated multitask learning framework, FATHOM, to learn separate models for each sensor device and keep data locally.
- We design a hierarchical attention mechanism within the framework. At the local device level, task-specific attentions are developed to evaluate personal feature

[1]In this paper, we define a model learned for an individual device as one task.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

correlations. At the global level, a temporal attention layer on shared representations is created to evaluate cross-device temporal correlations.
- We perform extensive empirical experiments and visual analytics to evaluate our model. The proposed approach outperforms a wide range of baselines on three multi-modal sensor datasets from different domains with multi-binary labels or multi-continuous labels.

II. Related Work

A. Multi-task learning.

Multi-task learning (MTL) is designed for simultaneous training of multiple related tasks with the same prediction targets [4], [21]. Leveraging common information across related tasks has shown to be effective in improving the generalization performance of each task [4]–[6], [21]. MTL is particularly useful when there are a number of related tasks but some tasks have limited amounts of quality training data. Task relationships are modeled by sharing layers/units of neural networks [4], sharing subset of features [23]–[26], sharing a low dimensional common subspace [7], [27], assuming a clustering among tasks [8], [22], or sharing representation by structured regularization [29], [30]. In this work we do not make any assumptions on task relationships beforehand and learn the common representation with attention mechanisms directly from the data.

Specifically, for sensor analytics, a multi-task multilayer perceptron (MLP) model [9] was developed to recognize different human activities from mobile sensors, and this approach outperformed a standard logistic regression (LR) model [10]. However, the MLP and LR approaches do not leverage the underlying temporal dependencies and inter-feature correlations of sensor data.

B. Attention-based deep network.

Fundamentally, neural networks allocate importance to input features through the weights of the learned model. In the context of deep learning, attention-based encoder-decoder model allows a network to assign different levels of importance to various inputs by adjusting the weights [2]. This leads to a better feature representation. Attention approaches can be roughly divided into Global Attention and Local Attention [3]. The global attention is akin to soft attention [20]; where the alignment weights are learned and placed "softly" over all patches in the source data. Local attention only selects one patch of the data to access at a time.

Multi-level attentions are studied for improving document classification [13] and predicting spatio-temporal data [15]. Our proposed hierarchical attention networks aim to learn task-specific attentions and global temporal attentions to infer key representations across both feature and time dimensions.

C. Federated learning.

Federated learning seeks to train a predictive model while training data is distributed across multiple nodes [17]. Compared to conventional distributed machine learning [31], [33], this framework is more robust to highly unbalanced data, unstable network connections, and a large number of client nodes. Prior work has been proposed to deal with a federated optimization problem [18], [19], and aim at learning a single model across a network. Different from these works, Smith *et al.* [12] provides a solution to statistical challenges in federated multi-task learning. We adopt this federated multi-task learning setting. As mentioned in the introduction, our method is different from Smith *et al.* [12], as we focus on learning features across tasks using a hierarchical attention mechanism. The proposed hierarchical attention approach first learns inner-feature correlations at each local task. Then it learns a shared representation across distributed tasks with a global temporal attention mechanism. By sending this shared representation back to each local device, we get updated device-specific representations.

TABLE I
Notations

Notation	Meaning
K	# of tasks
D	# of features in each task
M	# of labels in each task
N	length of total time steps in each task
T	size of time window
$\mathbf{X}^{(k)} \in \mathbb{R}^{N \times D}$	D input features and N time steps for task k
$\underline{\mathbf{X}}^{(k)} \in \mathbb{R}^{T \times D}$	input features in task k with window size T
$\mathbf{x}_t^{(k)} \in \mathbb{R}^D$	a feature vector at time step t for task k
$\mathbf{Y}^{(k)} \in \mathbb{R}^{N \times M}$	target label matrix of task k
$\underline{\hat{\mathbf{Y}}}^{(k)} \in \mathbb{R}^{T \times M}$	predicted label matrix for $\mathbf{X}_T^{(k)}$
$\mathbf{a}_d^{(k)}$, $\mathbf{a}_t$	attention weight of task-specific level and global temporal level, respectively
$\boldsymbol{\phi}_d^{(k)}$	task-specific level context vector
$\boldsymbol{\psi}_t^{(k)}$	temporal context vector of task k

III. Methods

A. Problem Definition

The proposed model can be applied to both classification and regression problems. Assume there are K devices. For each device k, let $\mathbf{X}^{(k)} \in \mathbb{R}^{D \times N}$ represent the input sensor data series where D is the dimension of input series and N is the length of time steps. $\mathbf{Y}^{(k)} \in \mathbb{R}^{M \times N}$ is the corresponding label matrix for all time steps where M is the dimension of potential class labels. Within the MTL paradigm, the objective is to jointly learn models for each of the K devices. This is achieved by minimizing a loss function across all the K learning tasks given by $\mathbf{E}_r = \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}(\hat{\mathbf{Y}}^{(k)}, \mathbf{Y}^{(k)})$ where $\mathcal{L}(\hat{\mathbf{Y}}^{(k)}, \mathbf{Y}^{(k)})$ is the loss function for task k, and $\hat{\mathbf{Y}}^{(k)}$ is the predicted class labels.

For multi-label classification problems, we adopt the loss function from [35]; that seeks to prevent over-fitting and makes

the model more adaptable with an additional regularization component added to the cross-entropy loss. This is given as:

$$\mathcal{L}(\hat{\mathbf{Y}}, \mathbf{Y}) = -(1-\alpha) \underbrace{\sum_{m=1}^{M}\sum_{t=1}^{N} y_t^m \log(\hat{y}_t^m)}_{a} - \frac{\alpha}{M} \underbrace{\sum_{m=1}^{M}\sum_{t=1}^{N} \hat{y}_t^m}_{b}, \tag{1}$$

where $\hat{y}_t^m, y_t^m$ are the predicted and true label for the m^{th} label at time step t, respectively. The part a of Equation 1 is the cross-entropy loss, and the second part b is the added uniform distribution, which is a measure of how dissimilar the predicted distribution is to uniform. $\alpha \in [0, 1]$ is an adjustable parameter and can be changed to control the amount of uniform distribution added. We performed a grid search on a validation set and found a value of $\alpha = 0.3$ showed the best performance across the benchmarks studied in this paper.

For regression problems, we seek to minimize the mean absolute error between the predicted and the true distributions for each task with N time steps and each time step has M outputs as:

$$\mathcal{L}(\hat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{M \times N} \sum_{m=1}^{M}\sum_{t=1}^{N} |\hat{y}_t^m - y_t^m|. \tag{2}$$

B. Preliminaries

We use a Long Short-Term Memory (LSTM) layer to process the generated feature representations from the task-specific attention and global temporal attention. With one LSTM layer after each attention component is not computational efficient as RNN/CNN free attention networks such as in [28]. However, model with these two LSTM layers will capture the temporal relations of time series data better and has better predictive performance. Consider a feature representation of task k as $\mathbf{E}^{(k)} = \{\mathbf{e}_1^{(k)}, \mathbf{e}_2^{(k)}, ..., \mathbf{e}_t^{(k)}, ..., \mathbf{e}_T^{(k)}\} \in \mathbb{R}^{T \times D}$, the LSTM updates the feature representation with function:

$$\mathbf{h}_t^{(k)} = f(\mathbf{h}_{t-1}^{(k)}, \mathbf{e}_t^{(k)}) \tag{3}$$

where $\mathbf{h}_t^{(k)}$ and $\mathbf{h}_{t-1}^{(k)}$ are the hidden states of time step t and $t-1$ for task k, respectively. $\mathbf{e}_t$ is the input at time step t. We use $f()$ to represent the update function of LSTM. We adopt the LSTM structure from [1].

For simplicity, we use notation LSTM() to represent the whole function update of an LSTM cell in the following sections.

C. Model Structure

The proposed model includes two main components: 1) task-specific local attention to learn feature representations of each device and 2) global temporal attention to extract a shared temporal representation across all devices. An illustration of our model structure can be found in Figure 1. $\mathbf{X}^{(1)}, ..., \mathbf{X}^{(K)}$ are the input data series of K devices. For each device k, we use a sliding window of size T to process the sequential data. A task-specific attention layer is applied on each input layer to capture local feature dependencies of each device.

Algorithm 1 FATHOM Framework

Input: $\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, ..., \mathbf{X}^{(K)}$ stored on K nodes separately

1: **for** iterations $i = 0, 1, ...$ **do**
2: **for** $k \in \{1, 2, ..., K\}$ in parallel over K nodes **do**
3: Calculate attention vector $\boldsymbol{\phi}_d^{(k)}$ and iterate each feature to get a matrix $\mathbf{\Phi}^{(k)}$
4: Pass $\mathbf{\Phi}^{(k)}$ to LSTM to get hidden representation $\mathbf{h}_T^k$
5: **end for**
6: Central node calculates $\mathbf{s}_T \leftarrow \mathbf{h}_T^1 \oplus \mathbf{h}_T^2 ... \oplus \mathbf{h}_T^K$
7: Compute global attention $\mathbf{a}_{1 \rightarrow T}$ based on $\mathbf{s}_T$
8: **for** $k \in \{1, 2, ..., K\}$ in parallel over K nodes **do**
9: Update attention vector $\boldsymbol{\psi}_t^{(k)} \leftarrow \mathbf{x}_t^{(k)} * \mathbf{a}_t^{(k)}$
10: Pass $\boldsymbol{\psi}_t^{(k)}$ to LSTM
11: Estimate output based on Eq. 12 and Eq. 13
12: **end for**
13: Calculate loss and backpropagation
14: **end for**
15: **return** $\hat{\mathbf{Y}}^{(1)}, \hat{\mathbf{Y}}^{(2)}, ..., \hat{\mathbf{Y}}^{(K)}$ K label matrices

After task-specific attention, the obtained attention vector is fed into the first LSTM layer to further learn a local feature representation. The computational process of this part is shown in lines 1-4 of Algorithm 1. Then we pass the concatenated local feature representations to the global temporal attention to learn a shared temporal representation (found in Lines 5-6 of Algorithm 1). The shared temporal representation is sent back to each device. For each device, by performing the element-wise multiplication of raw input features and the shared attention weights, we obtain the updated temporal attention vector for each device (lines 7-9 of Algorithm 1). A second LSTM layer is applied to each device after the global temporal attention layer. Finally, a classifier is used to predict the labels for each device. In the following sections we describe each part in detail.

1) Task-specific Attention.: Feature extraction at the input task level increases the probability of capturing task-related features. These features should be given higher weights in comparison to other features while computing a task-specific representation. However, it is unknown which part of the feature space has predictive information, so we choose a soft attention mechanism [3] to capture feature representations by attending to all input features from each task.

Consider $\underline{\mathbf{X}}^{(k)} = \{\mathbf{x}_{*,1}^{(k)}, ..., \mathbf{x}_{*,d}^{(k)}, ..., \mathbf{x}_{*,D}^{(k)}\} \in \mathbb{R}^{T \times D}$ as an input example for a given task k with D input series, where T is the time window size, and $\mathbf{x}_{*,d}^{(k)} \in \mathbb{R}^T$ is the d-th column in $\underline{\mathbf{X}}^{(k)}$. First we transform $\underline{\mathbf{X}}^{(k)}$ using a fully connected layer to obtain a hidden representation $\mathbf{Z}^{(k)}$ as:

$$\mathbf{Z}^{(k)} = \underline{\mathbf{X}}^{(k)} \mathbf{W}^{(k)} \in \mathbb{R}^{T \times D}, \tag{4}$$

where $\mathbf{W}^{(k)}$ is the trainable weight matrix. Let $\mathbf{z}_d^{(k)}$ be the d-th column of $\mathbf{Z}^{(k)}$, we compute the attention weight of the d-th input series by applying a softmax function:

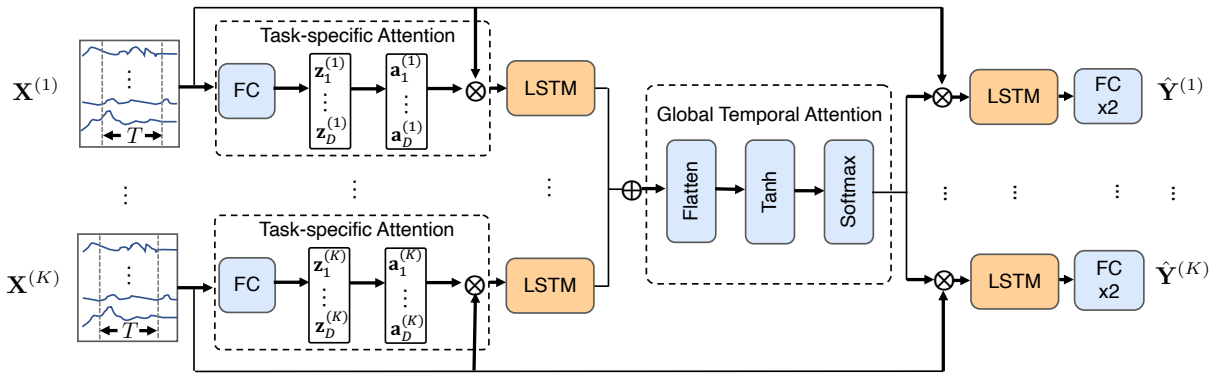


Fig. 1. Architecture illustration of the proposed federated Multi-task Hierarchical Attention Model (FATHOM). $\mathbf{X}^{(1)}, ..., \mathbf{X}^{(K)}$ indicate input data series of K tasks, T is the sliding window size, and $\hat{\mathbf{Y}}^{(1)}, ..., \hat{\mathbf{Y}}^{(K)}$ are the predicted labels. 'FCx2' indicates two fully connected layers, $\otimes$ indicates element-wise multiplication, $\oplus$ indicates tensors' concatenation.

$$\mathbf{a}_d^{(k)} = \frac{\exp(\mathbf{z}_d^{(k)})}{\sum_{g=1}^{D} \exp(\mathbf{z}_g^{(k)})} \in \mathbb{R}^T. \tag{5}$$

We measure the importance of features by computing the context vector with the element-wise multiplication of $\mathbf{x}_{*,d}^{(k)}$ and the attention weights $\mathbf{a}_d^{(k)}$. Then a $\tanh$ activation is applied to obtain the final attention vector at dimension d:

$$\boldsymbol{\phi}_d^{(k)} = \tanh(\mathbf{x}_{*,d}^{(k)} \otimes \mathbf{a}_d^{(k)}) \in \mathbb{R}^T. \tag{6}$$

Using eq. 6, we iterate each feature to get a matrix $\boldsymbol{\Phi}^{(k)} \in \mathbb{R}^{T \times D}$ across feature dimension D. Then we use a LSTM to update each row in $\boldsymbol{\Phi}^{(k)}$ get the hidden representation $\mathbf{h}_t^{(k)}$ at time step t:

$$\mathbf{h}_t^{(k)} = \text{LSTM}(\mathbf{h}_{t-1}^{(k)}, \boldsymbol{\Phi}_t^{(k)}). \tag{7}$$

2) Global Temporal Attention.: The attention distribution captured at task-specific levels focuses on a specific part of features for individual devices, which can only reflect the label information at a current time window. However, for time series data, there is usually a strong temporal correlation. Hence, it is essential to capture the temporal dependencies across time. The global attention component aims at learning a shared representation across all tasks at each time step. For task k, with iteration of $\mathbf{h}_t^{(k)}$ through the time window T we obtain the hidden representation $\mathbf{h}_T^{(k)}$ after the first LSTM layer. First we concatenate the hidden representation across K devices to get the shared hidden representation $\mathbf{S}_T$:

$$\mathbf{S}_T = \mathbf{h}_T^{(1)} \oplus \mathbf{h}_T^{(2)} \oplus ...\mathbf{h}_T^{(k)} ... \oplus \mathbf{h}_T^{(K)}. \tag{8}$$

We pass the shared hidden representation to a flatten layer to get the hidden representation $\mathbf{f}$. Different from task-specific attention, we apply a $\tanh$ nonlinearlity before softmax. By transforming $\mathbf{f}$ using a fully connected layer with T units, $\tanh$ is applied to obtain the time-step level context vector $\mathbf{u}_T$ by $\mathbf{u}_T = \tanh(\mathbf{f}) \in \mathbb{R}^T$.

Then we compute the global temporal attention score using a softmax function for each time step $t = 1, .., T$:

$$a_t = \frac{\exp(u_{t \in T})}{\sum_{j \in T} \exp(u_j)}. \tag{9}$$

The attention score a_t is obtained by normalizing the context score u_t at each time step t. We then iterate each feature d for a_t, and obtain the attention vector $\mathbf{a}_t \in \mathbb{R}^D$.

We measure the importance of each time-step by computing the attention vector with an element-wise multiplication of $\mathbf{x}_t^{(k)}$, which is the t-th row of $\underline{\mathbf{X}}^{(k)}$, and the attention vector $\mathbf{a}_t$:

$$\boldsymbol{\psi}_t^{(k)} = \mathbf{x}_t^{(k)} \otimes \mathbf{a}_t \in \mathbb{R}^D. \tag{10}$$

Here we obtain the extracted hidden representation for each task k at the time step t. Then we feed $\boldsymbol{\psi}_t^{(k)}$ of task k to a LSTM layer and get the hidden representation at time step t:

$$\mathbf{h'}_t^{(k)} = \text{LSTM}(\mathbf{h'}_{t-1}^{(k)}, \boldsymbol{\psi}_t^{(k)}). \tag{11}$$

Finally, we iterate each time step and get the hidden state $\mathbf{h'}_T^{(k)}$ at the last time step T. This hidden state is fed into two fully connected layers to get the predicted labels as below:

$$\tilde{\mathbf{h}}_T^{(k)} = \mathbf{W}_T^{(k)} \mathbf{h'}_T^{(k)} + \tilde{\mathbf{b}}_T^{(k)}, \tag{12}$$

$$\underline{\hat{\mathbf{Y}}}^{(k)} = \mathbf{V}_T^{(k)} \tilde{\mathbf{h}}_T^{(k)} + \mathbf{b}_T^{(k)}, \tag{13}$$

where $\tilde{\mathbf{h}}_T^{(k)}$ is the hidden state after the first fully connected layer, $\underline{\hat{\mathbf{Y}}}^{(k)}$ is the predicted labels of $\underline{\mathbf{X}}^{(k)}$, $\mathbf{W}_T^{(k)}, \mathbf{V}_T^{(k)} \in \mathbb{R}^{D \times T}, \tilde{\mathbf{b}}_T^{(k)}, \mathbf{b}_T^{(k)} \in \mathbb{R}^T$ are the learned parameters.

In FATHOM, we adopt a hierarchical attention structure for sensor data analytics. The task-specific attention is used to extract features with high predictive information, and the global temporal attention can explore dependencies and temporal relations among the shared feature representations across all input tasks. With this model structure, FATHOM is able to learn better feature representations and achieve overall high prediction performances on sensor data.

TABLE II
SUMMARY OF THE DATASETS

	ExtraSensory	Air Quality	FitRec
# of tasks	40	9	30
# of time step in each task	3,000	8,218	30,000
Sample rate(s)	20/60	3,600	10
Labels	c	r	r
Dimension	276	15	12

IV. EXPERIMENTS

A. Datasets

We use three real-world sensor datasets from different domains.

- **ExtraSensory Dataset** [2]**:** Mobile phone sensor data (e.g., location services, audio, accelerator) collected from 60 users [14]. We select 40 users with at least 3000 samples and use the provided 225-length feature vectors. We model the device of each user as a task and predict their activities (e.g., walking, talking, running).
- **Air Quality Dataset** [3]**:** Weather data collected from multiple weather devices distributed in 9 areas of Beijing, with features such as thermometer, barometer. We model weather device in each area as a separate task and use the observed weather data to predict the measure of six air pollutants (e.g., PM2.5, PM10) from May 1st, 2018 to May 31st, 2018.
- **FitRec Dataset** [4]**:** User sport records generated on mobile devices and uploaded to Endomondo, including sequential features such as heart rate, speed, and GPS as well as the sport type (e.g., biking, hiking). Following the feature processing in [34], we use data of randomly selected 30 users for heart rate and speed prediction. We model each user's device as a task.

We summarize the detailed statistics of each dataset used in our experiments in Table II. c and r show the dataset that is used for classification problems and regression problems in this paper, respectively.

B. Comparative Methods

We compare the proposed FATHOM approach to several single-task learning and multi-task learning approaches. We select the following state-of-the-art approaches as baselines:

- **Logistic Regression (LR)** [10]. This is a single task learning approach. Each task performs training and prediction with its own data. There is no parameter sharing among any tasks.
- **Multilayer Perceptron (MLP(16,16))** [9]. A multi-task MLP model with two hidden layers, where the hidden dimension used for both layers is 16.
- **Convolutional Recurrent Neural Network (CRNN)** [11]. A multi-task learning model that uses Convolutional Neural Networks to extract short-term basic patterns and find local dependencies among features.
- **M-Att** [32]. A hybrid multi-task attention model with a combination of Convolutional Neural Networks and Recurrent Neural Networks.
- **S-LSTM** A single task learning model with a single LSTM layer as a comparison with the LR approach.
- **M-LSTM**. A multi-task model with the first hidden layer of MLP(16,16) replaced by one LSTM layer which can better capture long-term dependencies in learning.

In particular, we perform ablation studies to assess the strengths of the different attention layers introduced in our proposed FATHOM.

- **FATHOM-ta**. FATHOM without the task-specific attention. This model is used to measure the importance of global temporal attention.
- **FATHOM-ga**. FATHOM without the global temporal attention. This model is used to measure the learning ability of task-specific attention.

C. Evaluation Metrics and Setup

We compare the performance of these models with the proposed FATHOM model. For the classification dataset, the labels are highly imbalanced and hence we report the F1 score, precision, recall, and Balanced Accuracy (BA) [9]. For the regression datasets, we evaluate performance using symmetric mean absolute percentage error (SMAPE) and mean absolute error (MAE).

1) Training Setup.: For each experiment, we split our dataset into training data, validation data and test data, in the proportions of 60%, 20%, and 20%, respectively. We use Keras and Tensorflow to implement all the approaches. Our models are trained with Adam optimizer for classification tasks, and RMSprop for the regression tasks.

2) Hyper-parameters.: Based on the performance on the validation set we choose the best group of parameters, retrain a model with the identified parameters, and report results on the test set. We set the hidden units of both LSTM layers to 64, and both the regular dropout and the recurrent dropout to 0.2. We also impose $l2$ constraints with value 0.001 on the weights within LSTM nodes to further reduce over-fitting. We use a batch size of 60 for ExtraSensory classification tasks, and 100 for the other two regression datasets. The initial learning rates are set to 0.001.

V. RESULTS

A. Comparative Performance

We demonstrate the prediction performance of the proposed approach in comparison to different baseline approaches. Table III shows the performance for the different models on ExtraSensory, Air Quality and FitRec Datasets.

For the classification performance on ExtraSensory Data, we observe that FATHOM significantly outperforms the other

[2]http://extrasensory.ucsd.edu/

[3]https://biendata.com/competition/kdd_2018/data/

[4]https://sites.google.com/eng.ucsd.edu/fitrec-project/home

TABLE III

COMPARATIVE PERFORMANCE ON THREE DATASETS. 'HRATE' REPRESENTS 'HEART RATE'. 'IMPROV.(A)' AND 'IMPROV.(B)' SHOW THE PERCENTAGE IMPROVEMENT OF FATHOM OVER THE WORST AND BEST BASELINE RESULTS, RESPECTIVELY. * INDICATES SIGNIFICANTLY BETTER THAN THE SECOND BEST SCORE ($p < 0.05$)

Methods	ExtraSensory				Air Quality		FitRec			
	Pr↑	Re↑	F1↑	BA↑	mae↓	smape↓	mae↓ (HRate)	mae↓ (Speed)	smape↓ (HRate)	smape↓ (Speed)
LR	0.57	0.60	0.52	0.72	30.13	1.23	14.56	13.86	0.64	0.61
MLP(16,16)	0.55	0.61	0.58	0.76	10.83	0.65	7.98	8.88	0.37	0.41
CRNN	0.43	0.68	0.54	0.78	10.43	0.64	9.05	9.38	0.35	0.35
M-Att	0.69	0.72	0.70	0.83	8.30	0.46	4.62	4.71	0.30	0.32
S-LSTM	0.79	0.71	0.74	0.84	9.11	0.52	13.51	12.96	0.58	0.58
M-LSTM	0.45	0.62	0.52	0.77	10.39	0.66	7.0	6.93	0.35	0.29
FATHOM-ga	0.50	0.61	0.54	0.77	15.67	0.73	7.59	7.11	0.34	0.31
FATHOM-ta	0.80	0.69	0.74	0.84	8.15	0.51	4.18	4.10	**0.19**	0.20
FATHOM	**0.89***	**0.77***	**0.82***	**0.88***	**7.30***	**0.44***	**3.93***	**4.0***	0.20	**0.17***
Improv.(a)%	106.97	28.33	57.69	22.22	75.77	64.22	70.91	69.13	68.75	70.69
Improv.(b)%	12.65	6.94	10.81	4.76	12.04	15.38	14.93	15.07	33.33	41.37

models in terms of precision, recall, F1, and Balanced Accuracy metrics. Given the highly imbalanced distribution for ExtraSensory data, FATHOM outperforms all baseline models in the range of 10.81%-57.69% on F1 score and 4.76%-22.22% on Balanced Accuracy. We observe that CRNN performs slightly better than multi-task LSTM (M-LSTM) and LR approaches, but not better than other models. The feature correlations captured by CNN on each task is loose and cannot represent temporal dependency effectively. M-Att model with one attention layer capturing the temporal correlations among input sequences achieves close performance as FATHOM-ta.

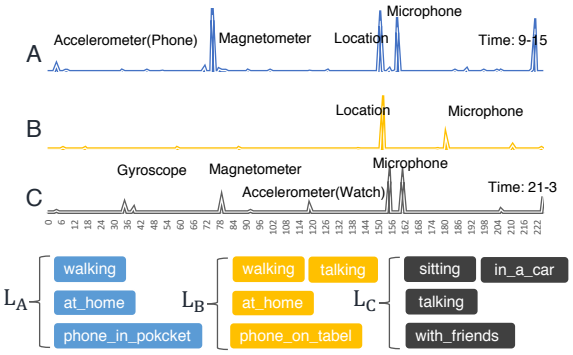
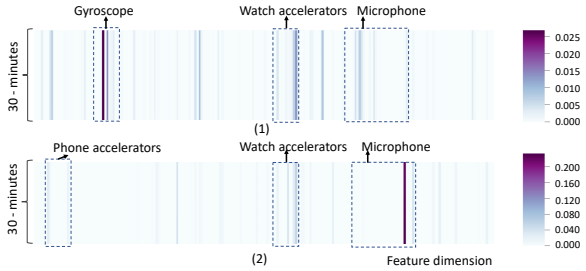
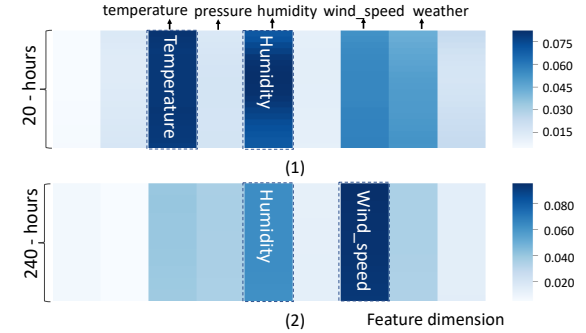


Fig. 2. Attention weights in feature dimensions captured with task-specific attention in ExtraSensory dataset. (a): A, B, C represent three different users. L_A, L_B, L_C represent their according labels.

From the regression results, we observe that FATHOM outperforms all baseline models in SMAPE by a range 15.38%-64.22%, and in MAE by a range 12.04%-75.77% for the Air Quality dataset. For FitRec, FATHOM achieves the best performance for three out of four results. All results on three datasets show that our model achieves the best performance, which indicates the effectiveness of the proposed method in both classification and regression problems.



(a) ExtraSensory



(b) Air Quality

Fig. 3. Sensor-specific attention matrix from the ExtraSensory Dataset (a) and Air Quality dataset (b). (a): predictions of 30-minutes time length of two users. (b) predictions of one task in 20-hours and 240-hours time length. Each column is the attention vector over the input series.

1) Ablation Study.: To further evaluate the two attention mechanisms in FATHOM we perform an ablation study by removing either the task-specific attention layer or the global temporal attention layer and denote them by FATHOM-ta and FATHOM-ga, respectively. From Table III, we observe that FATHOM-ta with the global temporal attention still achieves very good performance in comparison to other baseline approaches. However, FATHOM-ga does not perform well, because the feature representations learned by task-specific layer

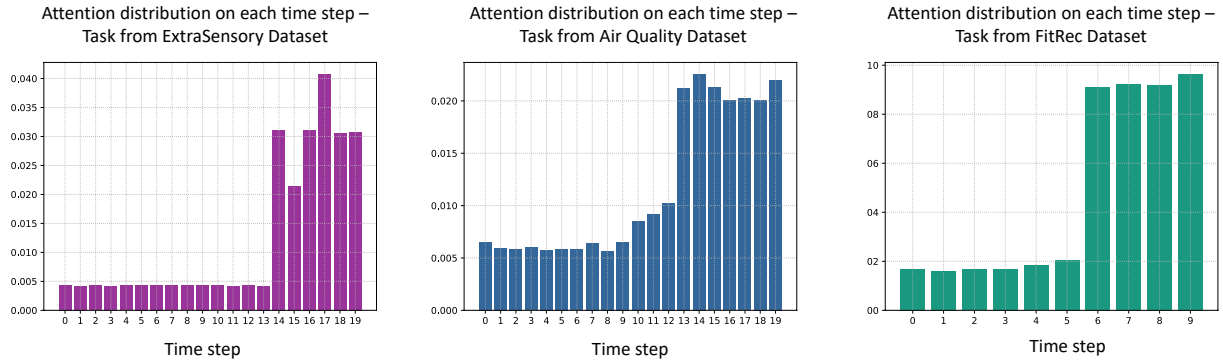


Fig. 4. Time-dimension attention distribution of three different tasks

fail to leverage the temporal correlations in the input data. The FATHOM model with both attention mechanism outperforms FATHOM-ga and FATHOM-ta by 51.85% and 10.81% in terms of F1-score on Extrasensory , respectively. We also observe that FATHOM-ta is close in performance for the other values with FATHOM. This is because training data of each user in FitRec is biased to one sport type (e.g., biking, hiking), therefore a shared global temporal feature representation is important for future prediction.

2) Single-task versus Multi-task learning.: We assess the benefits of multi-task learning in comparison to single-task learning models (See Table III). The single-task LR model has the worst performance on all three datasets. The single task S-LSTM model that captures temporal dependencies outperforms the LR model. However, the performance of jointly trained multiple task learning approaches with LSTM (M-LSTM) is worse compared to the S-LSTM model for ExtraSensory and Air Quality datasets. As mentioned before, for the FitRec dataset, the single task models get the worst performance because the training data for each task is imbalanced, so the model performance will be harmed if the user switched to another activity in test data. This in turn shows the benefit of multi-task learning. In general, multi-task learning approaches improve classification/regression performance but fail when the relationships among multiple tasks are not modeled well. FATHOM, on the other hand, outperforms the single task learning models because it is able to identify specific key features across different tasks and across different time steps.

B. Task-specific Attention Evaluation

To better understand the attention mechanisms and their abilities, we present several qualitative studies. Figure 2 shows the burstiness of features (spikes) captured by the task-specific attention from three different tasks at different time steps of ExtraSensory dataset. We observe a high correlation between the feature spikes and the corresponding labels. For example, for person A who is walking at home with a phone in pocket; the captured related features are phone accelerometers, magnetometer, location, microphone, and time. For person B, who is walking but talking to a phone on the table, there is no change of the magnetometer, no acceleration of the phone, and also a lower volume of voice. For person C who is driving a car and talking with friends, the task-specific attention captures the correlated features to corresponding group activities.

We take one task from the Air Quality dataset and two tasks from the ExtraSensory dataset to further visualize the variation of attention vectors across feature dimensions in Figure 3. From the attention weight matrix shown in Figure 3(a) we observe that the user of matrix (1) first lies down, then walks and talks with friends on a phone in pocket. The captured highly related features are phone gyroscope, watch accelerator, and microphone. The user of matrix (2) first grooms and gets dressed, then stays in a lab. We find that watch and phone accelerators have a strong correlation with body movement. The microphone is directly correlated with voice in the surroundings. Figure 3(b) represents that in a prediction window of 20 hours length, temperature and humidity have the highest weights among all the input features. Recall that the attention weights semantically indicate the relative importance of each local feature. We find that in the case of short term prediction, temperature and humidity affect the air pollutants most. This is because in the winter users in Beijing consume fuel for heating and humidity is usually low. While in a prediction window length of 240 hours, wind speed becomes the most important with the highest weight. In a dry season with low temperatures, the only effective way to disperse air pollutants is wind. All the above case studies show that our method is effective at capturing task-specific features that vary across time scales leading to interpretable results.

C. Global Attention Evaluation

Figure 4 shows the attention distribution from the central time attention layer of one task from the three datasets, respectively. The LSTM layer will allocate the weights to the last one or two time steps, thus the information of former steps will be lost. By applying the central time attention, the weight distribution does not just focus on the last step, but also spreads to former steps. Our observation is that temporal information is not lost and gets re-introduced leading to a stronger predictive performance.

VI. Conclusion

In this paper we present FATHOM, a novel federated multi-task model utilizing a hierarchical attention mechanism to generate more efficient device-specific feature representations. Task-specific attention is designed to capture feature correlations within each local task and global temporal attention is used to generalize inter-task feature representations across all tasks. We evaluate our proposed model on both classification and regression tasks. The results show that our approach improves prediction performance significantly compared to a wide range of state-of-the-art methods. We also show multiple qualitative case studies to interpret the attention mechanisms in our model. However, the proposed method works in a synchronous fashion. In the future, we plan to investigate other federated multi-task settings such as learning multiple tasks asynchronously, dealing with the problem of data heterogeneity, handling stragglers, and data privacy preserving mechanisms.

Acknowledgment

The authors would like to thank CloudLab for providing all computing resources needed in this work. All of the experiments in this paper are conducted with an Intel E5-2683 v3 56-core CPU at 2.00GHz [16].

References

[1] Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. Neural computation, 9(8), pp.1735-1780.

[2] Qin Y, Song D, Chen H, Cheng W, Jiang G, Cottrell G. A dual-stage attention-based recurrent neural network for time series prediction. arXiv preprint arXiv:1704.02971. 2017 Apr 7.

[3] Luong, M.T., Pham, H. and Manning, C.D., 2015. Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025.

[4] Caruana, R., 1997. Multitask learning. Machine learning, 28(1), pp.41-75.

[5] Evgeniou, T., Micchelli, C.A. and Pontil, M., 2005. Learning multiple tasks with kernel methods. Journal of Machine Learning Research, 6(Apr), pp.615-637.

[6] Bonilla, E.V., Agakov, F.V. and Williams, C.K., 2007, March. Kernel multi-task learning using task-specific features. In Artificial Intelligence and Statistics (pp. 43-50).

[7] Argyriou, A., Evgeniou, T. and Pontil, M., 2007. Multi-task feature learning. In Advances in neural information processing systems (pp. 41-48).

[8] Zhou, J., Chen, J. and Ye, J., 2011. Clustered multi-task learning via alternating structure optimization. In Advances in neural information processing systems (pp. 702-710).

[9] Vaizman Y, Weibel N, Lanckriet G. Context recognition in-the-wild: Unified model for multi-modal sensors and multi-label classification. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies. 2018 Jan 8;1(4):168.

[10] Vaizman Y, Ellis K, Lanckriet G. Recognizing detailed human context in the wild from smartphones and smartwatches. IEEE Pervasive Computing. 2017 Oct 31;16(4):62-74.

[11] Cirstea, R.G., Micu, D.V., Muresan, G.M., Guo, C. and Yang, B., 2018. Correlated time series forecasting using deep neural networks: A summary of results. arXiv preprint arXiv:1808.09794.

[12] Smith V, Chiang CK, Sanjabi M, Talwalkar AS. Federated multi-task learning. InAdvances in Neural Information Processing Systems 2017 (pp. 4424-4434).

[13] Yang, Z., Yang, D., Dyer, C., He, X., Smola, A. and Hovy, E., 2016, June. Hierarchical attention networks for document classification. In Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies (pp. 1480-1489).

[14] Vaizman, Yonatan, et al. Extrasensory app: Data collection in-the-wild with rich user interface to self-report behavior. Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. ACM, 2018.

[15] Liang Y, Ke S, Zhang J, Yi X, Zheng Y. GeoMAN: Multi-level Attention Networks for Geo-sensory Time Series Prediction. In *IJCAI* 2018 Jul 19 (pp. 3428-3434).

[16] Dmitry Duplyakin, Robert Ricci, Aleksander Maricq, Gary Wong, Jonathon Duerig, Eric Eide, Leigh Stoller, Mike Hibler, David Johnson, Kirk Webb, Aditya Akella, Kuangching Wang, Glenn Ricart, Larry Landweber, Chip Elliott, Michael Zink, Emmanuel Cecchet, Snigdhaswin Kar and Prabodh Mishra. *The Design and Operation of CloudLab*. Proceedings of the USENIX Annual Technical Conference (ATC), pp. 1-14. 2019.

[17] McMahan, H. Brendan et al. Communication-Efficient Learning of Deep Networks from Decentralized Data. AISTATS (2016).

[18] Konečný, J., McMahan, H.B., Yu, F.X., Richtárik, P., Suresh, A.T. and Bacon, D., 2016. Federated learning: Strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492.

[19] Konečný, J., McMahan, B. and Ramage, D., 2015. Federated optimization: Distributed optimization beyond the datacenter. arXiv preprint arXiv:1511.03575.

[20] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R. and Bengio, Y., 2015, June. Show, attend and tell: Neural image caption generation with visual attention. In International conference on machine learning (pp. 2048-2057).

[21] Pentina, A. and Lampert, C.H., 2017, August. Multi-task learning with labeled and unlabeled tasks. In Proceedings of the 34th International Conference on Machine Learning-Volume 70 (pp. 2807-2816). JMLR. org.

[22] Jacob, L., Vert, J.P. and Bach, F.R., 2009. Clustered multi-task learning: A convex formulation. In Advances in neural information processing systems (pp. 745-752).

[23] Chen, J., Zhou, J. and Ye, J., 2011, August. Integrating low-rank and group-sparse structures for robust multi-task learning. In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 42-50). ACM.

[24] Kim, S. and P Xing, E., 2010. Tree-guided group lasso for multi-task regression with structured sparsity.

[25] Zhou J, Yuan L, Liu J, Ye J. A multi-task learning formulation for predicting disease progression. InProceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining 2011 Aug 21 (pp. 814-822). ACM.

[26] Yang, X., Kim, S. and Xing, E.P., 2009. Heterogeneous multitask learning with joint sparsity constraints. In Advances in neural information processing systems (pp. 2151-2159).

[27] Zhang, Y. and Yeung, D.Y., 2012. A convex formulation for learning task relationships in multi-task learning. arXiv preprint arXiv:1203.3536.

[28] Shen, Tao, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. "Disan: Directional self-attention network for rnn/cnn-free language understanding." In Thirty-Second AAAI Conference on Artificial Intelligence. 2018.

[29] Evgeniou, T. and Pontil, M., 2004, August. Regularized multi–task learning. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 109-117). ACM.

[30] Zhou, J., Chen, J. and Ye, J., 2011. Malsar: Multi-task learning via structural regularization. Arizona State University, 21.

[31] Ma, C., Konečný, J., Jaggi, M., Smith, V., Jordan, M.I., Richtárik, P. and Takáč, M., 2017. Distributed optimization with arbitrary local solvers. optimization Methods and Software, 32(4), pp.813-848.

[32] Chen, Y. and Rangwala, H., 2019. Attention-based Multi-task Learning for Sensor Analytics. In proceeding of 2019 IEEE International Conference on Big Data. IEEE.

[33] Zhang, Yuchen, and Xiao Lin. Disco: Distributed optimization for self-concordant empirical loss. International conference on machine learning. 2015.

[34] Ni, J., Muhlstein, L. and McAuley, J., 2019, May. Modeling Heart Rate and Activity Data for Personalized Fitness Recommendation. In The World Wide Web Conference (pp. 1343-1353). ACM.

[35] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z., 2016. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).