2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA) | 978-1-7281-8470-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICMLA51294.2020.00122

CRUFT: Context Recognition under Uncertainty using Fusion and Temporal Learning

Wen Ge Emmanuel Agu
Worcester Polytechnic Institute Worcester, MA
{wge, emmanuel}@wpi.edu

***Abstract*—Human context recognition (HCR), which involves determining a user's current situation (or context), has long been an important task in context-aware systems. With the widespread ownership of smartphones, HCR methods that utilize signals from its built-in sensors have recently received increased attention. We propose Context Recognition under label Uncertainty using Fusion and Temporal Learning (CRUFT), a novel method to recognize a diverse set of smartphone user contexts, including long-term human activities, short-term human activities, and phone placement (pocket or bag in which the smartphone is carried). Context recognition is formulated as a multi-label classification task. CRUFT uses both handcrafted features and auto-learned deep learning features extracted from raw time-series data in two separate arms. The handcrafted arm includes a Multi-Layer Perceptron (MLP), while the raw data arm utilizes a Convolutional Neural Network (CNN) along with a Bi-Directional Long Short Term Memory (Bi-LSTM) model that exploits temporal correlations in the input stream. As smartphone sensor readings, assigned timestamps, and labels can be wrong sometimes, CRUFT integrates an uncertainty module. CRUFT outperforms the state-of-the-art baselines achieving 94.25% in overall Balanced Accuracy (BA), which improves the best performing baseline by 2.7%. Our detailed analyses demonstrate the non-trivial contributions of each component in CRUFT.**

***Index Terms*—Human Context Recognition, Mobile Ubiquitous and mobile computing, Deep Learning**

I. Introduction

Context-Aware (CA) systems, which can adapt their behaviors based on the user's current context [1] have important applications in healthcare. In CA systems, recognition of the user's context (or situation) is an important task. We focus on HCR on smartphones as they are sensor-rich (e.g. with accelerometers, gyroscopes, and location sensors) and ubiquitously owned. Human context is often represented as a tuple <activity, phone placement> that includes their current activity and pocket or bag in which the phone is currently placed. Phone placement is important as the smartphone's sensor signal may vary for the same context or activity with different phone placements, making HCR challenging [1]. Using this tuple representation, smartphone HCR becomes a challenging multi-label classification task.

In this paper, we propose an improved HCR model, which exploits three observations we made about smartphone context data patterns: *1) Label correlation:* Distinct labels in the context tuple may be correlated or co-occur. For example, "Phone on Table, Facing Down" and "Sitting" tend to co-occur and have a high Pearson Correlation (PC) = 0.93. Conversely, some contexts have are unlikely to co-occur. For example, "Sitting" and "Walking" have low PC -0.15. Prior HCR systems learned context labels independently but did not exploit correlations between context labels. *2) Temporal context sequences:* Data instances tend to occur in bursts causing several consecutive context segments to be similar. *3) Label uncertainty:* Smartphone context data can be noisy due to perturbations of the phone in the real world, wrong timestamps, labels, and extremely imbalanced datasets, which present additional challenges for HCR systems. We exploit these three patterns in a single model to achieve significantly better performance than prior HCR systems.

We propose the Context Recognition under Uncertainty using Fusion and Temporal Learning (CRUFT), a deep learning method that recognizes a diverse set of smartphone user contexts, including long-term human activities, short-term human activities, and phone placement. CRUFT uses both handcrafted features and auto-learned deep learning features extracted from raw time-series data in two separate branches. Temporal patterns are learned by a Bi-Directional Long Short Term Memory (Bi-LSTM) model on the raw data arm. The two branches are fused by concatenating the hidden vectors generated by 1) MLP 2) CNN+(Attention+pooling)+Bi-LSTM. An additional fully connected layer is added for final prediction. CRUFT integrates an uncertainty module to deal with noisy smartphone data due to wrong assigned sensor timestamps and labels. Various pooling strategies are utilized to compress raw data better, and an attention mechanism is used to focus on the most predictive parts of the data. CRUFT outperformed the other state-of-the-art methods, achieving 94.25% Balanced Accuracy (BA) on a real-world context dataset.

Our work is related to Human Activity Recognition (HAR) research, including those that utilized data from multiple body-worn sensors [2].

However, our work focuses on HCR on smartphones. Prior smartphone HCR work uses MLP to classify handcrafted features extracted from in-the-wild data gathered from both a smartphone and smartwatch [3]. Neural networks have also been utilized for smartphone HCR [4], including combining MLPs and CNNs [5], fusing multimodal sensor data and classification using a bi-LSTM [2], using multiple sensor streams with shared weights [6] or converting sensor data to a 2D image that is then analyzed using CNNs [7].

Our research differs from the prior work above in the following ways:

1) *Learning temporal data correlations:* Prior work treats

978-1-7281-8470-8/20/$31.00 ©2020 IEEE
DOI 10.1109/ICMLA51294.2020.00122

each data instance independently. In contrast, CRUFT uses a Bi-LSTM to learn temporal correlations in sequential data and across labels using joint learning.

2) *Combining handcrafted and auto-learned features:* CRUFT incorporates and combines the advantages of handcrafted and auto-learned deep learning features to improve model performance.
3) *Combining local and temporal deep learning models:* MLP and CNN models extract local information while a Bi-LSTM learns temporal patterns. An attention mechanism focuses on the most predictive parts of the data. Pooling strategies were used for data compression.
4) *Uncertainty estimation module for mitigating real-world noise:* and reducing the impact of user movement and wrong labels and timestamps on prediction performance.

The main contributions of this paper are three-fold:

1) We propose CRUFT, a novel deep learning framework that solves the multi-label HCR problem. CRUFT combines a CNN that captures label correlations with a novel Bi-LSTM model that captures the *inter instance* temporal correlations between sequential data instances. For contexts such as continuous human motion that generate long streams of similar sequences, exploiting temporal relationships achieves significantly better performance than treating data instances independently. Prior work [3] only learned temporal correlations within each data instance (*intra-instance*).
2) We explored multiple methods for uncertainty estimation, which improved HCR in many scenarios.
3) We rigorously evaluate CRUFT, demonstrating that it outperforms state-of-the-art baselines, achieving an overall Balanced Accuracy (BA) of 94.25% on a real smartphone context dataset. Our ablation study demonstrates that the contributions of each component of CRUFT to overall model performance are non-trivial.

The remainder of this paper is organized as follows. Section II describes the context data collection and pre-processing steps for our experiments. Section III introduces our proposed approach. Section IV describes our evaluation and results. Finally, Section V concludes our work.

II. Context Dataset

We evaluated CRUFT and compared it to other baseline models using real-world context data gathered in the DARPA-funded Warfighter Analytics for Smartphone Healthcare (WASH) project. The over-arching goal of the WASH project is to passively assess the health status of smartphone users by utilizing high-specificity smartphone biomarkers or health tests performed in specific contexts. The scripted WASH dataset was collected from 109 subjects who visited contexts and performed activities in a scripted fashion while carrying a smartphone running an app that passively gathered sensor data. During data collection, each candidate placed their phone in different pockets/bags (in a bag, in the pocket, in hand, on a table, and facing up or down). Human proctors labeled each smartphone sensor data instance with 25 different binary labels, which were then used to compose the context tuple (See Table I).

TABLE I
WASH Scripted Context Labels

Category	Label	Support(sec)	Ratio(%)
Phone Placement	In Pocket	52,049	11.76
	In Hand	69,798	15.77
	In Bag	54,805	12.39
	On Table Face Down	20,493	4.63
	On Table Face Up	7,987	1.81
Long Term Activity	Lying Down	1,863	0.42
	Sitting	23,501	5.31
	Walking	129,834	29.34
	Sleeping	4,477	1.01
	Talking On Phone	2,945	0.67
	Bathroom	6,535	1.48
	Standing	13,246	2.99
	Jogging	4,366	0.99
	Running	3,769	0.85
	Stairs - Going Down	5,364	1.21
	Stairs - Going Up	1,890	0.43
	Typing	8,049	1.82
	Jumping	3,140	0.71
	Trembling	1,274	0.29
Short Term Activity	Laying Down (action)	1,216	0.27
	Sitting Down (action)	719	0.16
	Sitting Up (action)	873	0.20
	Sneezing	642	0.15
	Coughing	664	0.15
	Standing Up (action)	954	0.22

TABLE II
Handcrafted Features for 3-axial Sensor Measurements

Features	Description
Mean	$Mean_m = \frac{1}{N}\sum_{i=1}^{N} m_i$
Standard Deviation	$Std_m = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(m_i - \bar{m})^2}$
Moment-k	$\sqrt[k]{\frac{1}{N}\sum_{i=1}^{N}(m_i - \bar{m})^k}$, $k = 3, 4$
Percentile-k	the score at k percentile of magnitude, k=25, 50, 75
Value Entropy-k	$-\sum_{i=1}^{k} p_i \times log(p_i)$, a histogram of the magnitude values to k bins, k=20
Time Entropy	$-\sum_{i=1}^{N} abs(m_i) \times log(abs(m_i))$
Log Energy Band-[a,b]	$log(\sum_i fft(m_i)^2)$, $fftfreq(m_i)$, [a,b] = [0, 0.5], [0.5,1], [1,3], [3,5], [5,+∞] Hz
Spectral Entropy	$-\sum_{i=1}^{N} fft(abs(m_i)) \times log(fft(abs(m_i)))$
Period	Duration between two peak autocorrelations of m
Normalized Autocorrelation	Normalized highest autocorrelation of m after main lobe
Mean	$Mean_x = \frac{1}{N}\sum_{i=1}^{N} x_i$
Standard Deviation	$Std_x = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2}$
Inter-axis Correlation Coefficients	$Ro_{xy} = C_{xy}/\sqrt{C_{xx} \times C_{yy}}$, $C_{xy} = Cov(x, y)$

Two important points need to be clarified: 1) Each instance can be tagged with multiple labels. For example, a subject may be "talking on the phone" with "the phone in hand". 2) As shown in table I, this is an extremely imbalanced dataset in which positive labels are rare and sparse.

A. Handcrafted Features

The handcrafted features utilized are listed in Tables II and Table III, where magnitude $m = \sqrt{x^2 + y^2 + z^2}$. Table II lists features generated from spatial sensor data including the accelerometer, gyroscope, raw and unbiased magnetometers. Other features are listed in Table III. These features were previously utilized by Vaizman *et al* [3]. They were extracted from 5 sensors: accelerometer (Acc), gyroscope (Gyro), location (Loc), raw and calibrated version of the magnetometer (raw_Mag and proc_Mag), and phone state (PS). After removing identical features, 145 features were left. Continuous feature values were normalized by subtracting their mean and

TABLE III
OTHER FEATURES

Features	Description
Number of Updates	number of changed location
Log of Latitude-range	$log(max(latitude) - min(latitude))$
Log of Longitude-range	$log(max(longitude) - min(longitude))$
Min/Max Altitude	$min(altitude)/max(altitude)$
Min/Max Speed	$min(speed)/max(speed)$
Best Horizontal Accuracy	$min(horizontal\_accuracy)$
Best Vertical Accuracy	$min(vertical\_accuracy)$
(Log) Diameter	(Log of) Longest geographic distance between two locations
Screen Brightness, Light, Pressure, Relative Humidity, Battery Level, Proximity, Temperature Ambient	Low Frequency Measurements Directly Taken from Sensors Built in to the Phone
Battery State	{unknown, unplugged, not charging, discharging, charging, full}
On The Phone	{True, False}
Ring Mode	{normal, silent no vibrate, silent with vibrate}
Wifi Status	{True, False}
Time(Hour) of The Day	{[0,6], [3,9], [6,12], [9,15], [12,18], [15,21], [18,24]}

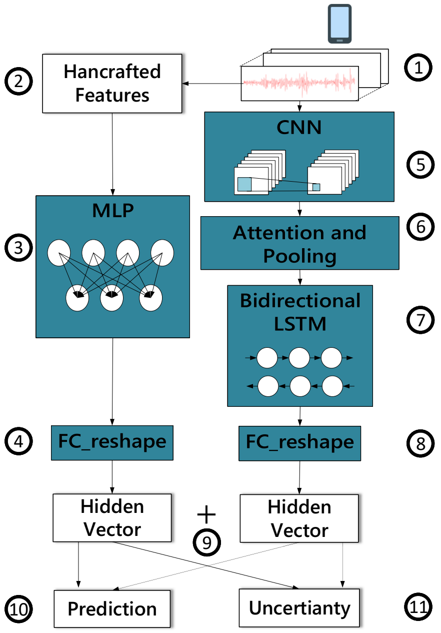

Fig. 1. Our CRUFT architecture.

dividing by their standard deviation. Categorical features were one-hot encoded.

B. Auto-learned Features and Splitting

Features were auto-learned from raw 3-axial accelerometer and gyroscope data $x_t = (acc_x^t, acc_y^t, acc_z^t, gyro_x^t, gyro_y^t, gyro_z^t)$, which were sampled to the desired length of 50 samples and then concatenated. To show that our model's inference generalizes well to new, previously unseen users, we used subject-level such that each subject's data was included in only one of the training, validation, or test sets. We created training (80%), test (10%), and validation (10%) splits.

III. OUR NOVEL CRUFT FRAMEWORK

Our proposed Context Recognition under Uncertainty using Fusion and Temporal Learning (CRUFT) framework is a Joint-Learning model (See Fig 1). Details of each layer are described in Table IV. Optimal parameter settings were determined using a grid search. The model has three main parts:

1) *Multi-Layer Perceptron (MLP)* that handles handcrafted features with a fully-connected reshaping layer to generate the hidden vector (shown as step 2-4).
2) *CNN with attention and pooling layer followed by a Bi-LSTM network* with a similar reshaping layer for automatically learning hidden vectors from sequential/time-series raw data (shown as step 5-8).
3) *Joint Fusion and Uncertainty Estimation* concatenates hidden vectors learned from the two previous steps and generates predictions and uncertainty (steps 9-11). For overlapping outputs, a simple voting strategy is used to make the final prediction and estimate uncertainty.

A. MLP with handcrafted features as input

The handcrafted features are fed into a one-hidden-layer MLP. A fully connected layer is then used to project the vector learned by MLP into the same level of magnitude as the hidden vector learned from the raw signal. This projection ensures that the hidden vectors from one arm of the design (parts 1 and 2) do not overwhelm or underwhelm the other part after concatenation. We apply dropout as an implicit Bayesian approximation [8], reducing both overfitting and model uncertainty [9].

B. CNN + Attention/Pooling + Bi-LSTM processing raw data

This part is composed of a CNN, followed by a bi-LSTM network. Several pooling methods were applied between CNN (local context) and Bi-LSTM (temporal context)

to condense the data and extract useful information. CRUFT takes $w = 10$ multiple consecutive, overlapping $o = 5$ instances as input.

In this setting, $X_t = [x_t, x_{t+1}, \ldots, x_{t+10}]$ and $X_{t+5} = [x_{t+5}, x_{t+6}, \ldots, x_{t+15}]$ would be two consecutive valid inputs to our model. For one input $X_t \in N^{10*6*50}$, each instance $x_t \in N^{1*6*50}, t \in (t, \ldots, t+w)$ is processed separately by the CNN to extract feature vectors $a_t \in N^{1*128*6}$. CNN's output is further condensed using three mapping/pooling strategies: self-attention, max pooling, and mean pooling. These three pooling results are then concatenated and fed to later networks. For each instance in A_t, the CNN outputs $b_t \in N^{384}$. CRUFT uses a bi-LSTM network with dropout to learn temporal patterns in the time-series context data. A linear layer is used to project the hidden vector generated by the Bi-LSTM into a lower hidden space, making it compatible with the output vector of part 1.

C. Joint Fusion and Uncertainty Estimation

Finally, we concatenate the representation vectors learned from both CRUFT arms using a hidden layer with dropout and LeakyReLU as the activation function. A linear projection maps it to $y_t = [y_0, y_1]$. The output can be written as:

$$output_t = [\mu(\hat{y_t}), \delta^2(\hat{y_t})]^T = [y_0, softplus(y_1)]^T \quad (1)$$

TABLE IV
DETAILED CRUFT FRAMEWORK ARCHITECTURE

Step Index	Layer Name	Previous Layer	Input Shape	Output Shape	Parameter Setting & Description
1	Input	-	-	-	Load raw data as input
2	Handcraft	1	-	10×145	Extracting handcrafted features
3	MLP	2	10×145	10×64	#hidden layer=1, dropout = 0.05
4	$FC\_reshape_1$	3	10×64	10×64	Reorganize weights and shape
5.1	$Conv_1$	1	$10 \times 6 \times 50 \times 1$	$10 \times 6 \times 25 \times 32$	kernel=(1,9), #kernel=32, stride=(1,2), padding=(0,4)
5.2	$MaxPool_1$	5.1	$10 \times 6 \times 25 \times 32$	$10 \times 6 \times 12 \times 32$	kernel=(1,2), stride=(1,2)
5.3	$Conv_2$	5.2	$10 \times 6 \times 12 \times 32$	$10 \times 6 \times 12 \times 64$	kernel=(1,3), #kernel=64, stride=(1,1), padding=(0,1)
5.4	$Conv_3$	5.3	$10 \times 6 \times 12 \times 64$	$10 \times 6 \times 12 \times 128$	kernel=(1,3), #kernel=128, stride=(1,1), padding=(0,1)
5.5	$MaxPool_2$	5.4	$10 \times 6 \times 12 \times 128$	$10 \times 6 \times 6 \times 128$	kernel=(1,2), stride=(1,2)
5.6	$Conv_4$	5.5	$10 \times 6 \times 6 \times 128$	$10 \times 1 \times 6 \times 128$	kernel=(6,1), #kernel=128, stride=(1,1), padding=(0,0)
6.1	Attention	5.6	$10 \times 1 \times 6 \times 128$	10×128	attention pooling of one dimension, squeeze dimension 1
6.2	Maxpool	5.6	$10 \times 1 \times 6 \times 128$	10×128	max pooling of one dimension, squeeze dimension 1
6.3	Meanpool	5.6	$10 \times 1 \times 6 \times 128$	10×128	mean pooling of one dimension, squeeze dimension 1
7	BiLSTM	6.1, 6.2, 6.3	$10 \times (128 + 128 + 128)$	10×128	#layer=2, dropout=0.05, hidden_dim=64, bidirectional
8	$FC\_reshape_2$	7	10×128	10×64	Reorganize weights and shape
9	Concatenate	4,8	(10×64) * 2	10×128	Concatenate output of two networks. dropout=0.25
10	Prediction	9	10×128	10×25	Generate predictions
11	Uncertainty	9	10×128	10×25	Generate uncertainties

where $softplus(y_1) = log(1 + exp(y_1))$ and $\mu(\hat{y_t})$ and $\delta^2(\hat{y_t})$ represents the mean and variance of the prediction respectively. The variance $\hat{y_t}$ measures the uncertainty of our prediction. In cases where multiple predictions are generated due to overlapping inputs, a simple "vote" is used to generate the final prediction. In the situation with chunk size $w = 10$ and overlap $o = 5$, the output of $X_t = [x_t, x_{t+1}, \ldots, x_{t+10}]$ and $X_{t+5} = [x_{t+5}, x_{t+6}, \ldots, x_{t+15}]$ are $Y_t = [\mu(\hat{y_t}), \mu(\hat{y_{t+1}}), \ldots, \mu(\hat{y_{t+10}})]$ and $Y_{t+5} = [\mu(\hat{y_{t+5}}), \mu(\hat{y_{t+6}}), \ldots, \mu(\hat{y_{t+15}})]$. Thus, the prediction for timestamp $t + 5$ is the average of $Y_{t+5}[0] = \mu(\hat{y_{t+5}})$ and $Y_t[5] = \mu(\hat{y_{t+5}})$. The Bi-LSTM network's output at timestamp t is learned from the hidden vector at timestamp t but is also influenced by vectors from other timestamps.

The uncertainty estimation method we utilized in CRUFT was previously proposed by RDeepSense [9] and APDeepSense [10] to estimate the model's uncertainty in a single pass. Other emerging uncertainty methods have been proposed including 1) Dropout-based methods [8]–[11] and 2) Ensemble methods [12]. *Dropout-based uncertainty estimation:* such as MCDrop [8] and ADF [11] interpreted dropout as a Bayesian approximation of the Gaussian process. These methods average the mean and variance of the neural networks model trained with different dropout values as a measure of the model's uncertainty. However, these methods are too computationally complex for resource-constrained mobile devices, require multiple passes and either overestimate or underestimate the model's uncertainty [9]. In the RDeepSense uncertainty estimation method, a scoring rule achieved a balance between overestimating and underestimating uncertainty.

D. Loss Function

We utilize a loss function adapted from Yao *et al* [9]. It is a linear combination of two parts: weighted mean square error$loss_{mse}$ and weighted negative log-likelihood $loss_{nll}$, controlled by hyper-parameter α. The higher the value of α, the more the model emphasizes making an accurate prediction. To mitigate the extreme imbalance in our context dataset, we introduced instance-pair weights $\omega_{n,l}$ into the loss function.

$$Loss_{mse} = 0.5 \times \sum_{n=1}^{N} \sum_{l=1}^{C} (\omega_{n,l}(y_{n,l} - \mu(\hat{y_{n,l}}))^2) \qquad (2)$$

$$Loss_{nll} = 0.5 \times \sum_{n=1}^{N} \sum_{l=1}^{C} (log\delta^2(\hat{y_{n,l}}) + \frac{(y_{n,l} - \mu(\hat{y_{n,l}}))^2}{\delta^2(\hat{y_{n,l}})}) \qquad (3)$$

$$Loss = (1 - \alpha) \times Loss_{nll} + \alpha \times Loss_{mse} + \lambda_w \|W\|_2^2 \qquad (4)$$

IV. EXPERIMENT

We evaluated our CRUFT and baseline models on scripted WASH data using different sliding *window size* and *step ratio* combinations. $step\ size = window\ size \times step\ ratio$.

A. Baselines

We compared CRUFT to state-of-the-art models, including both machine learning and deep learning methods. Some baselines are individual components of the CRUFT model.

- *ExtraSensory Multilayer Perceptron (MLP):* utilized by Vaizman*et al.* for context recognition on smartphones and smartwatches [3]. Its inputs are 145 handcrafted features extracted from the accelerometer, gyroscope, magnetometer, location, phone state, and low-frequency measurements. Labels were assigned weights corresponding to their inverse frequency ratio.
- *Random Forest (RF):* is an ensemble tree classifier, which has been utilized extensively in the HAR literature, including on wearable device sensors [13]. We used Random Forest (RF) to classify the same 145 handcrafted features as the ExtraSensory MLP.
- *CNN:* model automatically extracts features from raw accelerometer and gyroscope data. We use the same CNN architecture listed in Table IV, followed by a fully connected layer to make predictions.
- *Bi-LSTM:* uses the same raw data as the CNN and is followed by a fully connected layer to make predictions.
- *GaitAuth [14]:* was previously used to authenticate smartphone users from their gait. It uses a sequence of CNNs

TABLE V
MODEL PERFORMANCE ON SCRIPTED WASH, WINDOW SIZE=3, STEP RATIO=0.5

Category	**Model** **Label**	ExtraSensory	Bi-LSTM	CNN	GaitAuth	GaitIden	Random Forest	CRUFT
Phone Placement	Phone In Pocket	.905	.883	.892	.927	.912	.812	**.947**
	Phone In Hnad	.877	.835	.871	.871	.869	.841	**.889**
	Phone In Bag	.920	.905	.929	.930	.927	.901	**.967**
	Phone On Table-Face Down	.824	.807	.809	.810	.814	.555	**.881**
	Phone On Table-Face Up	.981	.977	.973	.973	.972	.859	**.989**
Average for Phone Placement		.901	.882	.895	.902	.899	.794	**.935**
Long Term Activity	Lying Down	.915	.801	.826	.863	.856	.500	**.937**
	Sitting	.814	.771	.789	.781	.784	.549	**.870**
	Walking	.950	.927	.933	.924	.938	.952	**.958**
	Sleeping	.981	.974	.972	.970	.967	.513	**.989**
	Talking On Phone	.937	.958	.977	.970	.973	.670	**.983**
	Bathroom	.851	.781	.773	.801	.800	.500	**.876**
	Standing	.814	.758	.785	.758	.748	.502	**.893**
	Jogging	.972	.974	.886	.977	.980	.577	**.982**
	Running	.980	.937	.957	.964	.970	.704	**.981**
	Stairs-Going Down	.932	.791	.885	.894	.865	.544	**.955**
	Stairs-Going Up	.942	.768	.842	.858	.839	.500	**.946**
	Typing	.982	.983	.987	.987	**.987**	.816	.965
	Jumping	.982	.949	.968	.967	.966	.866	**.988**
	Trembling	.960	.932	.968	**.980**	.954	.689	.978
Average for Long Term Activity		.929	.879	.896	.907	.902	.635	**.950**
Short Term Activity	Layding Down (action)	.906	.868	.911	.902	.895	.500	**.956**
	Sitting Down (action)	.859	.797	.848	.867	.864	.500	**.912**
	Sitting Up (action)	.931	.849	.898	.905	.897	.500	**.973**
	Sneezing	**.931**	.751	.755	.815	.715	.500	.841
	Coughing	.920	.766	.798	.883	.869	.500	**.940**
	Standing Up (action)	.888	.871	.880	.895	.865	.500	**.970**
Average for Short Term Activity		.906	.817	.848	.878	.851	.500	**.932**
Average for All Labels		.918	.865	.884	.899	.889	.634	**.943**

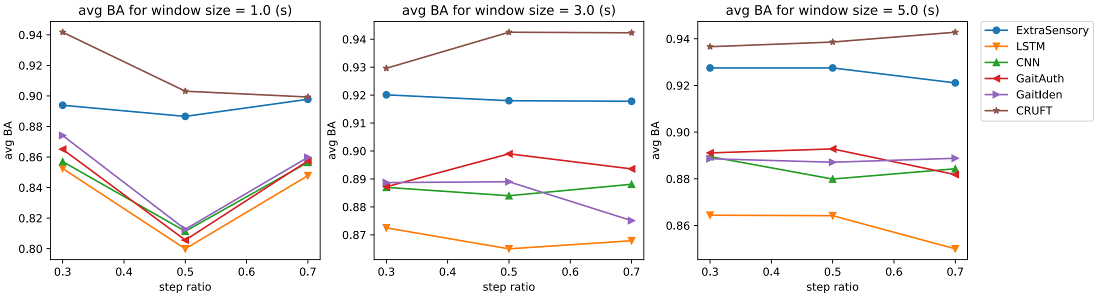

Fig. 2. Average Balanced Accuracy for different windows size for different step ratios

followed by LSTMs. The learned features are fed into a fully connected layer that predicts the user's context.

- *GaitIden [14]:* was previously used to identify smartphone users from their gait. It uses a CNN and an LSTM to extract features in parallel, concatenated, and fed to a fully connected layer for final prediction.

B. Experiment Settings

1) Hyper-parameters: Optimal values of all other hyper-parameters and settings were determined using grid search on the dataset with window size = 3, and step ratio = 0.5. The batch size was 128, learning rate to 0.001, and λ_w to $1e-5$. We used Adam as our default optimizer and l2_norm regularization.

The bi-LSTM hidden layer size was set to 64, dropout to 0.05, and the MLP hidden layer size to 64. The coefficient of the loss function α was set to 0.3. LeakyReLU was chosen as the activation function across all layers. Details of other parameters are listed in Table IV.

2) Evaluation Metric: As the scripted WASH dataset is extremely unbalanced, we evaluated our models using the Balanced Accuracy (BA) metric, which balances specificity and sensitivity. We computed BA macro for each context by averaging the BA of all its constituent labels. For each label,

$$BA = \frac{1}{2}(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}) \tag{5}$$

C. Results

1) Detailed performance on one specific data setting: For window size = 3 (secs), step ratio = 0.5, CRUFT outperformed all baselines in all general categories, on most labels, and overall with a 2.5% absolute improvement and 2.7% relative improvement against the best performing baseline (ExtraSensory MLP). Bold black indicates models with the best BA performance for each label, and the overall average BA is in the last row.

One exception was the label 'Sneezing' as it is difficult for CRUFT to learn useful information about a sound only from accelerometer and gyroscope sensor data. CRUFT is consistently over 6% better than other models at discriminating some challenging pairs of labels such as sitting and standing, which are frequently confused for each other. CRUFT

TABLE VI
ABLATION ON SCRIPTED WASH WITH WINDOW SIZE = 3 SECS

Step Size Model	0.9	1.5	2.1
CRUFT	**.930**	**.943**	**.942**
–MLP	.915	.917	.918
–Uncertainty Estimation	.928	.941	.936
–CNN+Attention/Pooling+bi-LSTM	.922	.925	.927

performs well on both long and short term human activity recognition. For Laying Down (action), Sitting Down (action), Sitting Up (action), and Standing Up (action), it outperforms baselines by over 5.0% on average. In contrast, Random Forest performed badly on short term activities.

2) General performance on all data settings: Figure 2 shows performance of all models for various window sizes. Random Forest was excluded as it performed significantly worse than other models. Even though we show only average BA due to space constraints, it is worth noting that CRUFT also outperforms baselines by a similar margin on individual labels. For window size = 1 and step size = 0.5, the average BA drops a lot mainly because it has the smallest number of instances among all datasets (150,882 instances vs 316,720, and 294,512 for step size of 0.3 (secs) and 0.7 (secs) respectively with a 1 second window size). Although all models' performance decreases when the step ratio = 0.5, our CRUFT and the ExtraSensory MLP models are more stable than baseline models that only utilize raw data. This demonstrates the power of using handcrafted features, which enables decent predictions even with insufficient data. Random Forest always has the lowest average BA, demonstrating that deep learning outperforms traditional machine learning models. RF also fails on Short Term Activities and other labels with small support size such as Lying Down, Bathroom, Stairs-Going Up. This shows that it suffers most when data is insufficient.

3) Ablation Study: We conducted an ablation study on all datasets to show that each part of our model contributes to the CRUFT model's success. From our results in Table VI, all three parts of our proposed CRUFT framework contributed positively, including MLP, CNN+Attention/Pooling+bi-LSTM, and Uncertainty estimation. Even after removing the raw time-series arm of CRUFT, which reduces CRUFT to an MLP with uncertainty, it still outperforms the pure MLP ExtraSensory model. This indicates that the uncertainty estimations in CRUFT contribute to its overall performance.

4) Impact of Uncertainty: We also evaluated how useful uncertainty estimations were by gradually removing instances from the test dataset, starting with predictions with the highest uncertainty down to those with lower uncertainty. We observed that removing highly uncertain predictions improved the model's overall BA. This implies that CRUFT could improve real-world HCR in highly noisy and uncertain in-the-wild scenarios by utilizing only context predictions that have high certainty.

V. CONCLUSION

In this paper, we proposed CRUFT, an HCR deep learning model comprising three parts: 1) MLP with handcrafted features, 2) CNN+Attention/Pooling+bi-LSTM handling raw sensor measurements, and 3) Uncertainty estimation. Our model learns both local and temporal correlations within the context data achieving a high Balanced Accuracy. Its uncertainty module makes it more robust to data and labeling noise on smartphones. CRUFT can discriminate the most confounding activities such as sitting and standing and significantly outperforms other state-of-the-art models by 5.6% and 7.9% BA, respectively. It performs well o both long term as well as short term human activities. The macro average BA over 25 labels outperformed the best baseline models by 2.7%. As the context dataset utilized in this paper was gathered in a scripted, controlled fashion, its sensor data and label uncertainty levels are lower than that of a dataset collected in-the-wild. In future work, we intend to demonstrate the power of CRUFT's uncertainty estimation module in mitigating real-world noise in an in-the-wild context dataset.

ACKNOWLEDGMENT

The research was sponsored by DARPA grant HR00111780032-WASH-FP-031. We thank the WPI WASH team for gathering the scripted dataset analyzed in this paper.

REFERENCES

[1] L. Sun, D. Zhang, B. Li, B. Guo, and S. Li, "Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations," in *ICUIC*. Springer, 2010, pp. 548–562.

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

[3] Y. Vaizman, N. Weibel, and G. Lanckriet, "Context recognition in-the-wild: Unified model for multi-modal sensors and multi-label classification," *ACM IMWUT*, vol. 1, no. 4, pp. 1–22, 2018.

[4] 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 *MobiCASE*. IEEE, 2014, pp. 197–205.

[5] V. Radu, C. Tong, S. Bhattacharya, N. D. Lane, C. Mascolo, M. K. Marina, and F. Kawsar, "Multimodal deep learning for activity and context recognition," *IMWUT*, vol. 1, no. 4, pp. 1–27, 2018.

[6] 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.

[7] N. Hatami, Y. Gavet, and J. Debayle, "Classification of time-series images using deep convolutional neural networks," in *ICMV 2017*, vol. 10696. Intl. Soc. Optics and Photonics, 2018, p. 106960Y.

[8] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approx.: Representing model uncertainty in deep learning," in *ICML*, 2016, pp. 1050–1059.

[9] S. Yao, Y. Zhao, H. Shao, A. Zhang, C. Zhang, S. Li, and T. Abdelzaher, "Rdeepsense: Reliable deep mobile computing models with uncertainty estimations," *ACM IMWUT*, vol. 1, no. 4, pp. 1–26, 2018.

[10] S. Yao, Y. Zhao, H. Shao, C. Zhang, A. Zhang, D. Liu, S. Liu, L. Su, and T. Abdelzaher, "Apdeepsense: Deep learning uncertainty estimation without the pain for iot apps," in *ICDCS*. IEEE, 2018, pp. 334–343.

[11] A. Loquercio, M. Segu, and D. Scaramuzza, "A general framework for uncertainty estimation in deep learning," *IEEE Robotics and Auto. Letters*, vol. 5, no. 2, pp. 3153–3160, 2020.

[12] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *NIPS*, 2017, pp. 6402–6413.

[13] L. Xu, W. Yang, Y. Cao, and Q. Li, "Human activity recognition based on random forests," in *Intl. Conf. Nat. Comp., Fuzzy Sys. and Knowledge Disc (ICNC-FSKD)*. IEEE, 2017, pp. 548–553.

[14] Q. Zou, Y. Wang, Q. Wang, Y. Zhao, and Q. Li, "Deep learning-based gait recognition using smartphones in the wild," *IEEE Trans. Inf. Forensics and Sec.*, vol. 15, pp. 3197–3212, 2020.