

Heterogeneous Hyper-Graph Neural Networks for Context-aware Human Activity Recognition

Wen Ge[§], Guanyi Mou[§], Emmanuel O. Agu, Kyumin Lee

Computer Science Department, Worcester Polytechnic Institute Worcester, MA, USA

{wge, gmou, emmanuel, kmlee}@wpi.edu

Abstract—Context-aware Human Activity Recognition (CHAR) is challenging due to the need to recognize the user’s current activity from signals that vary significantly with contextual factors such as phone placements and the varied styles with which different users perform the same activity. In this paper, we argue that context-aware activity visit patterns in realistic in-the-wild data can equivocally be considered as a general graph representation learning task. We posit that exploiting underlying graphical patterns in CHAR data can improve CHAR task performance and representation learning. Building on the intuition that certain activities are frequently performed with the phone placed in certain positions, we focus on the *context-aware human activity* problem of recognizing the $\langle \text{Activity, Phone Placement} \rangle$ tuple. We demonstrate that CHAR data has an underlying graph structure that can be viewed as a heterogeneous hypergraph that has multiple types of nodes and hyperedges (an edge connecting more than two nodes). Subsequently, learning $\langle \text{Activity, Phone Placement} \rangle$ representations becomes a graph node representation learning problem. After task transformation, we further propose a novel Heterogeneous HyperGraph Neural Network architecture for Context-aware Human Activity Recognition (HHGNN-CHAR), with three types of heterogeneous nodes (user, phone placement, and activity). Connections between all types of nodes are represented by hyperedges. Rigorous evaluation demonstrated that on an unscripted, in-the-wild CHAR dataset, our proposed framework significantly outperforms state-of-the-art (SOTA) baselines including CHAR models that do not exploit graphs, and GNN variants that do not incorporate heterogeneous nodes or hyperedges with overall improvements 14.04% on Matthews Correlation Coefficient (MCC) and 7.01% on Macro F1 scores.

Index Terms—Context-aware Human Activity Recognition, heterogeneous graph, hypergraph, graph neural networks

I. INTRODUCTION

Motivation: Context-aware Human Activity Recognition (CHAR) [1] tries to recognize the user’s activity while being aware of their current context. CHAR is an important task in Context-aware (CA) [2] Systems, targeting diverse real-life problems [3], [4]. To generate realistic, labeled CHAR datasets for supervised learning, apps installed on the smartphones (or smartwatches) of human volunteers¹, continuously gather sensor data (e.g., accelerometer) as they live their lives while performing various activities. Users are also prompted periodically to provide ground truth context-aware activity labels. While many definitions of user context exist in the literature [5], [6], in this paper, since certain activities are frequently

performed with the phone placed in certain positions, we focus on the specific *CHAR* problem of recognizing the $\langle \text{Activity, Phone Placement} \rangle$ tuple. The overarching goal of this paper is to create a pattern recognition model that can simultaneously infer a user’s current activity and phone placement from sensor data gathered from their smartphone. We focus on CHAR on smartphones, which are now nearly ubiquitously owned.

Prior Work: has explored improving Human Activity Recognition (HAR) by exploiting underlying graphical structures in activity data. Martin et al. [7] modeled human mobility patterns by using GPS coordinates to build multiple personalized graphs, and then predicted target HAR labels using two Graph Convolution Networks (GCNs) [8]. Mohamed et al. [9] proposed HAR-GCNN, which exploited correlations between chronologically adjacent sensor measurements to predict missing activity labels. Selected activities were used to build a fully connected graph where each node contained sensors values and corresponding activity labels. A GCN was employed to learn node embeddings, followed by a sequence of CNNs to predict activity labels. One limitation of HAR-GCNN is that it is incapable of making predictions for nodes with completely missing labels as is common in in-the-wild HAR datasets.

Novelty: Our work differs from prior work in two ways:

1) **The CHAR task was not explored.** Prior work [7], [9] recognized activities without factoring in phone placement or inter-user activity performance style, which results in an oversimplification of the problem. Such HAR models would underperform when deployed in the wild [10]–[12].

2) **Graphs were built using data from specific sensors and labels.** Some sensor data can be privacy sensitive (e.g., GPS), and less than 50% of users are willing to grant permission access permission [13]. In contrast, our approach derives the graph directly from CHAR data based solely on label co-occurrence information observed in the training set.

Our approach: We propose a generalizable GNN-based approach that improves CHAR performance by exploiting graphical patterns and internal relationships between different entities in CHAR data without requiring external information. We propose 1) A *CHAR graph*: that has three types of graph entity/nodes (activities, phone placements, and users) and edges as the aggregated mean feature values of instances that have similar corresponding labels connecting them. 2) A method of encoding CHAR data, transforming the associated recognition task into the equivalent heterogeneous hypergraph

[§]Equal contribution

¹We use “volunteers”, “participants” and “users” interchangeably

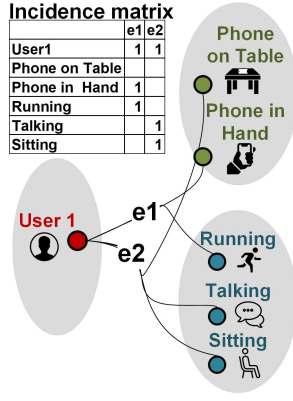


Fig. 1. Sample CHAR graph that has heterogeneous nodes and hyperedges

representation learning problem that is inspired by user-item interaction graphs commonly utilized in recommender systems [14]–[16], and 3) A novel deep heterogeneous hypergraph Graph Neural Network (GNN)-based learning model to solve the CHAR task.

The obtained CHAR graphical representation is heterogeneous [17] because the activity, phone placement, and user attributes that we define as nodes in our graph, are actually different types of nodes. As participants may perform multiple activities with specific phone placement concurrently, connecting all activities performed by a given user to valid phone placements results in hyperedges [18]. Such a one-to-many mapping condition is formulated as a multi-label problem in which phone placement and activity labels may co-occur. The example CHAR graph shown in Fig. 1 has two hyperedges: $e1$ and $e2$, where hyperedge $e2$ represents the User1 putting the Phone On Table while Sitting and Talking at the same time. After graphical encoding and transformation, a CHAR **Heterogeneous HyperGraph** is obtained. In the original CHAR task, given an input signal representation, a CHAR model predicts associated context-aware activity labels. In the transformed task, given input data as a new hyperedge and the heterogeneous hypergraph discovered from the CHAR training set, a graph representation learning model can predict the labels as the most likely connecting nodes. Transforming the CHAR task into a graph representation learning task, facilitates explicit identification and exploitation of graphical relationships among three types of nodes (activity, phone placement, and user) and with the corresponding sensor data.

Our contributions: in this paper are as follows:

- A method for graphical encoding of CHAR data is proposed, defining three types of CHAR nodes (activity, phone placement, and user) and the corresponding signal representations as edges based on underlying relationships.
- Encoding the CHAR task into a graph representation learning problem is proposed, establishing that the transformed CHAR graph has heterogeneity and hypergraph properties.
- A novel **Heterogeneous HyperGraph Neural Network** for Context-aware **Human Activity Recognition** (HHGNN-CHAR) is proposed, which solves the above-mentioned heterogeneous hypergraph representation learning problem

in a supervised fashion.

- We rigorously evaluate our proposed HHGNN-CHAR and demonstrate that it outperforms SOTA baselines by 14.04% on Matthews Correlation Coefficient and 7.01% on Macro F1 scores on a realistic, in-the-wild CHAR dataset. An extensive ablation study also revealed the non-trivial contributions made by each component of HHGNN-CHAR and the novel adaptations of incorporating graph heterogeneity and hypergraph properties.

To the best of our knowledge, our work is the first to reformulate the CHAR task as a generalized graph learning problem, and we propose a specific GNN framework that significantly outperforms other non-GNN SOTA approaches.

II. PROPOSED FRAMEWORK

A. Notations and Specifications

For CHAR, **Model Input** and **Graph Information** are inputs to the **Model**, which generates predictions as **Model Output** and approximations of the actual **Model Target**. At a high level, model optimization during training, model parameters are adjusted to reduce the gap between **Model Output** and the **Model Target**. In subsequent sections, unless explicitly noted, the notation conventions followed are:

- **Model Input:** The batched input data to models are denoted as X . Each instance within the batch is x . X_{train} and X_{test} stand for the training and testing sets, respectively. The inputs are usually features extracted from smartphone sensor signals, which often undergo pre-processing steps before being consumed by the model.
- **Graph Information:** Graph nodes and edges are denoted as V and E , the incidence matrix (for each hyperedge, whether each node is connected) as G_H (Eq. 1). Fig. 1 shows an example incident matrix. The hyperedge weight matrix is denoted as G_W (frequency of each hyperedge), the hyperedge initial representation is G_{attr} , node degree matrix is $D_V = \sum_{e \in G_E} G_W(e) G_H(v, e)$ and hyperedge degree matrix is $D_E = \sum_{v \in V} G_H(v, e)$. The input V for each layer in the network is V_{in} while the output is V_{out} .

$$G_H(v, e) = \begin{cases} 0, & \text{if } v \notin e, \\ 1, & \text{if } v \in e \end{cases} \quad (1)$$

- **Model Target:** Batched target labels are denoted as Y and each label as lowercase y . Targets are the ground truth CHAR activity and phone placement labels.
- **Model Output:** $\hat{Y}$ and $\hat{y}$ are model-generated predictions.
- **The Model:** The general model is denoted as M and the model's parameters as θ_M .
- **Other Specifications:** A graph is denoted as G , users as U , phone placements as PP , and activities as ACT .

B. Graph Formation / Task Transformation

The Original Task: Given “input data” as smartphone sensor features X , performed by the simulation users U , an CHAR model M ought to generate “predictions” around the

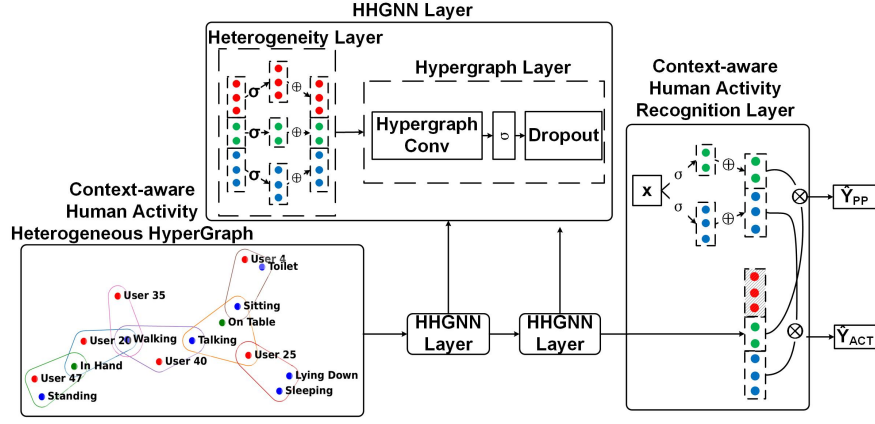


Fig. 2. An Overview of our HHGNN-CHAR Framework.

$< ACT, PP >$ pairs, represented by $\hat{Y}_{ACT}, \hat{Y}_{PP}$, where $\hat{Y}_{ACT} \in R^{|ACT|}$ and $\hat{Y}_{PP} \in R^{|PP|}$.

$$\hat{Y} = < \hat{Y}_{ACT}, \hat{Y}_{PP} > = \theta_M(X) \quad (2)$$

Traditional CHAR approaches directly model θ_M using machine learning / deep learning techniques such as Multi-Layer Perceptrons (MLP) or LightGBM to enable the model automatically learn the hidden correlations between X and Y . However, these approaches lack the explicit modeling of 1) Internal relationships within Y , especially Y_{PP} and Y_{ACT} ; 2) User-specific information that accounts for inter-user variability in visiting various context, which ultimately impacts its performance: correlation between U and Y .

The Transformed CHAR Task: Considering the lack of explicit modeling correlations among X, Y, U , we transform the CHAR task into a graphical representation learning problem. Given (X_{train}, Y_{train}) , we formulate an undirected heterogeneous hypergraph G_{train} with the following relationships:

$$\begin{aligned} G_{train} &= < V, E > \\ V &= \{v_i | i \in (U, PP, ACT)\} \\ E &= \{e_i | e_i \in V \times V^*\} \end{aligned} \quad (3)$$

where $*$ is noted for possible hyperconnections (connecting more than two nodes). Thus, a graph learning model M_G consumes the graph G as input and generates the learned node representations V_{rep} as output. The instance encoding model M_{enc} then transform the new feature vector x into the same feature dimensions as the node representation, denoted as x_{enc} . Intuitively, one may treat x as a hyperedge with unknown connections to nodes. A third decision-making model M_{dec} consumes both graph node representations and the new hyperedge information, and predicts its node connections.

$$\begin{aligned} V_{rep} &= M_G(G_{train}) \\ x_{enc} &= M_{enc}(x) \\ \hat{Y} &= < \hat{Y}_{ACT}, \hat{Y}_{PP} > = M_{dec}(V_{rep}(PP, ACT), x_{enc}) \end{aligned} \quad (4)$$

In the following sections, we will explain how each separate component in the deep learning model is designed.

C. Graph Neural Network Design: HHGNN-CHAR

Our HHGNN-CHAR network (Fig. 2), is composed of three types of fundamental sub-layers: Heterogeneity Layer, Hypergraph Layer, and Context-aware Human Activity Recognition Layer. The Heterogeneity Layer takes in node representations from either the constructed/learned Context-aware Human Activity Heterogeneous HyperGraph and encodes Heterogeneity into representations by projecting different types of nodes into their own corresponding hidden spaces. The Hypergraph Layer consumes the outputs from the Heterogeneity Layer and adds Hypergraph Convolutions plus activations and dropouts to address the hyperedges' characteristics. A combination of the Heterogeneity Layer and the Hypergraph Layer can be viewed as a HHGNN Layer block, which is stacked in order to learn the center node representations from more hops of its neighborhood. Lastly, the CHAR layer receives fine-tuned node representations from blocks of the HHGNN Layers and the task requires instance input, comparing them and deriving the possible nodes corresponding to the input instance input.

Heterogeneity Layer: is introduced to explicitly address the different types of nodes within the graph (i.e., Us, PPs, and ACTs). A given V , the input vector to the layer, is broken into node groups, which are handled by separate linear parameters and activations, and eventually, concatenate results as V_{out}

$$\begin{aligned} V_{in} &= [V_U, V_{PP}, V_{ACT}] \\ V_{out} &= [\sigma l_U(V_U) \oplus \sigma l_{PP}(V_{PP}) \oplus \sigma l_{ACT}(V_{ACT})] \end{aligned} \quad (5)$$

where $\oplus$ represents concatenation, l_* represents the linear layer, and σ represents a non-linear activation function (in practice, LeakyReLU is adopted).

Hypergraph Layer: is introduced to address the property that each hyperedge can connect more than two nodes. Each Hypergraph Layer is a sequence of a HyperConv [19] sub-layer, an activation sub-layer σ , and a dropout sub-layer β .

$$\begin{aligned} V_{hyperConv} &= D_V^{-1} G_H G_W D_E^{-1} G_H^T V_{in} \Theta \\ V_{out} &= \beta \sigma V_{hyperConv} \end{aligned} \quad (6)$$

where Θ is the learnable weight matrix in the convolutions.

CHAR Layer: Given a graph's learned node representations $V_{rep} \in R^{(|U|+|PP|+|ACT|) \times d_v}$ and new coming input data

$X \in R^{n \times d_x}$ as candidate hyperedges, the CHAR layer tries to predict the connecting nodes for each hyperedge $x_i, i = 1, 2, \dots, n$ within X . First, X is projected into the same dimension as V and the result is activated using non-linear transformations. Then matrix multiplication is performed between the input data and the learned node representation embeddings. The final outputs are the node predictions for each hyperedge.

$$X_{PP} = \sigma l_{PP}(X), X_{PP} \in R^{|PP| \times d_v} \quad (7)$$

$$X_{ACT} = \sigma l_{ACT}(X), X_{ACT} \in R^{|ACT| \times d_v}$$

$$\hat{Y}_{PP} = (X_{PP} \otimes V_{PP}^T)$$

$$\hat{Y}_{ACT} = (X_{ACT} \otimes V_{ACT}^T) \quad (8)$$

where $\otimes$ represents the matrix multiplication operation.

Putting it all together: Essentially, a GCN [8] can only capture the non-heterogeneous and non-hypergraph graphs leading to an inevitable loss of information [20]. Despite the existence of prior work that explicitly addresses hyperedges [19], the issue of heterogeneity within CHAR data has not previously been considered in a GNN. Thus, our proposed approach innovatively combines the Heterogeneity Layer and the Hypergraph Layer to achieve superior performance. Finally, the CHAR layer is used to predict $\hat{Y}$. In a supervised fashion, the model learns node representations and also predicts node connections giving hyperedges simultaneously.

III. EXPERIMENTS

A. Experiment Dataset

We evaluated HHGNN-CHAR on the publicly available Extrasensory [21] dataset, which contains 6,355,350 instances gathered from 60 participants. Each instance contains 170 features extracted from smartphone sensor data including accelerometer, gyroscope, location, phone state, audio and gravity sensors. 17 *Extrasensory* labels were considered (4 phone placement and 13 activity labels).

1) *Dataset Pre-processing:* The original dataset contained signals with varied durations that we sampled using a window length = 3 seconds and step size = 1.5 seconds, the values we experimentally found to be optimal. *Rules to fix labeling issues:* As a phone can only be carried in one position at a time, phone placement labels are mutually exclusive. Only phone placement labels of 0, 1, or *missing* are valid. In a pre-processing step, intuitive rules were applied to resolve ambiguous and duplicate labels, and remove conflicting activity and phone placement labels such as those that cannot co-occur.

B. Feature Extraction

Handcrafted features commonly utilized in CHAR [22], [23] were extracted. Features were then normalized using $z = (x - \mu)/s$ where μ and s are the mean and standard deviation of features in the training set, before being applied to the validation and testing sets. x and z are the original and transformed features, respectively.

TABLE I
MAJOR RESULTS ON HHGNN-CHAR AND BASELINES.

Models	Phone Placement		Activity		Overall	
	MCC	MacF1	MCC	MacF1	MCC	MacF1
ExtraMLP	0.784	0.813	0.636	0.608	0.673	0.659
CRUFT	0.758	0.868	0.616	0.783	0.651	0.804
GCN	0.824	0.907	0.675	0.813	0.712	0.837
LightGBM	0.835	0.913	0.710	0.837	0.741	0.856
HHGNN-CHAR	0.953	0.976	0.808	0.896	0.845	0.916

TABLE II
ABLATION STUDY ON HHGNN-CHAR.

Models	Phone Placement		Activity		Overall	
	MCC	MacF1	MCC	MacF1	MCC	MacF1
Hetero GCN	0.916	0.957	0.754	0.863	0.795	0.887
Hyper GCN	0.705	0.836	0.694	0.826	0.697	0.828
1-layer-HHGNN	0.942	0.970	0.804	0.894	0.839	0.913
HHGNN-CHAR	0.953	0.976	0.808	0.896	0.845	0.916

C. Baseline CHAR Models

We evaluated baseline models including ExtraMLP [22], CRUFT [24], GCN [8] and LightGBM [25] as well as our HHGNN-CHAR and its variants:

- **HHGNN-CHAR:** proposed approach using optimal hyperparameter values.
- **Hetero GCN:** HHGNN without hyperedges, to evaluate the contribution of hyperedges to HHGNN-CHAR performance.
- **Hyper GCN:** HHGNN without the heterogeneity layer to evaluate the contribution of the heterogeneity layer.
- **1-layer-HHGNN:** demonstrates that using two HHGNN layers in HHGNN-CHAR is optimal.

D. Evaluation Metrics

Considering that the CHAR dataset was extremely imbalanced, the two main evaluation metrics selected were Matthews Correlation Coefficient (MCC) and Macro F1 Score.

E. Experimental Setting

The dataset was randomly split into 60% for training, 20% for validation, and 20% for hold-out testing. HHGNN-CHAR was trained on the training set, and optimal hyperparameters were selected using grid search based on the best-performing model on the validation set. Evaluation results were reported on the hold-out testing set. The weighted binary cross entropy loss function was adopted as our target loss function (Eq. 9), where the instance-wise pair weights $\omega_{n,c}$ is inversely proportional to the label frequency and set to 0 if the label is missing. The weight ensures that positive and negative instances contribute equally to the loss. For graph representations, V and G_{attr} were initialized to the aggregated mean values of features and were fine-tuned during model training, while other matrices were set to initialization values. All other parameters were randomly initialized with a fixed random seed to enhance reproducibility. The performance of the proposed HHGNN-CHAR was evaluated and reported on three levels: 1) Results for each label, 2) Average results for the activity and phone placement categories, and 3) Average of all activities and phone placement labels.

$$Loss = \frac{1}{N} \sum_{n=1}^N \sum_{c=1}^C [-\omega_{n,c}(y_{n,c} \log \hat{y}_{n,c} + (1 - y_{n,c}) \log(1 - \hat{y}_{n,c}))] \quad (9)$$

F. Experiment Results

Results on *Extrasensory* is reported in Table I and Table II.

Overall Performance: In Table I, results for the best performing baseline model under each column is shown in *italic*. If the result of HHGNN-CHAR is significantly better than the best baseline result (gap larger than 0.01), it is marked in **bold**. HHGNN-CHAR outperformed the best baselines with 14.04% improvement on MCC and 7.01% improvement on MacF1 over the *Extrasensory* dataset.

Performance in each category: There were also consistent improvements for HHGNN-CHAR for both Phone Placement and Activity categories in Table I. HHGNN-CHAR had a 14.13% MCC improvement and 6.90% MacF1 improvement for predicting Phone Placement, and a 13.80% MCC improvement and 7.05% MacF1 improvement for predicting Activity, against the best performing baseline LightGBM.

Ablation Study: In Table II, when comparing HHGNN-CHAR to its variants, the heterogeneous design and hyperedges contributed non-trivially. When the hypergraph is converted into a traditional graph (i.e., Hetero GCN) by breaking hyperedges into multiple pair-wise edges, the performance of HHGNN-CHAR degraded due to information loss [20]. The inferior performance of Hyper GCN compared to HHGNN-CHAR also confirms that node embeddings learned in a unified space do not adequately capture heterogeneity characteristics of context-aware human activity data [26]. Admittedly, the gap between HHGNN-CHAR and the Hyper GCN is larger, indicating that Heterogeneity contributes to performance more than hyperedges. However, this may change in a larger, real-world dataset. Lastly, HHGNN-CHAR's performance decreased after removing a HHGNN layer. Intuitively, two HHGNN layers capture information in two hops of the graph, while one-layer HHGNN only learns representations from one-hop neighbor nodes. The network likely benefited from a two-layer design due to the complexity of in-the-wild CHAR data.

IV. CONCLUSION

To improve on prior approaches to the challenging Context-aware Human Activity Recognition (CHAR) task, we propose a novel graph learning approach, transforming original data-label connections into a Heterogeneous Hypergraph that explicitly encodes previously implicit relationships between context-aware activity labels. Consequently, the multi-label CHAR problem is transformed into a node identification problem given an unknown hyperedge. Transforming real-world datasets yields a graph with heterogeneous nodes and hyperedges. HHGNN-CHAR, a novel heterogeneous hypergraph deep learning model is proposed for resolving the newly transformed problem. To the best of our knowledge, ours is the first effort that formulates the CHAR task as a heterogeneous hypergraph problem without requiring external information. In rigorous experiments, our proposed approach outperformed SOTA baselines by 14.04% on Matthews Correlation Coefficient (MCC) and 7.01% on Macro F1 scores.

In future, we will evaluate HHGNN-CHAR using subject-level splitting and explore other backbone GNN models.

ACKNOWLEDGMENT

DARPA grant HR00111780032-WASH-FP-031 and NSF grant CNS-1755536 supported this research.

REFERENCES

- [1] T. Rault, A. Bouabdallah, Y. Challal, and F. Marin, "A survey of energy-efficient context recognition systems using wearable sensors for healthcare applications," *Perv. & Mob. Comp.*, 2017.
- [2] W.-P. Lee, "Deploying personalized mobile services in an agent-based environment," *Exp. Sys. & Appl.*, 2007.
- [3] K.-H. Chen, Y.-W. Hsu, J.-J. Yang, and F.-S. Jaw, "Evaluating the specifications of built-in accelerometers in smartphones on fall detection performance," *Inst. Sci. & Tech.*, 2018.
- [4] J. Lindqvist and J. Hong, "Undistracted driving: A mobile phone that doesn't distract," in *Proc. Hotmobile*, 2011.
- [5] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness," in *Int'l symp. handheld and ubiq. comp.*, 1999.
- [6] A. K. Dey, G. D. Abowd, and D. Salber, "A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications," *HCI*, 2001.
- [7] H. Martin, D. Bucher, E. Suel, P. Zhao, F. Perez-Cruz, and M. Raubal, "Graph convolutional neural networks for human activity purpose imputation," in *Spatiotemporal workshop co-located with NIPS*, 2018.
- [8] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.
- [9] A. Mohamed, F. Lejarza, S. Cahail, C. Claudel, and E. Thomaz, "Hargenn: Deep graph cnns for human activity recog. from highly unlabeled mobile sensor data," in *Proc. PerCom*, 2022.
- [10] S. Suh, V. F. Rey, and P. Lukowicz, "Adversarial deep feature extraction network for user independent har," in *PerCom*, 2022.
- [11] F. Attal, S. Mohammed, M. Dedabrishvili, F. Chamroukhi, L. Oukhellou, and Y. Amirat, "Physical har using wearable sensors," *Sensors*, 2015.
- [12] M. Berchtold, M. Budde, H. R. Schmidtke, and M. Beigl, "An extensible modular recognition concept that makes ar practical," in *AAAI*, 2010.
- [13] A. Dogrucu, A. Perucic, A. Isaro, D. Ball, E. Toto, E. A. Rundensteiner, E. Agu, R. Davis-Martin, and E. Boudreaux, "Moodable: On feasibility of instantaneous depreswension assessment using machine learning on voice samples with retrospectively harvested smartphone and social media data," *Smart Health*, 2020.
- [14] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in *Proc. ACM SIGIR*, 2020.
- [15] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proc. ACM SIGIR*, 2019.
- [16] Y. Zhu, Z. Guan, S. Tan, H. Liu, D. Cai, and X. He, "Heterogeneous hypergraph embedding for doc. recomm," *Neurocomputing*, 2016.
- [17] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, "Heterogeneous graph neural network," in *Proc. ACM SIGKDD*, 2019.
- [18] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, "Hypergraph neural networks," in *Proc. AAAI*, 2019.
- [19] S. Bai, F. Zhang, and P. H. Torr, "Hypergraph convolution and hypergraph attention," *Pattern Recognition*, 2021.
- [20] D. Yang, B. Qu, J. Yang, and P. Cudre-Mauroux, "Revisiting user mobility and social relationships in lbsns: a hypergraph embedding approach," in *The WWW Conf.*, 2019.
- [21] Y. Vaizman, K. Ellis, and G. Lanckriet, "Recognizing detailed human context in the wild from smartphones and smartwatches," *IEEE Pervasive Computing*, 2017.
- [22] Y. Vaizman, N. Weibel, and G. Lanckriet, "Context recognition in-the-wild: Unified model for multi-modal sensors and multi-label classification," *ACM IMWUT*, 2018.
- [23] W. Ge and E. O. Agu, "Qcruft: Quaternion context recog. under uncertainty using fusion & temporal learning," in *Proc. ICSC*, 2022.
- [24] W. Ge and E. Agu, "Cruft: Context recog. under uncertainty using fusion and temporal learning," in *Proc. ICMLA*, 2020.
- [25] "Lightgbm: A highly efficient gradient boosting decision tree," *Proc. NIPS*, 2017.
- [26] D. Yang, B. Qu, J. Yang, and P. Cudre-Mauroux, "Lbsn2vec++: Heterogeneous hypergraph embedding for loc.-based soc. nets," *TKDE*, 2020.