

Article

Detection and Validation of Macro-Activities in Human Inertial Signals Using Graph Link Prediction

Christoph Wieland ^{1,*}  and Victor Pankratius ² ¹ Bosch Sensortec GmbH, 72770 Reutlingen, Germany² Robert Bosch GmbH, 70469 Stuttgart-Feuerbach, Germany; victor.pankratius@bosch.com

* Correspondence: christoph.wieland2@bosch.com

Abstract: With the continuous development of new wearable devices, sensor-based human activity recognition is enjoying enormous popularity in research and industry. The signals from inertial sensors allow for the detection, classification, and analysis of human activities such as jogging, cycling, or swimming. However, human activity recognition is often limited to basic activities that occur in short, predetermined periods of time (sliding windows). Complex macro-activities, such as multi-step sports exercises or multi-step cooking recipes, are still only considered to a limited extent, while some works have investigated the classification of macro-activities, the automated understanding of how the underlying micro-activities interact remains an open challenge. This study addresses this gap through the application of graph link prediction, a well-known concept in graph theory and graph neural networks (GNNs). To this end, the presented approach transforms micro-activity sequences into micro-activity graphs that are then processed with a GNN. The evaluation on two derived real-world data sets shows that graph link prediction enables the accurate identification of interactions between micro-activities and the precise validation of composite macro-activities based on learned graph embeddings. Furthermore, this work shows that GNNs can benefit from positional encodings in sequence recognition tasks.

Keywords: human macro-activity recognition; HAR; activity sequences; activity validation; graph link prediction; graph neural network; GNN

**Citation:** Wieland, C.; Pankratius, V.Detection and Validation of
Macro-Activities in Human Inertial
Signals Using Graph Link Prediction.
Sensors **2024**, *24*, 1282. <https://doi.org/10.3390/s24041282>

Academic Editor: Ki H. Chon

Received: 24 January 2024

Revised: 8 February 2024

Accepted: 13 February 2024

Published: 17 February 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Human activity recognition (HAR) with wearable sensors is a growing research field in pervasive computing [1]. For instance, humans can track sports activities [2–4] or household activities [5,6] to monitor, support, and improve their daily routines. Published benchmark data sets form an important foundation for the development and implementation of new HAR applications. These data sets contain signals from accelerometers, gyroscopes, or other sensors that are available in wearable devices [4,5,7,8]. Typically, these signals have a single primary activity label. Most published work focuses on improving the classification rates for these activities using increasingly sophisticated deep learning models [9,10].

Besides the recognition of plain human activities, another challenge for sensor-based HAR has emerged in recent years: the detection of *macro-activities* (or *complex activities*) [6,11]. Macro-activities are complex human activities that are composed of various subsequent *micro-activities* (or *meta-activities*) [6,11]. Depending on the use case, micro- and macro-activities can have distinct levels of granularity. Liu et al. refer to the lowest level of movements as *motion units* [12]. Some research projects also investigate the recognition of concurrent or coupled human activities [13].

In table tennis, for example, training exercises (macro-activities) are sequences of strokes (micro-activities) [14]. Furthermore, a single table tennis stroke involves a back-swing, a stroke movement, and a swing-out movement (motion units) [15]. The distinction between human micro- and macro-activities, however, is a more general problem that

is not particularly restricted to the sports domain. In 2020, a macro-activity recognition competition [16] challenged participants to accurately detect macro-activities and their micro-activities in cooking scenarios. For example, the macro-activity preparing salad consists of various micro-activities, such as taking some vegetables, washing them, slicing them, and putting them in a bowl [6]. Although the distinction between micro- and macro-activities can provide profound information about human behavior [11], such competitions are the exception in state-of-the-art HAR research. A possible explanation for the lack of micro–macro HAR research is the lack of data, as most public HAR data sets only contain either micro- or macro-activity labels [4,5,7,8].

Another challenge in sensor-based HAR is determining when a macro-activity begins and when it ends. Data sets, such as the cooking activity data set [6], typically provide this information by defining time intervals (windows) that represent portions of specific macro-activities. Unfortunately, this information is not directly accessible in real-world scenarios. Therefore, this work explores the adaptation of graph link prediction [17] to detect and facilitate the tracking of valid micro-activity sequences in inertial sensor signals. In this context, link prediction determines whether the next micro-activity can be appended (linked) to the current sequence of micro-activities, thereby forming a valid sequence or marking the beginning of another sequence. This paper builds on graph neural networks (GNNs) [18] because they have previously shown promising results on link prediction tasks [18–20]. Additionally, GNNs are underrepresented in the HAR domain, with only a few existing papers on GNN-based HAR (e.g., [21–24]). This study offers an excellent opportunity to gain a deeper understanding into the application of GNNs within the HAR domain.

This work presents a GNN-based neural network architecture with three components. The components detect and validate micro-activity sequences that form complex macro-activities. In addition, we present an end-to-end approach for integrating a novel macro-activity recognizer into an existing micro-activity recognition workflow. We evaluate our concept on data that we have derived from table tennis exercises [3] and cooking activities [6]. For the table tennis data, we consider training exercises as macro-activities and the corresponding stroke types as micro-activities. For the cooking data, we consider recipes as macro-activities and preparation steps as micro-activities. In both cases, the micro-activities are located within short sliding windows of signal data. The detection of such homogeneous micro-activity windows is enabled by change point detection [25], adaptive sliding windows [26], or joint activity segmentation [27].

This paper has the following structure. Section 2 gives a brief overview of the related work. Section 3 describes our proposed method in detail. Section 4 evaluates our approach on two data sets. Section 5 discusses important considerations regarding the application of our approach. Section 6 provides a conclusion.

2. Related Work

Peng et al. [28] presented AROMA, a multi-task learning framework for micro–macro HAR with sensor signals. AROMA uses a CNN to extract task-dependent features from micro-activity windows and an LSTM to classify macro-activities. The LSTM takes fixed-length macro-activity samples that contain a fixed number of subsequent micro-activity CNN embeddings. Peng et al. trained the two structures jointly to capture related features from both tasks.

Hu et al. [29] presented a semi-supervised approach for classifying micro-activities that leverages only a small amount of labeled micro-activity data for training. Since macro-activities implicitly contain information about their underlying micro-activities, the presented framework additionally integrates labeled macro-activity data in the training process to compensate for the missing micro-activity labels. Thus, the framework learns relationships between micro- and macro-activities to perform micro-activity predictions. Since macro-activity labels are easier to obtain and micro-activity labels are limited, the presented approach makes the data preparation for micro–macro HAR much easier.

Xie et al. [11] presented an approach to micro- and macro-activity recognition that is based on template matching. They first take a three-axis acceleration signal and calculate movement angle profiles. Then, they leverage these movement angle profiles to extract separate macro-activity intervals from the signals. Afterward, they split each extracted macro-activity into a sequence of micro-activities and classify it using template matching and dynamic time warping. The resulting sequence of matched templates represents the macro-activity. The authors label the macro-activity using the least edit distance between the classified micro-activity sequence and given macro-activity profiles.

Alia et al. [16] summarized the “Cooking Activity Recognition Challenge”. The task was to classify micro- and macro-activities in three cooking recipes based on predefined windows of accelerometer and motion capture data. Each data set contained 3 s windows with their macro-activity labels (the recipe) and a list of corresponding micro-activities, e.g., take, cut, and put. Several teams have developed a variety of machine learning approaches to solve these two tasks. Mao et al. [30] employed GNNs to classify micro-activities based on skeleton data.

The field of GNNs in sensor-based HAR is still in its infancy, with only a few research papers published. For instance, Wieland et al. [21] transformed sliding windows of signal data into graphs that connect vertices based on temporal relationships between successive time steps. The authors classified such graphs using TinyGraphHAR, a powerful GNN model for sensor-based HAR. Further approaches can be found in [22–24].

3. Materials and Methods

Our proposed method for detecting and validating macro-activities relies on graph link prediction: the goal is to determine which graph vertices are related [18] and to connect those vertices with edges. We also apply this concept one layer of abstraction up to macro-activity recognition to assess whether successive micro-activities form valid micro-activity sequences.

We now explain important concepts regarding the application of GNNs to micro-macro HAR. This chapter describes the transformation of micro-activity sequences into graph representations (Section 3.1), a novel GNN model architecture (Section 3.2), and the processing steps in our model (Section 3.3).

3.1. Mapping Micro-Activity Sequences to Graphs

As a prerequisite for graph link prediction, we need to transform a sequence $M_T = \{m_1, \dots, m_T\}$ of micro-activities m_t with $t \in \{1, \dots, T\}$ into a graph representation $G_T = (V_T, A_T)$ as illustrated in Figure 1.

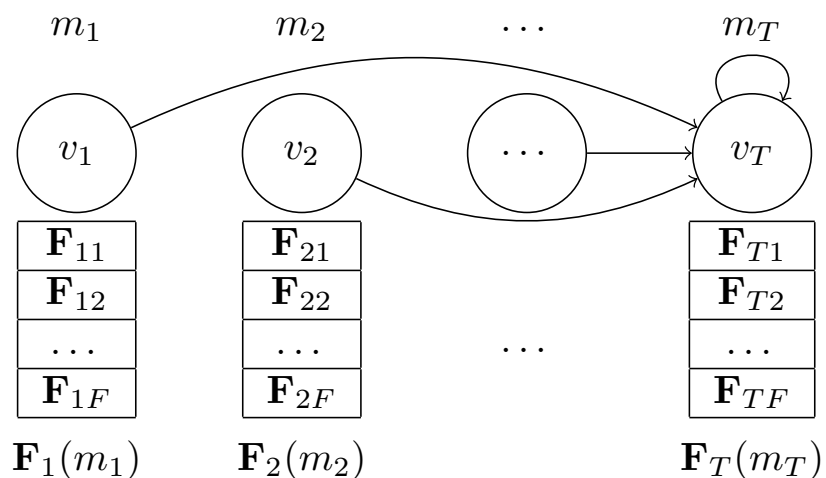


Figure 1. The graph G_T of a micro-activity sequence M_T . The micro-activities $m_t \in M_T$ represent vertices $v_t \in V_T$ with F -dimensional feature embedding vectors \mathbf{F} . Each v_t has one outgoing edge that connects it to the last vertex v_T .

The micro-activities m_t are represented as graph vertices v_t . Each v_t contains an F -dimensional micro-activity embedding vector \mathbf{F} that is either obtained by a micro-activity classifier or a comparable feature extraction technique (we use UMAPs [31] for simplicity). Hence, the set of graph vertices V_T has the shape $T \times F$.

The adjacency matrix A_T of G_T is a square $T \times T$ matrix with zeros in all columns except the last one $a[:, T] = [1]^T$ (see Table 1). Thus, our proposed micro-activity graph is equivalent to a directed star graph with a self-loop at the central vertex v_T .

Table 1. The sparse adjacency matrix A_T of G_T . The last column contains 1s. The other cells contain zeros.

A_T	v_1	...	v_{T-1}	v_T
v_1	0	.	0	1
v_2	0	.	0	1
...
v_T	0	.	0	1

The idea behind this star structure is that the GNN aggregates the individual micro-activity features in its final vertex. Thus, the last vertex becomes a rich representative of the given micro-activity sequence. Furthermore, the sparsity of the graph (and its adjacency matrix A_T) allows for efficient storage and processing, because only T edges exist in the graph. However, this graph representation ignores the temporal information of the micro-activity sequences. To account for the chronological order of micro-activities, we follow the ideas in [32] and enrich the vertex features with positional encodings before we process the graph with our GNN. The utilization of positional encodings streamlines the graph by eliminating the necessity to encode temporal dependencies along the graph edges.

3.2. Model Architecture

Our proposed multi-task model consists of three components: the graph embedding block \mathbf{E} , the graph validator \mathbf{V} , and the link predictor \mathbf{L} . Figure 2 illustrates the interplay of \mathbf{E} , \mathbf{V} , and \mathbf{L} . All inner layers leverage the ReLU activation function. The output layers of the binary tasks \mathbf{V} and \mathbf{L} use the sigmoid activation function.

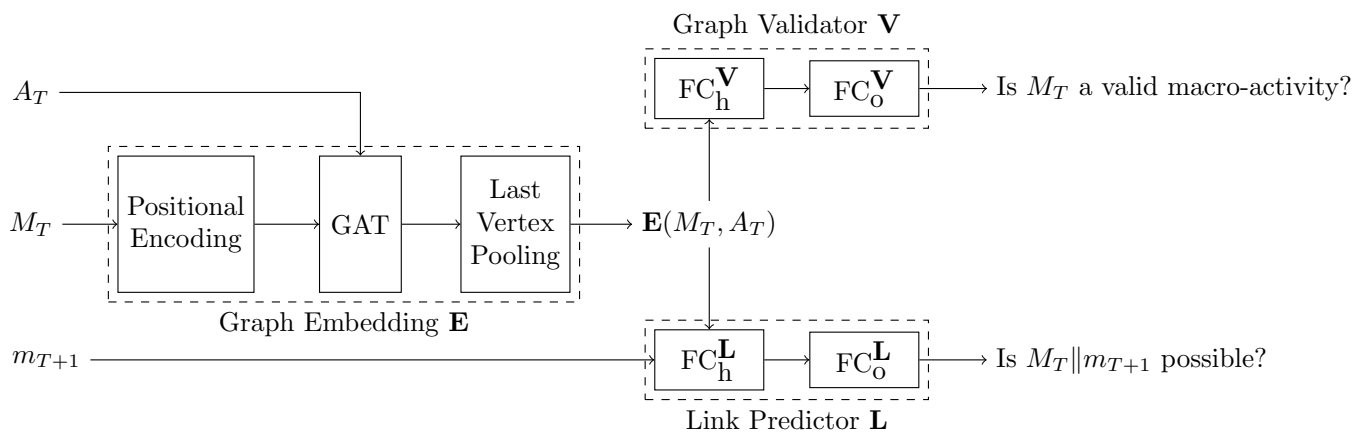


Figure 2. Our macro HAR GNN consists of the graph embedding block that calculates an embedding \mathbf{E} of a micro-activity graph $G_T = (V_T, A_T)$ using a graph attention layer (GAT), the graph validator that checks whether $\mathbf{E}(G_T)$ represents a valid macro-activity, and the link predictor that checks whether $M_T || m_{T+1}$ forms a possible micro-activity sequence, where m_{T+1} is the next macro-activity. FC_h : h fully connected hidden layers; FC_o : fully connected output layers.

The *graph embedding block E* forms the input layer to the neural network. It converts a graph $G_T = (V_T, A_T)$ of micro-activities M_T into a single embedding vector $\mathbf{E}(G_T)$. However, the sparse graph structure does not include information about the chronological

order of its micro-activities. To compensate for that, we enrich the vertex features with the positional encodings from [32]:

$$PE(t, f) = \begin{cases} \cos\left(\frac{t}{10,000^{2f/F}}\right) & \text{if } f \text{ is even,} \\ \sin\left(\frac{t}{10,000^{2f/F}}\right) & \text{if } f \text{ is odd,} \end{cases} \quad (1)$$

where $t \in T$ is the temporal index of a vertex $v_t \in V_T$ (i.e., a micro-activity $m_t \in M_T$), F is the size of the feature space, and $f \in F$ is the index of a feature. We update the vertex features V'_T with the element-wise sum of the initial vertex features V_T and their positional encodings PE :

$$V'_T = V_T \oplus PE. \quad (2)$$

Afterward, a single graph attention layer (GAT) [33] with five attention heads and $2F$ channels aggregates the information of all micro-activities in the last (T -th) graph vertex according to their relevance to the macro-activity. Finally, we extract the rich graph embedding $E(G_T)$ from the last vertex using a custom Last Vertex Pooling operation.

The *graph validator* V checks whether V_T represents a *valid* macro-activity or not. If a macro-activity, for example, consists of the three micro-activities a , b , and c , then the only valid embedding is $E([a, b, c])$. Other permutations, e.g., $[a]$, or $[a, b]$, are *invalid*. To make this decision, V passes $E(G_T)$ through $h = 3$ hidden fully connected layers FC_h^V with $2F$ hidden features and a binary output layer FC_o^V .

Definition 1 (valid and invalid micro activity sequences). *The graph validator V returns 1 if the input graph G_T represents a **valid** macro-activity, i.e., a micro-activity sequence that represents a complete macro-activity. V returns 0 if the input graph G_T is an **invalid** macro-activity, i.e., G_T is either a subsequence of a macro-activity or not part of a macro-activity.*

The *link predictor* L checks whether appending the next micro-activity m_{T+1} to the current micro-activity sequence M_T constructs a *possible* micro-activity sequence or not. For this purpose, L concatenates the sequence embedding $E(G_T)$ and m_{T+1} and passes this new vector through $h = 3$ hidden fully connected layers FC_h^L with $2F$ hidden features and a binary output layer FC_o^L . Sticking with the graph validator example, a link is possible if it forms the micro-activity sequences $[a, b, c]$, $[a, b]$, or $[a]$. In this example, all other permutations of a , b , and c are *impossible*.

Definition 2 (possible and impossible links). *The link predictor L returns 1 if the concatenation of the input graph G_T and the next micro-activity m_{T+1} results in a **possible** micro-activity sequence. **Possible** micro-activity sequences include both partial micro-activity subsequences and complete macro-activities. L returns 0 if the concatenation $G_T \parallel m_{T+1}$ is **impossible**, i.e., m_{T+1} cannot be appended to V_T as a new vertex v_{T+1} .*

3.3. End-to-End Classification Pipeline

Given a sequence of $T > 0$ micro-activity embeddings M_T and the micro-activity embedding of the next micro-activity m_{T+1} , our approach follows the scheme in Figure 3 to identify macro-activities within micro-activity graphs:

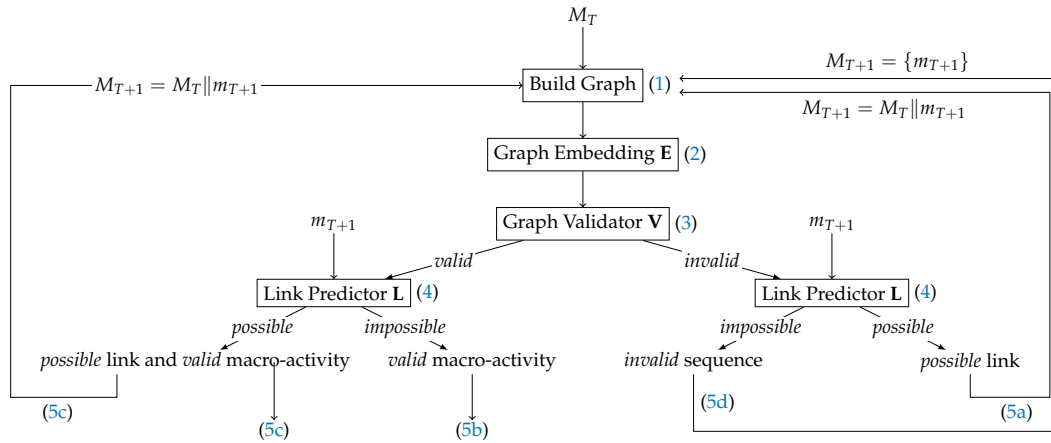


Figure 3. End-to-end classification pipeline. Inputs: initial micro-activity sequence M_T with $T > 0$ micro-activities and the next micro-activity embedding m_{T+1} . The terms *valid* and *invalid* are defined in Definition 1, and the terms *possible* and *impossible* are explained in Definition 2.

1. Construct a micro-activity graph G_T from the micro-activity embeddings M_T (cf. Section 3.1).
2. Generate the graph embedding $E(G_T)$ using the graph embedding block E .
3. Validate $E(G_T)$ using the graph validator V .
4. Check whether $V_T || m_{T+1}$ is possible using the link predictor L .
5. Now, several actions are possible:
 - (a) If $V(E(G_T))$ is invalid but $L(E(G_T))$ is possible, set $M_{T+1} = M_T || m_{T+1}$ and go back to (1).
 - (b) If $V(E(G_T))$ is valid but $L(E(G_T))$ is impossible, return $E(G_T)$.
 - (c) If $V(E(G_T))$ is valid and if $L(E(G_T))$ is possible, return $E(G_T)$. In addition, set $M_{T+1} = M_T || m_{T+1}$ and go back to (1).
 - (d) If $V(E(G_T))$ is invalid and if the link is impossible, G_T either represents an unknown micro-activity sequence or an interrupted macro-activity. Thus, interrupt the current micro-activity sequence but keep its embedding $E(G_T)$ for later use or inspection. Set $M_{T+1} = \{m_{T+1}\}$ and go back to (1).

4. Results

To evaluate our proposed method for identifying and validating macro-activities in wearable sensor signals, we rely on two real-world data sets: the cooking activity data set [6] and the table tennis data set [3]. Section 4.1 introduces the data sets. Sections 4.2 and 4.3 describe the data preparation steps and the training configuration. Section 4.4 evaluates our approach on the two prepared data sets.

4.1. Data Sets, Macro-Activity Labels, and Micro-Activity Labels

The cooking activity data set [6,16] provides accelerometer signals from four subjects during the preparation of three meals (sandwich, salad, and cereals). Four smart devices collected signals at four body locations (smartphone at the right arm and the left hip, smartwatches at the wrists). The four participants had to prepare the dishes using specified ingredients; however, the subjects could choose their preferred order of preparation. In total, the subjects have performed nine distinct micro-activities to prepare the dishes: take, wash, cut, put, mix, open, pour, peel, and *other*. In our work, we omit *other* micro-activities as we cannot assign them to the individual recipes. This results in eight concrete micro-activities.

The cooking activity data set provides 468 snippets of sensor data. Each snippet has a duration of 30 s and represents at least one micro-activity, but the temporal boundaries between the micro-activities within a snippet are unknown. Furthermore, there is no information about the sequential order of the snippets. Thus, we do not know the order of the snippets and to which execution of a recipe a particular snippet belongs. Fortunately,

the cooking activity data set description provides general information on which micro-activities are part of each recipe. Based on this information, we have derived seven valid micro-activity sequences (macro-activities): three for sandwiches, three for cereals, and one for fruit salad. The sandwich recipes consist of 11 micro-activities, the cereal recipes contain 9 micro-activities, and the fruit salad recipe has 14 micro-activities. Appendix A.1 lists the micro-activity sequences per recipe.

The table tennis data set [3] contains human inertial signals of two table tennis players while executing table tennis strokes. The players collected the signals with a Fossil Gen 5E smartwatch at their racket-holding wrist. The data set encompasses 457 time series and 8 different table tennis stroke types: drive, loop, block, push in forehand and backhand. Each time series contains 5 to 15 identical table tennis strokes. Thus, we need additional knowledge on common table tennis stroke sequences to form versatile micro-activity sequences. We gained this knowledge by analyzing common table tennis exercise descriptions from [14]. The exercises and their corresponding stroke sequences define valid micro-activity sequences. We ignore all exercises with stroke types that are not part of the eight stroke types (micro-activities) in the table tennis data set. As a result, we identify 55 unique table tennis exercises (macro-activities). The most complex exercise contains 10 strokes (micro-activities). The shortest valid micro-activity sequence has only one stroke type. Appendix A.2 lists the micro-activity sequences.

4.2. Data Preparation

4.2.1. Micro-Activity Distributions

A prerequisite of our macro-activity recognition GNN is creating a graph structure from raw activity data (cf. Section 3.1). However, the initial data sets lack well-defined micro-activity embeddings that suit the recognition of macro-activities on wearables. We apply the following steps to create representative, multivariate (i.e., F -dimensional) distributions to generate random embeddings for each micro-activity class m_i :

1. Split the signals into non-overlapping windows with 1 s of data. Each window represents one or more micro-activity classes.
2. Fit a seeded UMAP [31] with F components on the windows and extract the UMAP embeddings per micro-activity class.
3. For each micro-activity class m_i , calculate the mean values of their UMAP embeddings $\mu_i = \mu(m_i)$ and generate Gaussian micro-activity distributions $N(\mu_i, \sigma)$ with a pre-defined standard deviation σ .

We repeat this procedure several times with varying random seeds $\in \{1, 2, 3\}$ to account for random statistical behavior. We select small numbers of components $F \in \{5, 15, 25\}$ to reduce the computational requirements of the GNN. Furthermore, we investigate different standard deviations $\sigma \in \{0.05, 0.15, 0.25\}$ to evaluate the behavior of the GNN with varying levels of uncertainty.

Table 2 compares the similarities of the micro-activity embeddings per data set. The values represent the average level of similarity across the three random seeds. We define the similarity measure s as follows:

$$s = \frac{1}{\binom{n}{2}} \sum_{i=1}^{n-1} \sum_{j=i+1}^n \frac{1}{d[\mu_i, \mu_j]}, \quad (3)$$

where n represents the number of distinct micro-activity classes within a given data set and $d[i, j] = \|\mu_i - \mu_j\|$ is the Euclidean distance between the mean UMAP embedding vectors of two micro-activities m_i and m_j .

Table 2. Average similarities of the micro-activity embeddings per data set. Lower values indicate better distinguishability between micro-activities.

Data Set	Features F		
	5	15	25
Cooking Activity	1.7235	1.0429	0.8657
Table Tennis	0.5071	0.3024	0.2322

For both data sets, the similarity scores shrink with an increasing number of embedding features F . Thus, the embeddings with larger F 's are more distinctive.

The micro-activity embeddings of the cooking activity data set exhibit a higher level of similarity compared to those in the table tennis data set because the cooking activity signals often encompass multiple inseparable micro-activities (cf. Section 4.1). Consequently, the sliding windows of the cooking activity data set may carry multiple micro-activity labels that reduce the distinguishability of the UMAP embeddings of different micro-activities.

Figures 4 and 5 illustrate the mean UMAP embeddings μ_i of both data sets for different feature counts F in the 2D-plane after applying a two-dimensional principal component analysis (PCA). The diagrams also include circles around the mean vectors representing the standard deviations $\sigma \in \{0.05, 0.15, 0.25\}$. The plots confirm the conclusions from Table 2, as smaller F s exhibit more overlaps and smaller distances between micro-activities.

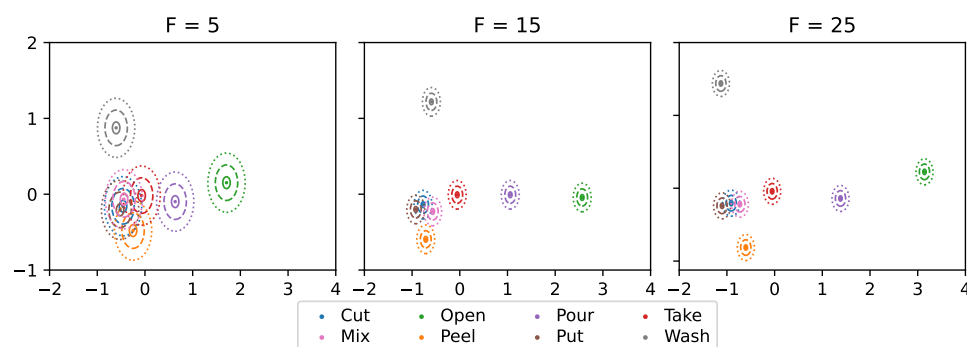


Figure 4. Two-dimensional PCA plot of the mean UMAP embeddings (seed 1) of the cooking activity data set (dots: centroids, solid/ dashed/ dotted circles: standard deviations $\sigma \in \{0.05, 0.15, 0.25\}$).

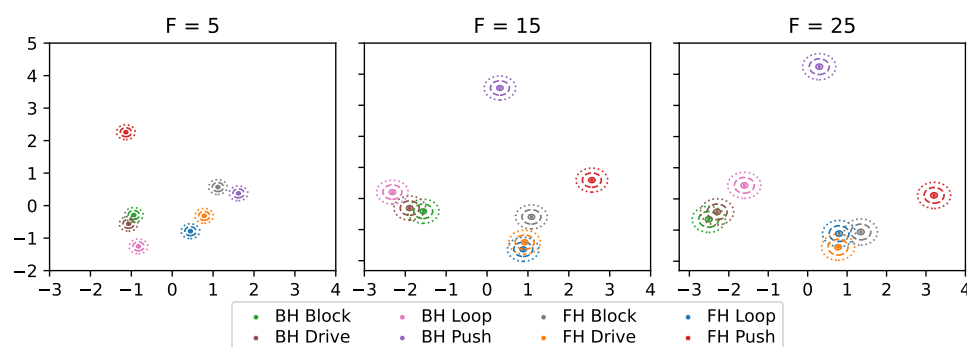


Figure 5. Two-dimensional PCA plot of the mean UMAP embeddings (seed 1) of the table tennis data set (dots: centroids, solid/ dashed/ dotted circles: standard deviations $\sigma \in \{0.05, 0.15, 0.25\}$).

4.2.2. Sequence Embeddings and Data Split

We have derived seven valid micro-activity sequences for the cooking data set and 55 valid micro-activity sequences for the table tennis data set (cf. Section 4.1). Next, we apply Algorithm 1 on each micro-activity sequence to generate all their possible subsequence embeddings.

Algorithm 1 Generate possible subsequence embeddings.

Input M_T : micro-activity sequence
 $N(\mu, \sigma)$: micro-activity distributions
Output M'_T : embeddings of all possible subsequences of M_T that start with m_1 (the first micro-activity in M_T)

- 1: $E_T \leftarrow$ empty list
- 2: **for** micro-activity class $m_t \in M_T$ **do**
- 3: $e_t \leftarrow$ draw embedding for m_t from $N(\mu(m_t), \sigma)$
- 4: $E_T.append(e_t)$
- 5: **end for**
- 6: $M'_T \leftarrow$ empty list
- 7: **for** micro-activity embedding $e_t \in E_T$ **do**
- 8: $m'_t \leftarrow E_T[t]$
- 9: $M'_T.append(m'_t)$
- 10: **end for**
- 11: **return** M'_T

Algorithm 1 takes a valid micro-activity sequence M_T and the micro-activity distributions $N(\mu, \sigma)$ as inputs and returns all possible subsequence embeddings M'_T . For instance, possible subsequences of the table tennis exercise $M_T = \{\text{drive, loop, push}\}$ are $M'_T = \{\{\text{drive}\}, \{\text{drive, loop}\}, \{\text{drive, loop, push}\}\}$. Thus, the algorithm returns embedding sequences for all three subsequences in M'_T .

We repeat this procedure 30 times for each micro-activity sequence M_T . We divide the resulting embeddings per micro-activity sequence according to a 60/20/20 rule (60% of the embeddings form the training set, the remaining 40% are evenly distributed between the validation and test sets).

4.2.3. Graphs and Graph Labels

For each subsequence embedding, we create one micro-activity graph according to Section 3.1. Each of these graphs requires two labels: the graph validator label and the link predictor label.

Graphs representing valid micro-activity sequences, i.e., complete macro-activities, receive the graph validator label 1. Graphs representing incomplete subsequences receive the graph validator label 0.

For the link prediction task, our GNN requires additional link embeddings m_{T+1} as inputs. For each graph G_T , we extract n_{pos} possible successor embeddings m_{T+1} from the next longer graphs G_{T+1} , where $G_{T+1} = G_T \parallel m_{T+1}$. These possible links receive the link predictor label 1. In addition, we randomly draw n_{imp} impossible successor embeddings from the previously generated embeddings:

$$n_{\text{imp}} = \begin{cases} 0.1 * n_{\text{total}} & \text{if } n_{\text{pos}} = 0, \\ \overline{n_{\text{pos}}} & \text{else,} \end{cases} \quad (4)$$

where n_{total} is the total number of previously generated micro-activity embeddings and $\overline{n_{\text{pos}}}$ is the average count of possible link embeddings per micro-activity. $n_{\text{pos}} = 0$ if a graph represents a valid micro-activity sequence that has no possible subsequent micro-activities. Impossible links receive the link predictor label 0.

This results in 284,916 graphs for the cooking activity data and 853,062 samples for the table tennis data. Table 3 lists the label distributions of the graph validator and the link predictor in the train splits of both data sets.

Table 3. Label distributions of the graph validator and the link predictor in the training splits of the cooking activity data set and the table tennis data set.

Data Set	Graph Validator		Link Predictor	
	Valid	Invalid	Possible	Impossible
Cooking Activity	16,254	217,134	41,796	191,592
Table Tennis	134,784	564,066	174,960	523,890

Due to the following reasons, the labels are highly imbalanced. For each complete micro-activity sequence M_T , we derive T subsequences with different lengths. Only the T -th sequence represents the complete macro-activity. Thus, most graphs represent incomplete, invalid micro-activity sequences. Moreover, we generate impossible link predictor samples for all micro-activity classes that cannot follow a subsequence based on our derived macro-activities. Especially for longer micro-activity sequences, there are only a few possible subsequent micro-activity classes. Therefore, the number of impossible link predictor samples grows quickly.

4.3. Model Training

Our multi-output model learns two loosely coupled tasks in parallel. The model updates the weights of **V** and **L** based on their respective binary cross-entropy losses and the weights of **E** with respect to the summed losses of all components. We balance the samples per class and task to ensure that unequal class distributions do not disturb the training. We use the Adam optimizer [34] with a learning rate of 0.001 for optimization and a batch size of 128. To reduce overfitting, we apply early stopping with patience 10 and a maximum number of 500 epochs.

4.4. Quantitative Analysis

We evaluate our GNN models using various data configurations. We alter the micro-activity embedding sizes F , vary σ during data generation, and investigate the effects of positional encodings on the graphs. To obtain robust results, we repeat each experiment three times with varying random seeds (1, 2, 3). Figure 6 presents the mean accuracies and the mean macro F1 scores for the cooking activity data and Figure 7 for the table tennis data. The plots also contain the standard deviations of the metrics.

4.4.1. Cooking Data (Figure 6)

The performances of **V** and **L** increase with a growing number of features F . We expect this behavior since adding more features to the UMAP embeddings reduces the similarity between the support vectors per micro-activity class (cf. Table 2). Thus, it improves the distinguishability of the samples.

Shrinking σ also improves the performances since a smaller σ results in narrower Gaussian distributions for data generation. Thus, it increases the inner similarities of the micro-activity classes.

Furthermore, adding positional encodings PE improves graph validation and link prediction across all experiments. Again, we expect this behavior since our graph transformation completely ignores the temporal relationships between subsequent micro-activities (cf. Figure 1). The differences between the addition and the omission of PE increase with growing σ and are most noticeable in the macro F1 scores of the graph validator **V**. Even with small $\sigma = 0.05$, the F1 scores of the graph validator without PE are up to 22.2 percentage points worse than with PE . Moreover, adding PE enables **V** to accurately validate graphs with an F1 score of up to 96.2%, even for the largest $\sigma = 0.25$. These results highlight the importance of positional encodings for the reliability of the models and their potential to improve GNN-based HAR models.

Another interesting finding is that the graph validator outperforms the link predictor across all settings. The differences between both components become larger with growing

σ . Thus, the link prediction task requires more precisely distinguishable micro-activity embeddings than graph validation. Nevertheless, L yields remarkable F1 scores of more than 89% for $F \geq 15$ features and $\sigma \leq 0.15$ thanks to the assistance of positional encodings.

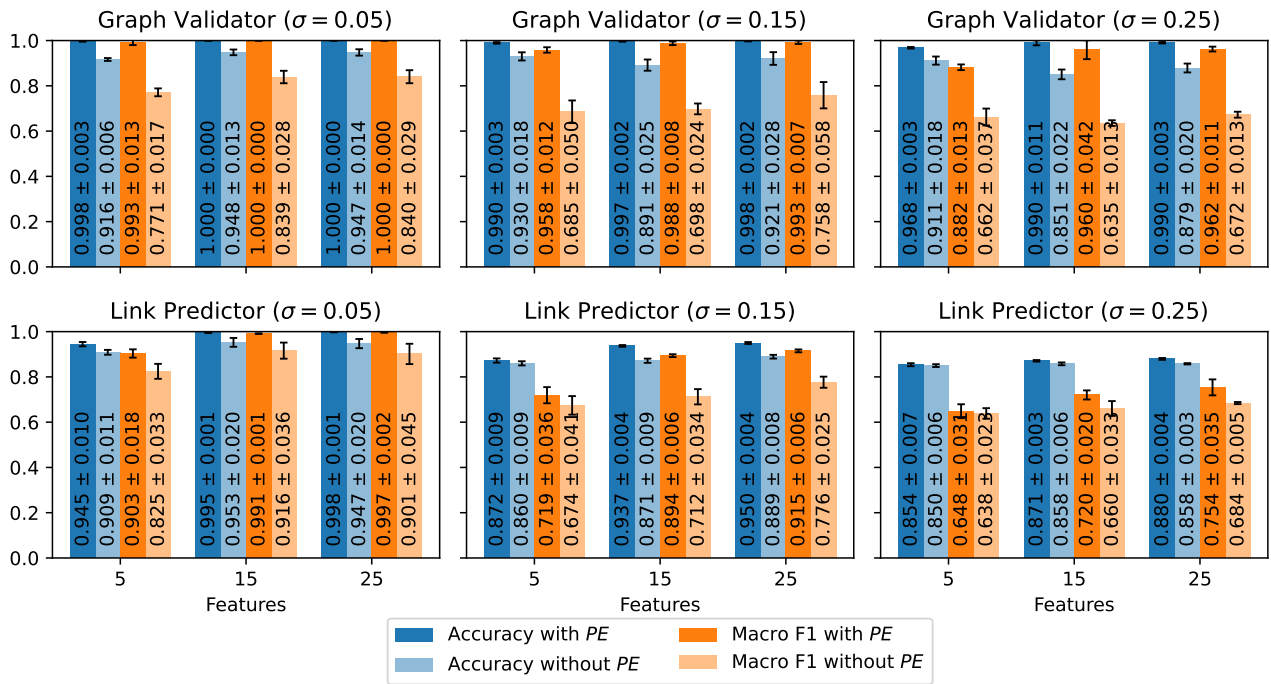


Figure 6. Test results of the graph validator and the link predictor on the cooking data set. The plots illustrate the mean accuracies and the mean macro F1 scores for different standard deviations σ and varying micro-activity embedding sizes F . Each plot compares the results with and without the positional embedding PE.

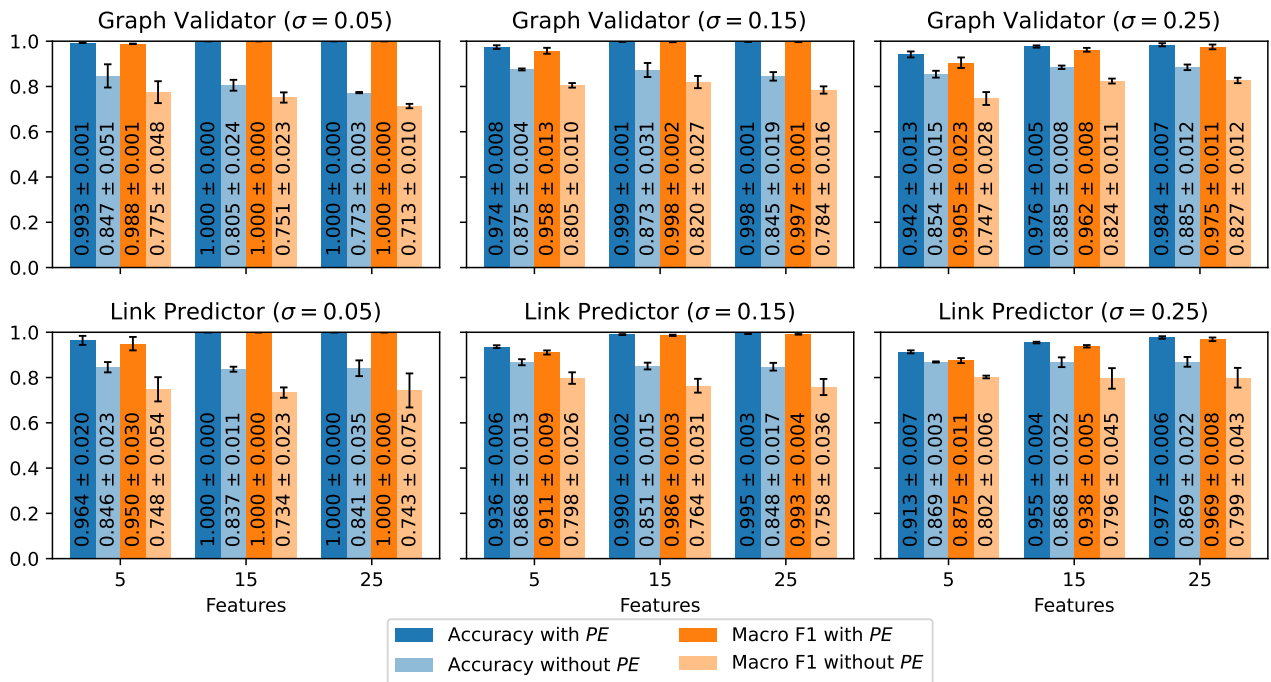


Figure 7. Test results of the graph validator and the link predictor on the table tennis data set. The plots illustrate the mean accuracies and the mean macro F1 scores for different standard deviations σ and varying micro-activity embedding sizes F . Each plot compares the results with and without the positional embedding PE.

4.4.2. Table Tennis Data (Figure 7)

The general observations from the cooking data set also hold for the table tennis data set. The accuracies and the F1 scores exhibit consistent improvements with growing F . However, due to more pronounced variations among individual micro-activity classes in the table tennis data set, the differences in metric scores across diverse F dimensions are notably smaller compared to the cooking activity data set.

Furthermore, shrinking σ has a similar effect in the table tennis scenario as for the cooking data set, but in general, the impact is smaller. This observation is especially evident for the link predictor. But, even with $\sigma = 0.25$, **L** reaches F1 scores greater than 93% for $F \geq 15$ features.

The performance metrics of the table tennis data also benefit from positional encodings. For instance, *PE* improves the macro F1 score of **V** by up to 28.7 percentage points and the F1 score of **L** by up to 25.7 percentage points for $\sigma = 0.05$. The impact of *PE* slightly shrinks with growing F . Nevertheless, the improvements from adding *PE* remain well above 15 percentage points for both model components. Again, these findings underline the importance of positional encodings in graph-based sequence recognition tasks but also emphasize the importance of a distinguishable feature space.

Similar to the cooking activity data set, the graph validator outperforms the link predictor in the table tennis scenario. However, the differences are less pronounced because **L** performs better overall, especially with $\sigma \leq 15$. With a difference of 3 percentage points, we observe the most significant gap for $F = 5$ and $\sigma = 0.25$. With $F = 5$ features, the differences shrink to less than 0.6 percentage points. Hence, even with the largest $\sigma = 0.25$, the gap between **V** and **L** becomes negligible, provided that the micro-activities are easily distinguishable.

5. Discussion

The unique graph structure eliminates the temporal component from the micro-activity sequences by directly connecting all micro-activities only to the last micro-activity in time. To overcome this limitation, we enrich the vertex features with positional encodings. This enhancement preserves the essential temporal information within the micro-activity graphs. Our evaluation underscores the effectiveness of this method, highlighting its ability to accurately verify micro-activity graphs.

One drawback of our approach is that it alters the requirements for data preparation compared to conventional sensor-based HAR scenarios. Instead of labeling windows solely with a single activity class, our method necessitates additional information for detecting and validating micro-activity sequences. Specifically, this includes knowledge about possible micro-activity sequences (i.e., which activity follows another). To our knowledge, sensor-based HAR data sets with high-quality micro- and macro-activity labels are currently unavailable. Consequently, we had to derive our evaluation data from two existing HAR data sets with limited details on the interplay between micro- and macro-activities.

Our data curation strategy involves the generation of both possible and impossible micro-activity sequences. This procedure rapidly inflates our micro-macro-activity data sets, particularly in scenarios where there is either a large or a small number of distinct micro-activity classes that can follow the current micro-activity sequence (many possible links or many impossible links). This issue emphasizes the need for future sensor-based HAR data sets to include detailed meta-information about micro- and macro-activities and their sequential order. Future data sets with continuous sensor signals and clearly defined activities will no longer require the generation of synthetic micro-activity sequences. Instead, the sequence information provides labels for possible links, while we only need to generate impossible links as negative samples for training the link predictor. Thus, such data sets will streamline the development of robust micro-macro HAR models. Moreover, our proposed solution to graph-based link prediction for detecting macro-activity sequences in sensor signals requires unambiguous micro-activity windows that serve as graph vertices. Current research suggests methods such as change point detection [25], adaptive sliding windows [26], or joint activity segmentation [27] to generate such homogeneous windows.

As described in Section 3.3 Step (5d), our proposal allows for the detection and storage of invalid, impossible micro-activity sequences. Storing incomplete sequences raises the question of how to deal with interrupted micro-activity sequences. One potential solution involves reloading previously interrupted micro-activity sequences after completing the subsequent macro-activity instead of initializing a new activity sequence. Depending on the number of previously interrupted micro-activity sequences, however, this approach may slow down the overall classification pipeline. Thus, developing a retention policy becomes crucial to manage memory usage and computational costs.

An alternative to the GNN-based prediction of possible links between micro-activities are hidden Markov models (HMMs). According to [35], HMMs are applicable to (sensor-based) HAR. However, detecting and validating human macro-activities with HMMs requires extensive exploration of how to transform micro-activities and their varying feature embedding vectors into HMM states and observations. Solving this challenge will enable a future comparison between GNN-based and HMM-based link prediction approaches.

6. Conclusions

In this work, we have explored the application of GNNs for predicting links between subsequent human micro-activities. We have presented a model architecture that leverages the power of GNNs to predict links between micro-activities and previously executed micro-activity sequences. This way, link prediction surpasses the limitations of basic activity classifications and opens up plenty of new possibilities for pervasive HAR applications. For instance, this approach enables athletes to carefully track their training exercises, including the validation of performed exercises or the suggestion of follow-up activities. This information allows players to optimize their executed activity sequences in a goal-oriented manner.

Furthermore, we have observed that positional encodings positively effect GNNs for sensor-based macro-activity recognition. Positional encodings eliminate the need to model the complex temporal relationships of sensor signals in graphs. In the future, we want to explore the potential benefits of combining positional encodings with graph neural networks or convolutional neural networks to enhance traditional sensor-based HAR tasks.

Author Contributions: Conceptualization, C.W. and V.P.; methodology, C.W.; software, C.W.; validation, C.W. and V.P.; formal analysis, C.W.; investigation, C.W.; data curation, C.W.; writing—original draft preparation, C.W.; writing—review and editing, C.W. and V.P.; visualization, C.W.; supervision, V.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Bosch Sensortec GmbH and Robert Bosch GmbH.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The cooking activity data set is openly available in [6]. The table tennis data set [3] is closed.

Conflicts of Interest: The authors were employed at Bosch Sensortec GmbH (C.W.) and Robert Bosch GmbH (V.P.).

Abbreviations

The following abbreviations are used in this manuscript:

HAR	Human Activity Recognition
V	Graph Validator
L	Link Predictor
E	Graph Embedding Block
GNN	Graph Neural Network
GAT	Graph Attention Layer
FC	Fully Connected Layer
PCA	Principal Component Analysis
PE	Positional Encoding

Appendix A. Micro-Activity Sequences

Appendix A.1. Cooking Activity Data Set

The cooking activity data set [6] contains 8 different cooking activities. For better readability, we assign the following numeric codes to the cooking activities. 0: cut, 1: mix, 2: open, 3: peel, 4: pour, 5: put, 6: take, 7: wash. With the help of [16], we have identified the following cooking activity sequences per recipe. Sandwich: 60567056705, 67056056705, 67056705605; cereal: 630565624, 656246305, 656305624; fruitsalad: 63056305630561.

Appendix A.2. Table Tennis Data Set

The table tennis data set [3] contains 8 different stroke types. For better readability, we assign the following numeric codes to the table tennis strokes types. 0: backhand block, 1: backhand drive, 2: backhand loop, 3: backhand push, 4: forehand block, 5: forehand drive, 6: forehand loop, 7: forehand push. With the help of [14], we have identified 55 unique table tennis exercises that train these 8 stroke types: 0, 1, 2, 3, 4, 5, 6, 7, 02, 11, 12, 22, 24, 32, 33, 36, 37, 55, 56, 66, 73, 76, 77, 032, 111, 116, 213, 300, 302, 315, 330, 355, 373, 512, 555, 600, 662, 700, 0044, 1230, 1256, 2226, 2266, 3377, 3460, 5674, 6662, 7632, 72626, 76767, 662551, 3266666, 51624073, 222666156, 7373202646.

References

1. Zhang, C.; Cao, K.; Lu, L.; Deng, T. A multi-scale feature extraction fusion model for human activity recognition. *Sci. Rep.* **2022**, *12*, 20620. [\[CrossRef\]](#) [\[PubMed\]](#)
2. Wang, Y.; Zhao, Y.; Chan, R.H.M.; Li, W.J. Volleyball Skill Assessment Using a Single Wearable Micro Inertial Measurement Unit at Wrist. *IEEE Access* **2018**, *6*, 13758–13765. [\[CrossRef\]](#)
3. Wieland, C.; Pankratius, V. Domain-Knowledge Enhanced Machine Learning for Table Tennis Stroke Characterization. In Proceedings of the 2023 IEEE World AI IoT Congress (AIIoT), Seattle, WA, USA, 7–10 June 2023; pp. 590–597. [\[CrossRef\]](#)
4. Sikder, N.; Nahid, A.A. KU-HAR: An open dataset for heterogeneous human activity recognition. *Pattern Recognit. Lett.* **2021**, *146*, 46–54. [\[CrossRef\]](#)
5. Reiss, A.; Stricker, D. Introducing a New Benchmarked Dataset for Activity Monitoring. In Proceedings of the 2012 16th International Symposium on Wearable Computers, Newcastle, UK, 18–22 June 2012; pp. 108–109. [\[CrossRef\]](#)
6. Lago, P.; Takeda, S.; Adachi, K.; Alia, S.S.; Matsuki, M.; Benai, B.; Inoue, S.; Charpillat, F. Cooking activity dataset with macro and micro activities. *IEEE Dataport* **2020**. [\[CrossRef\]](#)
7. Anguita, D.; Ghio, A.; Oneto, L.; Parra, X.; Reyes-Ortiz, J.L. Human Activity Recognition on Smartphones Using a Multiclass Hardware-Friendly Support Vector Machine. In *Ambient Assisted Living and Home Care: Proceedings of the 4th International Workshop (IWAAL 2012)*, Vitoria-Gasteiz, Spain, 3–5 December 2012; Bravo, J., Hervás, R., Rodríguez, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 216–223. [\[CrossRef\]](#)
8. Bachlin, M.; Plotnik, M.; Roggen, D.; Maidan, I.; Hausdorff, J.M.; Giladi, N.; Troster, G. Wearable Assistant for Parkinson's Disease Patients With the Freezing of Gait Symptom. *IEEE Trans. Inf. Technol. Biomed.* **2010**, *14*, 436–446. [\[CrossRef\]](#) [\[PubMed\]](#)
9. Zhou, Y.; Zhao, H.; Huang, Y.; Hefenbrock, M.; Riedel, T.; Beigl, M. TinyHAR: A Lightweight Deep Learning Model Designed for Human Activity Recognition. In Proceedings of the International Symposium on Wearable Computers (ISWC'22), Cambridge, UK, 11–15 September 2022; Association for Computing Machinery (ACM): New York, NY, USA, 2022. [\[CrossRef\]](#)
10. Bock, M.; Hölzemann, A.; Moeller, M.; Van Laerhoven, K. Improving Deep Learning for HAR with Shallow LSTMs. In Proceedings of the 2021 International Symposium on Wearable Computers (ISWC '21), New York, NY, USA, 21–26 September 2021; pp. 7–12. [\[CrossRef\]](#)
11. Xie, L.; Dong, X.; Wang, W.; Huang, D. Meta-activity recognition: A wearable approach for logic cognition-based activity sensing. In Proceedings of the IEEE Conference on Computer Communications (IEEE INFOCOM 2017), Atlanta, GA, USA, 1–4 May 2017; pp. 1–9. [\[CrossRef\]](#)
12. Liu, H.; Hartmann, Y.; Schultz, T. Motion Units: Generalized Sequence Modeling of Human Activities for Sensor-Based Activity Recognition. In Proceedings of the 2021 29th European Signal Processing Conference (EUSIPCO), Dublin, Ireland, 23–27 August 2021; pp. 1506–1510. [\[CrossRef\]](#)
13. Gehrig, D. Automatic Recognition of Concurrent and Coupled Human Motion Sequences. Ph.D. Thesis, Karlsruher Institut für Technologie (KIT), Karlsruhe, Germany, 2015. [\[CrossRef\]](#)
14. Newgy Industries Inc. *Robo-Pong 3050XL Owner's Manual*; Newgy Industries Inc.: Hendersonville, TN, USA, 2018.
15. Groß, B.U. *Tischtennis Basics*: [Alle Grundschrifttechniken in 30 Bildreihen; Aufschlüsse, Beinarbeit und Stellungsspiel; Praxis- und Trainingstipps von Richard Prause], 6th ed.; Meyer & Meyer: Aachen, Germany, 2015.
16. Alia, S.; Lago, P.; Takeda, S.; Adachi, K.; Benaissa, B.; Ahad, M.A.R.; Inoue, S. Summary of the Cooking Activity Recognition Challenge. In *Human Activity Recognition Challenge*; Springer: Singapore, 2021; pp. 1–13. [\[CrossRef\]](#)

17. Liben-Nowell, D.; Kleinberg, J. The Link Prediction Problem for Social Networks. In Proceedings of the 12th International Conference on Information and Knowledge Management (CIKM '03), New York, NY, USA, 3–8 November 2003; pp. 556–559. [\[CrossRef\]](#)
18. Zhang, M.; Chen, Y. Link Prediction Based on Graph Neural Networks. In *Advances in Neural Information Processing Systems: Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS'18), Montreal, Canada, 3–8 December 2018*; Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., Eds.; Curran Associates Inc.: Red Hook, NY, USA, 2018; Volume 31.
19. Afoudi, Y.; Lazaar, M.; Hmaidi, S. An enhanced recommender system based on heterogeneous graph link prediction. *Eng. Appl. Artif. Intell.* **2023**, *124*, 106553. [\[CrossRef\]](#)
20. Islam, M.K.; Aridhi, S.; Smail-Tabbone, M. A comparative study of similarity-based and GNN-based link prediction approaches. *arXiv* **2020**, arXiv:2008.08879.
21. Wieland, C.; Pankratius, V. TinyGraphHAR: Enhancing Human Activity Recognition With Graph Neural Networks. In Proceedings of the 2023 IEEE World AI IoT Congress (AllIoT), Seattle, WA, USA, 7–10 June 2023; pp. 47–54. [\[CrossRef\]](#)
22. Mondal, R.; Mukherjee, D.; Singh, P.K.; Bhateja, V.; Sarkar, R. A New Framework for Smartphone Sensor-Based Human Activity Recognition Using Graph Neural Network. *IEEE Sens. J.* **2021**, *21*, 11461–11468. [\[CrossRef\]](#)
23. Mohamed, A.; Lejarza, F.; Cahail, S.; Claudel, C.; Thomaz, E. HAR-GCNN: Deep Graph CNNs for Human Activity Recognition From Highly Unlabeled Mobile Sensor Data. In Proceedings of the 2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops), Pisa, Italy, 21–25 March 2022; pp. 124–126. [\[CrossRef\]](#)
24. Nian, A.; Zhu, X.; Xu, X.; Huang, X.; Wang, F.; Zhao, Y. HGCNN: Deep Graph Convolutional Network for Sensor-Based Human Activity Recognition. In Proceedings of the 2022 8th International Conference on Big Data and Information Analytics (BigDIA), Guiyang, China, 24–25 August 2022; pp. 422–427. [\[CrossRef\]](#)
25. Ni, Q.; Patterson, T.; Cleland, I.; Nugent, C. Dynamic detection of window starting positions and its implementation within an activity recognition framework. *J. Biomed. Inform.* **2016**, *62*, 171–180. [\[CrossRef\]](#) [\[PubMed\]](#)
26. Noor, M.H.M.; Salicic, Z.; Wang, K.I.K. Adaptive sliding window segmentation for physical activity recognition using a single tri-axial accelerometer. *Pervasive Mob. Comput.* **2017**, *38*, 41–59. [\[CrossRef\]](#)
27. Xia, S.; Chu, L.; Pei, L.; Yu, W.; Qiu, R.C. A Boundary Consistency-Aware Multitask Learning Framework for Joint Activity Segmentation and Recognition With Wearable Sensors. *IEEE Trans. Ind. Inform.* **2023**, *19*, 2984–2996. [\[CrossRef\]](#)
28. Peng, L.; Chen, L.; Ye, Z.; Zhang, Y. AROMA: A Deep Multi-Task Learning Based Simple and Complex Human Activity Recognition Method Using Wearable Sensors. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*; Association for Computing Machinery (ACM): New York, NY, USA, 2018; Volume 2. [\[CrossRef\]](#)
29. Hu, Z.; Yu, T.; Zhang, Y.; Pan, S. Fine-Grained Activities Recognition with Coarse-Grained Labeled Multi-Modal Data. In Proceedings of the Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers (UbiComp/ISWC '20 Adjunct), New York, NY, USA, 12–17 September 2020; pp. 644–649. [\[CrossRef\]](#)
30. Mao, D.; Lin, X.; Liu, Y.; Xu, M.; Wang, G.; Chen, J.; Zhang, W. Activity recognition from skeleton and acceleration data using cnn and gcn. In *Human Activity Recognition Challenge*; Springer: Singapore, 2021; pp. 15–25.
31. McInnes, L.; Healy, J.; Melville, J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv* **2020**, arXiv:1802.03426.
32. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is All you Need. In *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems*, Long Beach, CA, USA, 4–9 December 2017; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates Inc.: Red Hook, NY, USA, 2017; Volume 30.
33. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. *arXiv* **2017**, arXiv:1710.10903. [\[CrossRef\]](#)
34. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
35. Xue, T.; Liu, H. Hidden Markov Model and Its Application in Human Activity Recognition and Fall Detection: A Review. In Proceedings of the 10th International Conference on Communications, Signal Processing, and Systems, Changbaishan, China, 24–25 July 2021; Liang, Q., Wang, W., Liu, X., Na, Z., Zhang, B., Eds.; Springer: Singapore, 2022; pp. 863–869.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.