A Human Activity Recognition and Prediction System using simulated 3D virtual environments and deep neural networks: A review of the literature

Andrew Erasmus
University of Cape Town
Cape Town, South Africa
andrew.r.eras@gmail.com

ABSTRACT

Human Activity Recognition (HAR) and Human Activity Prediction (HAP) have gained interest in recent times with applications in healthcare, surveillance and elderly care. Currently, there exists a wide range of Deep Neural Network approaches to HAR and HAP operating on many publicly available datasets with ConvAE-LSTM and CNN-GRU being top performers. However, we have identified state-of-the-art Spatial-Temporal Graph Neural Networks (STGNNs) that capture both spatial and temporal dependencies in data much like sensor-based activity data. These STGNNs are an opportunity to improve activity recognition and prediction accuracies. In addition, we have identified 3D simulations like VirtualSmartHome which have the potential to enhance datasets and improve HAR and HAP algorithm performance and provide visual analysis opportunities.

KEYWORDS

Deep Neural Networks, Spatial-temporal Graph Neural Networks, 3D Graphics, Simulation, Human Activity Recognition, Human Activity Prediction

1 Introduction & Overview

Human Activity Recognition (HAR) systems process data, and classify them into Activities of Daily Living (ADLs). Human Activity Prediction (HAP) algorithms perform analysis on the underlying patterns in the data and infer the next activities a person is likely to perform.

Applications for this research include smart homes, the healthcare industry, surveillance, elderly care, entertainment and human-computer interaction [10, 13, 38]. Literature has shown that HAR and HAP have been performed on data sources such as video imagery [47, 54, 55], static images, and sensor-based data [33, 35]. Current research indicates that the use of Deep Neural Networks (DNNs) and Spatial-Temporal Graph Neural Networks (STGNNs) has significant benefits to the HAR and HAP problem [29, 34, 36, 46]. This is because neural networks excel at automatic feature extraction [29, 36, 46] through streams of multimodal data rather than arbitrary handcrafted features[36, 69]. STGNNs build on this by extracting temporal features from data input.

We will discuss this further in Section 2, where we will analyze current DNNs being used for HAR/HAP, giving a concrete and formulated description of the problem. W will then discuss public datasets used for experimentation, identify metrics used for algorithmic comparison, and provide a machine learning pipeline for the problem. In Section 3 we will discuss STGNNs, and their benefits to the HAR/HAP problem over traditional DNN methods. We will then discuss research on recent state-of-the-art STGNNs. Section 4 will discuss 3D HAR simulation platforms available today, and Section 5 will then highlight potential areas for further exploration in this field.

All of which have the goal to identify avenues for development of a 3D HAR platform that relies on STGNNs for recognition and prediction.

2 Deep Neural Networks for HAR and HAP

2.1 Current DNN Approaches for HAR

2.1.1 The use of DNNs in HAR

The HAR problem requires large and complex datasets containing a variety of ADLs. Sensor data used for HAR can include complexities such as ambiguity, noise and sparsity, making the problem increasingly difficult [34]. Throughout literature, DNNs have been increasingly used for HAR. This is because they can extract features from complex data automatically without the need for manually engineered features [36].

2.1.2 Mathematical Formulation of the HAR Problem

The HAR problem draws similarities to other problems in literature that made use of some DNN variation. These problems include traffic, weather and stock market prediction problems [7, 16, 21, 39, 61]. From literature [7, 21], we can adapt a formulation to define our HAR problem:

Let $X_i = \{x_i^1, x_i^2, x_i^3, \ldots, x_i^T\}$ denote data from sensor i up until time step T.

Let $X = \{X_1, X_2, X_3, \ldots, X_N\} \in \mathbb{R}^{N \times T}$ be the dataset of N sensors for T timesteps.

Our model input is data from the previous timesteps S as the following: $X_{(t-S):t} = \{x_i^{t-S}, x_i^{t-S+1}, \ldots, x_i^{t}\} \in \mathbb{R}^{N \times S}$ for each sensor i in the form:

$$X_{(t-S):t} = \begin{bmatrix} x_1^{t-S} & \cdots & x_1^t \\ \vdots & \ddots & \vdots \\ x_N^{t-S} & \cdots & x_N^t \end{bmatrix}$$

Our goal is then to output a classification of the activity at each time step t where $A_t \in \{A_1, A_2, \ldots A_K\}$ for K activities

We then can apply a DNN to learn and classify a single-label activity at time t based on the input: $A_t = DNN(X_{(t-S):t})$

From this we can apply the softmax function and 'arg max' functions in the form:

$$Y_t = arg\ max_k \left(\frac{e^{z_t^{(k)}}}{\sum_{i=1}^{K} e^{z_t^{(i)}}} \right)$$

Note that z_k is the logit or raw score for A_k at time step t. This will provide us with the output y_t, the predicted activity class at time t.

For model training, we must then compare our output Y_t with the true activity label Y_t^* using a categorial cross-entropy loss function:

$$L(\theta) = -\sum_{t=1}^{T}\sum_{k=1}^{K} Y_t^*(k) log(Y_t(k))$$

We will then minimize this loss and improve our model:

$$\theta^* = arg\ min\ L(\theta)$$

2.1.3 Publicly Available HAR Datasets

To carry out our recognition task, we need large amounts of data. This can be most effectively achieved by using publicly available datasets. These datasets have been summarized in **Table 1** of the Appendix, and will be discussed in 2.3.

2.1.4 Machine Learning Pipeline for HAR

We now will outline a machine learning pipeline (seen in **Figure 1**) for our HAR problem, with the influence of literature [7, 50] and datasets we have collected. The pipeline consists of:

1. **Data Collection** – Gather labelled sensor data from public datasets.
2. **Preprocessing** – Filter anomalies, reduce noise, and normalize sensor values.
3. **Define Inputs/Outputs** – Inputs: Sensor data over time; Output: Activity classification.
4. **Train-Test Split** – create a split with 80% training and 20% testing.
5. **Model Selection** – Choose a model like those presented in 2.1.5.
6. **Training** – Optimize using cross-entropy loss and gradient descent.
7. **Evaluation** – Assess performance against benchmarks and other literature models.

This pipeline enables accurate activity recognition from time-series sensor data.

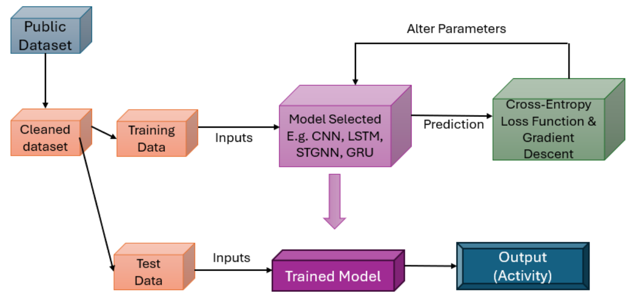

Figure 1. Proposed ML pipeline for the HAR problem

2.1.5 Current DNN Algorithms & Evaluation Metrics

We will now discuss current DNN approaches in literature to solve the HAR problem. The most common deep learning algorithms used to perform HAR in literature are variations of Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs) with specific focus to the Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) variations. Other less common approaches to this problem include Autoencoders and Transformers.

A. CNN

A CNN is a deep learning network that applies filters or kernels to a dataset and optimizes these filters to learn features. These filters are applied in convolutional layers. These layers are then pooled to reduce spatial dimensions and processed to predict and classify labels by fully connected layers [29].

Literature has shown that one-dimensional CNNs (1D-CNN) have been used to extract temporal features from time series for the purpose of HAR [18, 22]. Other studies have made use of 1D-CNNs with accelerometer and gyroscope data to identify human activities to satisfactory levels of success [25, 68]. 2D-CNNs have also been used in literature for HAR. Qi et al. [41] used smartphone data was compressed and passed through a 2D-CNN for activity recognition, while in Raj et al.[44] used accelerometer data passed into the proposed 2D-CNN for activity classification. One study used a combination of CNN, LSTM and autoencoder to leverage the complementing abilities of these algorithms, named "ConvAE-LSTM". This experiment yielded an accuracy of 98.14% based on the UCI-HAR dataset, with the CNN baseline measuring an accuracy of 93.96% [53].

B. RNN

A RNN is an architecture designed to process sequential data, and is well-suited to analyze time series. RNNs are designed to take the output of a neuron in one timestep and use that as the input to a neuron in the next timestep. This leverages temporal dependencies between neurons, as described by Kaseris et al. [29]. However, this works best for short-term dependencies and does not account for the long term, hence, RNNs have many variations [65]. For HAR, LSTM is one of the most common variations, which is where RNNs are combined with memory cells to capture dependencies in the long term [29]. Another RNN variation used in HAR is the GRU. GRU simplifies the LSTM architecture by having a simpler gate update mechanism. This improves efficiency but does not capture long-term dependencies as effectively [65]

In literature, LSTM and GRU are the most common RNNs. Bidirectional LSTM (Bi-LSTM) was used by Bollampally et. al [4] with wearable sensor data to achieve an accuracy score of 99.27%. Chen et al. utilizes an LSTM-based approach on accelerometer data in [12], producing a 92.1% accuracy for activity recognition. Liciotti et al. discuss and summarize different LSTM structures and their results on benchmark datasets at length in [34].

Simplifying LSTM structure, GRUs are also used for HAR. GRU has been combined with CNN by Khatun et al.[31] to capture temporal dependencies while having enhanced feature extraction. Gumaei et al. leveraged a hybrid of GRUs and simple recurrent units (SRUs) in [23] to improve accuracy of activity recognition over traditional SRUs achieving a 99.80% accuracy.

C. Autoencoders & Transformers

HAR literature primarily involves CNNs and RNNs. Alternatives like autoencoders and transformers have also been used. Stacked Denoising Autoencoder (SDAE) was used by Wang et al. in [56] to extract features automatically in an unsupervised manner. Human Activity Recognition Transformer has been applied by Ek et al. in [19] using smartphone sensor data to apply a usually heavyweight architecture to this problem. Guo et al. [24] adapts transformer models to include multi-layer convolutional layers to extract local and global features for activity classification.

A comprehensive summary of these algorithms is discussed in **Table 2** of the Appendix below.

2.2 Deep Learning for Human Activity Prediction (HAP)

2.2.1 DNN-based HAP

Reading the literature highlights that DNNs have mainly been used for HAR. The prediction that is mentioned in research is activity recognition at the current time step and does not forecast future activities. However, Jaramillo et al. [27] proposed a Sequence-to-Sequence model with aspects of LSTM and multi-head attention layers. This was to forecast signals based on accelerometer data and then was tested with a Bi-LSTM to predict future activities. With this they achieved 97.96% accuracy and 97.92% precision figures. Du et al. [17] perform activity prediction by first recording activity, performing high level recognition and then using an LSTM network, predict future activities based on activity logs and their weights. This framework has an accuracy of 78.3%. In addition, Wang et al. [57] utilize LSTMs to preform activity prediction based off smart home data. They found that LSTM outperforms GRU for predicting the next event, as well as the timestep that the next event will occur. However, GRU begins to perform better if predicting the next 3 activities rather than the direct next activity.

2.2.2 Mathematical Formulation of the HAP Problem

For the HAP problem, based on the structure in [17] we can extend our mathematical formulation to account for future predictions. With the same inputs as 2.1.2, we alter our goal to apply to the HAP problem.

Our goal is now to predict activity at future timesteps from (t+1) to (t+T'):

$$A_{(t+1):(t+T')} = \{A_{t+1,}A_{t+2}, \ldots, A_{t+T'}\}, \qquad A_t \in \{A_{1,}A_{2,} \ldots A_K\}$$

We then give our input from the previous S timesteps to the DNN to predict the activity at each future timestep:

$$A_{(t+1):(t+T')} = DNN(X_{(t-S):t})$$

We then apply the softmax and 'argmax' functions again to get our predicted activity output:

$$Y_t = arg\ max_k \left(\frac{e^{z_{t'}^{(k)}}}{\sum_{i=1}^{K} e^{z_{t'}^{(i)}}} \right)$$

We then can minimize loss and optimize our model:

$$L(\theta) = -\sum_{t'=t+1}^{t+T'} \sum_{k=1}^{K} Y_{t'}^{*}(k) log(Y_{t'}(k))$$

$$\theta^* = arg\ min\ L(\theta)$$

2.3 Summary and Discussion on DNN-based HAR and HAP

A large array of studies use publicly available datasets and a range of DNN approaches for the HAR problem . CNNs are one of the most widely used approaches for HAR but they have varying levels of success. CNNs capture local dependencies well but are limited with long-term temporal dependencies, situations in which LSTMs thrive [34, 67]. 1D and 2D-CNNs require large datasets and lots of training to achieve high accuracy [41, 42]. RNN variations like LSTMs and GRU show strong recognition capabilities and long-term dependency capturing. These variations consistently outperform CNNs in literature but have the drawback of being computationally expensive.

Many models are trained on smaller datasets in literature. Large publicly available datasets are necessary to produce robust recognition models, improve model accuracy and increase generalization.

Hybrid models like CNN-GRU [31] and ConvAE-LSTM [53] achieve some of the highest accuracy for the HAR problem out of all models discussed. This is because these models leverage strengths from the different networks they are comprised of to achieve higher accuracy. Bi-LSTM [4] is also noted to perform well with sequential activities in the HAR task.

The most common metrics used in literature include Accuracy, Precision, Recall and F1-score, as seen in **Table 2.** Confusion matrices are also common in literature where these metrics were discovered. These metrics allow benchmarking to occur for different approaches for HAR and HAP.

The HAP task is rare in literature. However, studies do exist that focus on the HAP problem [17, 27, 57]. These studies represent varying levels of success with great room for improvement. This area represents a large opportunity for future research.

3 Spatial-Temporal Graph Neural Networks (STGNNs) for HAR and HAP

3.1 Overview of STGNNs

Following on from our discussion of current DNN approaches for HAR and HAP, we will now discuss STGNNs. STGNNs are a type of deep learning model that focuses on modelling spatial dependencies in data through a graph structure, and models temporal dependencies as well [46]. Their role in sensor-based time series is that they can model data where observations are recorded from sensors over time while having an interconnected spatial structure. From this, they can extract relationships more effectively and make more accurate predictions.

In terms of the HAR and HAP problem, traditional DNN approaches do an adequate job in activity recognition but struggle with prediction, as seen in **Section 2**. Traditional DNN approaches have shown that they are not robust and generalizable due to their training dataset. For HAR, STGNNs introduce temporal dependencies (sensor readings over time) to a preexisting graph structure (the layout of sensors in an environment like a house). From this, they could provide more valuable information by learning the interdependent relationships between these sensors over time. This could improve the generalization and prediction capabilities of HAR models on real-world datasets, over preexisting DNN models. There has been an increased interest in dealing with time-varying graph data in literature [46]. Examples include social networks, and multivariate time series data like traffic networks. Existing hybrid DNN models like CNN-GRU [31] can model spatial patterns and capture time dependencies, but again, are not proven to be generalizable. By emphasizing interconnected relationships between sensors over time, and not treating them independently, STGNNs could improve generalizability and prediction capabilities for the HAR and HAP problems.

3.2 Current STGNN research

HAR and HAP are complex tasks. The data collected from sensors for these problems can be complex and multimodal as seen in **Table 1**. STGNNs can learn the spatial relationships between sensor data for the HAR problem and take temporal dependencies into account. Jin et al. [28] discusses trajectory prediction with STGNNs, where spatial and temporal correlations exist between movement patterns over time. HAR and HAP share many similarities with this application. For the HAR task, sensor readings change over time and are temporally dependent. STGNNs can also learn the spatial dependencies between sensors without prior knowledge and hence, STGNNs are ideal for the HAR/HAP problem.

Due to their significance and potential to be adapted to the HAR/HAP problem**,** an understanding of current state-of-the-art STGNN approaches is required. **Table 3** of the Appendix below highlights a summary of current state-of-the-art STGNNs in literature.

3.3 Summary and Discussion on STGNNs for HAR and HAP

Based on analysis of current state-of-the-art STGNNs summarized in **Table 3**, some trends can be identified. The most common benchmarking strategies used for STGNNs include Mean Average Error (MAE), Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE).

STGNNs are used for many different types of multivariate time series data. The most common applications of these STGNNs are traffic forecasting, weather forecasting and other prediction applications such as electricity consumption. STGNNs are widely applicable and thus, can be adapted for sensor-based HAR and HAP which has spatially dependent time series data. For STGNNs being used for HAR and HAP, most literature deals with skeletal-based data and transforms the data to a graph for recognition and prediction. An example being by Raj et al. in [43]. This is opposed to using STGNNs for sensor-based time series data for HAR, which could present an opportunity for further research.

In **Table 3**, most STGNNs employ a self-adaptive adjacency matrix to learn spatial dependencies within data automatically. This is except for STGCN. However, all STGNNs found have the limitation that they cannot handle dynamic graphs. Once the spatial dependencies have been learned they cannot be dynamically updated over time.

Through analysis we can see that Graph Convolutions (GCN) are the most common spatial component among STGNNs, with temporal components being some form of Temporal Convolution (TCN). Models like GWN and MTGNN automatically learn graph structures and can generalize better over different datasets. However, they can be computationally expensive, especially as the graph size increases (like StemGNN). STGNNs models like STGCN (with a fixed graph structure) are less flexible but can be more efficient, and WGN (with a low rank adjacency matrix) can also improve efficiency. To build on this discussion, we will next discuss 3D simulation platforms, and then discuss the implications of this with STGNNs as an opportunity to model the HAR/HAP problem.

4 3D HAR Simulation Platforms

4.1 3D HAR Simulation Platforms Available

For the HAR and HAP problem, visual representations of the recognition or environment in which data comes from are rare. For this, a 3D simulation would be required to demonstrate this information effectively. In literature, there exist various publicly available 3D simulation platforms. These simulation platforms have been summarized in **Table 4** of the Appendix.

4.2 Digital Twin Applications in HAR

3D simulations like VirtualHome [40] allow users to create a virtual environment to simulate real-world scenarios and create synthetic data. These environments are independent of the real world from which the data they simulate is sourced. Although 3D simulators are available in literature to generate data for HAR, Bouchabou et al. [5] discuss that these simulators are unable to produce varied datasets. This is an essential component for HAR and thus, a solution was proposed in the form of a digital twin. Digital twins are real-time digital representations of a physical process or environment [5]. In terms of the HAR/HAP problem, Bouchabou et al.[5]explains a digital twin would be a virtual reproduction of a house with its associated sensors (being fed in real-world data) so that avatars can perform ADLs. The purpose of this would be to improve HAR algorithms before real-world testing. The key difference between simulations and digital twins is that digital twins are linked directly with real world data and can reconcile the difference that exists between training and test data as described by Bouchabou et al. in [5].

4.3 Summary and Discussion

Table 4 summarizes publicly available 3D simulation platforms that have the potential to be adapted for HAR. Most of the simulation platforms are built with Unity and if they provide an API, the simulator's API is Python based. Direct integration with external datasets is uncommon for these simulators. Source code is freely available for modification to include external datasets, however. In addition, custom 3D environments can also be built in these simulators to emulate those used to record public datasets, and generate synthetic data as a result. Notable platforms that support this include OpenSHS, SESim, CHALET, VirtualHome and VirtualSmartHome [2, 5, 26, 40, 63]. Scripting is also common among these simulators and allows users various ways to define agent/avatar activities. In addition, OpenSHS, SESim, VirtualSmartHome and the simulator by Francilliette et al. [20] specifically allow for sensors to be placed in the 3D environment, which could be useful when needing to recognize and predict activity.

Based on their ability to create 3D environments, their rich feature set, their usability and extensibility, OpenSHS, SESim, Francilliette et al., and VirtualSmartHome look to be the most suitable for activity simulation and the HAR/HAP problem.

5 Further Exploration in this Field

From the discussions above, it is evident that current DNN approaches to the HAR problem exist, but are limited to the small datasets they rely on, and lack generalizability. In addition, the HAP problem has not been explored in great depth and serves as an opportunity for future research.

STGNNs are seen to suit the HAR and HAP problem, and many successful state-of-the-art models exist in literature for other applications. In terms of HAR, STGNNs have mainly been used for skeletal-based data and there is a lack of research for its use in sensor-based time series data, thus presenting another avenue for exploration. STGNNs serve as an opportunity to improve generalizability of HAR and HAP algorithms over real-world datasets.

Since real-world datasets are limited and can lack quality and diversity, 3D simulations and the potential of digital twins serve as a great opportunity to enhance HAR and HAP algorithms. The opportunity is that there is potential to create datasets with these platforms and to develop more robust HAR and HAP algorithms. In addition, these 3D platforms have the potential to augment existing datasets to enhance our understanding of predictions. These platforms serve as a great opportunity in the HAR/HAP field to improve prediction explanations, understanding, and visualize algorithm performance by means of qualitative visual analysis.

6 Conclusions

In this paper, we have introduced the HAR and HAP problems, provided mathematical formulations to define them and outlined a machine learning pipeline to solve these problems. We have discussed publicly available HAR datasets, current DNN approaches and performance metrics in literature to perform activity recognition and prediction. The most notable DNN algorithms include hybrid approaches like ConvAE-LSTM and CNN-GRU. We have also discussed STGNNs, their applicability to HAR over traditional DNN approaches, as well as associated benchmarking metrics. From this, we identified state-of-the-art STGNN techniques like GWN, STGCN, MTGNN, StemGNN and WGN. We then discussed current 3D simulation platforms and the concept of the digital twin in the HAR context. Notable simulators include VirtualSmartHome, SESim and the simulator created by Francilliette et al. From this, we identified the promising opportunity in this field, to develop a 3D AI driven HAR and HAP simulation platform.

ACKNOWLEDGMENTS

The author would like to thank Associate Professor Deshen Moodley for his insight and guidance.

REFERENCES

[1] Alemdar, H., Durmaz Incel, O., Ertan, H. and Ersoy, C. 2013. ARAS Human Activity Datasets in Multiple Homes with Multiple Residents. *Proceedings of the ICTs for improving Patients Rehabilitation Research Techniques* (2013).

[2] Alshammari, N., Alshammari, T., Sedky, M., Champion, J. and Bauer, C. *OpenSHS: Open Smart Home Simulator*.

[3] Alshammari, N., Alshammari, T., Sedky, M., Champion, J. and Bauer, C. 2017. OpenSHS: Open Smart Home Simulator. *Sensors*. 17, 5 (May 2017), 1003. DOI:https://doi.org/10.3390/s17051003.

[4] Bollampally, A., Kavitha, J., Sumanya, P., Rajesh, D., Jaffar, A.Y., Eid, W.N., Albarakati, H.M., Aldosari, F.M. and Alharbi, A.A. 2024. Optimizing Edge Computing for Activity Recognition: A Bidirectional LSTM Approach on the PAMAP2 Dataset. *Engineering, Technology & Applied Science Research*. 14, 6 (Dec. 2024), 18086–18093. DOI:https://doi.org/10.48084/etasr.8861.

[5] Bouchabou, D., Grosset, J., Nguyen, S.M., Lohr, C. and Puig, X. 2023. A Smart Home Digital Twin to Support the Recognition of Activities of Daily Living. *Sensors*. 23, 17 (Sep. 2023). DOI:https://doi.org/10.3390/s23177586.

[6] Bouchard, K., Ajroud, A., Bouchard, B., Bouchard, K., Ajroud, A., Bouchard, B., Bouzouane, A. and Bouchard, {kevin 2012. *SIMACT: a 3D Open Source Smart Home Simulator for Activity Recognition with Open Database and Visual Editor*.

[7] Bui, K.H.N., Cho, J. and Yi, H. 2022. Spatial-temporal graph neural network for traffic forecasting: An overview and open research issues. *Applied Intelligence*. 52, 3 (Feb. 2022), 2763–2774. DOI:https://doi.org/10.1007/s10489-021-02587-w.

[8] Cao, D., Wang, Y., Duan, J., Zhang, C., Zhu, X., Huang, C., Tong, Y., Xu, B., Bai, J., Tong, J. and Zhang, Q. 2021. Spectral Temporal Graph Neural Network for Multivariate Time-series Forecasting. (Mar. 2021).

[9] Chang, A., Dai, A., Funkhouser, T., Halber, M., Nießner, M., Savva, M., Song, S., Zeng, A. and Zhang, Y. 2017. Matterport3D: Learning from RGB-D Data in Indoor Environments. (Sep. 2017).

[10] Chang, M.-C., Krahnstoever, N., Lim, S. and Yu, T. 2010. Group Level Activity Recognition in Crowded Environments across Multiple Cameras. *2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance* (Aug. 2010), 56–63.

[11] Chavarriaga, R., Sagha, H., Calatroni, A., Digumarti, S.T., Tröster, G., Millán, J. del R. and Roggen, D. 2013. The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognition Letters*. 34, 15 (Nov. 2013), 2033–2042. DOI:https://doi.org/10.1016/j.patrec.2012.12.014.

[12] Chen, Y., Zhong, K., Zhang, J., Sun, Q. and Zhao, X. 2016. LSTM Networks for Mobile Human Activity Recognition. *Proceedings of the 2016 International Conference on Artificial Intelligence: Technologies and Applications* (Paris, France, 2016).

[13] Cook, D.J. 2012. Learning setting-generalized activity models for smart spaces. *IEEE Intelligent Systems*. 27, 1 (Jan. 2012), 32–38. DOI:https://doi.org/10.1109/MIS.2010.112.

[14] Cook, D.J., Crandall, A.S., Thomas, B.L. and Krishnan, N.C. 2013. CASAS: A Smart Home in a Box. *Computer*. 46, 7 (Jul. 2013), 62–69. DOI:https://doi.org/10.1109/MC.2012.328.

[15] Cumin, J., Lefebvre, G., Ramparany, F. and Crowley, J.L. 2017. A Dataset of Routine Daily Activities in an Instrumented Home. 413–425.

[16] Davidson, M. and Moodley, D. 2022. ST-GNNs for Weather Prediction in South Africa. *Communications in Computer and Information Science* (2022), 93–107.

[17] Du, Y., Lim, Y. and Tan, Y. 2019. A novel human activity recognition and prediction in smart home based on interaction. *Sensors (Switzerland)*. 19, 20 (Oct. 2019). DOI:https://doi.org/10.3390/s19204474.

[18] Duffner, S., Berlemont, S., Lefebvre, G. and Garcia, C. 2014. 3D gesture classification with convolutional neural networks. *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (May 2014), 5432–5436.

[19] Ek, S., Portet, F. and Lalanda, P. 2023. Transformer-based models to deal with heterogeneous environments in Human Activity Recognition. *Personal and Ubiquitous Computing*. 27, 6 (Dec. 2023), 2267–2280. DOI:https://doi.org/10.1007/s00779-023-01776-3.

[20] Francillette, Y., Boucher, E., Bouzouane, A. and Gaboury, S. 2017. The Virtual Environment for Rapid Prototyping of the Intelligent Environment. *Sensors*. 17, 11 (Nov. 2017), 2562. DOI:https://doi.org/10.3390/s17112562.

[21] Gaibie, A., Amir, H., Nandutu, I. and Moodley, D. 2025. Predicting and Discovering Weather Patterns in South Africa Using Spatial-Temporal Graph Neural Networks. *Communications in Computer and Information Science* (2025), 144–160.

[22] Grzeszick, R., Lenk, J.M., Rueda, F.M., Fink, G.A., Feldhorst, S. and ten Hompel, M. 2017. Deep Neural Network based Human Activity Recognition for the Order Picking Process. *Proceedings of the 4th International Workshop on Sensor-based Activity Recognition and Interaction* (New York, NY, USA, Sep. 2017), 1–6.

[23] Gumaei, A., Hassan, M.M., Alelaiwi, A. and Alsalman, H. 2019. A Hybrid Deep Learning Model for Human Activity Recognition Using Multimodal Body Sensing Data. *IEEE Access*. 7, (2019), 99152–99160. DOI:https://doi.org/10.1109/ACCESS.2019.2927134.

[24] Guo, X., Kim, Y., Ning, X. and Min, S.D. 2025. Enhancing the Transformer Model with a Convolutional Feature Extractor Block and Vector-Based Relative Position Embedding for Human Activity Recognition. *Sensors*. 25, 2 (Jan. 2025), 301. DOI:https://doi.org/10.3390/s25020301.

[25] Hamidi, M. and Osmani, A. 2021. Human Activity Recognition: A Dynamic Inductive Bias Selection Perspective. *Sensors*. 21, 21 (Nov. 2021), 7278. DOI:https://doi.org/10.3390/s21217278.

[26] Ho, B., Vogts, D. and Wesson, J. 2019. A Smart Home Simulation Tool to Support the Recognition of Activities of Daily Living. *Proceedings of the South African Institute of Computer Scientists and Information Technologists 2019* (New York, NY, USA, Sep. 2019), 1–10.

[27] Jaramillo, I.E., Chola, C., Jeong, J.G., Oh, J.H., Jung, H., Lee, J.H., Lee, W.H. and Kim, T.S. 2023. Human Activity Prediction Based on Forecasted IMU Activity Signals by Sequence-to-Sequence Deep Neural Networks. *Sensors*. 23, 14 (Jul. 2023). DOI:https://doi.org/10.3390/s23146491.

[28] Jin, G., Liang, Y., Zhang, J., Zheng, Y., Fang, Y. and Huang, J. Spatio-Temporal Graph Neural Networks for Predictive Learning in Urban Computing: A Survey. DOI:https://doi.org/10.48550/arXiv.2303.14483.

[29] Kaseris, M., Kostavelis, I. and Malassiotis, S. 2024. A Comprehensive Survey on Deep Learning Methods in Human Activity Recognition. *Machine Learning and Knowledge Extraction*. 6, 2 (Jun. 2024), 842–876. DOI:https://doi.org/10.3390/make6020040.

[30] van Kasteren, T.L.M., Englebienne, G. and Kröse, B.J.A. 2010. Activity recognition using semi-Markov models on real world smart home datasets. *Journal of Ambient Intelligence and Smart Environments*. 2, 3 (2010), 311–325. DOI:https://doi.org/10.3233/AIS-2010-0070.

[31] Khatun, Mst.A., Yousuf, M.A. and Moni, M.A. 2023. Deep CNN-GRU Based Human Activity Recognition with Automatic Feature Extraction Using Smartphone and Wearable Sensors. *2023 International Conference on Electrical, Computer and Communication Engineering (ECCE)* (Feb. 2023), 1–6.

[32] Kolve, E., Mottaghi, R., Han, W., VanderBilt, E., Weihs, L., Herrasti, A., Deitke, M., Ehsani, K., Gordon, D., Zhu, Y., Kembhavi, A., Gupta, A. and Farhadi, A. 2017. AI2-THOR: An Interactive 3D Environment for Visual AI. (Dec. 2017).

[33] Li, X., Zhang, Y., Li, M., Marsic, I., Yang, J. and Burd, R.S. 2016. Deep neural network for RFID-based activity recognition. *Proceedings of the Eighth Wireless of the Students, by the Students, and for the Students Workshop* (New York, NY, USA, Oct. 2016), 24–26.

[34] Liciotti, D., Bernardini, M., Romeo, L. and Frontoni, E. 2020. A sequential deep learning application for recognising human activities in smart homes. *Neurocomputing*. 396, (Jul. 2020), 501–513. DOI:https://doi.org/10.1016/j.neucom.2018.10.104.

[35] Minh Dang, L., Min, K., Wang, H., Jalil Piran, M., Hee Lee, C. and Moon, H. 2020. Sensor-based and vision-based human activity recognition: A comprehensive survey. *Pattern Recognition*. 108, (Dec. 2020), 107561. DOI:https://doi.org/10.1016/J.PATCOG.2020.107561.

[36] Nweke, H.F., Teh, Y.W., Al-garadi, M.A. and Alo, U.R. 2018. Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges. *Expert Systems with Applications*. Elsevier Ltd.

[37] Ordóñez, Fco.J., De Toledo, P. and Sanchis, A. 2013. Activity Recognition Using Hybrid Generative/Discriminative Models on Home Environments Using Binary Sensors. *Sensors*. 13, 5 (Apr. 2013), 5460–5477. DOI:https://doi.org/10.3390/s130505460.

[38] Paola, D. Di, Naso, D., Milella, A., Cicirelli, G. and Distante, A. 2010. Multi-sensor surveillance of indoor environments by an autonomous mobile robot. *International Journal of Intelligent Systems Technologies and Applications*. 8, 1/2/3/4 (2010), 18. DOI:https://doi.org/10.1504/IJISTA.2010.030187.

[39] Pillay, K. and Moodley, D. 2022. Exploring Graph Neural Networks for Stock Market Prediction on the JSE. *Communications in Computer and Information Science* (2022), 95–110.

[40] Puig, X., Ra, K., Boben, M., Li, J., Wang, T., Fidler, S. and Torralba, A. 2018. VirtualHome: Simulating Household Activities via Programs. (Jun. 2018).

[41] Qi, W., Su, H., Yang, C., Ferrigno, G., De Momi, E. and Aliverti, A. 2019. A Fast and Robust Deep Convolutional Neural Networks for Complex Human Activity Recognition Using Smartphone. *Sensors*. 19, 17 (Aug. 2019), 3731. DOI:https://doi.org/10.3390/s19173731.

[42] Ragab, M.G., Abdulkadir, S.J. and Aziz, N. 2020. Random Search One Dimensional CNN for Human Activity Recognition. *2020 International Conference on Computational Intelligence (ICCI)* (Oct. 2020), 86–91.

[43] Raj, M.S.S., George, S.N. and Raja, K. 2024. Leveraging spatio-temporal features using graph neural networks for human activity recognition. *Pattern Recognition*. 150, (Jun. 2024). DOI:https://doi.org/10.1016/j.patcog.2024.110301.

[44] Raj, R. and Kos, A. 2023. An improved human activity recognition technique based on convolutional neural network. *Scientific Reports*. 13, 1 (Dec. 2023), 22581. DOI:https://doi.org/10.1038/s41598-023-49739-1.

[45] Ronao, C.A. and Cho, S.-B. 2016. Human activity recognition with smartphone sensors using deep learning neural networks. *Expert Systems with Applications*. 59, (Oct. 2016), 235–244. DOI:https://doi.org/10.1016/j.eswa.2016.04.032.

[46] Al Sahili, Z. and Awad, M. *Spatio-Temporal Graph Neural Networks: A Survey*.

[47] Saleem, G., Bajwa, U.I. and Raza, R.H. 2023. Toward human activity recognition: a survey. *Neural Computing and Applications*. 35, 5 (Feb. 2023), 4145–4182. DOI:https://doi.org/10.1007/s00521-022-07937-4.

[48] Savva, M., Chang, A.X., Dosovitskiy, A., Funkhouser, T. and Koltun, V. 2017. MINOS: Multimodal Indoor Simulator for Navigation in Complex Environments. (Dec. 2017).

[49] Savva, M., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V., Malik, J., Parikh, D. and Batra, D. 2019. Habitat: A Platform for Embodied AI Research. (Apr. 2019).

[50] Sharen, H., Jani Anbarasi, L., Rukmani, P., Gandomi, A.H., Neeraja, R. and Narendra, M. 2024. WISNet: A deep neural network based human activity recognition system. *Expert Systems with Applications*. 258, (Dec. 2024). DOI:https://doi.org/10.1016/j.eswa.2024.124999.

[51] Song, S., Yu, F., Zeng, A., Chang, A.X., Savva, M. and Funkhouser, T. 2016. Semantic Scene Completion from a Single Depth Image. (Nov. 2016).

[52] Tapia, E.M., Intille, S.S. and Larson, K. 2004. Activity Recognition in the Home Using Simple and Ubiquitous Sensors. 158–175.

[53] Thakur, D., Biswas, S., Ho, E.S.L. and Chattopadhyay, S. 2022. ConvAE-LSTM: Convolutional Autoencoder Long Short-Term Memory Network for Smartphone-Based Human Activity Recognition. *IEEE Access*. 10, (2022), 4137–4156. DOI:https://doi.org/10.1109/ACCESS.2022.3140373.

[54] Vishwakarma, S. and Agrawal, A. 2013. A survey on activity recognition and behavior understanding in video surveillance. *The Visual Computer*. 29, 10 (Oct. 2013), 983–1009. DOI:https://doi.org/10.1007/s00371-012-0752-6.

[55] Vrigkas, M., Nikou, C. and Kakadiaris, I.A. 2015. A review of human activity recognition methods. *Frontiers Robotics AI*. Frontiers Media S.A.

[56] Wang, A., Chen, G., Shang, C., Zhang, M. and Liu, L. 2016. Human Activity Recognition in a Smart Home Environment with Stacked Denoising Autoencoders. 29–40.

[57] Wang, C. and Peng, Z. 2020. Deep Learning Model for Human Activity Recognition and Prediction in Smart Homes. *2020 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)* (Jan. 2020), 741–744.

[58] Wilson, T., Tan, P.-N. and Luo, L. 2018. A Low Rank Weighted Graph Convolutional Approach to Weather Prediction. *2018 IEEE International Conference on Data Mining (ICDM)* (Nov. 2018), 627–636.

[59] Wu, Y., Wu, Y., Gkioxari, G. and Tian, Y. 2018. Building Generalizable Agents with a Realistic and Rich 3D Environment. (Jan. 2018).

[60] Wu, Z., Pan, S., Long, G., Jiang, J., Chang, X. and Zhang, C. 2020. Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks. (May 2020).

[61] Wu, Z., Pan, S., Long, G., Jiang, J. and Zhang, C. *Graph WaveNet for Deep Spatial-Temporal Graph Modeling*.

[62] Wu, Z., Pan, S., Long, G., Jiang, J. and Zhang, C. 2019. Graph WaveNet for Deep Spatial-Temporal Graph Modeling. (May 2019).

[63] Yan, C., Misra, D., Bennnett, A., Walsman, A., Bisk, Y. and Artzi, Y. 2018. CHALET: Cornell House Agent Learning Environment. (Jan. 2018).

[64] Yan, C., Misra, D., Bennnett, A., Walsman, A., Bisk, Y. and Artzi, Y. 2018. CHALET: Cornell House Agent Learning Environment. (Jan. 2018).

[65] Yang, S., Yu, X. and Zhou, Y. 2020. LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as an Example. *2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI)* (Jun. 2020), 98–101.

[66] Yu, B., Yin, H. and Zhu, Z. 2018. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence* (California, Jul. 2018), 3634–3640.

[67] Zeng, M., Mengshoel, O.J., Nguyen, L.T., Yu, B., Zhu, J., Wu, P. and Zhang, J. 2014. *Convolutional Neural Networks for human activity recognition using mobile sensors*.

[68] Zhou, B., Yang, J. and Li, Q. 2019. Smartphone-Based Activity Recognition for Indoor Localization Using a Convolutional Neural Network. *Sensors*. 19, 3 (Feb. 2019), 621. DOI:https://doi.org/10.3390/s19030621.

[69] Zuo, X., Yang, X., Dou, Z. and Wen, J.R. 2019. RUCIR at TREC 2019: Conversational Assistance Track. *28th Text REtrieval Conference, TREC 2019 - Proceedings* (2019).

Appendix:

Table 1 – Summary of publicly available HAR datasets

Name	Sensor Description	# Participants	# Activities	Activity Types	Environment Description	Datatype
CASAS SHiB [14]	14-30 sensors - Motion (Objects and Telephone sensors)	20	10 to 15	Bed-toilet transition, Cook, Eat, enter home, leave home, personal hygiene, phone, relax, sleep, work	3-bedroom apartment with a kitchen, living room and a bathroom) - also performed on >30 houses	Date, Time, SensorID, Value (bin. Or num)
CASAS testbeds [13]	20-86 sensors depending on testbed	1 to 3	11	Eating, grooming, cooking, drinking, taking medicine, etc.	7 different house layouts	Various sensor values
CASAS Aruba [13, 14]	34 sensors	1	12	Bed to toilet, eating, housekeeping, etc.	2-bedroom house with office	Binary Values
Opportunity [11]	Accelerometer, Gyroscope and magnetometer	4	17	Open and close door, open and close fridge, open and close dishwasher, open and close drawer, clean table, drink from cup, Toggle switch, Groom, prepare coffee, Drink coffee, prepare Sandwich, eat sandwich, Clean up	A room simulating a studio flat	Text file
Orange4Home [15]	236 Heterogeneous Sensors	1	20	Entering, leaving, preparing food, cooking, washing dishes, eating, watching tv, computing, using toilet, going upstairs, going downstairs, using the sink, dressing, reading, cleaning, etc.	87 square meter two-story home	Binary, integer, real number, categorical
ARAS [1]	20 Wireless Binary Sensors	4	27	Sleeping, brushing teeth, watching tv, eating …	2 houses	Binary values
VanKasteren [30]	14-21 sensors	1	8	Cooking, sleeping, and bathing	3 different houses	Binary values
Ordinez [37]	12 sensors	1	11	Open and close cupboards and doors, use of appliances, flushing toilet, walking	2 different houses	Binary values
MIT [52]	77-84 sensors	1	9 to 13	Meal preparation, sleeping, eating	2 houses	Binary and scalar values

Table 2 – Summary of current DNNs used in HAR

Source	DNN Approach	Dataset	Limitations	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Note
[34]	CNN	CASAS Milan, Cairo, Kyoto2, Kyoto3, Kyoto4	Could not capture long term temporal dependencies well, outperformed by LSTM structure	77.68	76.33	77.67	76.67	-
[45]	CNN	UCI-HAR	Small dataset used to capture only 6 different activities. Quite a limited testing of the model which would impact performance on larger datasets. Classification faulters on non-moving activities	94.79%	-	-	-	95.75% accuracy achieved when tested with additional handcrafted features
[42]	Random Search 1D-CNN	UCI-HAR	Random hyperparameter tuning employed and authors had to manually change the search range. Random aspect could include redundant features they found.	95.40	-	-	-	-
[34]	LSTM	CASAS Milan, Cairo, Kyoto2, Kyoto3, Kyoto4	Misclassified some activities consistently. E.g. Bed to toilet, and eating	93.42	93.67	93.67	93.33	Best performance metrics used
[53]	ConvAE-LSTM	WISDM, UCI, PAMAP2, OPPORTUNITY	Autoencoder and convolutional autoencoder have faster processing speeds	98.14	-	-	97.67	Metrics given for the UCI-HAR dataset tests

[4]	Bi-LSTM	PAMAP2	Some misclassifications are observed in the model between activities that share similar movement patterns. The model relies heavily on diverse and quality of training data.	99.27	99.00	99.00	99.00	-
[31]	CNN-GRU	UCI-HAR, OPPORTUNITY, mHealth	Generalizability of model to new datasets not covered. Complex model could be computationally expensive.	99.53	-	-	99.19	Model performed best on the mHealth dataset
[23]	SRU-GRU	mHealth	Model is more complex and took more computation time that traditional SRU models. Activities predicted were not complex and not in real time	99.80	-	-	99.60	-
[56]	SDAE	VanKasteren	Performs well in terms of time-slice accuracy but performs fairly poorly in terms of class accuracy. While it outperformed benchmark models in the study, the accuracy score is quite low compared to other DNN approaches to HAR	85.32	-	-	-	-
[19]	Transformer	UCI, MotionSense, HHAR, RealWorld, SHL Preview	While it outperformed other transformer-based approaches, the CNN-LSTM architecture still much faster for computation on mobile devices	-	-	-	94.59	Metrics reported on the combine datasets
[24]	Transformer with Convolutional Feature Extractor Block	KU-HAR, UniMiB SHAR, and USC-HAD	Base transformer model underperforms for activity recognition so was necessary to include a multi-layer convolutional network to improve predictions. Increasing the number of convolutional layers decreased classification performance	96.80	97.50	97.51	97.50	Results on the KU-HAR dataset reported here

Table 3 – Summary of STGNNs in Literature

Source	Year	Model	Application	Spatial Component	Temporal Component	Performance Metrics	Limitations	Notes
[62]	2019	Graph WaveNet (GWN)	Traffic Network Forecasting	GCN (Graph Convolution)	TCN (Temporal Convolution)	MAE, RMSE, MAPE	Model was faster than DCRNN but slower than STGCN. Model still assumes that some graph structure exists even if it is not the case (could lead to overfitting). Cannot learn dynamic spatial dependencies.	Employs a self-adaptive adjacency matrix to uncover hidden graph structure in data without prior knowledge. TCN extracts long term temporal dependencies from the data. This model achieved best results (lowest error) among all models tested on the traffic data.
[8]	2021	Spectral Temporal Graph Neural Network (StemGNN)	General multivariate time series forecasting	GFT (Graph Fourier Transform)	DFT (Discrete Fourier Transform) to 1D convolution to Gated Linear Unit to Inverse DFT	MAE, MAPE, RMSE	StemGNN is computationally expensive for large graphs. GFT and DFT can be slow for large and complex graphs	Predefined graph structure is not necessary, relationships are learnt automatically. Outperforms Graph WaveNet and ST-GCN
[60]	2020	MTGNN	Electricity, financial, traffic and energy	GCN implementing a mix-hop propagation	Dilated inception layer and gated	MAE, RMSE, MAPE, RRSE,	Does not handle for dynamic graphs, the learned graph is fixed in training. Learning the	Also includes a self-adaptive adjacency matrix so does not require defined graph structure

			demand forecasting (Based of benchmark datasets in paper)	later (for neighbour information)	temporal convolution	Empirical Correlation Coefficient (CORR)	adjacency matrix requires lots of computation.	like other graph neural networks. Performs to similar levels as the Graph WaveNet model. Performs consistently well over different datasets
[66]	2018	STGCN	Traffic Forecasting but can be applied to other general spatio-temporal learning problems	Chebychev GNN	1D-CNN – uses 1D Gated Temporal Convolutions	MAE, MAPE, RMSE	Dependent on defined graph structure. Fixed graph structure does not learn the adjacency matrix like Graph WaveNet.	Predefined graph structure used for this approach. Both Chebychev polynomial approximation and 1st order approximation used in this study. Outperforms statistical and DNN approaches like ARIMA and FC-LSTM
[58]	2018	Low rank weighted graph convolutional network (WGN)	Weather prediction	GCN	LSTM	Coefficient of determinatio n	Graph structure fixed due to locations of weather stations. Model cannot handle stations being added or removed. Low rank adjacency matrices can be difficult to interpret.	Dynamically learns adjacency matrices between weather stations. Outperforms baseline methods. Learning edge weights gave noticeable performance improvements. Low rank adjacency matrix is computationally more efficient but the sparse adjacency matrices are still found to be useful.

Table 4 – Publicly Available 3D simulation platforms

Simulator	Source	Release Year	Development Environment	API	Dataset Integration	Usability Information
Ai Habitat	[49]	2019	Habitat-Sim (custom simulation environment)	Provided: Python (Habitat API)	Users can use their own 3D datasets	Detailed documentation, tutorials, and testing information available online, test datasets available online.
AI2Thor	[32]	2017	Unity	Provided: Python	Prebuilt with the iTHOR, RoboTHOR, ProcTHOR and ArchitecTHOR datasets	Editor available for 3D visual output. All datasets are interactive with several 3D house environments
MINOS	[48]	2017	No development environment – source code available only	Provided: Python	Has access to the SUNCG dataset [51] and the Matterport3D [9] dataset for the 3D environments (reference these)	Source code available. Used for goal directed navigation in indoor environments. Manipulation for HAR required
House3D	[59]	2018	Source code available – renderer based on OpenGL	Not Provided	Based on the SUNCG dataset [51]	Source code available. Used to teach agents navigation in 3D environments. Manipulation for HAR required.
CHALET	[64]	2018	Unity	Provided	Little dataset integration – users can only upload actions from a saved file to be performed.	Manipulation of source code necessary for HAR purposes. Work needs to be done to implement agent actions from external datasets passed in as a file,
OpenSHS	[3]	2017	Blender	Provided: Python	Generates datasets for the user	Blender editor available. Can design floor plan, types of smart devices or sensors used, assign activity labels and simulate to generate a dataset from the 3D environment
SESim	[26]	2019	Unity	Not provided	User can define their own activity scenarios in XML format. Database manager available, could be used to integrate external datasets through alterations. Synthetic datasets can be created.	Design support through Unity editor. Participants in the study identified that it was easily extensible. However, alterations need to be made in order to integrate external datasets through database manager.
SIMACT	[6]	2012	Java Monkey Engine (JME)	Provided: Java	XML file can be used to load user defined scenarios – data can be	Pre-built and customizable scripts – 3rd party application can be

					extracted from the simulator's database	integrated with the simulators database
Francilliette et al.	[20]	2017	Unity	Not provided	Datasets generated by simulation and does not integrate with external datasets. Generated datasets could emulate public datasets as can manipulate the environment to match the public dataset's environment.	Scripting available, editor available with Unity. Can build your own environment. Sensors available to simulate real sensors. Datasets can be generated from the simulation for experimentation. Behaviour trees can be defined for activity. Needs to be adapted for HAR purposes.
VirtualHome	[40]	2018	Unity	Provided: Python	No direct integration available. Converting public datasets into natural language scripts (allowed by the simulator) could allow for these datasets to be integrated.	Activity scripts available. Source code available since it is open source which could be manipulated for the HAR task. Simulation can be modified. External dataset environments can be recreated in the simulator through JSON files for configuration.
VirtualSmartHome	[5]	2023	Unity	Built off VirtualHome above	External datasets are not directly integrated but their environments can be replicated when synthetic data is created. This synthetic data comes in the form of sensor logs that can be compared with the ground truth sensor logs from the datasets.	Digital twin of environment can be created in the smart home environment. Public datasets environments can be replicated and used as ground truth. Agent activities recorded in virtual sensor logs that can be used for the HAR task.