A Human Activity Recognition and Prediction System using simulated 3D Virtual Environments and Deep Neural Networks: A Review of the Literature

Temiloluwa Aina
Supervisor: Professor Deshen Moodley

ABSTRACT

This literature review synthesizes recent advancements in sensor-based human activity recognition (HAR) and prediction (HAP) by integrating deep neural network methodologies. It begins by contrasting traditional supervised learning approaches that rely on handcrafted features with modern deep learning techniques, such as convolutional neural networks (CNNs) and long short-term memory networks (LSTMs), which automatically extract high-level representations from raw sensor data. The review further explores the emergence of spatial-temporal graph neural networks (STGNNs) that explicitly model the complex spatial and temporal relationships among heterogeneous sensor inputs, which may aid in enhancing the recognition of intricate activity patterns in smart home environments. In addition to recognition, the work examines prediction models that forecast subsequent activities by leveraging sequential data. A novel aspect of this review is the exploration of how 3D simulation platforms can offer immersive, real-time visualizations of activity recognition model performance. The discussion identifies key challenges such as class imbalance and sensor variability and it outlines potential future directions for developing more robust, generalizable HAR and HAP systems.

1. INTRODUCTION

Sensor-based human activity recognition (HAR) makes use of data from sensors, like those in smart homes or wearable devices like smartwatches, to automatically detect and classify what a person is doing [1]. For example, a sensor in a smart home kitchen may pick up that a person has switched on their stove and classify the activity as “cooking”. The ability to recognise human activity accurately can assist in applications where understanding human activity is important. Thus, HAR has been used in a wide range of areas such as smart homes [2] where recognising the activities of daily living (ADL) can lead to energy savings and enhanced comfort, or in healthcare [3] where HAR enables the remote tracking of elderly individuals or patients with chronic conditions, allowing for prompt interventions.

Human Activity Prediction (HAP) takes it a step further, using previous data to predict what someone will do next. For example, if a person’s recent sensor data corresponds to “standing up from chair,” a HAP model might predict that the next activity will be “walking”. HAP can also be formulated as an “early prediction” problem where the goal is to anticipate an ongoing activity before it fully unfolds [4].

Sensor-based human activity recognition is generally approached as a supervised learning problem, where models learn to map sequences of sensor data to corresponding activity labels, such as “walking” or “cooking.” Early research predominantly employed traditional machine learning and statistical methods, like Decision Trees, Bayesian networks, and Multilayer Perceptrons, to address this challenge [5]. More recent studies have focused on training Deep Neural Networks, including Deep Fully Connected Networks, Convolutional Neural Networks, and Recurrent Neural Networks, for improved performance [6]. Additionally, Spatial-Temporal Graph Neural Networks (STGNNs) have emerged as a promising alternative due to their ability to model data with both spatial and time dependencies [7].

3D simulation environments enable intuitive visualization of HAR model performance by overlaying activity classifications onto lifelike avatars. This immersive approach allows users to quickly identify misclassifications and assess how these models capture dynamic transitions in human activity, transforming abstract performance metrics into clear, actionable insights.

2. Overview of Previous Work

2.1 The Supervised Learning Problem

As discussed earlier, human activity recognition (HAR) can be approached as a supervised learning problem, specifically, a multi-class classification problem. In this formulation, the model receives a sequence of sensor readings

$$S = \{S_1, \ldots, S_n\}$$

where each S_i is a t-dimensional (with t representing the sensor recording interval) vector representing the readings from the i^{th} sensor over time:

$$S_i = \{S_{i,0}, \ldots, S_{i,t}\}$$

$S_{i,k}$ is the sensor reading from sensor i at time k.

More concisely, the input takes the form:

$$S = \begin{pmatrix} S_{1,0} & S_{2,0} & \cdots & S_{n,0} \\ \vdots & \vdots & \ddots & \vdots \\ S_{1,t} & S_{2,t} & \cdots & S_{n,t} \end{pmatrix}$$

The task is to build a model $\mathcal{F}$ that can correctly classify a person’s activities from a set of activities $A = \{A_1, \ldots A_k\}$ where A_i could for example be “cooking”, given the sequence of sensor readings S:

$$\mathcal{F}(S) = \hat{A} = \{\hat{A}_0, \ldots, \hat{A}_t\}$$

where $\hat{A}_i$ is the model’s classification of the activity that took place at time i. In section 2.2 we discuss some of the deep neural network approaches that have been used to construct $\mathcal{F}$.

The objective is to learn the model $\mathcal{F}$ that minimizes the difference between the set of activities the model thinks occurred $\hat{A}$ and the set of activities that actually occurred A^* = $\{A_0, ..., A_t\}$ where A_i is the activity that actually took place at time i. This is done by minimizing a loss function $\mathcal{L}(\mathcal{F}(S)$, A*) where $\mathcal{L}$ quantifies the difference between

$\mathcal{F}(S) = \hat{A}$ and A* [8].

For multi-class classification, the loss function $\mathcal{L}$ typically used is cross-entropy loss.

While some HAR methods can be applied across different kinds of sensor data, most are designed for specific types. There are three different kinds of sensors [9]: body-worn sensors (sensors worn by users to capture body movements, e.g. smartwatches), object sensors (sensors attached to objects to monitor their movements, e.g. accelerometer attached to a cup), and ambient sensors (sensors integrated into the environment to capture user interactions, e.g. sound sensors) [8]. Our focus is on hybrid sensor data that blends different sensor modalities which are typically utilized in smart home environments.

2.2 Deep Neural Networks

The adoption of deep learning approaches to tackle HAR was necessitated by the limitations of the traditional machine learning models discussed earlier. These models relied on feature extraction processes, such as time-frequency transformations [10], statistical techniques [11], and symbolic representations [12] to extract meaningful patterns from the data. This means that these models' features were hand-crafted and heuristic, relying heavily on human expertise and domain knowledge [6]. Extracting features in this manner allows for only shallow features, such as the mean, variance, frequency and amplitude to be learned [13]. These models are only suitable for detecting low-level activities such as walking or running, and they struggle to identify high-level or context-aware activities, such as making coffee [14].

Deep Neural Networks eliminate the need for feature extraction as a pre-modelling step as these models inherently learn the features needed to distinguish between human activities. Furthermore, deep neural networks can derive high-level representations from their deeper layers, enhancing their suitability for tackling complex activity recognition tasks [8]. Below is a brief overview of different kinds of DNNs that have been applied to the HAR task:

2.2.1 Convolutional Neural Network

Convolutional Neural Networks (CNNs) are a class of DNN that has been widely used in HAR to automatically learn features from raw sensor data [3]. CNNs are feed-forward networks that apply convolution filters to detect local patterns. In HAR with ambient sensors, these models typically treat the sensor data as temporal sequences or images. CNNs naturally capture local dependencies (nearby sensor events) and are scale-invariant to time shifts, which is useful for recognizing activities that may start at different times [15]. Gochoo et al. [16] demonstrated a CNN approach by converting ambient sensor data sequences into binary images and applying a 2D CNN, showing that image-based CNN classifiers can successfully recognize activities. In an extension to this work [17], researchers even used color encoding to include sensor type information in the images. Other work has applied 1D CNNs directly on raw sensor timelines, for instance Singh et al. [18] used a 1D CNN on smart home sensor streams and achieved comparably high recognition accuracy, thanks to the CNNs' strong feature extraction capabilities. These studies found that CNN models can perform on par with or even better than recurrent models for many activities, efficiently learning features from raw ambient data. Overall, CNN-based HAR models have achieved high accuracy while requiring minimal manual feature design, making them a popular choice in recent years.

2.2.2 Long Short-Term Memory Networks

LSTMs are a type of Recurrent Neural Network (RNN) that include gating mechanisms (input, output, forget gates) to manage long-term information. This architecture allows LSTMs to retain important context over longer sequences. In HAR, LSTMs can learn the temporal dynamics of activities, for instance, remembering that after a fridge door opens, a stove sensor activation might indicate cooking. By gating the flow of information, LSTMs can decide to keep or forget sensor events as needed, capturing long-range dependencies more effectively than standard RNNs. LSTMs have been very successful in sensor based HAR. Liciotti et al. [19] explored various LSTM architectures on smart home datasets and showed that LSTM-based models significantly outperformed traditional HAR approaches like Hidden Markov Models and Conditional Random Fields, without requiring handcrafted features. Similarly, Sedky et al. [20] reported LSTM models surpassing classical classifiers like AdaBoost on ambient sensor data. Researchers have continued to refine LSTM models for HAR. Park et al. [21] introduced a deep LSTM network with residual connections and an attention mechanism to highlight important sensor events. The residual links helped combat vanishing gradients in very deep sequences, and attention allowed the model to focus on critical moments (e.g. a unique sensor trigger that signals a specific activity). Medina-Quero et al. [22] tackled the challenge of varying activity durations by combining LSTMs with adaptive "fuzzy" time windows, allowing the model to automatically adjust how far back in time to consider sensor data. These enhancements modestly improved accuracy (pushing toward ~95% on certain benchmarks), although truly complex or interleaved activities remain difficult. In summary, LSTMs are a cornerstone of recent HAR systems, providing robust sequential modeling that has boosted accuracy over earlier methods.

2.2.3 Transformers and Attention Mechanisms

More recently, attention-based models and Transformers have been explored for HAR. The self-attention mechanism allows modeling long-range dependencies more effectively than RNNs [23]. For instance, a Transformer can attend to relevant sensor readings across a window to decide the activity, potentially capturing relationships that CNNs or RNNs might miss (like a subtle pattern at the beginning of a window that correlates with another pattern at the end). Some studies applied Transformers to wearable sensor data [23], reporting improvements in recognizing complex activities.. While still emerging, Transformers for HAR show promise in handling multimodal inputs and longer sequences effectively.

Table 1 in the appendix is a tabular summary of recent studies on using deep learning techniques for human activity recognition. The studies show that deep learning models for activity recognition in smart home environments show impressive performance when they are finely tuned to specific datasets. Advanced architectures like CNN+LSTM with self-attention, graph neural networks, and Transformers enable these models to capture complex temporal and spatial patterns in sensor data. Techniques such as self-supervised learning and sensor embedding further boost performance, particularly when working with limited labeled data. However, many of these high-performing models often struggle to generalize across different environments. Their success tends to be confined to the particular conditions and sensor configurations of datasets like CASAS Aruba, Milan, or Kyoto (see details about these datasets in the Table 2), and many are designed for single-resident scenarios. Issues such as class imbalance, variability in sensor setups, and the need for extensive preprocessing and parameter tuning highlight the challenges of applying these models universally without additional adaptation or retraining.

2.3 Evaluation Metrics for Activity Recognition

As previously mentioned, HAR is typically evaluated as a multi-class classification problem. Accuracy (the proportion of correctly classified instances) is the most commonly reported metric, giving a quick overall performance measure. However, accuracy can be misleading if activity classes are imbalanced (e.g. “walking” may constitute a large fraction of data compared to rarer activities like “falling”). Therefore, metrics such as precision, recall and the F1 score are typically used as well [15]. These metrics are computed per class or averaged and are especially important in activity recognition when some classes are more important or rarer (e.g. falls). Precision is the fraction of predicted instances of a class that are actually that class, while recall (or sensitivity) is the fraction of actual instances of a class that the model correctly detected. The F1 score is the harmonic mean of precision and recall. Researchers often report macro-averaged F1 or weighted F1 to summarize performance across activities.

2.4 Typical Machine Learning Pipeline for Activity Recognition

HAR systems generally follow the following pipeline [8]:

1. **Data Acquisition:** Raw data is collected from sensors such as accelerometers, gyroscopes, smartwatches, smartphones, depth cameras, or ambient IoT sensors. This data is often multi-channel time series. Data acquisition also involves synchronization of multiple sensors and handling missing data or noise.

2. **Preprocessing and Segmentation:** In more traditional machine learning pipelines, the continuous sensor streams were segmented into intervals suitable for classification. This was typically done by using a sliding time window (e.g. 2-5 seconds long). Preprocessing also included filtering noise, normalizing sensor readings, or transforming axes. In deep learning pipelines, heavy manual feature extraction is minimized, but some signal processing steps like resampling or denoising might be applied.

3. **Feature Extraction (Automated by DNNs):** As previously discussed, traditional pipelines computed hand-crafted features (mean, variance, frequency-domain features, etc.) from each window. In deep learning approaches, the network itself takes raw or minimally processed input and learns internal feature representations [3].

4. **Training:** The core of the pipeline is the model training process. The model is trained on a portion of the dataset (training set) by minimizing a loss function (usually classification cross-entropy) using optimizers like Stochastic Gradient Descent (SGD) or Adam. Training involves multiple epochs over the data, with techniques like early stopping or validation to prevent overfitting.

5. **Evaluation:** The trained model is evaluated on held-out data using the metrics discussed above (accuracy, F1, etc.). Often k-fold cross-validation or leave-one-subject-out validation is used in HAR research to assess generalization to unseen data points [24]. The pipeline might be iteratively refined (tuning window size, network hyperparameters, etc.) to improve these metrics.

2.5 Human Activity Prediction

Du et al. [25] studied how to perform the task of Human Activity Prediction. In their paper, they define human activity prediction as a time sequence prediction problem, where the goal is to predict the next activity an inhabitant will perform based on their past activities. The authors believed that inhabitants performed different activities in relatively fixed patterns. For example, someone might always watch TV after having dinner. To model this, they utilize LSTMs because of these networks' ability to memorize both long and short-term knowledge, aligning with human behavior. The input to the LSTM model is the activity log, represented as a sequence of past activities. Their work mentions that the model considered not only the current activity but also several past activities to make a prediction. The output of the LSTM model was the prediction of the next activity. The paper also applied the LSTM method to model object-usage habits to predict the next object that might have been used. It then found the relevant activities associated with that object. Finally, the paper mentioned finding the intersection of the two prediction results (based on activity sequence and object usage) to further improve the overall prediction performance. In essence, their work framed human activity prediction as forecasting the subsequent action in a sequence of daily living activities, leveraging the temporal dependencies and patterns learned from historical activity and object usage data through LSTM networks.

3. Spatial Temporal Graph Neural Networks

Spatial-Temporal Graph Neural Networks (STGNNs) are a class of neural network models designed to handle data that is structured as a graph and evolves over time [7].

In an STGNN, the input is a sequence of graphs $G_1, G_2, \ldots, G_T$ where each graph is defined as $G_t = (V, E, X_t)$:

- V: The set of nodes.
- E: The edges connecting the nodes.
- X_t: The node feature values at time t.

The goal of an STGNN is to learn patterns along both the spatial dimension (the graph structure at each time point) and temporal dimension (the changes occurring over time). Fundamentally, STGNNs combine the techniques of Graph Neural Networks (GNNs), which perform computations over graph nodes and edges (such as graph convolution or message passing), with temporal sequence modeling methods, which can include recurrent units, temporal convolution, or attention mechanisms. A typical STGNN layer might consist of a graph convolution step (updating node representations based on neighbors at the same time step) followed by a temporal operation that propagates information forward to the next time step. For example, Yan et al.'s [26] Spatial Temporal Graph Convolution Network (ST-GCN) for skeletons applies graph convolution to capture how joints are connected in each pose and uses temporal convolution to track changes across poses. The effect is a deep model that can learn, for example, how the coordination of limb movements (spatial pattern) and the timing of those movements (temporal pattern) together characterize an action. More generally, STGNNs have been applied to problems like traffic flow forecasting (nodes are sensors on roads), where they excel at capturing both the physical network structure and time dynamics [7].

3.1 Differences between STGNNS and DNNs

Traditional deep neural networks for time-series (like the CNNs and RNNs discussed earlier) typically assume the input is an ordered sequence or a grid (image) [26]. They do not explicitly model relationships among sensors beyond what can be inferred from the data ordering. In contrast, STGNNs ingest structured relationships in the model architecture itself. For example, if we have 10 sensors in a smart home, a standard CNN might treat the 10-channel time series in a generic way, whereas an STGNN can have a graph where each sensor is a node and edges represent, for example, spatial proximity in the home. This means that STGNNs allow us to bake in domain knowledge of the system's topology. Traditional DNNs would need to learn any such structure implicitly, whereas STGNNs start with that structure, potentially making learning more efficient and interpretation more direct.

STGNNs also allow us to model non-Euclidean data [7]. Many deep learning models (CNNs especially) excel on Euclidean structured data (grids like images and sequences like 1D signals). Graphs are non-Euclidean (arbitrary connections). STGNNs are designed for that scenario, making them powerful for scenarios where data isn't naturally a grid, or the grid is too restrictive. For HAR, consider ambient sensors scattered in a home: they are not on a regular grid, but they do have a connectivity (rooms, doors). STGNNs can naturally model that, whereas a CNN treating the sensor array as a vector would ignore those connections.

A conventional RNN might learn temporal patterns but would treat all sensors as part of one flat input vector. It might not distinguish which sensor is where. In contrast, an STGNN could learn patterns like "Sensor A and B activating in quick succession indicates activity X" by virtue of its graph message passing. It can capture the interaction between sensors explicitly.

3.2 Benefits of STGNNs for Human Activity Recognition

In smart home HAR, often an activity involves interactions among objects and rooms (for example, "making tea" might trigger sensors in the kitchen: kettle sensor, cup cabinet sensor, fridge sensor if getting milk). Representing the home as a graph allows the model to learn, for example, certain sensors firing in sequence or simultaneously (spatially connected events) indicate a specific activity pattern. This is beneficial for complex multi-sensor environments. Early experiments have shown graph-based models outperform flat models in smart homes [27].

Graph Neural Networks (GNNs) also allow for model explainability in tasks like HAR [28]. Using a graph-based approach may make it easier to interpret what the model has learned by looking at which edges or nodes are influential. For example, Graph Attention Networks (a GNN variant) allows one to analyze attention weights on edges [29]. This could help explain HAR decisions ("the model focused on the relationship between the door sensor and motion sensor to recognize 'leaving home'").

3.3 Studies showing the use of STGNNs

Table 3 in the appendix shows recent studies that have explored the use of STGNNs in various domains.

The architectures in the table can be applied to object/ambient sensor-based human activity recognition (HAR) by representing the input sensor data as a graph network. This approach requires either developing a heuristic to determine sensor connectivity or integrating a component, similar to that proposed by Wu et al. [30], that automatically learns the graph structure.

4. 3D Virtual Environment

3D simulation platforms, like VirtualHome [45], create a detailed 3D virtual environment where household activities can be mimicked and analyzed. It accepts high-level instructions that define a sequence of actions, such as opening a door, interacting with objects, or performing daily routines, and then animates avatars to execute these tasks in a virtual home. 3D simulation platforms have been largely used to generate synthetic data that HAR models can be trained on [2].

In contrast, a digital twin is a dynamic, real-time virtual replica of a specific physical environment [2]. In the context of HAR, a digital twin mirrors the exact layout, sensor configurations, and even the behavioral nuances of a real home, continuously updated by live sensor data. For example, VirtualSmartHome [2] simulates a real smart apartment to minimize the "reality gap" between simulated and real sensor outputs, thereby enhancing the transferability of models trained on synthetic data to real-world applications.

Integrating a 3D simulation platform with a machine learning model for human activity recognition allows for a real-time overlay of classifications on an avatar executing various activities. This dynamic approach goes beyond static metrics like the F1 score by providing a visual, immersive assessment of model performance. It enables observers to see how the model interprets different actions in a realistic setting, making it easier to identify errors and assess its handling of subtle or overlapping behaviors. Ultimately, this visualization transforms abstract numerical metrics into tangible insights, thereby facilitating more effective model refinement and troubleshooting.

Table 4 in the appendix lists some of the prominent 3D simulation platforms in use today.

5. Discussions and Conclusions

This review has demonstrated that the integration of deep neural network architectures with sensor-based human activity recognition and prediction systems represents a significant leap forward from traditional machine learning approaches. By transitioning from handcrafted features to automated feature learning via deep learning models such as CNNs, LSTMs and Transformers, researchers have achieved impressive improvements in accurately classifying and forecasting human activities. These advancements are particularly critical in smart home and healthcare applications where timely and precise activity detection can have practical, life-enhancing implications.

The literature reveals that while CNNs and LSTMs provide robust methods for learning local dependencies and long-term sequential patterns, respectively, the emergence of STGNNs offers a promising alternative by explicitly modeling the spatial and temporal relationships among heterogeneous sensors. This capability is especially beneficial in complex environments where sensor placement and interconnectivity play crucial roles in activity recognition. Moreover, STGNNs can aid in model explainability by providing insights into which sensor interactions most significantly contribute to recognition performance, thereby offering a more interpretable framework for understanding model decisions.

The incorporation of simulated 3D virtual environments may open new avenues for immersive model evaluation and real-time performance visualization. These platforms may help provide tangible insights that can accelerate model refinement and troubleshooting.

Despite these advancements, challenges remain. Issues such as class imbalance, sensor variability, and limited generalizability across different environments continue to hinder the universal applicability of current systems. Addressing these challenges will require the development of more robust, adaptable models.

In summary, the emergence of advanced deep learning methodologies marks a significant milestone in HAR and HAP research. Future efforts should focus on enhancing model architectures while rigorously validating performance using both simulated and real-world data. This integrated approach promises to yield more reliable, generalizable, and interpretable systems that effectively support smart living and healthcare applications.

6. References

[1] S. Gupta. Deep learning based human activity recognition (HAR) using wearable sensor data. Int. J. Inf. Manag. Data Insights 2021, 1, 100046

[2] D. Bouchabou, J. Grosset., S.M. Nguyen, C. Lohr, and X. Puig, 2023. A smart home digital twin to support the

recognition of activities of daily living. *Sensors*, *23*(17), p.7586.

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

[4] I. E. Jaramillo, C. Chola, J.-G. Jeong, J.-H. Oh, H. Jung, J.-H. Lee, W. H. Lee, and T.-S. Kim. 2023. Human Activity Prediction Based on Forecasted IMU Activity Signals by Sequence-to-Sequence Deep Neural Networks. *Sensors* 23 (2023), 6491. https://doi.org/10.3390/s23146491

[5] O. D. Lara and M. A. Labrador. 2013. A Survey on Human Activity Recognition Using Wearable Sensors. *IEEE Communications Surveys & Tutorials* 15, 3 (2013), 1192–1209.

[6] K. Chen, D. Zhang, L. Yao, B. Guo, Z. Yu, and Y. Liu. 2018. Deep Learning for Sensor-Based Human Activity Recognition: Overview, Challenges and Opportunities. *J. ACM* 37, 4, Article 111 (Aug. 2018), 40 pages. https://doi.org/10.1145/1122445.1122456

[7] K. H. N. Bui, J. Cho, and H. Yi. 2022. Spatial-Temporal Graph Neural Network for Traffic Forecasting: An Overview and Open Research Issues. *Appl. Intell.* 52 (2022), 2763–2774. https://doi.org/10.1007/s10489-021-02587-w

[8] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu. 2017. Deep Learning for Sensor-Based Activity Recognition: A Survey. *Pattern Recognition Letters* 119 (2017). https://doi.org/10.1016/j.patrec.2018.02.010

[9] R. Chavarriaga, H. Sagha, A. Calatroni, S.T. Digumarti, G. Tröster, J.d.R. Millán, D. Roggen, The opportunity challenge: a benchmark database for on– body sensor-based activity recognition, Pattern Recognit. Lett. 34 (15) (2013) 2033–2042.

[10] T. Huynh and B. Schiele. 2005. Analyzing Features for Activity Recognition. In *Proceedings of the 2005 Joint Conference on Smart Objects and Ambient Intelligence: Innovative Context-Aware Services – Usages and Technologies*. ACM, 159–163.

[11] A. Bulling, U. Blanke, and B. Schiele. 2014. A Tutorial on Human Activity Recognition Using Body-Worn Inertial Sensors. *ACM Computing Surveys* 46, 3 (2014), 33.

[12] J. Lin, E. Keogh, S. Lonardi, and B. Chiu. 2003. A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. In *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*. ACM, 2–11.

[13] J. B. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy. 2015. Deep Convolutional Neural Networks on Multichannel Time Series for Human Activity Recognition. In *Proceedings of IJCAI*, Buenos Aires, Argentina, 25–31.

[14] Q. Yang. 2009. Activity Recognition: Linking Low-Level Sensors to High-Level Intelligence. In *Proceedings of IJCAI*, 20–25.

[15] D. Bouchabou, S. M. Nguyen, C. Lohr, B. LeDuc, and I. Kanellos. 2021. A Survey of Human Activity Recognition in Smart Homes Based on IoT Sensors Algorithms: Taxonomies, Challenges, and Opportunities with Deep Learning. *Sensors* 21, 18 (2021), 6037. https://doi.org/10.3390/s21186037

[16] M. Gochoo, T. H. Tan, S. H. Liu, F. R. Jean, F. S. Alnajjar, and S. C. Huang. 2018. Unobtrusive Activity Recognition of Elderly People Living Alone Using Anonymous Binary Sensors and DCNN. *IEEE J. Biomed. Health Inform.* 23 (2018), 693–702.

[17] T. H. Tan, M. Gochoo, S. C. Huang, Y. H. Liu, S. H. Liu, and Y. F. Huang. 2018. Multi-Resident Activity Recognition in a Smart Home Using RGB Activity Image and DCNN. *IEEE Sens. J.* 18 (2018), 9718–9727.

[18] D. Singh, E. Merdivan, S. Hanke, J. Kropf, M. Geist, and A. Holzinger. 2017. Convolutional and Recurrent Neural Networks for Activity Recognition in Smart Environments. In *Proceedings of Towards Integrative Machine Learning and Knowledge Extraction*. Springer, Berlin/Heidelberg, Germany, 194–205.

[19] D. Liciotti, M. Bernardini, L. Romeo, and E. Frontoni. 2020. A Sequential Deep Learning Application for Recognising Human Activities in Smart Homes. *Neurocomputing* 396 (2020), 501–513.

[20] M. Sedky, C. Howard, T. Alshammari, and N. Alshammari. 2018. Evaluating Machine Learning Techniques for Activity Classification in Smart Home Environments. *Int. J. Inf. Syst. Comput. Sci.* 12 (2018), 48–54.

[21] J. Park, K. Jang, and S. B. Yang. 2018. Deep Neural Networks for Activity Recognition with Multi-Sensor Data in a Smart Home. In *Proceedings of the 2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, Singapore, Feb. 5–8, 2018, 155–160.

[22] J. Medina-Quero, S. Zhang, C. Nugent, and M. Espinilla. 2018. Ensemble Classifier of Long Short-Term Memory with Fuzzy Temporal Windows on Binary Sensors for Activity Recognition. *Expert Syst. Appl.* 114 (2018), 441–453.

[23] Tao, S.; Goh, W.L.; Gao, Y. A Convolved Self-Attention Model for IMU-based Gait Detection and Human Activity Recognition. In Proceedings of the 2023 IEEE 5th International Conference on Artificial Intelligence Circuits and Systems (AICAS), Hangzhou, China, 11–13 June 2023; pp. 1–5.

[24] A. Jordao, A. C. Nazare Jr., J. Sena, and W. R. Schwartz. 2018. Human Activity Recognition Based on Wearable Sensor Data: A Standardization of the State-of-the-Art. arXiv:1806.05226 (2018).

[25] Y. Du, Y. Lim, and Y. Tan. 2019. A Novel Human Activity Recognition and Prediction in Smart Home Based on Interaction. *Sensors* 19, no. 20 (2019): 4474. https://doi.org/10.3390/s19204474

[26] S. Yan, Y. Xiong, and D. Lin. 2018. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI '18)*, 744–750.

[27] P. Srivatsa and T. Plötz. 2024. Using Graphs to Perform Effective Sensor-Based Human Activity Recognition in Smart Homes. *Sensors* 24, no. 12 (2024): 3944. https://doi.org/10.3390/s24123944

[28] M. Fiori, D. Mor, G. Civitarese, and C. Bettini. 2025. GNN-XAR: A Graph Neural Network for Explainable Activity Recognition in Smart Homes. arXiv:2502.17999 (2025). https://doi.org/10.48550/arXiv.2502.17999

[29] J. Tang, L. Xia, and C. Huang. 2023. Explainable Spatio-Temporal Graph Neural Networks. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23). Association for Computing Machinery, New York, NY, USA, 2432–2441. https://doi.org/10.1145/3583780.3614871.

[30] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang. 2020. Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks. In *Proceedings of the 26th International Conference on Knowledge Discovery & Data Mining (KDD 2020)*, ACM, 753–763.

[31] H. Chen, C. Gouin-Vallerand, K. Bouchard, S. Gaboury, M. Couture, N. Bier, and S. Giroux. 2024. Enhancing Human Activity Recognition in Smart Homes with Self-Supervised Learning and Self-Attention. *Sensors* 24 (2024), 884. https://doi.org/10.3390/s24030884

[32] D. Bouchabou, S. M. Nguyen, C. Lohr, I. Kanellos, and B. Leduc. 2021. Fully Convolutional Network Bootstrapped by Word Encoding and Embedding for Activity Recognition in Smart Homes. In *Proceedings of DL-HAR 2021: 2nd International Workshop on Deep Learning for Human Activity Recognition*, Yokohama, Japan, Jan. 2021, 111–125. https://doi.org/10.1007/978-981-16-0575-8_9

[33] R. G. Ramos, J. D. Domingo, E. Zalama, and J. Gómez-García-Bermejo. 2021. Daily Human Activity Recognition Using Non-Intrusive Sensors. *Sensors* 21 (2021), 5270. https://doi.org/10.3390/s21165270

[34] X. Huang, S. Zhang. 2023 Human Activity Recognition based on Transformer in Smart Home. *Proceedings of the 2023 2nd Asia Conference on Algorithms, Computing and Machine Learning*.

[35] R. A. Hamad, L. Yang, W. L. Woo, B. Wei. 2020. Joint Learning of Temporal Models to Handle Imbalanced Data for Human Activity Recognition. *Applied Sciences*, *10*(15), 5293. https://doi.org/10.3390/app10155293

[36] D. Cook. 2011. Learning setting-generalized activity models for smart spaces. IEEE Intelligent Systems.

[37] H. Alemdar, H. Ertan, O. D. Incel and C. Ersoy. 2013. "ARAS human activity datasets in multiple homes with multiple residents," *2013 7th International Conference on Pervasive Computing Technologies for Healthcare and Workshops*, Venice, Italy, pp. 232-235.

[38] Ordoñez, F.J. 2013. *Activities of Daily Living (ADLs) Recognition Using Binary Sensors [Dataset]*. UCI Machine Learning Repository. https://doi.org/10.24432/C5J02M

[39] D. Cook, M. Schmitter-Edgecombe. 2009. Assessing the quality of activities in a smart environment. Methods Inf Med. 2009;48(5):480-5. doi: 10.3414/ME0592. Epub 2009 May 15. PMID: 19448886; PMCID: PMC2759863.

[40] van Kasteren, T. L. M., Noulas, A., Englebienne, G., & Kröse, B. J. A. 2008. Accurate activity recognition in a home setting. In *Proceedings of the 10th International Conference on Ubiquitous Computing* (pp. 1–9). ACM. https://doi.org/10.1145/1409635.1409637

[41] J. Ye, H. Jiang, and J. Zhong. 2023. A Graph-Attention-Based Method for Single-Resident Daily Activity Recognition in Smart Homes. *Sensors* 23 (2023), 1626. https://doi.org/10.3390/s23031626

[42] Y. Mao, G. Zhang, and C. Ye. 2024. A Spatio-Temporal Graph Transformer Driven Model for Recognizing Fine-Grained Human Activity. *Alexandria Engineering Journal* 104 (2024), 31–45.

[43] L. Pan, J. Lu, and X. Tang. 2024. Spatial-Temporal Graph Neural ODE Networks for Skeleton-Based Action Recognition. *Scientific Reports* 14 (2024), 7629. https://doi.org/10.1038/s41598-024-58190-9

[44] B. Yu, H. Yin, and Z. Zhu. 2018. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI 2018)*, 3634–3640.

[45] X. Puig, K. Ra, M. Boben, J. Li, T. Wang, S. Fidler, and A. Torralba. 2018. Virtualhome: Simulating household activities via programs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '18)*, 8494–8502.

[46] N. Alshammari, T. Alshammari, M. Sedky, J. Champion, C. Bauer. 2017. OpenSHS: Open smart home simulator. Sensors 2017, 17, 1003.

[47] K. Bouchard, A. Ajroud, B. Bouchard, and A. Bouzouane. 2012. SIMACT: A 3D open source smart home simulator for activity recognition with open database and visual editor. *Int. J. Hybrid Inf. Technol.* 5 (2012), 13–32.

[48] J.W. Lee, S. Cho, S. Liu, K. Cho, and S. Helal. 2015. Persim 3D: Context-driven simulation and modeling of human activities in smart spaces. *IEEE Trans. Autom. Sci. Eng.* 12 (2015), 1243–1256.

[49] Y. Francillette, E. Boucher, A. Bouzouane, and S. Gaboury. 2017. The virtual environment for rapid prototyping of the intelligent environment. *Sensors* 17 (2017), 2562.

7. Appendix

Table 1: Summary of recent studies of DNNs in HAR

Authors	**Deep Learning Method**	**Activities Predicted**	**Dataset**	**Strengths**	**Limitations**	**F1 Score**
Hui Chen et al. [31]	CNN + LSTM with self-attention (rooted in the self-supervised learning framework SimCLR)	10 ADLs (e.g. meal prep, sleeping, bathing, etc.)	CASAS Aruba-1, Aruba-2 & Milan	Leverages unlabeled data via contrastive pre-training; excels with limited labels and transfers well to new homes.	Struggles with class imbalance (rare activities); focuses on single-resident data (not multi-resident).	95–97% (with only 20–40% labeled data) (performance varied across datasets used)
Prateek Srivats and Thomas Ploetz [27]	Graph Neural Network with attention (learns sensor–sensor relations)	11–27 daily activities (CASAS smart home ADLs like cooking, eating, toileting)	CASAS Kyoto, Aruba, Cairo, Milan	Captures spatiotemporal sensor correlations using graph attention; robust to missing sensor data and no fixed window needed	Higher model complexity and not explicitly tested for multi-resident differentiation (treats home as single graph).	~78.3% - 92.4% (performance varied across datasets used)
Daniele Liciotti et al. [19]	Sequential LSTM models (uni-LSTM, bi-LSTM, ensemble of LSTMs)	9–15 ADLs (e.g. cooking, eating, housekeeping, etc.)	CASAS Milan, Cairo, Kyoto homes	LSTM captures temporal patterns without hand-crafted features, outperforming earlier ML approaches. Ensemble of LSTMs improves robustness	No single LSTM model emerged as the best performer across all CASAS datasets. Factors such as the number of residents, sensor types, etc. influence performance, highlighting challenges when applying the approach to varied real life environments.	74.33% - 94% (performance varied across datasets and models used)
Damien Bouchabou et al. [32]	Fully Convolutional Network with sensor embedding (FCN + word2vec encoding of sensors)	12–16 ADLs (incl. “Other” for unlabeled periods)	CASAS Aruba & Milan	End-to-end learns sensor embeddings to add context (treats sensors like “words”); achieves state-of-art accuracy, even with “Other” class present.	Including “Other” (no activity) class still challenges precision; method tested on single-resident scenarios	≈99% (near-perfect on Aruba; ~98% on Milan).
Raúl Gómez-Ramos et al. [33]	Bidirectional LSTM (real-time sliding window)	11–15 usual daily activities (medication, eating, sleeping, etc.)	CASAS Milan	Real-time recognition with minimal delay; careful preprocessing (windowing, stacking events, time-of-day features) yields high accuracy	Limited to single-resident data; requires extensive data cleaning and tuning (sliding window size, regularization) for best results.	95% (Bi-LSTM model F1≈0.95 on test set).
Xinmei Huang, Shenmin Zhang [34]	Transformer model (self-attention on sensor event sequence)	10 ADLs (common daily tasks in home)	CASAS Aruba	Captures long-range dependencies in sensor sequences via self-attention; achieved perfect precision/recall on majority of activities.	Requires large training data to avoid overfitting (Transformer is data-hungry); some infrequent activities still had lower f1 scores due to few samples.	~96% (6 of 10 activities reached F1=100%).
Rebeen A. Hamad et al. [35]	Joint learning using LSTM and 1D CNN	9–15 ADLs (with class imbalance: rare vs. frequent activities)	Ordonez ADL and van Kasteren Datasets	Introduces a combined RNN-CNN approach to address class imbalance, boosting detection of infrequent activities without sacrificing overall accuracy.	Although the approach improved detection of infrequent classes (compared to other models), it still performed poorly on these classes	64.46% - 76.51% (Performance varied across datasets used)

Table 2: Publicly available datasets used for HAR model training

Dataset	Activity Classes	Sensors Used	Participants	Notes
CASAS Aruba (WSU) [36]	11 daily activities (e.g., Sleeping, Eating, Bed-to-Toilet, Leave Home)	31 PIR motion sensors, 3 door-open sensors, 5 temperature sensors, 3 light sensors	1 resident (single older adult)	~220 days of annotated data. Publicly available and widely cited benchmark for ambient ADL recognition.
ARAS (Ambient Recognition of ADLs) [37]	27 activities of daily living (e.g., various cooking, hygiene, work, leisure, etc.)	20 binary sensors (motion, door/cabinet contacts, pressure mats, etc.) in each home	2 houses, each with 2 residents (multi-occupant)	Collected over ~2 months (1 month per house). Public multi-resident dataset from Boğaziçi Univ., Turkey. Often used to test multi-occupancy activity recognition algorithms.
Ordonez ADL (UCI) (Ordóñez et al., 2013) [38]	10 activities (e.g., Leaving, Toileting, Showering, Sleeping, Breakfast, Lunch, Dinner, Snack, TV, Grooming)	12 wireless binary sensors: PIR motion (in shower, kitchen), magnetic door sensors (entry door, fridge, cabinets), pressure mats (bed, chair), flush sensor (toilet), appliance use (microwave, toaster	2 residents (in separate single-person homes; 14 and 21 days data)	Publicly available UCI *Activities of Daily Living Using Binary Sensors* dataset. Widely used for evaluating HAR algorithms on binary ambient sensor data.
CASAS Tulum (WSU) [36]	14 activities (e.g., Bathing, Eating, Bed-Toilet transition, Watching TV, Working (at table/living room), Entering/Leaving Home, etc.)	36 sensors: 31 PIR motion sensors + 5 temperature sensors in a two-bedroom apartment	2 residents (an older married couple; multi-resident home)	~98 days of annotated data. Publicly released for multi-resident ADL recognition. Often used to evaluate multi-user activity clustering and recognition algorithms.
CASAS Cairo (WSU) [36]	13 activities (e.g., Bed-to-Toilet, Breakfast, Night Wandering, Medication, Sleep for Resident1 /Resident2, etc.)	32 sensors: 27 PIR motion sensors + 5 temperature sensors in the apartment	2 residents (an elderly pair) + 1 pet (cat)	~57 days recorded. Multi-resident activity data, includes some unique events (e.g., night wandering). Publicly available; frequently used for evaluating algorithms that differentiate two inhabitants' activities.
CASAS Milan (WSU) [39]	~15 daily living activities (e.g., Sleeping, Taking Medicine, Meal preparation, Leaving home, etc.)	~30 ambient sensors installed in a smart home (primarily PIR motion detectors throughout rooms, plus a front door contact sensor and a few ambient temperature sensors)	Single resident (an elderly female living alone) + one pet dog; occasional short visits by family members (multi-occupant only during those visits)	Natural daily life dataset collected in a real home over ~3 months (≈92 days). Contains ≈433,656 sensor events with 2,310 annotated activity instances. Often used to study activity patterns with interleaved tasks and the impact of sporadic multi-resident interactions (visitors).
CASAS Kyoto (WSU) [39]	5 scripted ADLs performed in trials: Making phone calls, Washing hands, Cooking, Eating, and Cleaning. Each volunteer carried out these five activities in the smart home environment.	About 39 sensors used including motion sensors, item/object usage sensors on appliances and cabinets (e.g., kitchen utensil sensors, medicine cabinet sensor), a water flow sensor, a stove burner ignition sensor, and a telephone use sensor.	20 volunteers (each acting as a single resident during their session). Participants performed the activities one at a time in the testbed (each trial involves one person in the home).	A controlled experiment dataset from the CASAS project. Each participant's session was recorded in the instrumented apartment; combined data covers 86 days of trials (split into 120 files). All activities are annotated (ground truth provided) and the data is publicly available. Often used for benchmarking activity recognition algorithms under single-user, non-overlapping scenarios (no pets or additional residents present during trials).
van Kasteren Dataset [40]	7 Activities of Daily Living: Leave House, Toileting, Showering, Sleeping, Preparing Breakfast, Preparing Dinner, Preparing a Beverage (with "Idle" periods when none of these activities occur).	14 binary state-change sensors (wireless) attached to household objects that change state when used by the resident. These include door contact sensors (e.g. front door, fridge, cabinets), a toilet flush sensor, etc. (No wearable sensors; all are ambient binary sensors.)	Single-resident homes. The original dataset was collected in one apartment with a 26-year-old male living alone.	Collected over 28 days. Annotations total 245 activity instances across ~2,120 sensor events. The data (provided in raw sensor event format) is publicly available and has become a common benchmark dataset for evaluating activity recognition methods. All scenarios involve one person (no pets, no concurrent multi-residents). The dataset was later expanded to an additional single-occupant home.

Table 3: Recent studies exploring the use of STGNNs

Model	**Description**	**Original Application**
TLGAT (Time- & Location-oriented Graph Attention Network) [41]	Graph Attention Network that learns time-oriented and location-oriented dependencies between smart-home sensors. It treats a sliding window of sensor events as a fully connected graph and uses dual attention modules (temporal and spatial) to capture correlations in irregular sensor event sequences. Significantly improves activity recognition by leveraging sensor layout and time-of-day patterns.	Single-resident smart home activity recognition (ambient sensors in a smart home)
STGT (Spatial-Temporal Graph Transformer) [42]	A graph-transformer hybrid model that fuses Graph Attention with Transformer-style sequence modeling. It organizes sensor data into a fine-grained graph structure and uses a transformer encoder–decoder to jointly learn non-Euclidean spatial relationships and long-term temporal dependencies.	Sensor based HAR using wearable sensors
STG-NODE (Spatio-Temporal Graph Neural ODE) [43]	An architecture that integrates graph neural nets with continuous-time ordinary differential equation (ODE) modeling. It constructs a graph of entities (e.g., body joints or sensors) and uses a neural ODE solver to model the temporal evolution of node features.	Skeleton-based human action recognition (graphs of human body joints over time)
Fully Graph Convolutional Network [44]	The model is built from spatial-temporal blocks (blocks that handle both space and time). Each block uses a graph convolution (to pick up spatial patterns, like relationships between different points) and two 1D convolutions (to learn how sequences change over time). Instead of using traditional recurrent neural networks for time, it uses a type of 1D convolution that respects the order of time (causal convolution) along with a mechanism called gated linear units to manage the flow of information.	Forecasting the flow of traffic using sensor data
Spatial-temporal graph structure learning [30]	This model automatically extracts uni-directed relations among variables through a graph learning module addressing the challenge of unknown graph structures. It consists of a graph learning module, graph convolutional module (using the adjacency matrix learnt) and a temporal convolution module.	This model was used in the domains of traffic, solar energy, electricity consumption, and financial exchange rates forecasting.

Table 4: 3D Simulation Platforms for HAR

Platform	Open Source	Dataset Integration	Application Focus	3D Environment	API / Interface	Replay Existing Data?
VirtualHome [45]	Yes	Generates simulation data. Not originally designed to ingest sensor data	General household tasks and embodied AI for computer vision and task-learning	Unity3D game engine	Python API for scripting high-level actions, with options for first-person mode control	Not natively – requires extensions (e.g., VirtualSmartHome) for sensor-based replay)
OpenSHS [46]	Yes	Primarily for synthetic dataset generation. Users design a virtual home and simulate ambient sensor events. Not designed to directly import external datasets	Smart home HAR research.	Blender Game Engine (OpenGL)	Interactive first-person avatar control and script-based simulation (no separate external API required)	No – intended for generating new data rather than replaying existing logs
SIMACT [47]	Yes	Uses predefined scenarios to generate sensor events. Custom scenarios defined via XML. No direct plug-and-play for external real-world datasets	Synthetic smart home activity simulation for HAR	JMonkeyEngine (custom 3D engine in Java)	XML-based scenario scripts and a Java API	Not natively – geared toward scripted activity playback. External dataset replay requires converting data into scenario format.
Persim-3D [48]	No	Generates synthetic sensor sequences validated against real smart-home data. Does not ingest external logs	Context-aware ADL simulation for smart homes.	Unity3D engine	C# (Unity) implementation. Activities configured via code/internal editor (no external API reported)	No – designed to generate new sensor datasets rather than replaying existing logs
Francillette et al. [49]	Yes	Generates synthetic sensor data which is logged to a database. No support for pre-existing data to be ingested.	Human Activity Recognition Research	Unity3D engine	Database-driven API for external interaction. The simulator uses SQLite for both logging sensor data and for actuator control: one database stores sensor events, and another is used as a channel for external programs to send commands to virtual actuators. This means an external application or AI agent can read the sensor outputs and inject actuator actions (e.g. turning devices on/off) by updating the database, and the simulation will reflect those changes.	No – designed for synthetic data generation