Deep Learning for Human Activity Recognition and Prediction

Andrew Erasmus
University of Cape Town
ersand012@myuct.ac.za

Kishalan Pather
University of Cape Town
pthkis001@myuct.ac.za

Temiloluwa Aina
University of Cape Town
anxtem001@myuct.ac.za

1. Abstract

Human Activity Recognition (HAR) and Human Activity Prediction (HAP) are key tasks in domains ranging from smart homes to healthcare. This project explores the effectiveness of deep learning models, including conventional Deep Neural Networks (DNNs) and advanced Spatial-Temporal Graph Neural Networks (STGNNs), for sensor-based HAR and HAP using publicly available human activity datasets. We begin by establishing baselines with Convolutional Neural Networks and Long Short-Term Memory networks. These will be compared to a baseline and three state-of-the-art STGNN models, which leverage spatial–temporal relationships among sensors. Performance will be evaluated both quantitatively and qualitatively, the latter using network analysis on learned adjacency matrices. To assess model trustworthiness and interpretability, predictions will be visualized in a Unity3D-based environment where activity outputs are overlaid on lifelike avatars. Through comparative analysis, we aim to show that STGNNs offer superior recognition and prediction accuracy along with richer interpretability. The expected outcome is a robust, generalizable machine learning pipeline and an interactive 3D evaluation tool for more explainable HAR/HAP applications.

2. Project Description

2.1 Introduction

Sensor-based HAR uses data from sensors to detect and classify what a person is doing [1]. For example, a sensor in a kitchen may detect stove use and classify the activity as "cooking". Accurate recognition is essential in domains where understanding human behavior is valuable. 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.

HAP uses 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 HAR 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 and Bayesian networks, to address this challenge [5]. More recent studies have focused on training DNNs, such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory Networks (LSTMs), 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].

STGNNs are special types of neural networks for time-evolving, graph-structured data [7], widely used in sensor-based time series tasks like weather and traffic forecasting [7, 8]. By jointly learning spatial and temporal dependencies, they yield more robust performance than traditional DNNs [9]. Unlike CNNs, which excel on Euclidean grids (images, sequences), STGNNs natively handle non-Euclidean graphs [7], making them ideal when data is not on a regular grid. For HAR, ambient home sensors form a connectivity graph that STGNNs capture naturally, whereas a CNN would ignore those relationships. Moreover, the graph structure learned can be examined to uncover the underlying dynamics and key pathways of human activity.

3D environments enable intuitive visualization of human activity [10] and thus can be used to gauge whether HAR and HAP models produce sensible outputs. By overlaying activity classifications onto lifelike avatars one can gain a clearer view of model performance in different scenarios. This visualization could allow one to quickly identify misclassifications and assess how HAR and HAP models capture dynamic transitions in human activity, transforming quantitative performance metrics into clear, actionable insights.

2.2 Project Overview

This project will systematically evaluate cutting-edge techniques for HAR and HAP. First, we will develop baseline Convolutional Neural Network and Long Short-Term Memory models to establish performance benchmarks. To visually assess model performance, we will create a visualization of the sensor data in a 3D environment, superimposing activity classifications and future predictions onto lifelike avatars. We will then train a baseline STGNN alongside 3 state-of-the-art STGNN variants, comparing their results to the CNN and LSTM baselines using quantitative metrics. Finally, we will conduct a qualitative analysis of the graph structures each STGNN learns, comparing the advanced models against the baseline STGNN, to reveal how they capture the underlying dynamics of human activity.

3. Problem Statement and Research Objectives

3.1 Problem Statement

Despite significant advances in deep learning for sensor-based HAR, the complementary task of HAP remains relatively understudied. Most existing work concentrates on recognition using CNNs or LSTMs, with only a handful of studies exploring prediction and even fewer leveraging architectures that jointly model spatial and temporal dependencies. STGNNs have demonstrated superior performance on non-Euclidean, time-evolving data, yet there has been a lack of research to systematically evaluate their robustness and generalizability in this context.

Traditionally, the evaluation of HAR and HAP models has relied heavily on quantitative metrics, which offer limited insight into the spatial-temporal patterns these models learn. Moreover, there has been little exploration of how 3D environments can serve as model-checking tools to assess the explainability, interpretability, and trustworthiness of predictions. While metrics like precision and recall provide a broad summary of performance, they fall short in revealing whether a model performs reliably across diverse real-world scenarios. By projecting model outputs onto a lifelike avatar within a 3D environment, we can visually assess whether the learned spatial-temporal dynamics align with actual human activity. In essence, 3D visualization enables a deeper, more intuitive understanding of model behavior and helps determine whether the classifications are reasonable and trustworthy.

3.2 Research Objectives

1. Implement, train, and quantitatively evaluate and compare baseline Deep Neural Network and Spatial-Temporal Graph Neural Network models on selected CASAS datasets.
2. Implement a 3D visualization environment for the selected CASAS datasets.
3. Perform a qualitative evaluation of the baseline DNN and STGNN models by: (1) performing visual analysis (using the 3D environment) to assess the trustworthiness, interpretability, and explainability of the model outputs, and (2) analyzing the graph structure learned by the baseline STGNN through network analysis.
4. Implement, evaluate, and compare advanced STGNN models to the baseline DNNs and the baseline STGNN model through quantitative measures and visual analysis (using the 3D environment).
5. Analyse the advanced STGNNs by applying network analysis techniques to their learned adjacency matrices, in order to uncover the dynamics of human activity in the selected CASAS datasets.

4. Procedures and Methods

4.1 The Supervised Learning Problem

HAR is framed as a supervised multi-class classification problem, where the model receives a sequence of sensor readings $S = \{S_1, \ldots, S_n\}$ with each $S_i = \{S_{i,0}, \ldots, S_{i,t}\}$ representing the time-series data from sensor i from time = 0 to time = t. 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 goal is to learn a function $\mathcal{F}(S) = \hat{A} = \{\hat{A}_0, \ldots, \hat{A}_t\}$, where $\hat{A}_i$ denotes the model's classification of the activity that took place at time i., from a set of possible activities $A = \{A_1, \ldots, A_k\}$ (e.g., "cooking").

The model $\mathcal{F}$ is trained to minimize the loss $\mathcal{L}(\mathcal{F}(S), A^*)$, where $A^* = \{A_0, \ldots, A_t\}$ is the ground truth activity sequence, and $\mathcal{L}$ measures the discrepancy between the model's classifications and the actual activities.

HAP focuses on forecasting future activities based on recent sensor patterns. The input remains the recent sequence of sensor readings $S = \{S_1, \ldots, S_n\}$, and the output is a predicted sequence of future activities $\hat{A}_{future} = \{\hat{A}_{t+1}, \ldots, \hat{A}_{t+h}\}\}$, where h is the prediction horizon.

Our research will train and evaluate CNN, LSTM, and STGNNs to learn the most accurate function $\mathcal{F}$ for classifying and predicting human activities from sensor data.

In our search for the most accurate $\mathcal{F}$, we will carry out the following components:

4.2 Data Collection

To train and optimize models for the HAR task, we require large amounts of publicly available sensor-based human activity data. For this, we intend to use the CASAS datasets from the Washington State University smart home project [11]. The CASAS datasets include a collection of testbeds which each contain a diverse range of Activities of Daily Living (ADL). Each dataset includes sensor data files, a layout of the environment used, and documentation describing the data provided. The motivation for using this dataset is that it is well documented, extensively used, diverse and readily available.

4.3 Exploratory Data Analysis (EDA):

We will perform an exploratory data analysis (EDA) to establish a statistical baseline of the human activity patterns in the chosen datasets. Specifically, we will:

1. Identify dominant activities by computing and visualizing activity frequency distributions (e.g., bar charts).
2. Construct a transition matrix to capture empirical activity-state transition probabilities and highlight common sequences of behavior.
3. Rank sensors by activation frequency to identify the most active devices.
4. Assess inter-sensor relationships through pairwise statistical correlations, visualized as a heatmap.

This EDA will provide insights into activity patterns and sensor interactions, serving as a baseline to determine what additional information the STGNNs were able to uncover.

4.4 Preprocessing:

We have identified that there are a variety of different sensors and activities used in the different testbeds in the CASAS datasets. To avoid abnormal model behaviour it is crucial that we perform preprocessing on this data. This includes:

- Filtering anomalies that could interfere with prediction results
- Ensuring our data has no missing values
- Reducing noise
- Categorizing our data into their respective sensor sources.
- Segmenting the data into the training, validation and test splits as the validation set is crucial for hyperparameter tuning.

For training, we will then need to format our datasets to be suitable for the specific model being used.

4.5 Baseline Model Development & Training:

We will establish DNN baselines using selected CNN and LSTM architectures, which have demonstrated strong performance on

HAR tasks in prior studies. Additionally, we will include the STGNN-based Graph WaveNet [12] as a baseline for later comparison. All models will be trained on pre-processed training data from the CASAS datasets.

To optimize performance, we will conduct hyperparameter tuning (e.g., learning rate, layer depth, and units per layer) to minimize misclassification error on the validation set. To mitigate overfitting and ensure robust model selection, we will use k-fold cross-validation.

Misclassification error will be evaluated using the cross-entropy loss function, consistent with prior work such as Jaramillo et al. [4].

4.6 Model evaluation through quantitative metrics:

We will evaluate the performance of our baseline models quantitatively using the following metrics:

- **Accuracy:** The proportion of correctly classified instances. This metric provides an overall performance measure. However, this metric is misleading when the activity classes are imbalanced (as is the case with activity recognition). Thus, we will also report the following metrics:
- **Precision:** Per-class precision is measured as the number of true positives for a particular activity class *a* (instances when an activity *a* was correctly classified) divided by the total number of instances classified as *a* (true positives plus false positives for *a*). To summarize model-wide performance, one can average these per-class precisions either by taking their unweighted mean (macro-precision) or by pooling the true and false positive across all classes before calculating the ratio (micro-precision).
- **Recall:** Per-class recall for an activity class *a* is measured as the number of true positives for *a* divided by the total number of actual instances of *a* (true positives plus false negatives for *a*). As with precision, to assess overall performance you can aggregate these per-class recalls by calculating the macro- and micro-recall.
- **F1-Score:** The harmonic mean of precision and recall which aims to balance the two.

4.7 3D Virtual Environment:

Our chosen 3D environment platform is VirtualHome [13]. It utilizes the Unity3D game engine to create a realistic 3D environment of a person in a smart home. It allows for a wide range of activities such as picking up items, making coffee, etc., and its Python API enables simple scripting of these activities. In this component of the project, we will begin scripting scenarios (replicating our sensor data) in VirtualHome. We will then modify the VirtualHome source code to allow for our model's classification of activities to be overlayed on the avatar within the 3D environment.

To help evaluate the model more effectively, we will add an annotation step to mark and label cases where the model gets things wrong. These failures will be saved and organized in the 3D environment so we can easily find and study them. This will help us spot patterns in the model's mistakes and improve its performance over time.

The 3D environment will be used to evaluate the trustworthiness of the baseline models' outputs. By visually tracing each classification and prediction, we can gain clearer insights into the models' performance across various scenarios. This will help assess whether model outputs are trustworthy in all situations.

4.8 STGNN Model Training and Optimization

Each member will champion an a state-of-the-art STGNN. The selected models include:

1. Dynamic Spatial-Temporal Aware Graph Neural Network (DSTAGNN) [14]: A recent architecture that has gained some popularity. This model makes use of a Spatial-Temporal Aware Graph formed directly from historical data. This captures the degree of spatial association among nodes dynamically, without a predefined static adjacency matrix. Uses an attention mechanism and multi-receptive field gated convolution, suited for the complex spatial-temporal dependencies in HAR/HAP.
2. Adaptive Graph Convolutional Recurrent Network [15] (AGCRN): Originally designed for traffic forecasting but has potential in the HAR/HAP space. AGCRN builds an adaptive adjacency matrix compared to traditional graph convolution networks that use a fixed adjacency matrix. This allows the model to learn relationships between locations based on the data itself.
3. Attention Based Spatial-Temporal Graph Convolutional Networks (ASTGCN) [16]: It is a relatively recent and well-established model that has also been used in traffic forecasting which is similar to sensor-based data. The attention modules allow the model to focus on the most relevant sensors and time steps to improve recognition accuracy.

All selected models will be trained for both activity recognition and activity prediction tasks.

4.9 STGNN Evaluation against DNNs

Each member will evaluate their own STGNN in the same manner as outlined previously with quantitative metrics for accuracy, precision, recall and F1-score. Each member will compare their model separately to Graph WaveNet as the baseline STGNN, as well as to the baseline DNN models.

Each member will then analyze whether their model can be trusted in all scenarios by utilizing our 3D environment much like the analysis mentioned previously.

4.10 Network Analysis and Knowledge Discovery

Building upon our explanatory data analysis, we will seek to gain a deeper understanding of the dynamics of human activity by examining the graph structures learned by our trained STGNNs. Each team member will extract the adjacency matrix from their chosen STGNN and directly compare it to the matrix learned by the Graph WaveNet model. We will evaluate these matrices using three qualitative measures:

- Node centrality [17]: Quantifies the importance of each node (sensor) within the network. High-centrality nodes are those through which many strong connections pass, indicating they play a pivotal role in activity

propagation (how sensor activations ripple through the learned graph).
- Adjacency matrix heatmap: A visual representation of edge weights between every pair of nodes. Darker cells indicate stronger learned connections, making it easy to spot clusters of tightly linked sensors.
- Dependency map: Illustrates inferred directional relationships among sensors. By tracing how activity in one node depends on or influences others, we can uncover the model's implied information flow.

This qualitative graph analysis will reveal which sensor relationships each model prioritizes and thus shed light on why certain classifications or predictions are made. Previous work by Gaibie et al. [8], applied adjacency heatmaps and dependency maps to uncover the dynamics of weather flow predictions, and we intend to adopt the same approach to elucidate the spatial-temporal patterns learned by our STGNNs for HAR and HAP.

4.11 Model & 3D Environment Enhancement

Based on the needs, limitations and insights identified during the development of our models and 3D environment, each team member will build upon their STGNN model and the base 3D environment to extend their functionalities. Examples of such enhancements include experimenting with different time horizons to assess long-term prediction capabilities, utilizing the 3D environment for synthetic data generation and analyzing the impact of this on model performance, representing the model's adjacency matrix visually within the 3D environment, highlighting areas in the 3D environment with high prediction error, identifying and displaying a series of potential activity chains within the 3D environment and lastly, implementing a step-by-step playthrough of model predictions over time. Each member will be responsible for their own enhancements to promote further knowledge discovery and deepen application domain understanding.

5. Ethical, Professional and Legal Issues:

This research makes use of publicly available sensor datasets that have been published under open licenses and contain no information that could be traced back to individuals. The 3D environment being developed is for internal use, as a tool to evaluate whether model outputs are trustworthy and thus no user study is required. As a result, no ethics clearance is required as this project falls outside the scope of activities requiring institutional review.

All intellectual property, including the source code, models and 3D environment assets will be released under an open-source license.

6. Related Work

6.1 Traditional Deep Neural Networks

Early work in sensor-based HAR and HAP leveraged conventional deep learning architectures such as CNNs and Recurrent Neural Networks (RNNs), in particular LSTMs. CNNs extract local spatial-temporal features through stacked convolution and pooling operations, proving effective at capturing patterns in fixed-size input windows, while LSTMs model long-range temporal dependencies by maintaining internal memory states. Studies such as Ramesh et al. [18] and Chen et al. [19] report HAR accuracies of 86.7% and 96.12%, respectively, using CNN-based pipelines, whereas Du et al. [20] employed an LSTM for HAP and achieved a 78.3% accuracy, highlighting challenges in modeling long-term temporal dynamics for prediction tasks.

6.2 Spatial-Temporal Graph Neural Networks

STGNNs extend graph neural networks by integrating specialized modules that jointly learn spatial relationships (through graph convolutions) and temporal evolution (through recurrent or convolutional sequence models). Given a sensor network graph, spatial graph convolutions aggregate information from neighboring nodes via learned adjacency weights, while temporal modules (e.g., gated recurrent units or temporal convolutions) capture evolving node features over time. Attention mechanisms can further adaptively weight both spatial edges and historical time steps, enabling dynamic adjacency matrices that reflect context-dependent sensor interactions [7]. In the HAR/HAP domain, this approach naturally models the non-Euclidean connectivity among ambient sensors and yields richer representations than grid-based DNN. A notable STGNN variant is Graph WaveNet [12], which couples dilated temporal convolutions with a learnable adjacency matrix.

6.3 Network Analysis Techniques

Gaibie et al. [8] applied network analysis techniques to STGNNs for weather pattern prediction in South Africa, specifically using heatmaps and directed dependency graphs to uncover meteorological flows and key station relationships. Their work demonstrates how network analysis can surface domain-specific insights from STGNN representations, motivating our application of the same methods to human activity data.

6.4 3D Environment

Bouchabou et al. [21] introduced a VirtualHome: a Unity3D based framework with a virtual human avatar to generate synthetic sensor data for training and testing. However, the platform has only been used for synthetic data generation and has not yet been used as a tool to evaluate model HAR and HAP model performance.

7. Anticipated Outcomes

7.1 System:

We expect to produce a 3D virtual environment that can represent our sensor data accurately and can be integrated with our STGNN models to visually represent HAR model classifications and HAP model predictions. We expect the virtual environment and model to interact seamlessly and correctly visualize model predictions.

7.2 Research:

Gain a deeper understanding within the application domain of human activity through knowledge discovery. Through the analysis of our learned graphs, we expect to discover common activity chains and understand how activities unfold.

7.3 Expected Impact:

By leveraging knowledge discovery techniques, this work will uncover latent patterns in how people perform everyday household tasks and introduce a robust framework for qualitatively evaluating STGNN models (through network analysis) within the context of HAR and HAP.

7.4 Key Success factors:

Success will be defined by:

- Outperforming DNN baseline models with STGNNs.
- Developing a 3D environment that can visualize HAR and HAP model outputs.

- Demonstrating the trustworthiness, interpretability, and explainability of STGNN predictions through the 3D environment.

8. Project Plan

8.1 Project Outline

To conduct our research, we have decided to implement three core phases. Phase 1 involves data collection and preprocessing which will be used for the development and training of our baseline models, focusing on HAR. We then will perform exploratory data analysis on the datasets. This phase also includes the development of the 3D visualization platform, and lastly, the evaluation of our baselines' performance with metrics and visual analysis. Phase 2 involves implementing and optimizing the state-of-the-art STGNN models for HAR and HAP. We will then evaluate their performance against the baseline models using the associated metrics, with the main evaluation task being network/visual analysis and knowledge discovery. Phase 3 will involve the model and 3D environment enhancements discussed previously, based on the needs identified during earlier phases. This outline is highlighted in the timeline and Gantt chart (in the Appendix) and is aligned with our deliverables and milestones.

8.2 Risks

Table 1 in the appendix presents a detailed risk matrix outlining the project's potential risks alongside their monitoring, mitigation and management strategies.

8.3 Timeline

The timeline begins from the project proposal draft on the 24 April 2025 and ends with the School of IT showcase on the 27 October 2025. All the deliverables and milestones can be seen in the Gantt Chart (figure 1) found in the appendix. Red, green and purple represent each team member and blue represents a collective effort.

8.4 Resources Required

1. **Equipment:**
   - The following computer architecture will be utilized:
     - Intel i7 processor (and equivalent)
     - 16GB RAM
   - For intense model training tasks, we will be making use of the UCT High Performance Computing (HPC) cluster with the clearance of our advisor.
2. **Software**
   - PyTorch as our main software library
   - Unity3D for the 3D environment
3. **Datasets**
   - CASAS datasets
   - Examples: Kyoto, Aruba, Tulum, Cairo and Milan.
4. **3D simulator**
   - VirtualHome simulator

8.5 Deliverables and Milestones

The project's core outputs are:

- An academic paper detailing our findings
- 2 DNN and 4 STGNN models for HAR and HAP
- A 3D environment
- Supporting code for the models and 3D environment

The academic paper stands as the primary deliverable. Throughout the project's lifecycle, several intermediate deliverables will also be produced. They are:

- Three literature reviews
- Project proposal
- Draft academic paper
- Final academic paper
- System source code with documentation
- Project poster
- Project website

The project's milestones are defined by the conclusion of each phase, detailed in Section 8.1.

8.6 Work Allocation

As part of our phased approach, work will be allocated in the following manner:

Phase 1:

We will work together in this phase with dataset collection, preprocessing, data analysis, implementing the 3D visualization environment and writing up a report with our phase 1 results. Andrew and Temiloluwa will be responsible for the CNN and LSTM implementations respectively while Kishalan we implement Graph WaveNet. Temiloluwa will also perform the evaluation of the baseline models.

Phase 2:

In this phase each member will be responsible for their own state-of-the-art STGNN:

DSTAGNN - Andrew

ASTGCN - Kishalan

AGCRN – Temiloluwa

Tasks for each member include the development, optimization, and evaluation of their model against the baselines, and perform their own network analysis to be qualitatively compared to Graph WaveNet. This is for the purpose of knowledge discovery.

Phase 3:

Based on needs identified in Phase 2, each member will identify enhancements that can be made to their respective models as well as the base 3D environment. Each member will carry out their own enhancements for further knowledge discovery separately.

9. References

[1] S. Gupta. 2021. Deep learning based human activity recognition (HAR) using wearable sensor data. Int. J. Inf. Manag. Data Insights 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, 7586. https://doi.org/10.3390/s23177586

[3] M. Kaseris, I. Kostavelis, and S. Malassiotis. 2024. A comprehensive survey on deep learning methods in human activity recognition. Mach. Learn. Knowl. Extr. 6, 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, 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 Commun. Surv. Tutor. 15, 3, 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, 2763–2774. https://doi.org/10.1007/s10489-021-02587-w

[8] A. Gaibie, H. Amir, I. Nandutu, and D. Moodley. 2025. Predicting and discovering weather patterns in South Africa using spatial-temporal graph neural networks. Commun. Comput. Inf. Sci. 2326, 144–160. https://doi.org/10.1007/978-3-031-78255-8_9

[9] Z. Al Sahili and M. Awad. 2023. Spatio-temporal graph neural networks: A survey. Retrieved from https://arxiv.org/abs/2301.105

[10] 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, 1243–1256.

[11] D. J. Cook and M. Schmitter-Edgecombe. 2009. Assessing the quality of activities in a smart environment. Methods Inf. Med. 48, 5, 480–485. https://doi.org/10.3414/ME0592

[12] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang. 2019. Graph WaveNet for deep spatial-temporal graph modeling. Retrieved from https://arxiv.org/abs/1906.00121

[13] X. Puig, K. Ra, M. Boben, J. Li, T. Wang, S. Fidler, and A. Torralba. 2018. Virtualhome: Simulating household activities via programs. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR '18), 8494–8502.

[14] S. Lan, Y. Ma, W. Huang, W. Wang, H. Yang, and P. Li. 2022. DSTAGNN: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting. In Proc. 39th Int. Conf. Mach. Learn. (ICML), PMLR, 11906–11917. https://proceedings.mlr.press/v162/lan22a.html

[15] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang. 2020. Adaptive graph convolutional recurrent network for traffic forecasting. Adv. Neural Inf. Process. Syst. 33, 17804–17815.

[16] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan. 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI '19)*.

[17] F. Battiston, G. Cencetti, I. Iacopini, V. Latora, M. Lucas, A. Patania, J.-G. Young, and G. Petri. 2020. Networks beyond pairwise interactions: Structure and dynamics. Phys. Rep. 874, 1–92. https://doi.org/10.1016/j.physrep.2020.05.004

[18] L. S. Ramesh, B. Dhanalakshmi, and K. Selvakumar. 2024. Human activity recognition through images using a deep learning approach. https://doi.org/10.21203/rs.3.rs-4443695/v1

[19] K. Chen, L. Yao, D. Zhang, B. Guo, and Z. Yu. 2019. Multi-agent attentional activity recognition. Retrieved from: https://arxiv.org/abs/1905.08948

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

[21] 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. https://doi.org/10.3390/s23177586

10. Appendix

Table 1: Risk Assessment Matrix

Risk Description	Probability	Potential Impact	Risk Monitoring Strategy	Risk Mitigation Strategy	Risk Management Strategy
Model Overfitting/Underfitting and Convergence Problems: Training challenges with DNN and STGNN models	Medium–High	High	Monitor training and validation loss curves; use performance metrics and early stopping criteria	Employ regularization methods (dropout, weight decay); apply cross-validation and hyperparameter tuning; adjust model complexity accordingly	Quickly analyse training logs and error patterns to isolate the issue. Rollback to a previously stable model checkpoint if performance degrades sharply. Apply remedial re-training with adjusted hyperparameters.
Computational and Hardware Limitations: Insufficient compute or memory resources	Low	High	Track system resource usage (CPU, GPU, memory). Set up performance benchmarks during training phases.	Optimize code. Request access to UCT HPC cluster. Schedule training during low-demand period. Implement distributed processing.	Reschedule critical workloads using available resources. Communicate with HPC system administrators to increase available compute.
Integration Challenges with 3D Environment: Aligning sensor-based classifications with the avatar overlay	High	Medium–High	Perform regular integration tests between 3D environment modules and ML outputs. Compare 3D environment outputs to ground truth data	Develop a modular integration framework with clearly defined APIs. Introduce iterative testing and debugging cycles. Use 3D environment logs to pinpoint discrepancies.	Conduct a rapid post-failure review to identify misalignment sources. Isolate and correct integration points, possibly rolling back to the last stable configuration. Update integration protocols based on lessons learned.
3D Environment Software Bugs, Dependency & Version Control Issues: Version mismatches in libraries and environment setups	Medium	Medium	Implement automated tests and Continuous Integration/Continuous Deployment (CI/CD) pipelines. Conduct regular code reviews	Use containerization to standardize environments. maintain detailed documentation. Enforce strict version control practices	Rollback to a previously stable environment version immediately. Deploy emergency patches and re-run critical tests. Update and tighten dependency management procedures. Document the issue and resolution for future reference to prevent recurrence.
Project Completion Delays: The project not being completed on time	High	High	Track project timeline and milestones using project management tools. Conduct regular progress review meetings with supervisor and team.	Adopt a phased implementation approach to enable incremental delivery, continuous assessment, and adjustments to resource allocation and timelines.	Reassess the project timeline and adjust resource allocation promptly. Activate contingency plans, such as re-prioritizing tasks or adding manpower. Implement overtime or accelerated measures to bring the project back on schedule.
Team Member Departure: Unexpected loss of a team member	Low	High	Track team engagement and workload via regular meetings	Ensure that core groupwork components of the project are completed early on.	Drop the development and evaluation of the missing team member's STGNN and only include 2 state-of-the-art STGNNs in the final report.

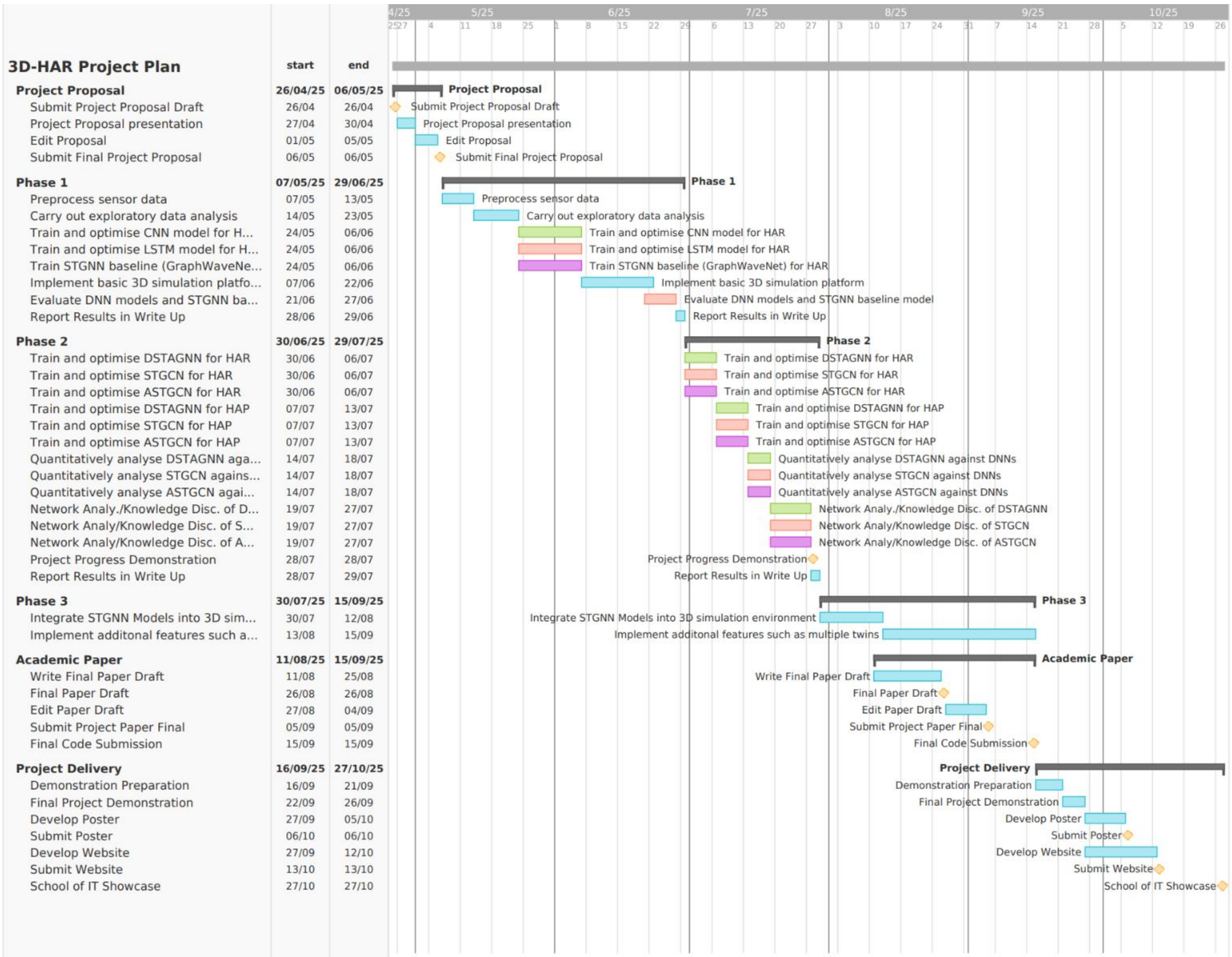

Figure 1: Gantt Chart with project milestones and deliverables.

*****Red represents Temiloluwa's work , green represents Andrew's work and purple represents Kishalan's work.