Temporal-Spatial Occupancy Estimation Using Transformer Encoders

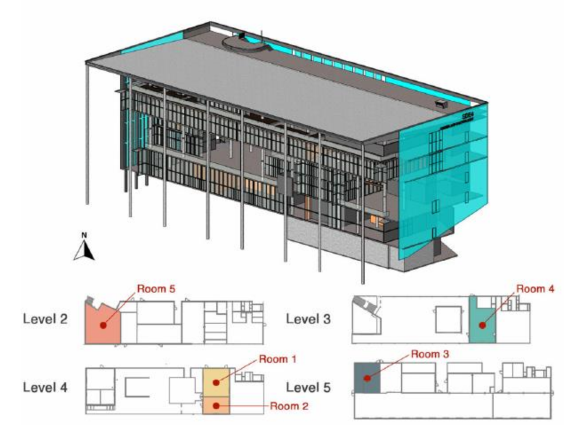
Somaya Ahmadi
Faculty Mentor: Sanish Rai
Department of Math and Computer Science, Longwood University

Background

- Occupancy Estimation [1] uses: energy saving, evacuation planning, crowd management, smart home.
- Transformer is a neural network architecture used for performing machine learning tasks . It is used in natural language processing and occupancy estimation.
- For this project, we used an encoder-only Transformer model to estimate the number of occupants in multiple rooms based on environmental sensor data.
- The model captures both temporal patterns from each room and spatial dependencies across rooms and it is highly parallelizable.
- The motivation was to predict occupancy across rooms and understand the temporal context within the data and the spatial context among the rooms.
- We were able to achieve accuracies as high as 92% for 5 rooms.

Dataset Information

- We used ROBOD dataset [2] which contains data from 5 rooms in a university building in Singapore.
- Each room has environmental sensors (HVAC, lighting, plug loads and fans, weather conditions) data collected over a total of 181 days at a resolution of 5 minutes.
- Room 1 and 2 are lecture rooms, Room 3 and 4 are office spaces, and room 5 is a library space.
- Dataset also contains ground truth occupancy count.



Experiment Setup

- **Transformer Models (Encoder-Only)**
  - Transformer models are designed to capture patterns in data using self-attention, which helps the model focus on important parts of the input sequence.
  - In our framework, we use temporal encoding to extract time-series context from the room's sensors and spatial encoding to extract spatial context among the rooms.

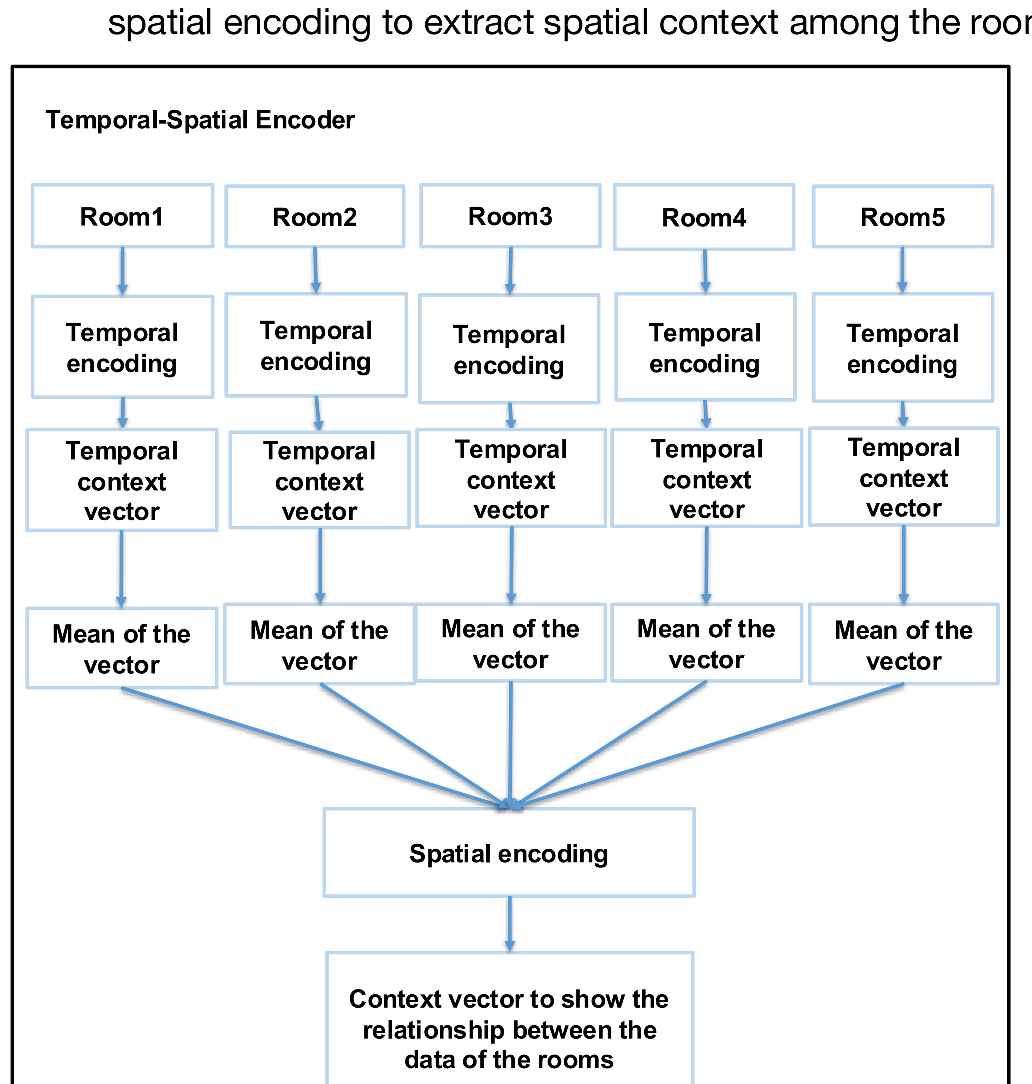


Figure 1. Temporal-Spatial Encoder framework

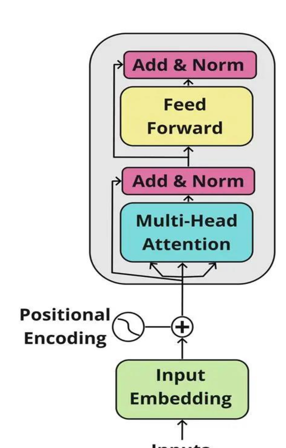


Figure 2. Transformer Encoder-Only [3]

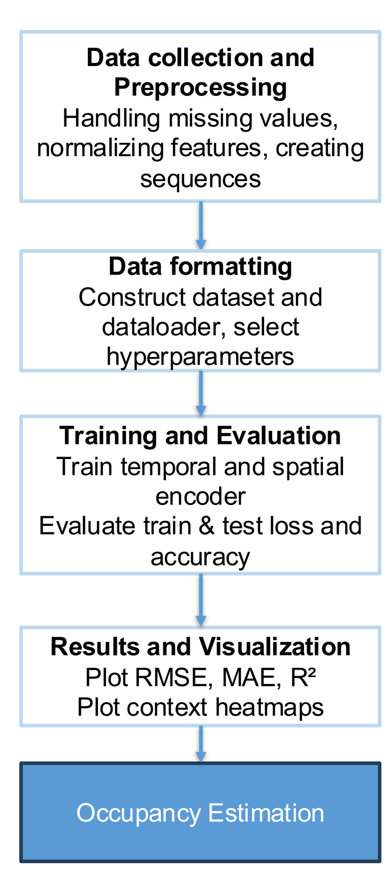


Figure 3. Model training and evaluation

- **Data Sequencing**
  - We wanted our data to be compatible with a time-series based machine to predict occupancy.
  - We converted the data into a series of sequences with a fixed size, an overlap, and a specified timestep to predict.

- **The Experiment**
  - We took training data and feed it to the model.
  - We then made predictions using the testing data and compared the results.
  - We can use the trained model for real-world applications.

Experiment Results

- **Experiments for finding the best hyperparameters**
  - Sequence length
  - Overlap
  - Hidden dimensions (temporal & spatial)
  - Number of layers (temporal & spatial)
  - Number of heads (temporal & spatial)
  - Epochs
  - Batch size (temporal and spatial)
  - Dropout rate (temporal & spatial)
- After training the model, we predicted occupancy one step into the future and achieved an average spatial accuracy of 92% across all rooms. We used 75% of the data for training and 25% for testing.
  - **Results using the best hyperparameters**

Temporal (Single-Room) Transformer Evaluation

Rooms	RMSE	MAE	R^2
Room1 (R1)	0.7645	0.2504	0.9793
Room2 (R2)	0.8487	0.3060	0.9396
Room3 (R3)	0.5894	0.2803	0.9439
Room4 (R4)	0.7828	0.5002	0.9482
Room5 (R5)	0.6718	0.3177	0.9392

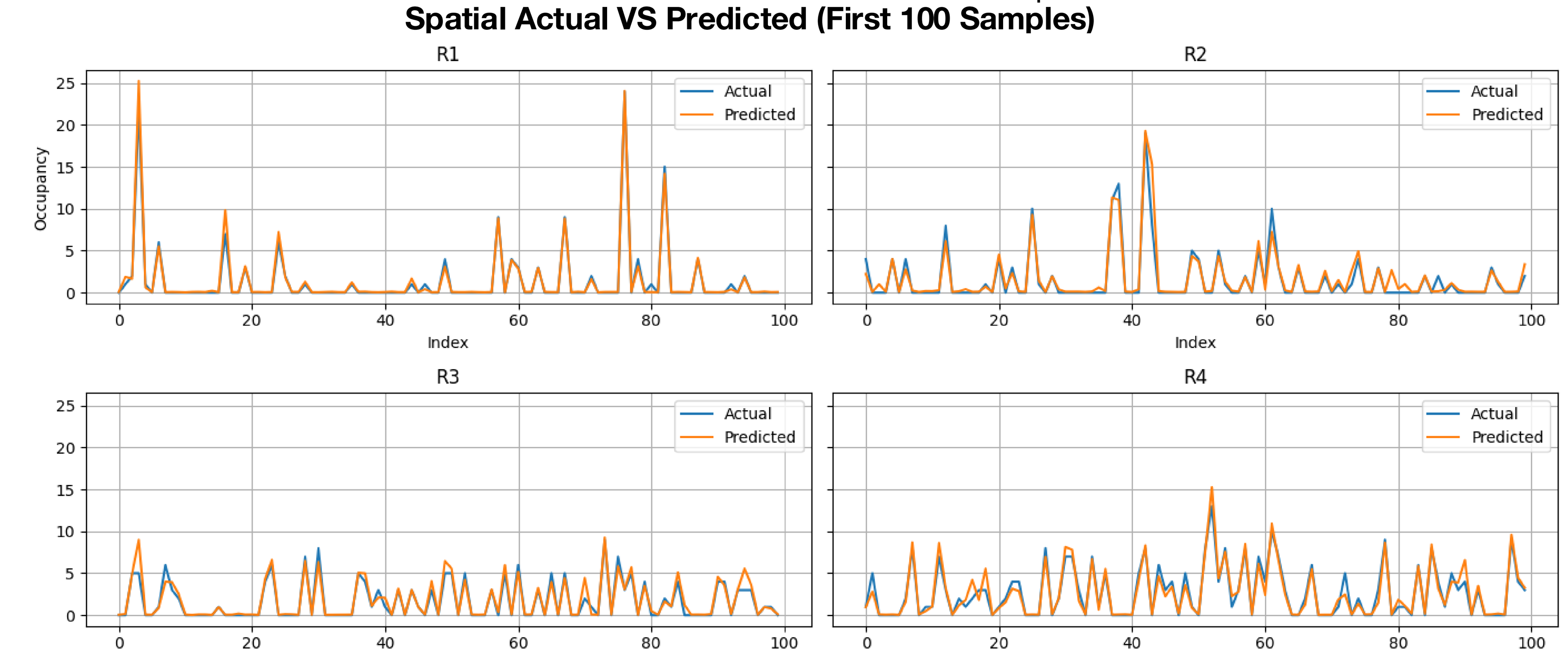
Spatial (Multi-Room) Transformer Evaluation

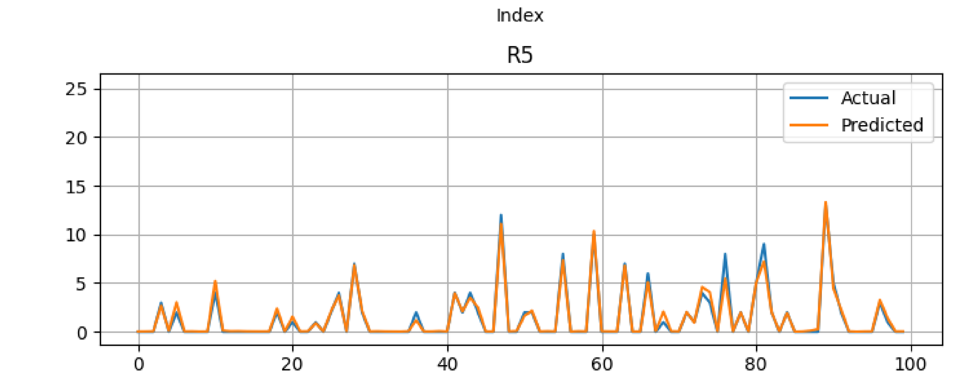
Rooms	RMSE	MAE	R^2
Room1 (R1)	1.0163	0.3649	0.9635
Room2 (R2)	1.4141	0.5537	0.8323
Room3 (R3)	0.5975	0.3229	0.9423
Room4 (R4)	0.8898	0.5867	0.9331
Room5 (R5)	0.6577	0.3417	0.9417

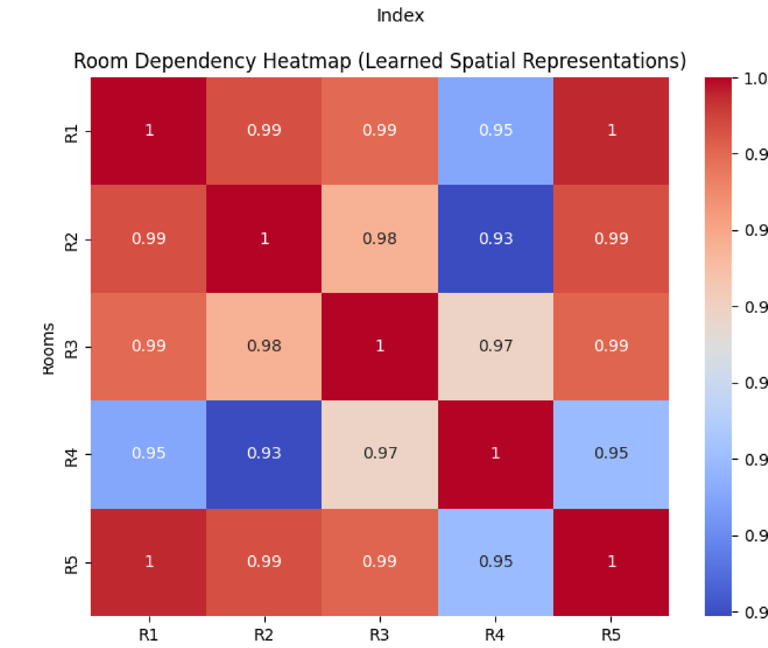
Total Runtime: 587.49 seconds (9.79 minutes)

- **Best hyperparameters:**
  - Sequence Length:
    - Each input sequence, contains 60 time samples of sensor data.
  - Overlap:
    - Each subsequent pair of sequences overlaps by 59 time samples.
  - Hidden Dimensions:
    - The temporal encoder uses a hidden size of 64 and the spatial encoder uses a hidden size of 32.
  - Number of Layers:
    - The temporal encoder has 3 layers, and the spatial encoder has 1 layer.
  - Number of Heads:
    - The temporal encoder uses 4 attention heads, and the spatial encoder uses 2 attention heads.
  - Epochs: 30
    - The training loop was run 30 times.
  - Batch Size: 64
    - For each training loop, data is fed into the machine at batches of 64 data inputs at a time.
  - Dropout:
    - Both temporal and spatial encoders use 10% dropout.

Spatial Actual VS Predicted (First 100 Samples)







Conclusions

- Using temporal and spatial encoders, we achieved up to 92% accuracy for the model we created.
- We used a temporal transformer encoder to help each room learn patterns over time.
- We added Spatial encoder to understand relationship and dependency between rooms.
- Even though the accuracy drops compared to non-transformer models, this model retains the time and space context, which can be useful to estimate occupancy when sensor data is missing for certain timesteps.
- The estimation will also use the spatial context for better occupancy in future.
- The training time is comparable to other models since transformers are highly parallelizable with GPU.

Future Directions

- For the future, here are some things we can do to improve:
  - Use better hyperparameters to achieve higher accuracy.
  - Test the model on a raspberry pi.
  - Attempt to forecast multiple timesteps per prediction.

Acknowledgements

- I want to thank Longwood University and the PRISM Program for providing this research opportunity for me.

References

[1] Aliero, Muhammad S., et al. “Non-Intrusive Room Occupancy Prediction Performance Analysis Using Different Machine Learning Techniques.” *MDPI*, Multidisciplinary Digital Publishing Institute, 6 Dec. 2022, doi.org/10.3390/en15239231.
[2] Tekler, Zeynep Duygu, et al. "ROBOD, room-level occupancy and building operation dataset." Building Simulation. Vol. 15. No. 12. Beijing: Tsinghua University Press, 2022.
[3] Delovski, B. “Intro to Transformers: The Encoder Block,” *Edlitera*, May 3, 2023. [Online]. Available: https://www.edlitera.com/blog/posts/transformers-encoder-block

Contact Information

- somaya.ahmadi@live.longwood.edu
- rais@longwood.edu