# Link Prediction with Graph Neural Networks

## Wyze Rule Recommendation

Francesco Conti, Machine Learning Engineer

# Link Prediction

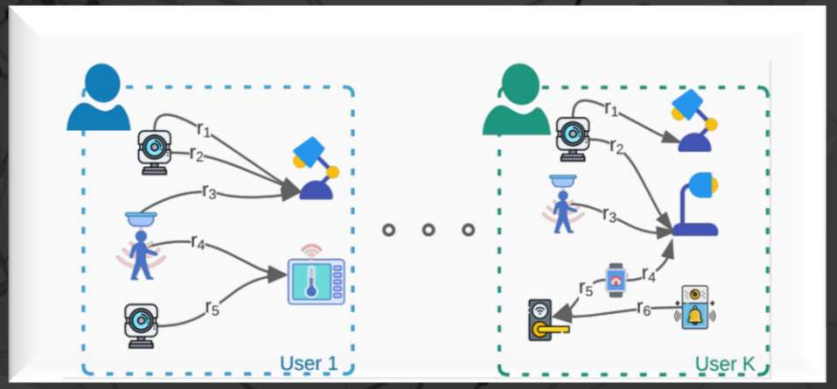
## Graph-based Modeling

Each user has a set of devices and a set of rules. Rules are identified by a trigger device, a trigger state, an action device and an action.

- **Nodes**: represented by individual devices;
- **Feature Nodes**: categorical variable encoding device model (e.g. Camera, MotionSensor).
- **Directed Edges**: each rule connects a trigger device to an action device.
- **Edge Features**: Incorporate trigger state and action as categorical variables.

## Rule Inference

- Problem reformulated as a Link Prediction Task.
- **Graph Neural Network (GNN);**
- GNN evaluates a **score** for each potential edge.




FedRule: Federated Rule Recommendation System with Graph Neural Networks (https://arxiv.org/pdf/2211.06812.pdf)

# Model Architecture

## Node Embedding block

- **Node Representation**: one-hot encoding of device models.
- **Edge Representation**: Employing embedding layers for edge features.
- **Aggregation**: Aggregating edge features to nodes for enriched representations.
- Implementing two **SAGEConv** layers to capture intricate relationships within the graph.
- At this stage, each node is associated with an embedding vector.

## Prediction Head block on node pairs

- **Node Pair Embedding**: Concatenating the embeddings of two nodes related to a candidate edge.
- **Neural Network**: Employing linear layers to process concatenated embeddings.
- **NN output**: 552 outputs, a value for each rule type with a frequency over 20 occurrences in the training-set.
- **Activation Function**: Applying a sigmoid activation to predict edge probabilities between nodes.

# Training Approach

- **Training Data:**
  - Each user serves as an individual training sample.
  - Users with fewer than two rules are filtered to ensure robust model training.
- **Batched Training:**
  - Implementing batched training for efficiency and enhanced learning dynamics.
- **Training Data Generation:**
  - **Positive Sampling**: Extracts an edge from each graph in the batch during training and predicting it.
  - **Negative Sampling**: Include not-real edges in the sampling process.
- **Loss Function:**
  - Binary-cross-entropy loss function differentiates between positive and negative samples.

# Conclusion

- **Performance:**
  - GNN achieved a Mean Rank score near to 0.45 in the private split.
- **Improvements:**
  - Performance can be enhanced by selecting optimal hyperparameters;
  - Focal loss;
  - Implemented using a free Google Colab account.
- **GitHub**
  - **Link:** https://github.com/conti748/wyze-rule-recommendation