Accurate Action Recommendations and Demand Response for Smart Homes via Knowledge Graphs

Wenzhi Chen, Hongjian Sun*
Department of Engineering
Durham University
Durham, UK, DH1 3LE
wenzhi.chen@durham.ac.uk
hongjian.sun@durham.ac.uk

Minglei You
Department of Electrical
and Electronic Engineering
University of Nottingham
Nottingham, UK, NG8 1BB
minglei.you@nottingham.ac.uk

Jing Jiang
Department of Mathematics, Physics
and Electrical Engineering
Northumbria University
Newcastle upon Tyne, UK, NE1 8ST
jing.jiang@northumbria.ac.uk

***Abstract*—Accurate action recommendations can enhance the convenience of daily life, such as automatically turning on the dining area lights during meals or playing music based on residential habits. Generating precise recommendations for the next household device actions is essential for future smart homes. This paper proposes an action recommendation system for household appliance scenarios by customizing the knowledge graph attention network in its sampling and aggregation, in which the usage habits, periods, and location information were used as common sense for graph modelling. The results of the recommendations can be explained by a designed method with the trained embeddings. Finally, with the recommendation expectation, appliances' comfort level and average power are modelled as a multi-objective optimization problem for participating in demand response. Simulations demonstrate that the proposed system can achieve 93.4% accuracy in recommendations and reduce the power consumption by 20% while providing reasonable explanations.**

***Index Terms*—Smart home, Interpretable recommendation, Knowledge graph, Demand response**

I. Introduction

A. Background

1) Domestic Appliances Recommendations: With the advent of smart home appliances, there has been a growing interest in replacing conventional automation with intelligent action recommendations [1]. Accurate action recommendations can enhance the convenience of daily life, such as automatically turning on the dining area lights during meals, closing curtains at night, or turning on the TV/music based on residential habits, especially for the elderly or patients with mobility difficulties [2]. When generating action recommendations, it is important to produce convincing and interpretable results. However, to the best of our knowledge, no paper focuses on interpretable models designed for household appliance action recommendations. This paper aims to fill this research gap.

2) Demand Response: With the participation of a flexible load, the household appliances' demand response can improve electrical efficiency [3]. Many household appliances such as fans, washing machines and ovens can be regarded as flexible loads with different power levels. While high-comfort household appliances often require higher power consumption, correspondingly, lower power consumption leads to discomfort. In this paper, the demand response means that residents can make a balance between comfort level and power consumption by setting the appropriate power level for each appliance, thus improving energy efficiency. Optimizing based on action recommendations to achieve demand response is another issue that this paper aims to address.

3) Knowledge Graph Recommendation System and Embedding: A knowledge graph is a heterogeneous graph that considers multiple types of nodes and edges. Typically, the data are stored in the form of a head-relation-tail, where the head and tail are nodes in the graph, and the relation represents the relationship between them. Knowledge graphs can easily represent irregular data, making them suitable for recommendation algorithms. The key step of the knowledge graph algorithms is the embedding aggregation. Graph embedding vectorizes nodes and edges in the graph. The trained embeddings reflect the characteristics of the original data, where the inner product of embeddings indicates the degree of correlation between two nodes. Embeddings can also undergo basic operations like addition, subtraction, and concatenation as basic elements operations.

B. Related works and Literature Review

1) Graph Recommendation Algorithms: In 2019, Knowledge graph convolutional networks (KGCN) [4] was proposed by Wang *et al.*, which was a typical example that combines knowledge graphs and convolution operations with neural networks for recommendations. Unlike KGCN, the Knowledge Graph Attention Networks (KGAT) algorithm [5] utilized graph attention networks for message passing, aggregating user and item embeddings, and recommendations.

2) Domestic Appliances Recommendation System: Few papers focused on recommending actions for household appliances. In 2022, Jeon *et al.* [6] used a Transformer-based model to encode the sequential behaviour of users and achieved accurate recommendations for the next action. Using GraphSage and federated learning, Yao *et al.* performed recommendations

This work was supported by the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie Grant Agreement No. 872172 TESTBED2 project, Royal Society IEC\NSFC\223152, Royal Society Short Industry Fellowship and European Space Agency Project 12893212.

for users in the same year [7]. However, these papers only used action sequence information but did not consider additional contextual information such as the usage time/location, or the comfort level of using household appliances. Both our approach and [7] utilized graph-based recommendation algorithms. The difference is that our recommended actions are triggered by electricity usage, leading to demand response-driven optimization.

3) The Interpretability of Recommendation Algorithms: The explanation of the algorithm's recommendations can be divided into in-process and post-hoc explanations. In 2017, Das *et al.* implemented an in-process explanation algorithm, reinforcement learning-based approaches were used and the most probable path was determined based on a pre-defined scoring function [8]. Based on the characteristics of the embedding inner products, In 2018, Ai *et al.* proposed a post-hoc explanation method that used the inner product multiplication of embeddings to represent the probability of feasible paths [9]. However, no household action recommendation model provided explanations for its recommendations.

C. Research Gaps and Motivation

The following gaps are identified from the literature review.

- How to incorporate information to recommend the next household appliance action accurately?
- How to generate the explanation for the household appliance action recommended results?
- How to optimize the power consumption and comfort level by setting the appropriate power for each appliance?

In response to these questions, this paper establishes a system that accurately recommends and optimises the user's next action. The contributions of this paper are as follows:

- The appliance information from the existing dataset is modelled as a knowledge graph, including electricity usage information, appliance location information, appliance usage sequences, etc. It contains a wealth of everyday household appliance usage and general knowledge, facilitating data mining for household activities.
- The KGAT algorithm is customized and improves sampling and aggregation to the traditional KGAT. It predicts the user's next possible action with the knowledge graph. Comparison experiments are conducted with traditional Deep neural network (DNN), Convolutional neural network (CNN), and Recurrent neural network (RNN) algorithms. The performance outperforms them.
- Explanations method for the recommended results are designed and implemented. The reasoning is based on embeddings, and the reasons for each recommended result are analyzed. This makes our recommendation system more trusted by users.
- Demand response optimization is performed based on the expected recommended actions, considering energy consumption and comfort, resulting in the power level settings for each appliance. The energy efficiency is enhanced by balancing energy consumption and comfort.

The remaining structure of this paper is as follows: Section II presents the system architecture, Section III introduces the proposed algorithm, Section IV presents the simulation and results of the system, and Section V provides the conclusion.

II. System Composition

This section introduces the system architecture including different modules and the attributes of each appliance.

The overall architecture is built upon a household appliance system, as is shown in Fig. 1. The system comprises appliances, sensors, a small-scale database, and a home energy manager. All the appliances are connected to the home's local area network, and sensors monitor them to collect energy consumption data. The data is sent to the local database for merging, processing and transforming to embeddings. The data in the database is then merged with the energy manager, which utilizes the collected data for recommendations. The recommended results are sent to the explanation module to generate explanations and to the optimization module to generate optimization results of the comfort level and energy consumption, assisting users in decision-making. The database data is not shared externally to ensure user privacy and federated learning can be employed.

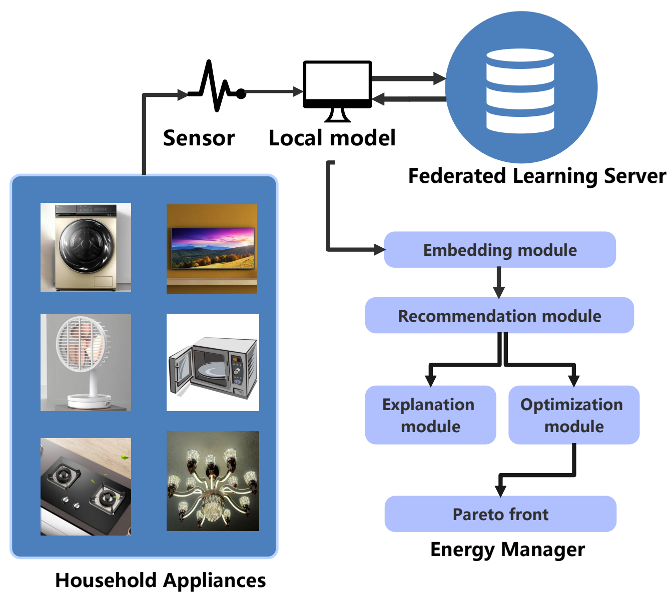

Fig. 1. The proposed system architecture and constituent modules.

Each appliance possesses the following attributes:

Rated Power: Each appliance has its rated power; for example, a Toaster has a rated power of 1500 Watts.

Appliance Location: Appliances are distributed across different rooms, including the kitchen, living room, office, bedroom, utility room, children's room and undefined locations.

Comfort Impact: Each appliance has a different impact on comfort. For instance, washing machines have a lower impact on comfort due to the demand response-ability, whereas appliances like microwaves or gas ovens have a higher impact.

Habitual Usage Time: Each appliance has specific habitual usage times. For example, gas stoves are most used during mealtime, while lights are frequently used at night.

Habitual Usage Sequence: Some appliances have a habitual usage sequence. For instance, the most commonly used appliance following a dryer might be a hair straightener.

III. Proposed Algorithm

This section introduces the proposed embedding, recommendation, explanation and optimization module.

A. Embedding Module

During the data processing, it is necessary to convert the data into embeddings, which use high-dimensional vectors to represent the discrete data. The transformation from data to embeddings facilitates the use of knowledge graph algorithms in subsequent steps. During backpropagation, its parameters are updated based on the gradient of the loss function to minimize the difference between predicted results and true labels. The following data was encoded to embeddings:

1) Appliances: Each appliance is assigned an embedding, resulting in a set of embeddings for different appliances.
2) Time zones: The daily time is divided into four time periods: midnight, morning, afternoon, and evening. Thus, a week can be divided into 4×7=28 time zones. These 28 time zones are encoded as a set of embeddings.
3) Locations: Appliances are distributed across different rooms, and an embedding is created for each room.

The embedding layer aims to train the embedding of (h,r,t) such that the embedding of the head e_{h} plus the embedding of relation e_{r} approach the embedding of tail e_{t} for any positive (h,r,t) in distance (for example L2 distance), which can be represented as

$$\min \quad g_{\text{(h,r,t)}} = ||W_{\text{r}} e_{\text{h}} + e_{\text{r}} - W_{\text{r}} e_{\text{t}}||_2^2 \ , \tag{1}$$

where $|| \cdot ||_2^2$ means the squared results of the L2 distance. $g_{\text{(h,r,t)}}$ represents the positive triples and the value of $g_{\text{(h,r,t)}}$ should be close to zero. However, for negative triples, which means the h, r or t can be replaced by other values that never happened. For example, $g_{\text{(h,r,u)}}$ is generated by replacing t with u arbitrarily and the value of $g_{\text{(h,r,u)}}$ should be infinity.

The input data is processed by the embedding module, which converts discrete inputs into continuous embeddings. These embeddings are then fed into the knowledge graph layer as input for training.

B. Recommendation Module

1) Knowledge Graph Attention networks: In KGAT, attention mechanisms are utilized to determine the aggregation weights of neighbouring node embeddings in

$$a_{\text{(h,r,t)}} = \text{Softmax}((W_{\text{r}} e_{\text{t}})^{\text{T}} \tanh(W_{\text{r}} e_{\text{h}} + e_{\text{r}})) \ , \tag{2}$$

where the $a_{\text{(h,r,t)}}$ is the calculated attention for knowledge graph aggregation. Softmax and tanh (hyperbolic tangent function) are activation functions. W_{r} is the relation transformation matrix. e_{h}, e_{r} and e_{t} are the head, relation and tail entity respectively. Then, it aggregates the embeddings of neighbouring nodes based on the weights in

$$e_{\text{Nh}} = \sum_{\text{(h,r,t)} \in \text{Nh}} a_{\text{(h,r,t)}} \times e_{\text{t}} \ , \tag{3}$$

where e_{Nh} is the aggregated entity, which means how much information that the tail entity e_{t} is going to pass to the head entity e_{t}, which is represented as

$$agg = \text{LeakyRelu}(W_{\text{trans}}(e_{\text{h}} + e_{\text{Nh}})) \ , \tag{4}$$

where agg is the information aggregation vector, which contains weighted information with the head and tails entities, and can then be used for recommendations by embedding dot product with the aimed user embeddings. LeakyRelu is the activation function. W_{trans} is the transformation matrix.

Algorithm 1: The proposed KGAT networks with improved next-action recommendation scenario

Load knowledge graph sequence embeddings $e^{\text{se}}_{\text{(h,r,t,time)}}$, time zone embeddings $e^{\text{ti}}_{\text{(h,r,t)}}$ and location embeddings $e^{\text{lo}}_{\text{(h,r,t)}}$ from processed datasets.
Initialize the list L.
for *episode* $= 1, M$ **do**
  for e_h^{se}, e_t^{se} *in* $e^{se}_{(h,r,t,time)}$ **do**
    Sampling dataset s_{h} for e^{se}_{h} and s_{t} for e^{se}_{t} according to the distribution of $e^{\text{se}}_{\text{(h,r,t)}}$, $e^{\text{ti}}_{\text{(h,r,t)}}$ and $e^{\text{lo}}_{\text{(h,r,t)}}$.
    Constructing adjacency matrics Adj_{h} with s_{h} and Adj_{t} with s_{t}.
    for *each* e_h, e_r *and* e_t *in* Adj_h **do**
      $a_{\text{h-(h,r,t)}} =$ $\text{Softmax}((W_{\text{r}} e_{\text{t}})^{\text{T}} \tanh(W_{\text{r}} e_{\text{h}} + e_{\text{r}}))$
      $e_{\text{h-Nh}} = \sum_{\text{(h,r,t)} \in \text{Nh}} a_{\text{h-(h,r,t)}} \times e_{\text{t}}$
      $agg_{\text{h}} = \text{LeakyRelu}(W(e_{\text{h}} + e_{\text{h-Nh}}))$
    end for
    for *each* e_h, e_r *and* e_t *in* Adj_t **do**
      $a_{\text{t-(h,r,t)}} =$ $\text{Softmax}((W_{\text{r}} e_{\text{t}})^{\text{T}} \tanh(W_{\text{r}} e_{\text{h}} + e_{\text{r}}))$
      $e_{\text{t-Nh}} = \sum_{\text{(h,r,t)} \in \text{Nh}} a_{\text{t-(h,r,t)}} \times e_{\text{t}}$
      $agg_{\text{t}} = \text{LeakyRelu}(W(e_{\text{h}} + e_{\text{t-Nh}}))$
    end for
    $V = \text{concatenate}(agg_{\text{h}}, agg_{\text{t}}, e^{\text{se}}_{\text{time}})$
    Store $O = \text{Relu}(\text{FullyConnectedLayer}(V))$ in L
  end for
end for
Output: L

2) Proposed Knowledge Graph Attention networks algorithm: Different from the traditional KGAT algorithm, some improvements have been customized for the action recommendation scenario, the pseudocode is shown as Algorithm 1. Firstly, the algorithm aims to predict the next potential appliance to be used based on known residents' appliance usage habits. In the traditional KGAT recommendation algorithm, only potentially recommended appliances undergo graph embedding aggregation because conventional KGAT treats the recommended appliance as an item and the known appliance as a user. However, in our problem, the known appliance usage can also be treated as an item and aggregated. This is expressed in the line 13-18 of the Algorithm 1, where the subscript t-Nh and h-Nh means the $e_{\text{h-Nh}}$ is generated by t or h, similar in subscript h-(h,r,t) and t-(h,r,t). In addition, we first collect each type of attribute, such as timezone, usage sequence, and location, ensuring equal chances of different kinds of data. Then random sampling is replaced with probability-based sampling according to the distribution. These samples will inform the knowledge graph algorithms about the studied case and affect the aggregation results. For example, if a particular appliance is most commonly used on Tuesday mornings, the probability of sampling Tuesday mornings for the adjacency matrix will be higher. This is expressed in the line 5-6.

The output L contains aggregated embedding agg_{h} for head embeddings and agg_{t} for tail embeddings. Then the embedding L are used as the input of the fully connected layers for training in line 20 of the Algorithm 1.

C. Explanation Module

In recommendation algorithms, interpretability is crucial for making the recommendation results convincing. The purpose of this module is to provide further explanations for the recommended results. It is well-known that the trained embeddings in graph neural networks can represent the meaning of the data [10], and paper [9] used that for making explanations. Based on this characteristic, we generate reasons for the recommended results, which consist of three types of reasons:

1) Habitual usage sequence, which means an appliance is often used after another one. 2) Habitual usage time, which means an appliance is often used at some specific time. 3) Usage at a habitual location, which means an appliance is used because it has location relationships with other appliances.

Firstly, all the connections between the recommended appliances and the known appliances within two hops are identified in the graph and find the embeddings of all the nodes and relationships involved. Based on TransE and the approach described in the paper [10] in (5).

$$e_{\text{t}} \approx e_{\text{h}} + e_{\text{r}} \ , \tag{5}$$

where the head embedding e_{h} plus relation embedding e_{r} should approach the tail embedding e_{t} as much as possible after training. Ideally, they are equal.

Here we propose a novel measure in formula (6)-(11) for the recommendation results in the context of household appliance action recommendations. From (5), the embedding of the reasons can be calculated in (6)-(8).

$$e^{\text{se}}_{\text{total}} = (e^{\text{se}}_{\text{h}} + e^{\text{se}}_{\text{r}})e^{\text{se}}_{\text{t}} \ , \tag{6}$$

$$e^{\text{ti}}_{\text{total}} = (\sum_{j=1}^{N}(e^{\text{ti}}_{\text{hj}} + e^{\text{ti}}_{\text{rj}})e^{\text{ti}}_{\text{tj}})\frac{1}{\text{N}} \ , \tag{7}$$

$$e^{\text{lo}}_{\text{total}} = (e^{\text{lo}}_{\text{h}} + e^{\text{lo}}_{\text{r}})e^{\text{lo}}_{\text{t}} \ , \tag{8}$$

$$R(\text{se}) = \frac{e^{\text{se}}_{\text{total}}}{e^{\text{se}}_{\text{total}} + e^{\text{ti}}_{\text{total}} + e^{\text{lo}}_{\text{total}}} \ , \tag{9}$$

$$R(\text{ti}) = \frac{e^{\text{ti}}_{\text{total}}}{e^{\text{se}}_{\text{total}} + e^{\text{ti}}_{\text{total}} + e^{\text{lo}}_{\text{total}}} \ , \tag{10}$$

$$R(\text{lo}) = \frac{e^{\text{lo}}_{\text{total}}}{e^{\text{se}}_{\text{total}} + e^{\text{ti}}_{\text{total}} + e^{\text{lo}}_{\text{total}}} \ , \tag{11}$$

where the superscript se, ti and lo refer to different reasons like sequence, timezone and location. The value $e^{\text{se}}_{\text{total}}$, $e^{\text{ti}}_{\text{total}}$ and $e^{\text{lo}}_{\text{total}}$ are the product value of the embeddings. From the characteristic embedding of TransE [9], a higher product value means the reason has a higher relationship with the answer, where $e^{\text{ti}}_{\text{total}}$ is the top N averages product of embedding of the reason timezone. $e^{\text{se}}_{\text{total}}$ and $e^{\text{lo}}_{\text{total}}$ only exist if two household appliances have a sequential relationship or are placed in the same location. In (9)-(11), the possibility R of different reasons is calculated.

D. Optimization Module

1) Multi-Objective Optimization and Pareto Front: When solving complex problems with multiple objectives and a large number of parameters, the Genetic algorithm (GA) is a promising algorithm for the solution [11], especially when these objectives involve a non-linear problem.

The Pareto front $P(Y)$ is the set of non-dominated solutions in multi-objective optimization, which is written as (12).

$$P(Y) = \{y' \in Y : \{y'' \in Y : y'' \succ y', y'' \neq y'\} = \emptyset\} \ , \tag{12}$$

where a system with the function $f : X \to \mathcal{R}^M$ is considered, X is a set of feasible decisions in the metric space $\mathcal{R}^M$ and Y is the feasible set of criterion vectors in $\mathcal{R}^M$, such that $Y = y = f(x), \forall x \in X$. If a point y'' strictly dominates another point y', written as $y'' \succ y'$.

2) Optimizations Based on the recommendation results: It is assumed that the power of each appliance is adjustable within a certain range, and the power of an appliance affects the comfort level as shown in (13).

$$\max \quad f^{\text{comfort}} = -\sum_{i=1}^{N} c_i(P_i - P_i^{\text{real}})^2 \ , \tag{13}$$

where f^{comfort} is the satisfactory function, whose value is the accumulated comfort score, number N is the kinds of appliances. The rated power of each appliance is P_i with $i = 1, \ldots, N$. Power significantly above or below the rated power can result in discomfort, while power close to the rated power can lead to higher comfort. The averaged power is calculated as f_{avgpower} through (14).

$$\min \quad f_{\text{avgpower}} = \sum_{i=1}^{N} P_i^{\text{real}} \ , \tag{14}$$

where number the actual power is P_i^{real}.

In (13) and (14), we consider two conflicting objectives: maximizing comfort and minimizing average power consumption. For this type of optimization problem, multi-objective optimization algorithms, such as genetic algorithms, can be used to find the Pareto front. The outcome of the optimization is to set using power for each appliance based on the expectation of the recommendation results. Through the optimization results, multiple feasible solutions between comfort level and average power consumption can be selected in the Pareto front.

IV. Simulations setup

A. Dataset and software

For datasets, UK-DALE (UK Domestic Appliance-Level Electricity) is a publicly available household electricity dataset used for power consumption monitoring and non-intrusive load recognition research [12]. We customized the UK-DALE dataset and transformed it into graph-structured data. Each appliance is treated as a node and adds an edge between two appliances if they have a sequential relationship, labelled as "used before/after." A knowledge graph was also established for when appliances are used, connecting time nodes with appliance nodes. The relationship can be represented as "used during the time period". Finally, a knowledge graph for the location of appliances can be established, connecting location nodes with appliance nodes and labelling the relationship as "located in the room". Other attributes, such as the rated power and comfort level, are added for each appliance. Finally, negative sampling is conducted. For software, the PyTorch framework was used for deep learning methods.

B. Results for Recommendations

The proposed improved KGAT algorithm has its comparison with other algorithms including the original KGAT, DNN, CNN, and RNN. In this section, the final recommended results are defined as the top three most likely appliances generated by the recommendation algorithm, and a recommendation score greater than 0.95 is also added to the recommendation list. Precision is determined by comparing with the labels.

Figure 2 shows a selected instance where all other conditions were the same, representing typical algorithm performances. It can be observed that DNN and CNN yielded notably inferior results. The performance of the RNN algorithm steadily improved over time, but the KGAT algorithm (Averaged accuracy: 87.3%) reached the highest precision comparable to RNN in the first round. The improved KGAT achieved the highest accuracy (Averaged accuracy: 93.4%) and provided the most accurate recommendations, which averaged 7% accuracy improved compared with the traditional KGAT algorithm though the gap is not so apparent in the figure 2.

C. Results for Explanations

1) Reason-Sequence: This reason means that 2 appliances are always used together. The recommended score on the horizontal axis is the inner product of embeddings from the recommendation algorithm. After using the hairdryer, the system's best-recommended result is the straightener. The probability analysis is shown in Fig. 3. This explanation aligns with common sense, as these two appliances are often used together after taking a shower.

2) Reason-Timezone: The second reason is that these two appliances are frequently used during the same time period, as shown in Fig. 4. According to the resident's habits on Tuesday nights, appliances such as lights, office lamps, and appliances for entertainment purposes, such as television, Hifi, and HPTC, are the most commonly used in our dataset.

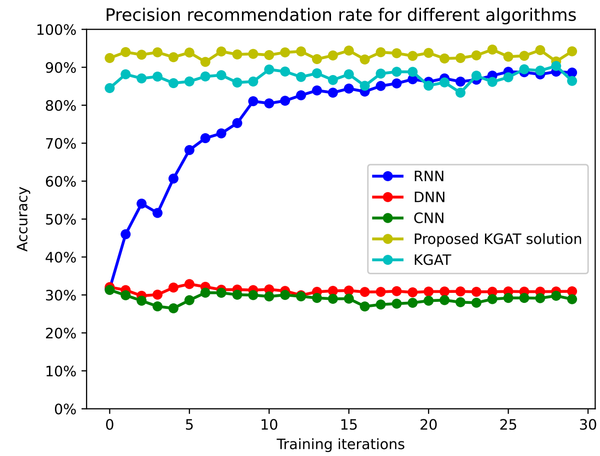

Fig. 2. Recommendation accuracy.

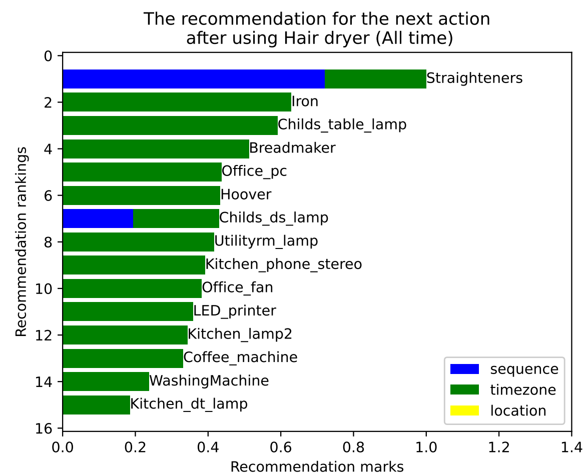

Fig. 3. Recommendation reason: sequence.

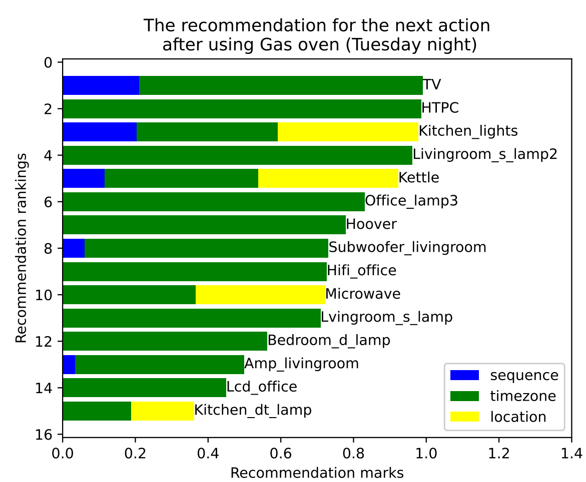

Fig. 4. Recommendation reason: timezone and location.

3) Reason-Location: This reason means both appliances are in the same location. As is shown in Fig. 4, when using the Gas Oven on Tuesday afternoon, the recommended appliance list includes the kettle, microwave, and other kitchen lights. This also aligns with common sense, as using appliances for cooking in the evening often involves turning on lights or using a kettle to boil water in the same location.

D. Results for Demand Response

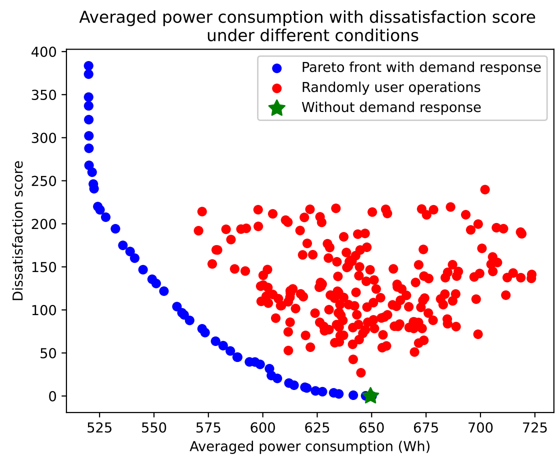

Fig. 5. Pareto front for demand response optimization.

It is assumed that the rated power of all appliances is adjustable within 0.8-1.2 times. In the simulation, it is assumed that there are three kinds of comfort levels: The first kind is adjustable appliances, including washing machines and dishwashers, the parameter c_i in (13) is set as 0.0001, which means it will produce less uncomfortable if the power is adjusted. The second kind is mainly entertainment appliances, including TV, HTPC, etc. The parameter c_i in (13) is set as 0.0003. The third kind is essential household appliances, including gas ovens, office PCs, etc. The parameter c_i in (13) is set as 0.0005, which means it will cause discomfort if the demand response adjusts it. The value of c_i is manually set based on life experience. The values 0.0001, 0.0003 and 0.0005 are chosen because they can normalize the dissatisfaction score to a range of 0-400. After optimization by the Generic Algorithm with the objective dissatisfaction calculated in (13) and the averaged power calculated in 14, the Pareto front generated by the optimization algorithm is shown in Fig. 5, where each point represents a set of device power settings.

In Fig. 5, the blue points represent the optimized Pareto Fronts, which have the most efficient energy consumption and comfort balance. The red points represent the values obtained by simulating user operations, assuming that the power range of each electrical appliance is a random multiple of 0.8 to 1.2 times the rated power, calculate the energy consumption and discomfort level (Averaged Power between 560 and 740 watt). It can be seen that the optimized Pareto Front blue points can achieve lower discomfort levels than the red points while saving energy consumption. The green star dots represent rated appliances' power (Averaged Power= 650 watt and Discomfort level= 0). It can be seen that compared to the green dots, the optimized power can be reduced by up to (650 watt-520 watt)/650 watt=20% with the proposed algorithm (The Averaged Power of the rightmost blue point is 520 watt).

According to the user's preference, the system can automatically assign power settings to each appliance based on the Pareto fronts towards a higher comfort level or energy-saving.

V. Conclusion

This paper presents a knowledge graph-based recommendation system for predicting the next appliance actions, along with explanations for the results of the recommendations and multi-objective optimization for demand response. The results demonstrate that the designed algorithm can accurately recommend the next possible action for users and provide explanations that align with common sense. By optimizing the expected outcomes of the recommendations, a Pareto front between comfort and average power consumption is achieved. Future work involves testing the algorithm in practical settings.

References

[1] M. Alaa, A. A. Zaidan, B. B. Zaidan, M. Talal, and M. L. M. Kiah, "A review of smart home applications based on internet of things," *Journal of network and computer applications*, vol. 97, no. 1, pp. 48–65, 2017.

[2] S. Das, S. Namasudra, S. Deb, P. M. Ger, and R. G. Crespo, "Securing iot-based smart healthcare systems by using advanced lightweight privacy-preserving authentication scheme," *IEEE Internet of Things Journal*, vol. 1, no. 1-15, pp. 1–1, 2023.

[3] D. Li, W.-Y. Chiu, H. Sun, and H. V. Poor, "Multiobjective optimization for demand side management program in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1482–1490, 2017.

[4] H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo, "Knowledge graph convolutional networks for recommender systems," in *Proceedings of the world wide web conference*. San Francisco, CA, USA, 2019, pp. 3307–3313.

[5] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "KGAT: Knowledge graph attention network for recommendation," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 950–958.

[6] H. Jeon, J. Kim, H. Yoon, J. Lee, and U. Kang, "Accurate action recommendation for smart home via two-level encoders and commonsense knowledge," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. Atlanta, USA, 2022, pp. 832–841.

[7] Y. Yao, M. M. Kamani, Z. Cheng, L. Chen, C. Joe-Wong, and T. Liu, "FedRule: Federated rule recommendation system with graph neural networks," *arXiv preprint arXiv:2211.06812*, vol. 6, no. 1, pp. 1–14, 2022.

[8] R. Das, S. Dhuliawala, M. Zaheer, L. Vilnis, I. Durugkar, A. Krishnamurthy, A. Smola, and A. McCallum, "Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning," *arXiv preprint arXiv:1711.05851*, vol. 1, no. 1, pp. 1–14, 2017.

[9] Q. Ai, V. Azizi, X. Chen, and Y. Zhang, "Learning heterogeneous knowledge base embeddings for explainable recommendation," *Algorithms*, vol. 11, no. 9, p. 137, 2018.

[10] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *Advances in neural information processing systems*, vol. 26, no. 1, pp. 1–14, 2013.

[11] S. Mirjalili and S. Mirjalili, "Genetic algorithm," *Evolutionary Algorithms and Neural Networks: Theory and Applications*, vol. 1, no. 1, pp. 43–55, 2019.

[12] J. Kelly and W. Knottenbelt, "The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five uk homes," *Scientific data*, vol. 2, no. 1, pp. 1–14, 2015.