A Spatio-Temporal Data-Driven Automatic Control Method for Smart Home Services

Jinrong Chen
College of Computer and Data Science, Fuzhou University
Fuzhou, China
Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing
Fuzhou, China
fzuchenjr@163.com

Zheyi Chen
College of Computer and Data Science, Fuzhou University
Fuzhou, China
Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing
Fuzhou, China
z.chen@fzu.edu.cn

Longhai Zheng
College of Computer and Data Science, Fuzhou University
Fuzhou, China
2984712343@qq.com

Xing Chen*
College of Computer and Data Science, Fuzhou University
Fuzhou, China
Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing
Fuzhou, China
chenxing@fzu.edu.cn

ABSTRACT

With the rapid development of smart home technologies, various smart devices have entered and brought convenience to people's daily life. Meanwhile, higher demands for smart home services have gradually emerged, which cannot be well satisfied by using traditional service provisioning manners. This is because traditional smart home control systems commonly rely on manual operations and fixed rules, which cannot satisfy changeable user demands and may seriously degrade the user experience. Therefore, it is necessary to capture user preferences based on their historical behavior data. To address the above problems, a temporal knowledge graph is first proposed to support the acquisition of user-perceived environmental data and user behavior data. Next, a user-oriented smart home service prediction model is designed based on the temporal knowledge graph, which can predict the service status and automatically perform the corresponding service for each user. Finally, a prototype system is built according to a real-world smart home environment. The experimental results show that the proposed method can provide personalized smart home services and well satisfy user demands.

CCS CONCEPTS

• **Human-centered computing**;

*corresponding author

WWW '22 Companion, April 25–29, 2022, Virtual Event, Lyon, France.

ACM ISBN 978-1-4503-9130-6/22/04…$15.00
https://doi.org/10.1145/3487553.3524705

KEYWORDS

Smart home, Temporal knowledge graph, Spatio-temporal data, Internet of things, Software Adaption

ACM Reference Format:
Jinrong Chen, Zheyi Chen, Longhai Zheng, and Xing Chen. 2022. A Spatio-Temporal Data-Driven Automatic Control Method for Smart Home Services. In *Companion Proceedings of the Web Conference 2022 (WWW '22 Companion), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3487553.3524705

1 INTRODUCTION

With the popularity of smart home, various smart devices have entered and brought convenience to people's daily life. Smart home technologies brought many conveniences to daily life through the diverse services provided by various smart devices [18]. At present, all major smart device manufacturers are committed to creating smart home ecosystems for their users. However, devices from different manufacturers are not able to interact with each other, and users who want to use a certain brand of smart devices must install that brand-specific APP to control the corresponding smart devices [1].

Meanwhile, higher demands for smart home services have gradually emerged. Smart home system with only simple device control function has lost its competitiveness, and users need more intelligent, senseless and personalized services [11]. To build a better smart home system without the collaboration of devices is impossible. For example, to automatically turn off the air conditioner in the bedroom and turn on the air conditioner in the living room when the user leaves bedroom and walks to the living room, the air conditioners in both rooms need to be linked with the user's positioning device. To satisfy complex user demands in different scenarios, it is necessary to make a higher level abstractions of devices and write corresponding service programs based on these abstractions. In addition, the needs of different users are not exactly the same, for example, each person has a different temperature acceptance level, therefore the threshold at which the air conditioner needs to be turned on varies. In order to satisfy individual user demands, existing solutions often allow users to set fixed device control rules.

However, there are several issues with existing methods. First, smart home devices from different manufacturers do not work together because each manufacturer is committed to constructing its ecosystem. The products of different brands are not interoperable and fragmented. It is difficult to form an integrated smart home experience with devices isolated. At the same time, the heterogeneity of smart devices is also a great challenge that developers have to face during building smart home system [8, 24]. Second, existing control methods of smart devices mostly rely on the user's remote control through terminal devices or fixed rules configured by the user to provide services. Remote control of smart home devices does not achieve true intelligence, and their device control commands still need to be issued manually by users. Although the preset rules have achieved a certain degree of intelligence, the rigid rules cannot adapt to the complex and changing environment. Moreover, numerous rule-setting process brings a certain using threshold and learning cost to users while using smart home. Some smart home systems even require users to spend a lot of time and energy to learn how to set rules [2]. Finally, user demands are based on the status of their own surrounding environment, which makes personalized smart home services inseparable from the perception of users' surrounding environment and its changes. Therefore, establishing a connection between the smart home system and environmental knowledge to provide real-time and flexible targeted services is essential for intelligent, senseless, and personalized smart home services.

At present, the data of individual devices alone cannot represent the user's surrounding environment well. The previous work models devices by introducing runtime knowledge graphs [6], abstracts the functions provided by each smart home device into multiple services with a single function and constructs a user-perceivable environment through a series of semantic rules. Based on the concepts above, the perceptible status of environment and device around the user can be reflected, and the control command of devices can be realized through a runtime knowledge graph. Some services based on fixed rules can be achieved through the existing runtime knowledge graph model. However, as mentioned above, in addition to the configuration of rules being very tedious, the preset rules do not take into account the contextual data of user behavior, which cannot well meet the changeable, complex, and personalized demands of users.

Therefore, a spatio-temporal data-driven smart home service control method is proposed, which can learn preference from users' historical behavior data, build a personalized model to predict users' operation, and use a runtime knowledge graph to automatically execute service operations for users. This method extends the original runtime knowledge graph model in the time dimension to provide historical data. First, we extend the knowledge graph to temporal knowledge graph with temporal capability. The TKG possesses the ability to provide the knowledge graph status data at any moment in the past specific period. We read and record the device status of each moment based on the time interval specified by the scenario and construct the knowledge graph model of the corresponding time slice. By constructing the knowledge graph time slices over a period, we can obtain the environmental status values and their changes perceived by users in that time slice. Finally, to solve the challenge of complex rule-setting and realizing the self-updating of rules, we propose a spatio-temporal data-driven automatic execution method for smart home services. The method models user preferences by collecting user-perceived environmental history data and corresponding service status through temporal knowledge graph (TKG), makes service status prediction based on the environmental status within a time window, and translates the prediction into specific device operations using a TKG. The method liberates users from complex and tedious smart home rule-setting operations, greatly improves the convenience of smart home usage and brings intelligent and senseless smart home user experience to users.

2 RELATED WORK

The smart home [13] is an important application scenario of an IoT [22] system. How to realize a smarter smart home automation system has been widely concerned by academia and industry. Smart home systems have been extensively researched in energy consumption, health monitoring. Li et al. [3, 16, 17] proposed an intelligent decision system based on machine learning and other methods to combine smart home and smart grid to reduce home energy consumption. De et al. [7] suggested a centralized smart building decision system based on wireless sensors and actuator networks to achieve energy-saving goals for large buildings. Khan et al. [15] proposed the analysis of sensor data history for pump load prediction in a smart home system. Forbes et al. [10] proposed to use behavior recognition in smart home systems for home health detection.

Smart home systems are built on top of various sensors and smart devices that provide smart home services. How to solve the communication problem between heterogeneous devices and decouple specific devices and upper-layer applications to reduce the complexity of development is an unavoidable problem for all IoT applications including smart homes. The current common approach to solve this problem is to incorporate a middleware layer3 [25]. Filho et al. [9] proposed an intelligent decision system based on the fog computing paradigm, which handles the differences in programming interfaces and connection methods of heterogeneous devices by adding a message queue layer based on MQTT [20].

In addition to the heterogeneity issues mentioned above, how to provide more personalized, intelligent, and senseless services in smart home systems has become a hot issue [2, 12, 21]. Ghiani [11] proposed a trigger-action rule-based smart home system, which automates smart home services by user's configuration of rules, based on manually configured rules that rely on user's expertise, while the manual configuration approach does not cope well with more complex and larger number rules. Chahuara et al. [4] suggested using voice control to improve the interaction quality of smart homes. Jin et al. [14] proposed a personalized indoor temperature control system based on deep learning, but its applicability is limited to temperature control services in smart homes and

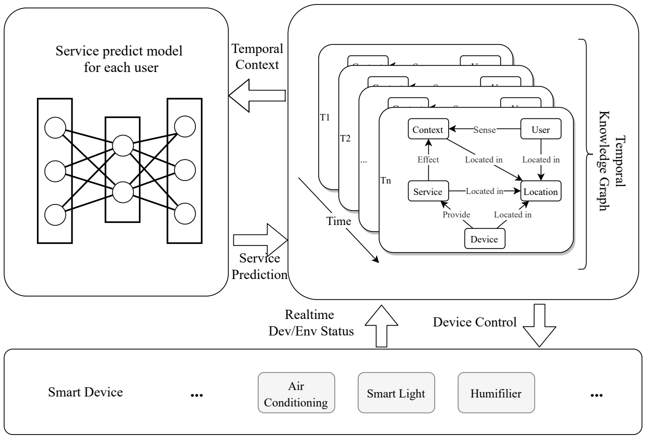

Figure 1: Method Overview

cannot be used for other smart home devices and services. Deephome [19] based on deep learning, learns behavioral habits from user's history, and predicts user behavior to automatically perform device operations for users. Deephome is better for intelligent and senseless services, but it is only for highly customized smart home environments.

3 SYSTEM ARCHITECTURE

To achieve higher intelligence, a spatio-temporal data-driven smart home service control method is proposed in this paper. The overall framework of this method is given in Figure 1. It is an extension of the previous work [6], which automatically operates smart home devices for users by learning their historical behavior records and predicting their possible future actions. The method consists of two aspects:

1) Smart home temporal knowledge graph construction method.

2) DNN-based decision model for smart home services.

First, a temporal data-driven knowledge graph construction method for smart homes is proposed. The TKG extends the existing runtime knowledge graph in the time dimension and describes the smart home scenario while recording the relationship and its changes between various entities in the scenario over time. By constructing the TKG of the smart home, the user-perceived environmental data change records and the user-perceived service status records within a period of time can be obtained based on the temporal data embedded in the TKG.

Second, a user-oriented smart home service prediction model is proposed. The model reads environmental status data perceived by each user within a certain time window and uses DNN to predict the service status perceived by the user at the next moment. Using the transformation rules of the TKG, the prediction results are transformed into the operations of corresponding devices and these operations are automatically executed for the users to realize an intelligent and senseless smart home service experience.

4 CONTEXT-AWARE RUNTIME KNOWLEDGE GRAPH FOR SMART HOME

Context-aware runtime knowledge graph for smart home is our prior work [6], the core concepts of the runtime knowledge graph are introduced in this section. Runtime knowledge graph constructs smart home scenario model by defining five smart home knowledge graph concepts and the relationship between these concepts. The runtime knowledge graph is defined as follows:

$$KG = \{(x_i, r_{i,j}, x_j) | x_i, x_j \in E, r_{i,j} \in R, i \neq j\} \tag{1}$$

$$E = \{User, Service, Location, Context, Device\} \tag{2}$$

$$R = \{Effect, LocatedIn, Sense, Provide\} \tag{3}$$

Where E is the set of all type instances, containing the five core concept *Context*, *Device*, *Service*, *User*, and *Location*, mentioned above. R is the set of relationships, including $Effect$, *LocatedIn*, *Sense* and *Provide*.

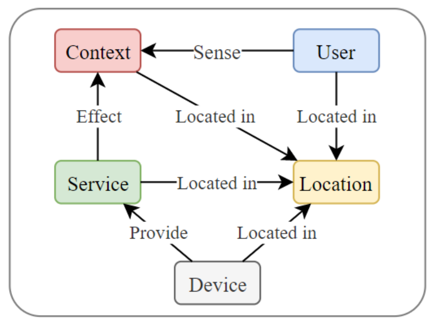

Figure 2: Concept Model of Runtime Knowledge Graph

As shown in Figure 2, *Location* is a specific area in the home, containing information about the physical location of the area. Other instances can be associated with specific locations, indicating that their scope of effectiveness or activity is in a specific location. A single attribute *LName* is used to represent the identifier location.

Device is an abstraction of a concrete smart device. Each device instance owns attributes that correspond to its name, location, running status, and information about the functions that the device can provide. A device is represented by a tuple $\langle DName, LName, Status, \{Key_1, Key_2, ..., Key_m\}\rangle$.

Service is the abstraction of the device function. As a device may provide more than one function, a device may correspond to multiple services. A service is defined as $\langle DName, LName, CType, Effect, Status, SValue \rangle$, the set of services is defined as S. Each service has an impact on the corresponding context, e.g. the cooling of air conditioners can reduce the room temperature, smart lights can increase the brightness, etc. The runtime knowledge graph indirectly controls the corresponding devices by controlling corresponding services to achieve the unified control of heterogeneous devices.

User is the object of service in the smart home, such as specific residents in the house. Each user instance contains the user's name and location, defined as $\langle UName, LName\rangle$, the set of users is denoted as U.

Context is the specific environmental status that a user can sense, such as temperature, humidity, brightness, etc., containing the user and location information associated with that context, defined as $\langle$ *CType*, *LName*, *CValue*$\rangle$, the set of context denoted as C. Each context instance can be associated with a specific user and location. The association expresses that the user perceives a certain environmental status at a certain location. Since the layout of each room in the house is diverse and the contexts of each room are also different, each room has its context instance. When the user moves from location A to B, the user will be unassociated from the context instance of location A and associated with the instance of B. The association is reflected by the relationship in the knowledge graph.

Runtime knowledge graph uses the runtime software architecture model [5, 6, 23] to construct a runtime model and maintain two-way synchronization between the knowledge graph and real world. i.e., while changes in the real world such as temperature raise and resident's location movement are reflected in the runtime knowledge graph, operations on service instances in the runtime knowledge graph are also transformed into operations of corresponding devices through execution rules. Based on the runtime knowledge graph model, it is easy to realize scenario-based device function control and environment status reading. Developers only need to configure specific contextual knowledge to automatically construct the runtime knowledge graph according to the rules.

5 TEMPORAL KNOWLEDGE GRAPH FOR SMART HOME

5.1 Temporal Knowledge Graph Model for Smart Home

In the real world application of smart home, the indoor environment status will change with the external environment state which makes environment uncomfortable. In order to counteract the changes, residents often utilize function of smart device, such as turning on the humidifier when feel the air is too dry, or turning on the air conditioning cooling function when it is hot. However, the current control methods generally require users to manually send their service requests, such as opening the corresponding applications on their phone then following several tedious steps or manually turning on each device by hand. Therefore, the challenge is how to achieve the automatic perception of user needs, so that the smart home control system can turn on the corresponding device functions for users autonomously. In a real world smart home scenario, the user's location will inevitably move between different rooms in the scenario. The environmental status sensed by the user and the devices to be controlled are determined by the user's location at that moment. How to deal with the different environmental status and devices perceived by the user at different locations and time is a problem that cannot be ignored. To capture changes in environmental status, sense user needs, and proactively provide various services to users, it is important to learn user preferences from their historical behavior data. However, existing runtime knowledge graph models cannot provide good support for historical behavioral data. Therefore, we extend the existing smart home runtime graph model so that it can reflect the process of environment and device state change in smart home scenario through historical data in addition to expressing the real-time status of smart home scenario. The following content introduces the concepts of the temporal knowledge graph model.

The smart home TKG model is shown in Figure 3. The model extends runtime knowledge graph model in the time dimension and is able to represent the changing process of concept instances and relationships within a certain time period, in addition to describing the concepts and relationships between them.

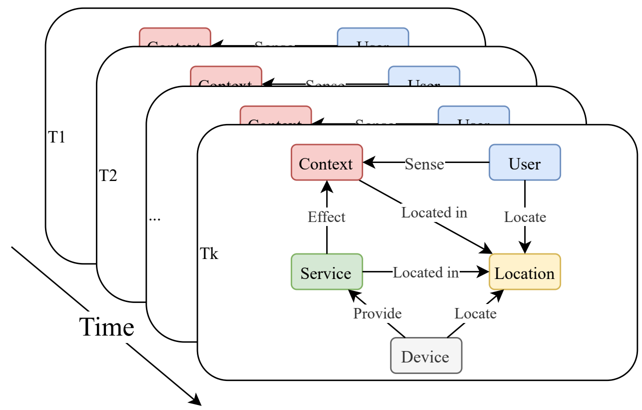

Figure 3: Temporal Knowledge Graph

A TKG can be understood as a knowledge graph stream with time tags, i.e., a collection of states of the knowledge graph at a given time interval ΔT in a given time period $\{T_1, \ldots, T_n\}$. The data includes knowledge graph state KG^k and time tag t^k at each moment, which can completely describe the state of the smart home scenario of corresponding moment, including the user's location, the user's perceivable environment status, the status of each smart device, etc. The continuous knowledge graphs can describe the change of smart home status in a period of time. The formal definition of the TKG is given below.

$$TKG = \{ \left(t^k, KG^k \right) | T_1 \leq t^k \leq T_n, t^{k+1} = t^k + \Delta T \} \tag{4}$$

5.2 Smart Home Control Model Base on Temporal Knowledge Graph

In order to achieve better control of smart home services, we need to focus only on the environmental status that user perceived. At the same time, it should be noted that the user's position at home changes constantly due to the user's activities, so the environmental status that the user can perceive may be different over time. Based on the TKG model mentioned above, we propose a user-oriented smart home service control model. The model extracts the service status and environment status sequences which user concerns and predicts the relevant service status for them by using the spatio-temporal semantic description capability of the TKG. The relevant concepts of the model are introduced below. Knowledge graph KG^t at time t is introduced:

$$KG^t = \{(x_i, r_{i,j}, x_j) | x_i, x_j \in E^t, r_{i,j} \in R^t, i \neq j\}. \tag{5}$$

We take each user $u(u \in E^t \wedge u.Type = User)$ from KG^t and define the subset of context Ctx_u^t that user u can perceive at moment t as follows:

$$Ctx_u^t = \{ctx_1, ctx_2, ..., ctx_w | \exists (u \xrightarrow{Sense} ctx_i) \in KG^t\}. \quad (6)$$

where w is the number of user interested context, and the type ($CType$)of any two ctx is different. In particular, it should be noted that the environmental status described in Ctx_u^t are partial environmental status in the whole smart home environment, the values of environmental status attributes perceived by a specific user at a specified moment, instead of the environmental state values of the entire smart home scenario. Due to the change of user's location, Ctx_u^t and Ctx_u^{t+1} may be completely different. The model proposed in this paper is built around the user, and the data relied on is always determined by the spatial location of the user. By constructing Ctx_u^t , we can extract the environmental status that the user can perceive at any moment. The spatial information is implicit in the data and transparent to the upper decision algorithm.

Based on Ctx_u^t , we can further deduce that the subset of Service instances Srv_u^t which have an impact on the Ctx_u^t at time point t.

$$Srv_u^t = \{srv_1, srv_2, ..., srv_r | \exists (srv_i \xrightarrow{Effect} ctx_j) \in KG^t, ctx_j \in Ctx_u^t\} \quad (7)$$

According to the definition of Ctx_u^t, Srv_u^t which can influence Ctx_u^t is equivalent to the environment that the user can perceive, i.e., the service instance that can provide functions to the user at time t, which is the service we want to control.

Based on the TKG as shown in Figure 4, a user-oriented decision model is proposed. The model takes context instances perceived by a specific user in the past period as input, predicts, and controls the service instances Srv_u^t that can be perceived at present utilizing a deep neural network model. Where l is a fixed time window length, w is the number of the user interested state types, r is the number of services to be controlled. In the proposed overall framework, each served user has an independent service prediction model that relies on the DNN-based algorithm introduced in the next section to implement service status prediction based on temporal data.

6 DNN-BASED SMART HOME SERVICE CONTROL

6.1 Input and Output of Smart Home Service Prediction

To achieve the service status decision mentioned above, using deep neural networks for service status prediction is proposed. How to get the output of the neural network from Ctx and Srv instances describes in this section.

According to the definition of the TKG, we extract the environmental status values ($CValue$) in each Ctx_u^t by a fixed order of environmental types to obtain the corresponding $CValue$ sequence E_u^t.

$$E_u^t = (c_1, c_2, ..., c_w), \quad (8)$$

where $c_i = ctx_i.CValue$, which is the value of ith environmental attribute. Thus, the input data X is the user-perceived environmental

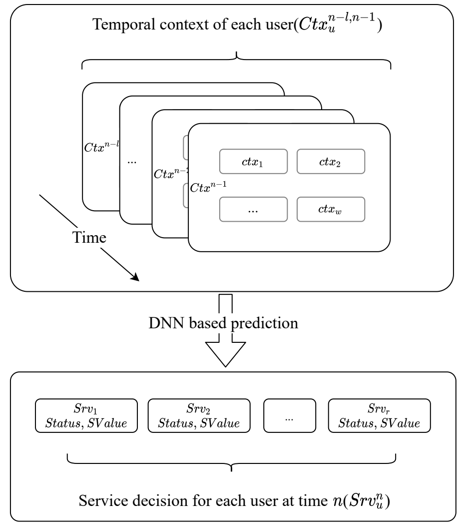

Figure 4: User-oriented Prediction Model for Smart Home

status context data within a certain time window:

$$X = [E^1, E^2, ..., E^k]^T. \quad (9)$$

In the existing runtime knowledge graph definition, if a service needs to be controlled, changing the running status of the service can be achieved by specifying its status value ($Status$) and parameter value ($SValue$). Thus, the output Y of the model is defined as:

$$Y = (s_1, v_1, s_2, v_2, ..., s_r, v_r), \quad (10)$$

where s_i and v_i are the corresponding $Status$ and $Svalue$ in Srv_u^t, r is the number of services to be controlled.

6.2 Network Structure

Based on the definition above, a multi-layer deep neural network (DNN) model is used to achieve the mining of the implicit relationship between input and output. Its overall structure is shown in Figure 5. The network consists of an input layer L_{in}, three fully connected hidden layers L_1, L_2 ,L_3 and an output layer L_{out} . The number of neurons in the input layer (N_{input}) is determined by number of environments p and length of time window k. The number of neurons in the output layer N_{output}, which is determined by the number of services to be predicted. The number of neurons in the three hidden layers are N_1, N_2 and N_3. The fully connected structure is used between layers. Treat L_{in} as L_0 and L_{out} as L_4 then output the of each layer can be described as:s

$$L_i = \text{Sigmoid}(L_{i-1} \cdot W_i + B_i). \quad (11)$$

The activation function used for each layer of the network is $Sigmoid$, which is defined as:

$$\text{Sigmoid}(a) = \frac{1}{(1 + e^{-a})}. \quad (12)$$

The mean squared error (MSE) is used as loss function:

$$MSE = \frac{1}{n} \sum (\hat{y}_i - y_i)^2. \quad (13)$$

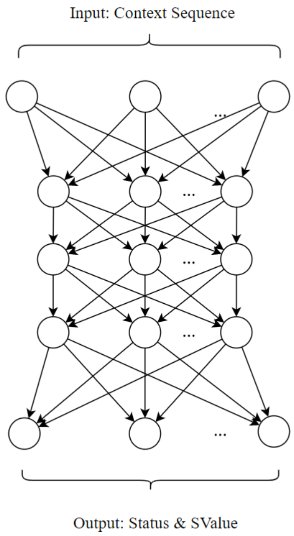

Figure 5: Network Structure

6.3 Network Training and Prediction

6.3.1 Training. Algorithm 1 describes the training process of the given DNN model. The decision model containing the user's behavioral habits can be obtained by training the network.

Algorithm 1 Model training

Input: Time window length k; Number of contexts q; Threshold for status th; Training Data set DS
Output: Trained Model M
adjust the number of neurons according to k and q;
initialize the weight matrices in W_1,W_2,W_3,W_{out} with random value in the range [0,1] ;
initialize biases B_1,B_2,B_3,B_{out} with zeros;
while more than 64 samples in D **do**
    randomly draw 64 samples from D as network input X;
    compute the output of each layer by equation 11;
    compute the loss according equation 13;
    adjust the weights and biases in the direction of the decreasing gradient;
return trained model M

6.3.2 Prediction. According to the weight W_1, W_2, W_3, W_{out} and bias B_1, B_2, B_3 and B_{out} trained from training process, the neural network is initialized, and the service status decision result *Status* is obtained by collecting the environment status matrix *Env* of the first k time points into the model under the support of temporal knowledge graph. The decision result *Status* is then output back to the TKG. With its control capability, TKG issues command to the corresponding smart home devices to achieve the control of specific device function which brings a senseless smart home user experience.

7 EVALUATION

In order to verify the effectiveness of the proposed method in smart home systems, a prototype system is built to evaluate the method for real world scenarios in this paper.

7.1 Temporal Knowledge Graph for Prototype

A smart home scenario prototype system is built as shown in Figure 6. It includes three areas: living room(L_1), bedroom(L_2) and balcony(L_3). Each area contains various smart devices for regulating the environment. Typical types of context we build in this scenario are temperature(C_{l_1}), humidity (C_{l_2}) and brightness(C_{l_3}). Each smart device in the scenario can provide *Monitor*, *Increase*, *Reduce*, and other related services to each context. Based on this scenario, we construct the smart home temporal knowledge graph as shown in the figure, which contains 3 location instances, 6 device instances, 9 context instances, 20 service instances, and 1 user instance U_1. Based on this instance scenario, we can build and record runtime knowledge graph to construct smart home temporal knowledge, which provides the basis of user model training and service state prediction. The model proposed in this paper constructs an independent prediction model (M_m) for each user, such as U_1 corresponds to M_1.

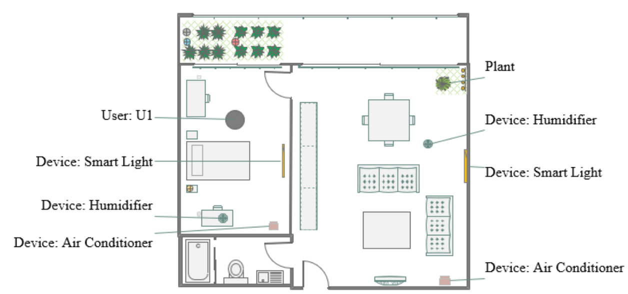

Figure 6: Prototype scenario diagram

Figure 7 demonstrates the temporal knowledge graph. The movement of U_1 from living room(L_1) and to bedroom(L_2) is reflected. The context, service, and corresponding device sensed by the user are on the left side at first. After the location of user changed, the instances on the right side are what the user is concerned.

7.2 Service Prediction

Due to user privacy and hardware limitations, this experiment is based on the publicly available smart home sensor dataset, the invited volunteers provide their behavioral data (work and rest schedules, environmental sensitivity, etc.) and the prototype system built according to public environmental change pattern to simulate user's real behavior to generate artificial dataset as the experimental dataset. The experimental code is written in Python 3 and based on the TensorFlow 2 deep learning framework. The experimental runtime environment is a personal computer with an Intel(R) Core(TM) i5-9500 CPU @ 3.00 GHz with 16.0 GB memory and a Windows 10 operating system installed.

The dataset contains two months of environmental and device data for a single-user household with a sampling interval of fewer

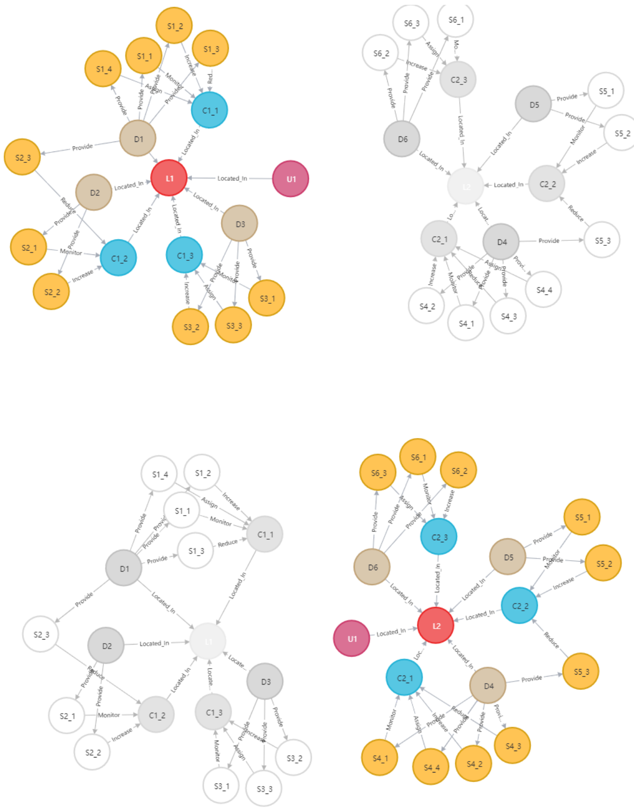

Figure 7: Temporal Knowledge Graph Demo

than 15 minutes. The environmental data contains temperature, humidity, and brightness. The device status data contains air conditioner, smart light, and air humidifier . The corresponding TKG is constructed for the environment data and device status data at each moment. The context and service instance perceived by the user at each moment is obtained through the abstraction ability of the TKG. Based on that context and service, we can further obtain the values of temperature, relative humidity, and brightness around the user at that moment, as well as the status data of air conditioners, humidifiers, and smart lights that have an effect on the user. Because the range of each data type varies, this experiment will normalize each type of data to the range of [0,1] by its type. The normalization formula is as follows:

$$\mathrm{X}' = \frac{X - X_{\min}}{X_{\max} - X_{\min}}. \tag{14}$$

To make it clear, we chose 3 typical services to predict. The correctness of prediction of each device is judged by both the $Status$ (s) and $SValue(v)$. The result of $Status$ is obtained through equation 15, where th is a preset threshold. We compare the result of the network and the label, only s' is the same and the square of the difference between v and label value $\hat{v}$ less than 0.02 is marked as

Table 1: Experiment result

Service	Precision	Recall	F1
Temperature Reduce	0.927	0.872	0.898
Humidity Increase	0.959	0.949	0.954
Brightness Increase	0.914	0.930	0.922
Average	0.933	0.917	0.925

a successful prediction.

$$s' = \begin{cases} 1 & \text{if } s \geq th, \\ 0 & \text{if } s < th. \end{cases} \tag{15}$$

A time window of $k = 6$ length is used in the entire time slice of the dataset and all environmental data within the window are used as neural network inputs to predict the device state at the next moment, yielding a total of 5946 samples. Proportionally, 80% of the dataset is used as the training set and 20% as the test set.

This experiment involves the accuracy of service status prediction, so $precision$, $recall$, and $F1$ are used as evaluation metrics to judge the prediction effectiveness of each service status separately. They are defined as:

$$percision = \frac{TP}{TP + FP} \tag{16}$$

$$recall = \frac{TP}{TP + FN} \tag{17}$$

$$F1 = \frac{2 \times percision \times recall}{percision + recall}. \tag{18}$$

where TP denotes the case where the user turns on the service and the prediction result is also on, FP denotes the case where user does not turn on the service and the prediction result is on, FN denotes the case where the user turns on the service but the prediction result is off, and TN denotes the case where user does not turn on the service and the prediction result is also off. Therefore, $precision$ indicates the proportion of samples in which the user actually turns on the service and the predicted result of the service is on; the $recall$ rate indicates the proportion of samples in which the predicted result of the model is on and the user wants the service to be on in the predicted result, and $F1$ is the summed average of $precision$ and $recall$.

7.3 Result

The result of the given test set is shown in Table 1. In terms of the three metrics, the average prediction of all services can reach 90%, with the temperature reduction service having slightly lower completeness, but it can also reach more than 87%.

To further investigate the reasons for the errors of these erroneous samples, this paper randomly sampled form all the erroneous samples and asked volunteers to re-judge the samples obtained from the sampling again. The proportion of samples in which volunteers thought the proposed method was misjudged is 30.8%. The proportion of volunteers who thought that the data of the given sample did not conform to their preference(i.e., the judgment result of our method was correct) is 53.8%. The proportion of samples which the volunteers thought that the situation would not change original device state was 15.4%. It can be seen that a large part of the error comes from the randomness and lagging nature of the human decision-making process.

8 CONCLUSION

To address the challenges of device heterogeneity faced in smart home, a spatio-temporal data-driven smart home service control method for temporal data is proposed, which provides historical data from user behavior through temporal knowledge graph and learns user preference from temporal data using a deep neural network. The result of the evaluation demonstrates the proposed method can successfully predict users' preferences and build a more intelligent, senseless, and personalized smart system. In future research, we will further explore the prediction method for temporal data to achieve a more intelligent prediction algorithm. In addition, the proposed system is also deployed to a real smart home scenario and the problems encountered in the actual scenario are investigated.

ACKNOWLEDGMENTS

This work was partially supported by the National Key R&D Program of China under Grant No. 2018YFB1004800 and the Natural Science Foundation of Fujian Province for Distinguished Young Scholars under Grant No. 2020J06014.

REFERENCES

[1] Fawzi Behmann and Kwok Wu. 2015. *Collaborative internet of things (C-IoT): for future smart connected life and business.* John Wiley and Sons, Inc, Hoboken.

[2] Ivan A. Berg, Oleg E. Khorev, Arina I. Matvevnina, and Alexey V. Prisjazhnyj. 2017. Machine learning in smart home control systems - Algorithms and new opportunities. *AIP Conference Proceedings* 1906, 1 (Nov. 2017), 070007. https://doi.org/10.1063/1.5012333 Publisher: American Institute of Physics.

[3] Khac-Hoai Nam Bui, Jason J. Jung, and David Camacho. 2018. Consensual Negotiation-Based Decision Making for Connected Appliances in Smart Home Management Systems. *Sensors* 18, 7 (July 2018), 2206. https://doi.org/10.3390/s18072206 Number: 7 Publisher: Multidisciplinary Digital Publishing Institute.

[4] Pedro Chahuara, François Portet, and Michel Vacher. 2017. Context-aware decision making under uncertainty for voice-based control of smart home. *Expert Systems with Applications* 75 (June 2017), 63–79. https://doi.org/10.1016/j.eswa.2017.01.014

[5] Xing Chen, Aipeng Li, Xue'e Zeng, Wenzhong Guo, and Gang Huang. 2015. Runtime model based approach to IoT application development. *Frontiers of Computer Science* 9, 4 (2015), 540–553.

[6] Yiyan Chen, Zhanghui Liu, Zhiming Huang, Chuangshumin Hu, and Xing Chen. 2019. A Services Development Approach for Smart Home Based on Natural Language Instructions. In *International Conference on Software Engineering and Knowledge Engineering*. 367–478. https://doi.org/10.18293/SEKE2019-173

[7] Claudio de Farias, Henrique Soares, Luci Pirmez, Flávia Delicato, Igor Santos, Luiz Fernando Carmo, José de Souza, Albert Zomaya, and Mischa Dohler. 2014. A control and decision system for smart buildings using wireless sensor and actuator networks. *Transactions on Emerging Telecommunications Technologies* 25, 1 (Jan. 2014), 120–135. https://doi.org/10.1002/ett.2791

[8] Laith Farhan, Sinan T. Shukur, Ali E. Alissa, Mohmad Alrweg, Umar Raza, and Rupak Kharel. 2017. A survey on the challenges and opportunities of the Internet of Things (IoT). In *2017 Eleventh International Conference on Sensing Technology (ICST)*. 1–5. https://doi.org/10.1109/ICSensT.2017.8304465 ISSN: 2156-8073.

[9] Geraldo P. Rocha Filho, Rodolfo I. Meneguette, Guilherme Maia, Gustavo Pessin, Vinícius P. Gonçalves, Li Weigang, Jó Ueyama, and Leandro A. Villas. 2020. A fog-enabled smart home solution for decision-making using smart objects. *Future Generation Computer Systems* 103 (Feb. 2020), 18–27. https://doi.org/10.1016/j.future.2019.09.045

[10] Glenn Forbes, Stewart Massie, Susan Craw, Lucy Fraser, and Graeme Hamilton. 2020. Representing Temporal Dependencies in Smart Home Activity Recognition for Health Monitoring. In *2020 International Joint Conference on Neural Networks (IJCNN)*. 1–8. https://doi.org/10.1109/IJCNN48605.2020.9207480 ISSN: 2161-4407.

[11] Giuseppe Ghiani, Marco Manca, Fabio Paternò, and Carmen Santoro. 2017. Personalization of Context-Dependent Applications Through Trigger-Action Rules. *ACM Transactions on Computer-Human Interaction* 24, 2 (April 2017), 14:1–14:33. https://doi.org/10.1145/3057861

[12] Shivani Jadon, Arnav Choudhary, Himanshu Saini, Utkarsh Dua, Nikhil Sharma, and Ila Kaushik. 2020. *Comfy Smart Home using IoT.* SSRN Scholarly Paper ID 3565908. Social Science Research Network, Rochester, NY. https://doi.org/10.2139/ssrn.3565908

[13] Li Jiang, Da-You Liu, and Bo Yang. 2004. Smart home research. In *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826)*, Vol. 2. 659–663 vol.2. https://doi.org/10.1109/ICMLC.2004.1382266

[14] Jing Jin, Shaolong Shu, and Feng Lin. 2019. Personalized Control of Indoor Air Temperature Based on Deep Learning. In *2019 Chinese Control And Decision Conference (CCDC)*. 1354–1359. https://doi.org/10.1109/CCDC.2019.8833088 ISSN: 1948-9447.

[15] Nida Saddaf Khan, Sayeed Ghani, and Sajjad Haider. 2018. Real-Time Analysis of a Sensor's Data for Automated Decision Making in an IoT-Based Smart Home. *Sensors* 18, 6 (June 2018), 1711. https://doi.org/10.3390/s18061711 Number: 6 Publisher: Multidisciplinary Digital Publishing Institute.

[16] Ding Li and Sudharman K. Jayaweera. 2014. Reinforcement learning aided smart-home decision-making in an interactive smart grid. In *2014 IEEE Green Energy and Systems Conference (IGESC)*. 1–6. https://doi.org/10.1109/IGESC.2014.7018632

[17] Ding Li and Sudharman K. Jayaweera. 2015. Distributed Smart-Home Decision-Making in a Hierarchical Interactive Smart Grid Architecture. *IEEE Transactions on Parallel and Distributed Systems* 26, 1 (Jan. 2015), 75–84. https://doi.org/10.1109/TPDS.2014.2308204 Conference Name: IEEE Transactions on Parallel and Distributed Systems.

[18] Ching-Hu Lu. 2015. IoT-enabled smart sockets for reconfigurable service provision. In *2015 IEEE International Conference on Consumer Electronics - Taiwan*. 330–331. https://doi.org/10.1109/ICCE-TW.2015.7216927

[19] Bo Mao, Ke Xu, Yuehui Jin, and Xiaoliang Wang. 2018. DeepHome: A Control Model of Smart Home Based on Deep Learning. *Chinese Journal of Computers* 41, 12 (2018), 2689–2701.

[20] Mqtt.Org. 2021. MQTT - The Standard for IoT Messaging. Retrieved 2021-10-20 from https://mqtt.org/

[21] Geraldo P. R. Filho, Leandro A. Villas, Heitor Freitas, Alan Valejo, Daniel L. Guidoni, and Jó Ueyama. 2018. ResiDI: Towards a smarter smart home system for decision-making using wireless sensors and actuators. *Computer Networks* 135 (April 2018), 54–69. https://doi.org/10.1016/j.comnet.2018.02.009

[22] Biljana L. Risteska Stojkoska and Kire V. Trivodaliev. 2017. A review of Internet of Things for smart home: Challenges and solutions. *Journal of Cleaner Production* 140 (Jan. 2017), 1454–1464. https://doi.org/10.1016/j.jclepro.2016.10.006

[23] Hui Song, Gang Huang, Franck Chauvel, Yingfei Xiong, Zhenjiang Hu, Yanchun Sun, and Hong Mei. 2011. Supporting runtime software architecture: A bidirectional-transformation-based approach. *Journal of Systems and Software* 84, 5 (2011), 711–723.

[24] Summia Taj, Uniza Asad, Moeen Azhar, and Sumaira Kausar. 2019. Interoperability in IOT based smart home: A review. *Review of Computer Engineering Studies* 5, 3 (Sept. 2019), 50–55. https://doi.org/10.18280/rces.050302

[25] Ying Zhang, Guohui Tian, Senyan Zhang, and Cici Li. 2020. A Knowledge-Based Approach for Multiagent Collaboration in Smart Home: From Activity Recognition to Guidance Service. *IEEE Transactions on Instrumentation and Measurement* 69, 2 (Feb. 2020), 317–329. https://doi.org/10.1109/TIM.2019.2895931 Conference Name: IEEE Transactions on Instrumentation and Measurement.