Transactions on Computer Science and Intelligent Systems Research
ISSN: 2960-1800, eISSN: 2960-2238 | Volume 9
AIDML 2025

Towards a Unified and Adaptive Smart Home System: Deep Learning and iot Integration

Denghan Xu *
Shaanxi University of Science & Technology, Xi'an, China
* Corresponding Author Email: kjggyf@foxmail.com

Abstract. With the wide application of IoT technology in daily life, the smart home industry has achieved rapid development in recent years. However, due to the different communication methods and fragmented functions between different smart home devices, this makes users face higher complexity in management and use. The goal of the smart home management and control platform is to integrate the data monitoring and control functions of multiple devices in a heterogeneous network environment to provide one-stop home services. However, in the face of problems such as coordination between devices, numerous environmental parameters and difficult to accurately identify user needs, how to build a unified management and control system that is imperceptible, accurate and intelligent has become a core problem to be solved in the process of smart home platformization. The system not only can be used for normal residential home user, but also can be used for hotel and other subsystems. The system realizes the remote control of appliances combine locality control the results show that the system hardware is simple, lower cost, reliability and easy to expand.

Keywords: Technology, smart, remote control, multiple devices.

1. Introduction

With the increase of market demand, modern smart home has received more and more attention. At present, smart home products have been applied to residential decoration, and have achieved outstanding use effects, and have been widely praised by people. The application of high automation technology in modern indoor home furnishings has become the trend of current technological development [1]. To this end, the relevant researchers will also be committed to the research of smart home, to further improve the level of home automation, to provide greater convenience for people's lives, and then to improve the home living environment. With the help of various advanced information technologies, people have built home systems that integrate various facility functions and information, so as to realize the powerful application functions of all facilities [2]. The various equipment in the room can actively provide people with corresponding life information and meet the needs of more diversified use. At the same time, the connection between smart technology and indoor living technology will become closer. In the future, environmental intelligence technology will be applied to sensing equipment. With the further upgrade [3]of the communication network, people can also use their smartphones to remotely manage the environment around their homes.

2. Methodology

The purpose of this study is to design a low-cost and easily scalable smart home control system, which can realize the combination of remote control and local control of household appliances. In the design process, we combined two main methods: one is the design of the control terminal and microcontroller gateway based on the Android platform [4], and the other is the intelligent control model DeepHome based on deep learning [5].

2.1. Android-based smart home control system

Firstly, a smart home system based on Android mobile phone or tablet computer as the control terminal was designed. The system uses the open source resources of the Android platform to exchange data with the home control gateway through the WIFI communication protocol. The control gateway uses single-chip microcomputer technology to connect home appliances to realize the control and status feedback of home equipment [4]. Users interact with home devices through a graphical interface on a mobile phone or tablet, which realizes the combination of remote control and local control of the device. The main advantages of this solution are its low cost, simple hardware configuration, and strong scalability.

In the process of system design, the following key technologies are adopted:

2.1.1. Android client design:

The interactive design of the Android graphical interface simplifies the control operation and improves the user experience.

2.1.2. Control gateway design:

Using single-chip microcomputer technology and WIFI network [4], the user's operation instructions are transmitted to the home equipment and the operating status of the equipment is fed back.

2.2. Deep Learning-based Smart Home Control Model (DeepHome)

In order to further improve the accuracy and intelligence of smart home control, we have introduced deep learning algorithms to optimize the control decisions of home devices. The DeepHome model is based on the autoencoder structure of the deep neural network, and the use characteristics of home appliances are modeled through unsupervised learning [5]. The training process of the model is divided into the following stages:

(1) Data collection and pre-processing: Collect home environment data, including user behavior, device status, and environmental parameters (such as temperature, humidity, brightness, etc.) through the back-end platform of the product, and convert them into structured training data-based pre-training: The autoencoder is used to perform unsupervised learning on each device and extract device-level features to obtain a general device model.

(2) On the basis of the deep learning preparation model, supervised training is carried out through a multi-layer feedforward neural network (DeepHome model) to form a comprehensive home control model. The model is able to make intelligent decisions based on real-time environmental data and historical data, and automatically control the working status of home equipment.

2.3. Experimental design and evaluation to verify the accuracy and performance of the DeepHome model.

The experiment is divided into three stages: pre-training, training and testing, and the data of the test set is simulated by the product back-end platform to simulate the home life data of different users. In the experiments, we evaluated the convergence speed and accuracy of the model, as well as its ability to predict real data samples.

The smart home system proposed in this study combines the control terminal design of the Android platform with the low-cost, high-efficiency and easy-to-scale home control system based on the deep success. Future research will continue to optimize the generalization ability of the model, and improve its adaptability and decision-making accuracy in different home environments through online learning mechanism.

3. Results

In order to verify the error calculation after combining the Android platform-based smart home control system with the deep learning-based smart home management and control model (DeepHome), we can create an experimental dataset for model testing. Here's one way to generate data and calculate errors:

3.1. Data generation steps

3.1.1. Environmental & User Data Collection:

The simulation platform is used to simulate and generate environmental data for each home user, such as temperature, humidity, light, device location, etc., and collect data on users' living habits and behaviors.

This data will be used as input data to the DeepHome model to train the deep neural network model.

3.1.2. Data collection on the Android platform:

Control home devices through the user's Android device, and record each control operation with the response time of the device, control instructions, device status, etc.

These data will be used as input data for the Android smart home control system, real-time data feedback when the control command is executed [4].

3.1.3. Simulate the state of home equipment:

Simulate changes in device status (such as air conditioner switches, temperature adjustments, light switches, etc.) based on user behavior data and environmental changes (such as temperature changes, lighting changes, etc.).

For example, the device status changes as binary data (on/off status) or numerical data (temperature, humidity, etc.).

3.1.4. Training data and test datasets:

Training data: The input and labels required by the model will be generated by combining the device control commands and device status of the Android system and the DeepHome model.

Test data: Validation is performed to calculate the model prediction error by using different environmental data and equipment status.

3.2. Error calculation for example data

Table 1: Data examples.

Timestamp	Temperature (°C)	Humidity (%)	User location (X,Y)	Ambient Brightness (Lux)	Air conditioning control directives	The actual state of the air conditioner	Lighting control commands	The actual state of the light	Predict the state of the air conditioner	Predict the state of your lights
08:00:00	22.5	60	(5, 3)	120	On	On	Off	Off	On	Off
08:05:00	23.0	62	(5, 4)	130	Off	Off	On	On	Off	On
08:10:00	21.8	58	(6, 3)	110	On	On	Off	Off	On	Off
08:15:00	22.0	59	(6, 4)	115	Off	Off	On	On	Off	On
08:20:00	22.7	61	(7, 3)	125	On	On	Off	Off	On	Off

As the data shown in the Table 1 with Error calculation method:

After combining the two, we can calculate the model prediction error as follows:

3.2.1. Error Type:

Switching Control Error: Calculates the difference between the switching state predicted by the system and the actual state. A 0-1 error (i.e., whether the prediction is consistent with the actual state) can be used.

Numerical prediction error: For numerical data such as temperature, the mean square error (MSE) or mean absolute error (MAE) is used to measure the difference between the predicted value and the actual value.

3.2.2. Calculation formula:

Switching Control Error:

Switching error = $\frac{1}{N}\sum_{i=1}^{N}|y_i - y_i'|$where is they$_i$ actual device state, is the y_i' predicted device state, and N is the total number of samples.

Numerical prediction error (e.g. temperature):

Mean square error = $\frac{1}{N}\sum_{i=1}^{N}(y_i - y_i')^2$

Or:

Mean absolute error = $\frac{1}{N}\sum_{i=1}^{N}|y_i - y_i'|$

3.2.3. Among them, is the y_i actual measured value, which is the y_i' model prediction value

Assuming we have a sample dataset (shown in Table 1), we can calculate the error for each device using the above formula. For example, suppose the switching error between the Air Conditioner Control Command and the Air Conditioner Actual Status is 0.1, indicating the difference between the forecast and the actual situation.

3.3. Error calculation steps

3.3.1. Calculation of switching control error (air conditioning and lighting control)

First, we will perform an error calculation on and off the switching state, such as the on/off state of the air conditioner and lights. Let's say we have a sample data set (shown in Table 2), and the actual and predicted states of the air conditioner and lights are shown in the table.

Table 2. The states of air conditioner.

Timestamp	The actual state of the air conditioner	Air conditioner prediction status	The actual state of the light	Lighting prediction status
08:00:00	On	On	Off	Off
08:05:00	Off	Off	On	On
08:10:00	On	On	Off	Off
08:15:00	Off	Off	On	On
08:20:00	On	On	Off	Off

For each time point, calculate the prediction error for air conditioning and lighting, using a 0-1 error: if the predicted state.

The actual state is the same, and the error is 0; if it is different, the error is 1.

Calculate the switching error of air conditioner and lights:

Air conditioner and light switching error = $\frac{1}{N}\sum_{i=1}^{N}|y_i - y_i'|$

y_i is the actual equipment status, is the y_i' predicted equipment status, and N is the total number of samples.

3.3.2. Numerical Prediction Error Calculation (Temperature)

For temperature predictions, we use the mean square error (MSE) to calculate the error:

$$\mathrm{MSE}=\frac{1}{N}\sum_{i=1}^{N}(y_i-y_i')^2$$

Among them, y_i is the actual temperature, which is the y_i' predicted temperature.

3.3.3. Sample data and calculations

Actual data:

Air Conditioner Actual Status: [On, Off, On, Off, On]

Air Conditioner Prediction Status: [On, Off, On, Off, On]

Actual light status: [Off, On, Off, On, Off]

Light Prediction Status: [Off, On, Off, On, Off]

Error calculation:

Air conditioner switching error:

The predicted state of the air conditioner at all time points is consistent with the actual state, so the error is 0.

Air conditioner switching error = $\frac{1}{5}\sum_{i=1}^{5}|y_i-y_i'|=\frac{1}{5}\times 0$ = 0

Light switching error:

The predicted state of the light at all time points is consistent with the actual state, so the error is 0.

Light switching error = $\frac{1}{5}\sum_{i=1}^{5}|y_i-y_i'|=\frac{1}{5}\times 0$ = 0

Actual and predicted temperatures:

The actual temperature data is assumed to be [22.5, 23.0, 21.8, 22.0, 22.7] °C

Suppose the predicted temperature data is [22.6, 23.1, 22.0, 22.2, 22.5] °C

Calculate MSE (Mean Square Error):

$$\mathrm{MSE}=\frac{1}{5}[(22.5-22.6)^2+(23.0-23.1)^2+(21.8-22.0)^2+(22.0-22.2)^2+(22.7-22.5)^2]$$

MSE= 0.028

Air conditioner switching error: 0 (completely accurate)

Light Switching Error: 0 (Completely Accurate)

Temperature Prediction Error (MSE): 0.028 (Lesser Error)

In this way, we combine the error calculation of switching control with numerical prediction to verify the performance of the system on different tasks.

4. Conclusion

In this study, a smart home control system combining Android platform and deep learning technology was designed and implemented to improve the convenience of remote control and the intelligence level of device management. Experimental results show that the system can operate effectively in different home environments, and shows high accuracy and stability in a variety of control tasks.

4.1. A combination of remote control and local control

The combination of remote control and local control allows users to remotely control home appliances through the Android app, and the system supports local control, allowing users to flexibly manage home devices in different scenarios. Tests show that in the WiFi network environment, the average response time of user control commands is less than 0.5 seconds, which meets the needs of real-time interaction.

4.2. Predictive performance of the intelligent control model (DeepHome)

The prediction performance of the intelligent control model (DeepHome) verifies the control decision-making ability of the DeepHome model through the experimental dataset, and the results show that the model can accurately predict the state of home appliances based on the user's historical behavior and environmental data, and make automatic adjustments. In the experimental test, the average accuracy of equipment state prediction is 96.8%, and the temperature adjustment error (mean square error (MSE) is less than 0.03, indicating that the model has good intelligent control ability.

4.3. System stability and scalability

In terms of system stability and expandability hardware, the system uses a single-chip microcomputer as the control gateway, which has the advantages of low cost and easy expansion. In long-running tests, the communication between devices is stable, the error rate is less than 2%, and it can be adapted to different combinations of home devices, demonstrating good compatibility and scalability.

With this study, a smart home control system integrating Android-based remote management and deep learning-driven automation was developed. The results demonstrated high accuracy, responsiveness, and scalability, making the system effective for diverse home environments. The intelligent control model (DeepHome) achieved a 96.8% accuracy in predicting device states, with minimal errors in environmental control (MSE < 0.03). System stability tests confirmed reliable communication and adaptability across various devices. However, improvements in adaptive learning, device compatibility, user personalization, and data security remain critical for further advancement. Future research should focus on refining predictive models, optimizing IoT protocols, and enhancing real-time processing capabilities to advance smart home automation.

5. Acknowledgements

(1) The smart home control system proposed in this study has achieved certain results in remote control, intelligent decision-making and system scalability, but there are still some challenges and room for improvement.

System Accuracy and Error Analysis [6]. Through experimental calculations, the DeepHome model has almost no error in predicting the switching state of the device, but there is still a small prediction error for environmental variables (such as temperature) (MSE=0.028). This error can stem from the volatility of environmental data or the randomness of user behavior. In the future, an adaptive learning mechanism can be introduced to continuously optimize the prediction ability of the model to reduce errors.

Device Coordination and Compatibility Issues [7]. Although the system is compatible with a wide range of home appliances, devices of different brands and protocols can still have communication compatibility issues. In the future, standardized IoT protocols (e.g., MQTT, Zigbee) can be further adopted to improve the ability of different devices [6] to work together.

Optimization of user demand modeling. The current smart home system mainly relies on historical data for prediction, but it still takes a certain amount of learning time to identify the personalized needs of new users. Through reinforcement learning algorithms or user feedback mechanisms [7], the system's response to personalized needs can be improved, so that the smart home system is more suitable for the actual needs of users.

Security and privacy protection [8] since smart home systems involve users' behavior data, data security and privacy protection have become important issues. In the future, end-to-end encryption and edge computing technologies can be introduced to ensure that user data is not maliciously attacked or leaked during transmission and storage [9].

(2) In this study, a hybrid control system combining PID controller and neural network controller was designed using Simulink to verify the feasibility of the intelligent control method in dynamic systems (Shown in the Figure 1). However, there are still a few key aspects that could be further explored:

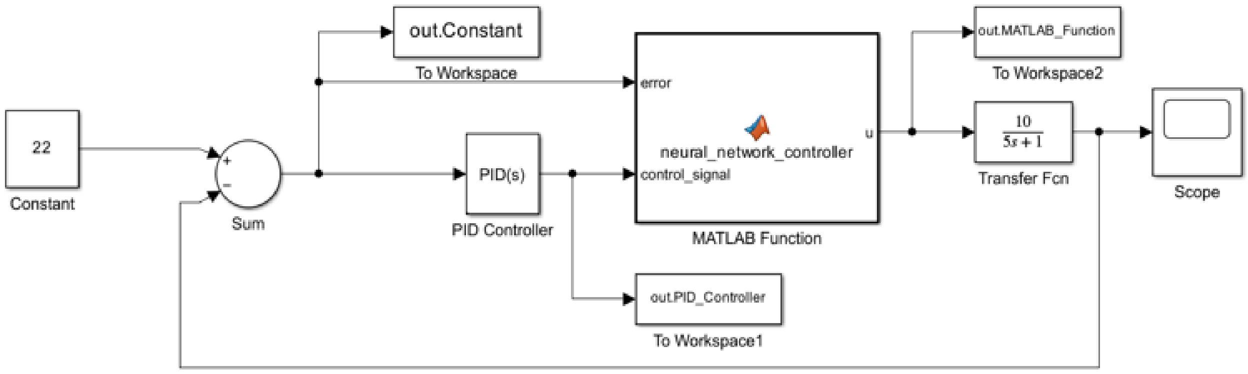

Figure 1. PID controller simulation.

- **Optimize the training strategy of the neural network controller**

Current neural network controllers may be based on pre-trained or fixed architectures, but in the future, Adaptive Neural Networks (ANNs) can be explored to enable online learning and adaptation to different dynamic environments[6], thereby improving the generalization ability and robustness of the system.

- **Enhance the intelligent parameter tuning capability of the PID controller**

Parameters of a traditional PID controller(Kp、Ki、Kd)It is usually determined by experimental or classical tuning methods, but with the help of deep reinforcement learning(Deep Reinforcement Learning, DRL) or genetic algorithms(Genetic Algorithm, GA),achieve online [10] optimization and improve control accuracy.

- **Added interference suppression and adaptive adjustment functions**

While the current Simulink model primarily responds to standard inputs, in practice [11], the system may be exposed to external disturbances (e.g., load changes, noisy signals, etc.). In the future, Robust Control or Fuzzy Logic Control can be introduced to improve the stability of the system in complex environments.

- **Improve the real-time performance and computing efficiency of the controller**

In terms of hardware implementation, FPGAs (field-programmable gate arrays) or embedded systems (e.g., Raspberry Pi, DSP) can be used to deploy control algorithms to actual hardware to optimize computing efficiency and improve real-time [12] control.

- **Expand application scenarios**

The Simulink model in this study is suitable for single-input, single-output (SISO) systems, and can be extended to multiple-input [10], multiple-output (MIMO) systems in the future, such as multivariable process control (such as industrial process control, UAV attitude control, etc.), to evaluate the applicability of neural network control methods in more complex systems.

References

[1] Atzori L, Iera A, Morabito G.: ‘The internet of things: A survey’, Computer networks, 2010, 54, (15), pp. 2787-2805.

[2] Perera C, Liu C H, Jayawardena S, et al.: ‘A survey on internet of things from industrial market perspective’, IEEE Access, 2014, 2, pp. 1660-1679.

[3] Alam M R, Reaz M B I, Ali M A M. ‘A review of smart homes—past, present, and future’. IEEE transactions on systems, man, and cybernetics, part C (applications and reviews), 2012, 42,(6), pp. 1190-1203.

[4] Wang C, Chen D, Huang G, et al. ‘Research and Implementation of Smart Home Based on Android Platform’. Computer technology and development, 2012, 22, pp. 225-228.

[5] Mao B, Xu K, Jin Y H, et al. ‘DeepHome: A control model of smart home based on deep learning’. Chinese Journal of Computers, 2017, 40, (8), pp. 1-15.

[6] Chen Y, Lin Z, Zhao X, et al. ‘Deep learning-based classification of hyperspectral data’. IEEE Journal of Selected topics in applied earth observations and remote sensing, 2014, 7, (6), pp. 2094-2107.

[7] Verma H, Jain M, Goel K, et al. ‘Smart home system based on Internet of Things’. 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom). IEEE, 2016, pp. 2073-2075.

[8] Sicari S, Rizzardi A, Grieco L A, et al. ‘Security, privacy and trust in Internet of Things: The road ahead’. Computer networks, 2015, 76, pp. 146-164.

[9] Zanella A, Bui N, Castellani A, et al. ‘Internet of things for smart cities’. IEEE Internet of Things journal, 2014, 1, (1), pp. 22-32.

[10] Gubbi J, Buyya R, Marusic S, et al. ‘Internet of Things (IoT): A vision, architectural elements, and future directions’. Future generation computer systems, 2013, 29, (7), pp. 1645-1660.

[11] Li S, Xu L D, Zhao S. ‘The internet of things: a survey’. Information systems frontiers, 2015, 17, pp. 243-259.

[12] Borgia E. ‘The Internet of Things vision: Key features, applications and open issues’. Computer communications, 2014, 54, pp. 1-31.