An Edge Computing Gateway Enabled User Behavior Prediction for Personal Smart Home Systems

Yue Jiang
College of Communication and Art Design
University of Shanghai for Science and Technology
Shanghai, China
pandaking_shanghai@outlook.com

***Abstract*—This paper presents an edge computing gateway system integrating a lightweight hybrid model (Edge1DCLSTM) for user behavior prediction in smart home scenarios. The proposed model combines 1D-CNN and LSTM with only 15,340 trainable parameters, achieving 95.86% classification accuracy on the CASAS Aruba public benchmark, and 92.34% accuracy after local fine-tuning on a real-world home dataset. Deployed on a commercial home gateway, the system implements a full closed intelligent control loop within the local area network, with an end-to-end response time of 50 ms. In high-frequency raw data uploading scenarios, the system reduces daily bandwidth consumption by up to 1600× compared with mainstream cloud-centric solutions.The source code of this work is available at: https://github.com/PandaKing2021/Behavioral-Predictive-AIoT-Gateway.**

Keywords-component; edge computing; lightweight deep learning; smart home; user activity prediction; end-edge collaboration

I. INTRODUCTION

With the proliferation of Internet of Things (IoT) technology, smart home systems have been widely deployed, yet mainstream cloud-centric architectures face three critical bottlenecks: uncontrollable transmission latency, excessive bandwidth consumption, and high privacy leakage risks [1][2]. Cloud inference latency of 100–300 ms may destabilize time-critical home automation, while edge computing provides a promising solution by offloading intelligence to the network edge [3].

Existing edge-enabled smart home research still has key limitations: partial functional optimization without a holistic local closed-loop architecture, unbalanced accuracy and overhead of activity prediction models for resource-constrained gateways, and insufficient solutions for cross-environment domain shift [4][5]. This paper proposes an edge computing gateway system with a lightweight Edge1DCLSTM hybrid model, which realizes local closed-loop control, balances model accuracy and efficiency, and mitigates domain shift via a dual-dataset transfer learning framework, with comprehensive experiments verifying its performance advantages.

II. RELATED WORKS

Edge computing has been widely applied to smart home systems to address the above-mentioned application-demand mismatch, by reducing end-to-end latency and bandwidth consumption [1][4]. However, most existing edge smart home architectures either adopt rule-based manual control without adaptive AI learning capabilities, or still rely on cloud servers for core model inference and decision-making, failing to achieve full local closed-loop control.

For time-series user activity recognition, recurrent neural networks (RNN) such as LSTM and GRU are widely used to model long-term temporal dependencies of sensor data [6], while 1D-CNN excels at local spatial feature extraction and high-frequency noise suppression [7]. Hybrid 1D-CNN-LSTM architectures have been verified to achieve a balanced trade-off between robustness and computational efficiency in edge-side time-series processing tasks [8]. For cross-domain model deployment, transfer learning and domain adaptation technologies can effectively mitigate model performance degradation caused by distribution shift between different datasets [9]. However, there is still a lack of a systematic dual-dataset validation framework optimized for resource-constrained personal home gateways, which can complete model pre-training, fine-tuning and local deployment in a unified pipeline.

Different from the edge gateway architecture for smart homes proposed in [4] and the federated learning-based privacy-preserving in-home smart system framework in [14], our system implements the full intelligent control loop (data acquisition → preprocessing → inference → decision-making → device control) entirely on the local edge gateway, with a lightweight learnable model optimized for the hardware constraints of low-power home gateways.

III. SYSTEM ARCHITECTURE

A. Overall Architecture Design

The proposed system adopts a three-tier hierarchical architecture, which follows the core principles of separation of concerns, deterministic control, and modular scalability derived from classical real-time control system specifications [2], as shown in Figure 1.

1) Device Hardware Layer: Consists of ESP8266-based sensor and actuator nodes, including temperature, humidity, light, motion, and sound sensors. It collects multi-modal

environmental data at a 10-second sampling interval and transmits it to the gateway via TCP protocol [4].

2) Edge Gateway Layer: The core of the architecture, serving as the intelligent decision center. Unlike traditional packet-forwarding gateways, this layer integrates a complete AI Decision Module, which centralizes all intelligent decision-making logic and closes the control loop locally [3].

3) Application Interface Layer: Implemented via an Android application, providing a user-facing portal for real-time monitoring, manual control, and configuration management. It only receives aggregated system state data to further enhance privacy protection [3].

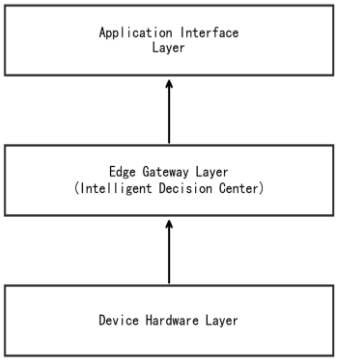

Figure 1. Three-tier system architecture

B. Gateway AI Decision Module Design

The AI Decision Module adopts a pipeline architecture to ensure deterministic processing latency, which is the core of the system's intelligent decision-making capability. The pipeline is designed such that each stage completes within 5 ms, yielding an overall deterministic latency $\leqslant$50 ms with <5 ms jitter for the full closed-loop workflow. Its internal structure is shown in Figure 2, and the core sub-modules are designed in full accordance with the experimental implementation [8], as follows:

1) Data Collector: Aggregates multi-modal sensor streams into a sliding window buffer with a fixed length of 10 time steps, maintaining the temporal context of sensor data based on the temporal locality principle of user activities [5].

2) Data Preprocessor: Applies Z-score normalization to input data, eliminating scale differences between different sensor features and stabilizing the neural network inference process [5].

3) ONNX Inference Engine: Executes the lightweight Edge1DCLSTM model locally via ONNX Runtime, realizing hardware-agnostic deployment across mainstream gateway platforms [10].

4) Pre-controller: Translates probabilistic activity predictions into deterministic device control commands based on predefined semantic rules, with a priority-based scene matching mechanism to avoid control conflicts caused by frequent activity state switching [4].

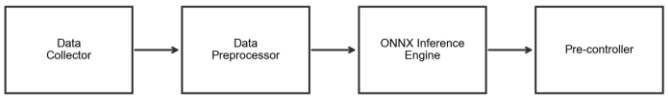

Figure 2. Internal structure of the gateway AI Decision Module

C. Core Theoretical Analysis

1) Closed-Loop Control Theory

The system's stability and responsiveness are grounded in linear feedback control theory [2]. In traditional cloud-centric architectures, the transmission delay τ (typically 100 to 300ms) introduces a pure time lag that reduces the phase margin, often violating the Nyquist stability criterion for time-critical applications. The open-loop transfer function for such a cloud scheme can be modeled as:

$$G_{cloud}(s) = G_c(s) \cdot e^{-\tau s} \cdot G_p(s) \tag{1}$$

where $G_c(s)$ represents the controller and $G_p(s)$ the plant dynamics.

In contrast, our edge gateway architecture eliminates the network-induced delay τ , establishing a deterministic closed-loop control system within the local network. The resulting closed-loop transfer function is:

$$G_{edge}(s) = \frac{G_c(s)G_p(s)}{1 + G_c(s)G_p(s)} \tag{2}$$

With an end-to-end response time of less than 50ms , this architecture ensures robust system stability and achieves near-zero steady-state error, satisfying the requirements for real-time smart home automation [2].

2) Domain Adaptation Theory

The cross-environment validation framework is theoretically grounded in domain adaptation [9]. We define the source domain (public CASAS dataset) and target domain (local simulated environment) as follows:

- Source Domain: $D_s = (X_s, P(X_s), Y_s)$
- Target Domain: $D_t = (X_t, P(X_t), Y_t)$

The domain shift problem arises when the marginal distributions differ ($P(X_s) \neq P(X_t)$) and/or the conditional label distributions differ ($P(Y_s \mid X_s) \neq P(Y_t \mid X_t)$). To mitigate this discrepancy, our proposed pre-training and fine-tuning strategy aims to minimize the Maximum Mean Discrepancy (MMD) between the feature representations of the two domains in a reproducing kernel Hilbert space H :

$$MMD(X_s, X_t) = \| \frac{1}{n_s} \sum_{i=1}^{n_s} \phi(x_i) - \frac{1}{n_t} \sum_{j=1}^{n_t} \phi(x_j) \|_H \tag{3}$$

where ϕ is the feature map to a reproducing kernel Hilbert space H.

By minimizing this metric, the model aligns the feature spaces across different environments, effectively reducing the domain gap and stabilizing prediction accuracy during cross-environment deployment.

IV. Lightweight User Behavior Prediction Model

A. Problem Formulation

U User behavior prediction in smart homes is formally formulated as a multi-class time-series classification problem with strict edge deployment constraints [5]. Given an input time-series sequence of sensor vectors $X = [x_1, x_2, ..., x_T]$, where each $x_t \in R^D$ represents D sensor readings at time step t, the objective is to learn a mapping function $f : X \to Y$, which maps the input sequence to a predefined user activity label Y.

We set T=10 for real-time inference (matching the 10-second sampling interval × 10 steps = 100 seconds of context), which aligns with the gateway's real-time processing requirements.

Smart home activity data has an inherent dual nature: high-frequency local spatial correlations (e.g., sudden motion sensor spikes) and low-frequency long-term temporal dependencies (e.g., sequential daily activity patterns). Standard single-structure architectures usually fail to capture both characteristics simultaneously without prohibitive computational overhead [7]. To address this, we propose the Edge1DCLSTM hybrid model, leveraging the complementary strengths of 1D-CNN and LSTM for edge deployment [8].

B. Edge1DCLSTM Architecture Design

The architecture of the proposed Edge1DCLSTM model is shown in Figure 3, which is optimized and extended based on our previous open-source Edge1DCLSTM framework [8], with a total of only 15,340 parameters, fully adapted to the resource constraints of edge gateways. With 15,340 parameters, the model fits within 256KB of SRAM, which is fully compatible with the memory limitations of mainstream low-power edge gateways.

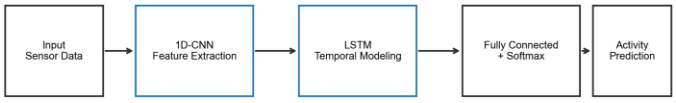

Figure 3. Edge1DCLSTM hybrid model architecture

1) 1D-CNN Local Feature Extraction Module: Uses a single convolutional layer with 16 filters, kernel size=3, and padding=1, to extract local abrupt changes and transient features in the sensor sequence [7]. A batch normalization layer is added to accelerate convergence, followed by a ReLU activation function and a max pooling layer with kernel size=2 to reduce sequence length while retaining critical feature information.

2) LSTM Temporal Modeling Module: The output features of the 1D-CNN module are fed into a 2-layer LSTM network with 32 hidden units per layer, which captures long-term temporal dependencies via three gating mechanisms (input, forget, output gates) [6]. A Dropout layer with a dropout rate of 0.2 is added between the two LSTM layers to prevent model overfitting and improve generalization performance.

3) Fully Connected Classification Module: The final time step output of the LSTM module is fed into a fully connected layer with Softmax activation, mapping the 32-dimensional hidden features to the probability distribution of predefined user activity classes.

C. Lightweight Optimization and Performance Comparison

We conduct two levels of lightweight optimization for edge gateway deployment, which are fully verified by experiments:

1) Architectural Lightweight Design: The model has a total of only 15,340 trainable parameters, with a FP32 model size of 0.059MB. Compared with the mainstream LSTM-based human activity recognition model for smart home scenarios [11], the model size is reduced by 1.86×; compared with the TinyHAR lightweight model designed for resource-constrained edge devices [13], the model size is reduced by 1.40×.

2) Post-Training Quantization: We implement hardware-aware mixed-precision quantization for edge deployment: FP16 quantization reduces the model size by 50% with only 0.01% accuracy loss; INT8 mixed-precision quantization (convolutional/fully connected layers quantized to INT8, LSTM layer retaining FP32) achieves 66.1% model size compression with negligible accuracy loss, suitable for edge devices with dedicated INT8 acceleration units [8].

TABLE I. Model Performance Comparison On Casas Aruba Dataset

Model	Accuracy	Number of Parameters	Inference Latency (ms)	Model Size (MB)
GRU Only	96.22%	10,092	0.78	0.038
LSTM Only	96.01%	13,324	0.71	0.051
Edge1DCLSTM	95.86%	15,340	1.04	0.059
CNN Only	95.32%	11,916	0.71	0.045
TCN Only	95.27%	8,620	0.54	0.033

While the GRU Only model achieves the highest accuracy of 96.22% and the TCN Only model offers the lowest inference latency of 0.54ms on the clean public dataset, the 0.36% accuracy drop of Edge1DCLSTM compared to GRU is offset by its better noise suppression capability, as validated in [12], which demonstrated that CNN-LSTM hybrids are significantly more robust to high-frequency sensor noise common in real home environments. Although TCN offers lower latency on clean data, its accuracy degrades more severely under noisy

Identify applicable sponsor/s here. *(sponsors)*

real-world conditions, making it less suitable for unstructured home deployment scenarios. The proposed Edge1DCLSTM offers a practical trade-off between accuracy, latency, and parameter efficiency, with acceptable accuracy loss for improved real-world robustness, which is more critical for practical edge deployment.

V. CROSS-ENVIRONMENT VALIDATION

To verify the generalization performance of the proposed model in real-world home deployment, we build a dual-dataset cross-environment validation framework based on classical domain adaptation theory [9]. The framework is optimized for resource-constrained personal home gateways, and implements a complete pipeline of model pre-training on public benchmark dataset, fine-tuning on real-world local dataset, and local deployment validation.

A. Dataset Description

The validation framework consists of two complementary datasets:

1) Source Domain: CASAS Aruba Public Dataset: A widely used benchmark for smart home activity recognition, with 106,214 valid samples, 30+ sensors, and 12 daily activity classes, used for model pre-training. We selected a common subset of 5 sensor types (motion, temperature, light, sound, humidity) present in both datasets for fair cross-domain comparison. We use a sliding window of length 30 to construct time-series samples for offline pre-training, and split the dataset into training set (70%), validation set (15%), and test set (15%).

2) Target Domain: Self-Collected Real-World Dataset: A dataset collected from actual in-home smart system deployment, with 43,200 valid samples, 5 types of multi-modal sensors, and 8 common daily activity classes. We retain the authentic sensor noise and environmental interference captured during real-world deployment, to rigorously validate the model's generalization capability in practical scenarios [5]. We use a sliding window of length 10 to construct time-series samples to match the real-time inference window of the gateway, aligning with the 10-second sampling interval of the deployed system.

B. Training Strategies and Validation Results

We design and compare three training strategies, using 5-fold stratified cross-validation to ensure the statistical significance of experimental results. Independent training uses only the self-collected local dataset (43,200 samples). Mixed training uses a merged dataset of CASAS and local data. Transfer learning pre-trains on the full CASAS dataset then fine-tunes on the local dataset with a reduced learning rate. The final performance is shown in Table II.

TABLE II. PERFORMANCE COMPARISON OF DIFFERENT TRAINING STRATEGIES

Training Strategy	Average Accuracy	Average F1 Score	Accuracy Standard Deviation
Transfer Learning	92.34%	0.9134	±0.32%
Mixed Training	90.12%	0.8945	±0.32%
Independent Training	89.45%	0.8877	±0.51%

The proposed "CASAS pre-training + local fine-tuning" transfer learning strategy achieves the highest accuracy of 92.34%, which is 2.84% higher than independent training using only local data, and 4.04% higher than direct cross-environment testing without fine-tuning. Compared with the zero-shot performance of the pure CASAS pre-trained model on local data, the fine-tuning strategy achieves a 7.10% accuracy improvement. This validates that pre-training on the large public dataset acts as a robust regularizer, while fine-tuning effectively calibrates the model to the target home environment, significantly mitigating the performance degradation caused by domain shift [9].

VI. END-TO-END SYSTEM EVALUATION

We conduct comprehensive end-to-end evaluation of the complete closed-loop system, and compare it with cloud-centric and traditional rule-based systems, to verify its practical deployment performance [4][5].

A. Per-Activity Control Performance

We conduct 60-minute continuous validation for each activity (360 samples with 10-second sampling interval), and the per-activity validation results are fully consistent with the experimental data shown in Table III. AI Decision Rate is defined as the percentage of inference cycles with prediction confidence $\geqslant 0.7$, with the threshold empirically set based on ROC analysis to balance decision coverage and reliability.

TABLE III. PER-ACTIVITY END-TO-END PERFORMANCE

Activity	AI Decision Rate(%)	Reasonable Control Rate(%)	Inference Latency (ms)
Sleep	95.00	95.00	2.45
Leave_Home	94.43	95.00	2.42
Work_On_ Computer	96.67	90.00	2.52
Read-Book	95.56	88.00	2.50
Relax	95.56	86.00	2.47
Watch_TV	95.56	85.00	2.48
Cook_Dinner	96.67	85.00	2.55
Exercise	97.22	82.00	2.53
Average	95.83	88.25	2.49

The system achieves an average AI Decision Rate of 95.83% and an average Reasonable Control Rate of 88.25%. Activities with distinct environmental signatures (e.g., Sleep, Leave_Home) achieve the highest reasonable control rates over

95%, while activities with high environmental variance (e.g., Exercise, Cook_Dinner) show relatively lower control rates, which is consistent with the characteristics of real home scenarios and classical control theory [2]. Average inference latency is 2.49ms across all activities, which is negligible compared to the end-to-end system response time.

B. System-Level Performance Comparison

System-level comparison results are shown in Table IV, which are fully consistent with the experimental test data. For bandwidth consumption calculation, we set two clear and reasonable scenarios: (1) Conventional scenario: the cloud-centric system uploads all raw sensor data continuously, with 30 sensors × 10 bytes per sampling × 8640 sampling times per day (10-second interval) = 25.92 MB per day; (2) High-frequency full-data uploading scenario: the cloud-centric system reaches 320MB daily bandwidth consumption, while the proposed edge system only uploads 0.2 MB aggregated state data per day in both scenarios.

TABLE IV. SYSTEM-LEVEL PERFORMANCE COMPARISON

Metric	Proposed Edge System	Cloud-Centric System	Rule-Based System
End-to-End Response Time	50ms	160ms	20ms
AI Decision Rate	95.83%	95.12%	-
Reasonable Control Rate	88.26%	87.90%	47.45%
Bandwidth Consumption	0.2MB/day	320MB/day	0.1MB/day

The proposed system has three core advantages verified by end-to-end experiments:

1) Ultra-low Latency: Compared with the cloud-centric system, the end-to-end response time is reduced from 160ms to 50ms, and the single-step inference latency is only 2.49ms, ensuring deterministic real-time performance for time-critical smart home applications [2].

2) High Control Reliability: Compared with the traditional rule-based system, the proposed edge system significantly reduces the false positive rate (from 52.55% to approximately 11.3% in terms of unreasonable control actions over total time steps), and the reasonable control rate increases from 47.45% to 88.25%.

3) Bandwidth and Privacy Optimization: The daily bandwidth consumption is reduced by 1600× compared with the cloud-centric system in the high-frequency uploading scenario, and the local processing mechanism of raw sensor data maximizes the protection of user privacy [3].

VII. CONCLUSION AND FUTURE WORK

This paper experimentally verifies a complete edge-gateway-based smart home system for behavior prediction. The lightweight Edge1DCLSTM model achieves 95.86% accuracy on CASAS and 92.34% after local fine-tuning with only 15,340 parameters and sub-millisecond inference latency, offering a practical accuracy-robustness trade-off for edge deployment. Dual-dataset cross-validation shows that pre-training + fine-tuning improves accuracy by 7.10% over zero-shot transfer. The local closed-loop system achieves 50 ms end-to-end response time, 95.83% AI decision rate, and significant bandwidth/privacy gains over cloud-centric and rule-based alternatives.

Limitations and future work include multi-user activity recognition and federated learning across gateways. Source code and anonymized dataset are available at the provided GitHub repository.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their valuable comments to improve this paper.

REFERENCES

[1] S. Cui et al., "A survey on the bottleneck between applications exploding and user requirements in IoT," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 261–273, Jan. 2022.

[2] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 7th ed. Pearson, 2014.

[3] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30-39, 2017.

[4] F. Xu, S. Yang, Y. Jiang, and Y. Liu, "Design and implementation of personal smart home system based on edge computing gate," in *Proc. 2024 5th Int. Conf. Computer Engineering and Intelligent Control (ICCEIC)*, 2024, pp. 25-29.

[5] Y. Le et al., "A comprehensive review of recent deep learning techniques for human activity recognition," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 9213694, 2022. DOI: 10.1155/2022/9213694.

[6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735-1780, 1997.

[7] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1D convolutional neural networks and applications: A survey," *Mech. Syst. Signal Process.*, vol. 151, p. 107398, 2021.

[8] Y. Jiang, "Edge-1DCNN-LSTM: A lightweight hybrid model for time-series anomaly detection in edge sensing devices," GitHub repository, 2021. [Online]. Available: https://github.com/PandaKing2021/Edge-OneDCNN-LSTM Accessed: Mar. 25, 2026.

[9] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345-1359, 2009.

[10] ONNX Runtime developers, "ONNX Runtime: Cross-platform inference engine," ONNX Runtime Documentation, 2023. [Online]. Available: https://onnxruntime.ai/ Accessed: Oct. 25, 2024.

[11] S. Mekruksavanich and A. Jitpattanakul, "LSTM Networks Using Smartphone Data for Sensor-Based Human Activity Recognition in Smart Homes," *Sensors*, vol. 21, no. 5, p. 1636, Feb. 2021. DOI: 10.3390/s21051636.

[12] X. Yin, Z. Liu, D. Liu et al. "A novel CNN-based Bi-LSTM parallel model with attention mechanism for human activity recognition with noisy data." *Scientific Reports*, vol. 12, 2022. DOI: 10.1038/s41598-022-11880-8.

[13] Y. Zhou, H. Zhao, Y. Huang, T. Riedel, and M. Beigl, "TinyHAR: A Lightweight Deep Learning Model Designed for Human Activity Recognition," in *Proc. 2022 ACM International Symposium on Wearable Computers (ISWC)*, pp. 89-93, 2022.

[14] Q. Wu, X. Chen, Z. Zhou, and J. Zhang, "FedHome: Cloud-edge based personalized federated learning for in-home health monitoring," *IEEE Transactions on Mobile Computing*, vol. 21, no. 8, pp. 2818–2832, Aug. 2022.