

Context-Aware Edge Computing for Service Response and Affective Interaction Design in Smart Home IoT Systems

Xiang Li, Yue Wang, and Yaxin Chen

College of Textile and Clothing Engineering, Xinjiang Light Industry Vocational and Technical College, Urumqi 830021, China

(Received 11 June 2026; Revised 07 September 2026; Accepted 20 September 2026; Published online 15 October 2026)

Abstract: Smart home Internet of Things (IoT) systems require service-response decisions that account for uncertainty, device availability, service risk, and edge latency rather than relying solely on activity recognition. To address this, we propose an edge-constrained context-aware service-response framework. It integrates activity prediction, recognition confidence, device state, service risk, and context into a service-oriented context state. An uncertainty-aware policy then selects auto, ask, or none. Experiments on Center for Advanced Studies in Adaptive Systems (CASAS)-Aruba and Activity Recognition with Ambient Sensing (ARAS) compare the method with traditional Human Activity Recognition (HAR), deep HAR, lightweight edge HAR, rule-based, context-aware reasoning, and uncertainty-removal baselines. Macro-F1, Service Response Accuracy (SRA), and Edge Response Latency (ERL) are primary metrics, with Ask-F1 and Auto Error as supplementary indicators. Results show improved response action consistency under rule-mapped labels while maintaining context recognition and edge-side decision-making within latency constraints. Response labels are rule-constructed and do not reflect real user preferences. Further validation with real service logs and long-term user feedback is needed.

Keywords: context state modeling; device availability; low-burden feedback; response action selection; uncertainty awareness

I. INTRODUCTION

Smart home Internet of Things (IoT) systems perceive household states through heterogeneous sensors, including door contacts, infrared sensors, temperature and humidity sensors, home appliances, and wearable devices [1]. These systems support services such as lighting adjustment, security reminders, elderly assistance, and daily service recommendation [2]. As smart home systems become increasingly autonomous, their task extends beyond identifying user activities. They must also determine whether a service should be executed automatically, confirmed by the user, or suppressed.

In China, smart home IoT research has expanded from basic automation toward aging-friendly services, context-aware interaction, intelligent sensing, and adaptive household assistance. Existing studies have examined aging-friendly interaction and system acceptance among older adults [2,3]. Wireless sensing and human-machine interaction have also been incorporated into smart home environments to improve monitoring and control [4]. However, increasingly context-dependent services create a further challenge: recognized activities must be transformed into appropriate service responses under uncertainty, device constraints, and different levels of risk.

Existing research has established foundations in activity recognition, edge inference, and context-aware interaction [5–7]. These capabilities, however, are often optimized separately, and an activity label does not directly determine an appropriate service action [8]. For example, recognizing nighttime getting up does not necessarily mean that lighting should be activated automatically. The decision may also depend on confidence, location, device availability, and

disturbance risk. Under low-confidence, multi-resident, sensor-missing, device-constrained, or high-risk conditions, response selection requires joint consideration of uncertainty, service relevance, device state, risk, and edge latency [9,10].

This creates a gap between context perception and service execution. Activity recognition can identify a likely household state, while edge inference can provide timely computation, but neither alone determines whether an automatic service action is appropriate. A practical response mechanism must therefore combine perceptual outputs with service-side constraints before device execution.

To address this problem, this study treats activity recognition as an intermediate input rather than the final output. An edge-constrained context-aware service-response framework combines activity prediction and confidence with contextual information, candidate service relevance, device availability, and service risk. An uncertainty-aware edge response policy then selects among auto, ask, and none under a latency budget. A low-burden feedback correction mechanism further allows acceptance, rejection, delayed reminders, or preference adjustments to modify subsequent response parameters. In the current experiments, these feedback signals are rule-constructed rather than collected from long-term real user interactions.

This study makes three contributions. First, it develops a service-oriented context state representation that connects activity prediction and recognition confidence with contextual information, device availability, service relevance, and service risk. This representation bridges activity recognition and executable smart home services. Second, it proposes an uncertainty-aware edge service-response policy. The policy jointly considers confidence, service relevance, device availability, service risk, and latency constraints when selecting auto, ask, or none. Third, it introduces a low-burden feedback correction mechanism that uses lightweight feedback

Corresponding author: Xiang Li (e-mail: xiangliedu@phc.ac.cn).

signals to adjust subsequent response parameters. The mechanism also provides an interface for future validation using real service logs and long-term user feedback.

The rest of the paper is organized as follows. Section II reviews related work on smart home activity recognition, edge computing, context reasoning, affective interaction, and user feedback. It then clarifies the research gap addressed in this study. Section III presents the proposed edge-constrained context-aware service-response framework, including the problem formulation, service-oriented context state representation, uncertainty-aware response policy, feedback correction mechanism, training strategy, and complexity analysis. Section IV describes the experimental setup and reports the main results, followed by statistical, representation, interpretability, failure case, ablation, and threshold sensitivity analyses. Section V discusses the main findings, practical implications, and limitations of the proposed framework. Finally, Section VI concludes the paper and outlines directions for future work.

II. RELATED WORK

Research relevant to smart home service-response decision-making can be grouped into three areas: activity recognition and context awareness, edge computing and context-aware service reasoning, and user-centered interaction and feedback. These areas provide perceptual, computational, and interaction foundations but generally address different stages of the service process. This section reviews their contributions and limitations and positions the proposed framework accordingly.

A. SMART HOME ACTIVITY RECOGNITION AND CONTEXT AWARENESS

Smart home activity recognition infers daily activities from heterogeneous sensor events, including door contacts, infrared sensors, environmental sensors, switch states, and wearable devices [11]. These sensing systems provide the perceptual basis for context-aware household services.

Public datasets are widely used to develop and evaluate activity recognition methods. Center for Advanced Studies in Adaptive Systems (CASAS) provides real-world sensor streams for activity recognition and behavioral analysis, while Activity Recognition with Ambient Sensing (ARAS) extends this setting to multi-resident environments [12]. Existing computational approaches include traditional machine learning, sequential modeling, and deep learning [13,14].

These methods primarily answer what the user is doing, whereas service-oriented systems must additionally decide what action should be taken. An activity label alone cannot determine whether a service should be executed, confirmed, or suppressed [15]. Recognition confidence, environmental context, device availability, and service risk may produce different decisions for the same recognized activity. Human Activity Recognition (HAR) therefore provides the perceptual foundation, but an additional service-oriented layer is required to transform activity information into executable responses [16].

Figure 1 summarizes the transition from sensing and activity recognition to service-response decision-making.

As shown in Fig. 1, activity recognition and context awareness are intermediate stages rather than final service outputs. Service execution additionally requires factors such as recognition confidence, device availability, risk, and contextual suitability.

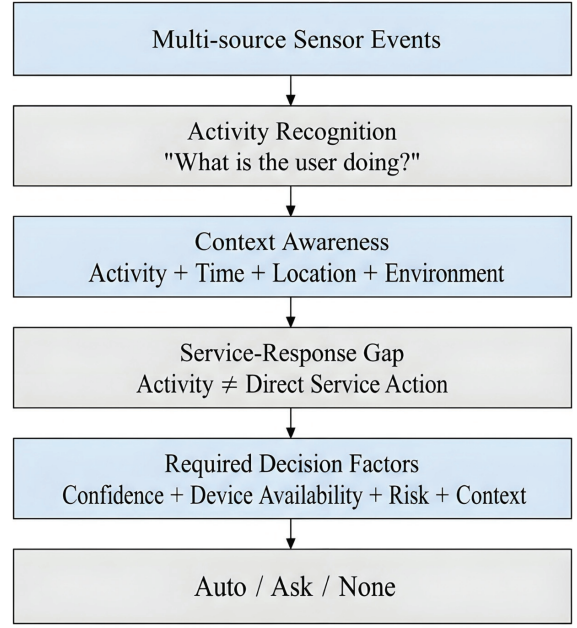


Fig. 1. Research progression from smart home sensing to service-response decision-making.

B. EDGE COMPUTING, CONTEXT REASONING, AND SMART HOME SERVICE RESPONSE

Edge computing enables local inference and rapid response while reducing cloud dependence and supporting lower latency, improved privacy, and greater network availability [17,18]. These characteristics are particularly useful in smart homes, where sensor events may require timely local processing.

Lightweight on-device inference and TinyML methods further demonstrate that resource-constrained devices can support smart home perception tasks [19]. However, their main objective is generally efficient inference rather than deciding whether a recognized state should trigger a particular household service.

In parallel, ontology reasoning, rule-based systems, and context-aware control transform sensor and environmental information into interpretable service conditions and predefined actions [20,21]. These methods extend beyond activity recognition by incorporating contextual variables into service triggering.

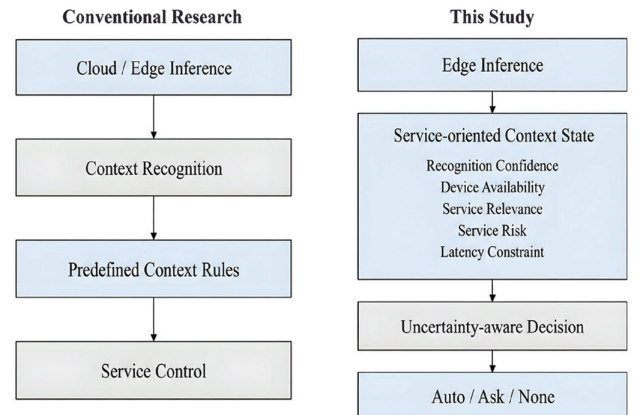


Fig. 2. Evolution from edge/context-aware control to uncertainty-aware service response.

Nevertheless, edge inference and context reasoning address different aspects of the service process. Edge inference mainly concerns efficient local computation, while context reasoning commonly depends on predefined mappings. Neither alone fully addresses response selection when recognition is uncertain, sensing is incomplete, devices are unavailable, or incorrect automatic execution involves elevated risk [22].

Figure 2 contrasts conventional edge/context-aware approaches with the response-decision scope considered in this study.

Conventional approaches often terminate at local inference or rule-based control. In contrast, this study introduces a service-response decision layer in which activity prediction is combined with confidence, device availability, service relevance, service risk, and latency constraints. These factors jointly determine whether the system selects auto, ask, or none.

C. AFFECTIVE INTERACTION, USER FEEDBACK, AND SERVICE CORRECTION

Smart home systems must consider not only activity perception and service execution but also users' acceptance of system responses [23,24]. This becomes increasingly important as systems move from passive monitoring toward proactive and automatic service provision.

Studies on user experience, smart home acceptance, and elderly oriented interaction emphasize users' sense of control, interaction burden, and willingness to adopt intelligent home technologies [25,26]. Smart home experience and technology acceptance may also depend on perceived disruption, personal needs, and individual preferences [27,28]. These findings indicate that technically feasible services may still be unsuitable if they create unnecessary interruption or conflict with user expectations.

However, feedback is often treated as an outcome of user experience evaluation rather than an input to later service-response decisions. Under low-confidence or high-disturbance-risk conditions, lightweight confirmation feedback can instead help correct subsequent responses.

Accordingly, affective interaction in this paper does not refer to emotion recognition. It denotes a low-burden feedback correction process in which acceptance, rejection, delayed reminders, or preference adjustments influence subsequent response parameters. The present experiments use rule-constructed feedback rather than long-term feedback from real users. Therefore, the results evaluate response correction rather than personalized affective modeling or real user preference learning.

D. RESEARCH GAP AND POSITIONING

The reviewed studies provide four complementary foundations for smart home intelligence. HAR primarily outputs activity categories, while rule-based and ontology-based systems map predefined contexts to service triggers. Edge HAR emphasizes local inference and latency, whereas user-centered interaction research focuses on experience, acceptance, and feedback [29]. These directions, however, do not individually provide a unified mechanism for selecting service actions under simultaneous uncertainty, device, risk, context, and latency constraints.

CASAS- and ARAS-based HAR provides the perceptual basis for identifying household activities. However, activity recognition alone cannot determine whether the appropriate response is auto, ask, or none. Context-aware and ontology-based control can automate service triggering, but predefined mappings remain limited under low confidence, missing sensors, unavailable devices, or conflicting risk

conditions. User-perceived risk and acceptance further indicate that service suitability cannot be reduced to activity classification alone [30].

The proposed framework is therefore distinguished primarily by its decision target. Unlike conventional HAR, activity prediction is treated as an intermediate representation rather than the final output. Unlike rule-based or ontology-based control, the response policy explicitly incorporates uncertainty and service risk. Unlike lightweight edge HAR, the framework does not propose a new lightweight recognition backbone. Instead, edge computing provides the environment for service-response selection under a latency budget. Finally, lightweight user feedback is incorporated as an auxiliary correction signal rather than merely an evaluation outcome.

The framework consequently connects three stages that are often treated separately: context perception, service-response decision-making, and response correction. Activity and confidence are combined with context, service relevance, device availability, and risk before response selection, while feedback adjusts subsequent policy behavior. These elements motivate the methodology presented in Section III.

III. METHODOLOGY

Activity recognition results in smart home IoT systems cannot be directly transformed into service-response actions. To address this problem, this paper proposes an edge-constrained context-aware service-response framework. The framework takes multi-source sensor events, device states, and environmental context as inputs and constructs a service-oriented context state. It then selects automatic execution, confirmation request, or no action by considering recognition confidence, service relevance, device availability, service risk, and latency constraints. This paper focuses on the transformation from service-response decision-making rather than on designing a new activity recognition backbone or lightweight network architecture. The framework also introduces a low-burden feedback correction mechanism to assist in updating subsequent response policies. This chapter presents the problem definition, overall framework, core modules, training strategy, and algorithmic procedure in sequence.

A. PROBLEM DEFINITION

This paper studies edge-side service-response decision-making in smart home IoT scenarios. The system receives a continuous event stream composed of household sensors, device states, and environmental context. Its objective is not merely to recognize user activities but also to determine whether to execute a service automatically, request confirmation, or take no action under edge latency constraints. Service response is therefore formulated as a three-action decision problem. The state contains activity prediction, recognition confidence, device availability, service risk, and contextual information. The action space consists of auto, ask, and none.

Let the i -th sample be defined as:

$$x_i = \{E_i, D_i, C_i\} \quad (1)$$

where E_i denotes multi-source sensor events within a time window. D_i represents the states of IoT devices, including lighting, air conditioners, door locks, and security devices. C_i denotes temporal, spatial, and environmental context. Here, i denotes the index of an event-window sample rather than an individual sensor or switching event. The observations in E_i may include binary switching-state events from motion detectors, door contacts, and switch-type sensors, as well as continuous measurements such as temperature and

humidity. The inputs are therefore not uniformly defined as switch quantities. Binary and continuous observations are encoded according to sensor type and aggregated within the same time window to form $\mathbf{x}_i$. Device states in $\mathbf{D}_i$ may similarly be binary, categorical, or state-dependent.

The activity recognition label is denoted as $y_i \in \mathcal{Y}$, and the service-response label is denoted as $r_i \in \mathcal{R}$, where

$$\mathcal{R} = \{\text{auto}, \text{ask}, \text{none}\} \quad (2)$$

representing automatic execution, confirmation request, and no action, respectively. The model outputs include the activity prediction $\hat{y}_i$, recognition confidence p_i , candidate service relevance s_{ij} , service risk q_{ij} , and final response action $\hat{r}_i$, where j denotes a candidate service or the corresponding IoT device.

Commonly used smart home activity datasets focus primarily on activity labels and may not directly provide service-response action labels. Therefore, service mapping rules are constructed based on activity type, device availability, environmental context, and service risk. These rules are then used to assign a response label to each sample. Some device availability and service risk variables are constructed in a rule-based manner according to activity context, candidate service requirements, and safety or disturbance risk. They do not represent real service execution states directly provided by the original datasets. These labels are used only for service-response decision modeling and evaluation and are not equivalent to real user preferences. Edge-side decisions are also constrained by a latency budget τ , namely: $t_i \leq \tau$.

The objective of this paper is to improve the consistency between the predicted response action $\hat{r}_i$ and the rule-mapped response label r_i , while maintaining basic context recognition capability under the edge latency constraint. This objective can be summarized as:

$$\max f(\hat{y}_i, \hat{r}_i), \text{s.t. } t_i \leq \tau \quad (3)$$

where $f(\cdot)$ denotes an integrated objective determined by context recognition correctness and service-response consistency. The specific optimization objective is defined in the training strategy section.

B. OVERALL FRAMEWORK

To transform activity recognition results into smart home IoT service responses, this paper proposes an edge-constrained context-aware service-response framework, as shown in Fig. 3. Instead of directly mapping activity labels to actions, the framework considers five decision factors: service relevance, device availability, service risk, uncertainty, and edge latency. These factors jointly determine the selection of auto, ask, and none.

The framework consists of four parts: multi-source IoT event representation, service-oriented context state modeling, uncertainty-aware edge response decision-making, and low-burden feedback correction.

Figure 3 shows how sensor-level observations are progressively converted into service decisions. The activity recognition module produces activity prediction $\hat{y}_i$ and confidence p_i , with activity prediction serving as an intermediate representation rather than the final service output. These results are combined with contextual information, device availability, and service risk to construct the service-oriented context state. Candidate service relevance s_{ij} links the recognized household activity with potentially executable services.

The service-oriented context state is then processed by the uncertainty-aware edge response policy. Recognition confidence and service risk determine whether an action is sufficiently reliable for automatic execution or should require confirmation. Device availability removes infeasible services, while the latency constraint restricts decisions to those compatible with the prescribed edge-side budget. Response utility U_{ij} integrates these considerations before the policy selects auto, ask, or none.

Auto is used when service relevance and confidence are sufficiently high and risk remains acceptable. Ask is selected when confidence is insufficient, risk is elevated, or the service condition is

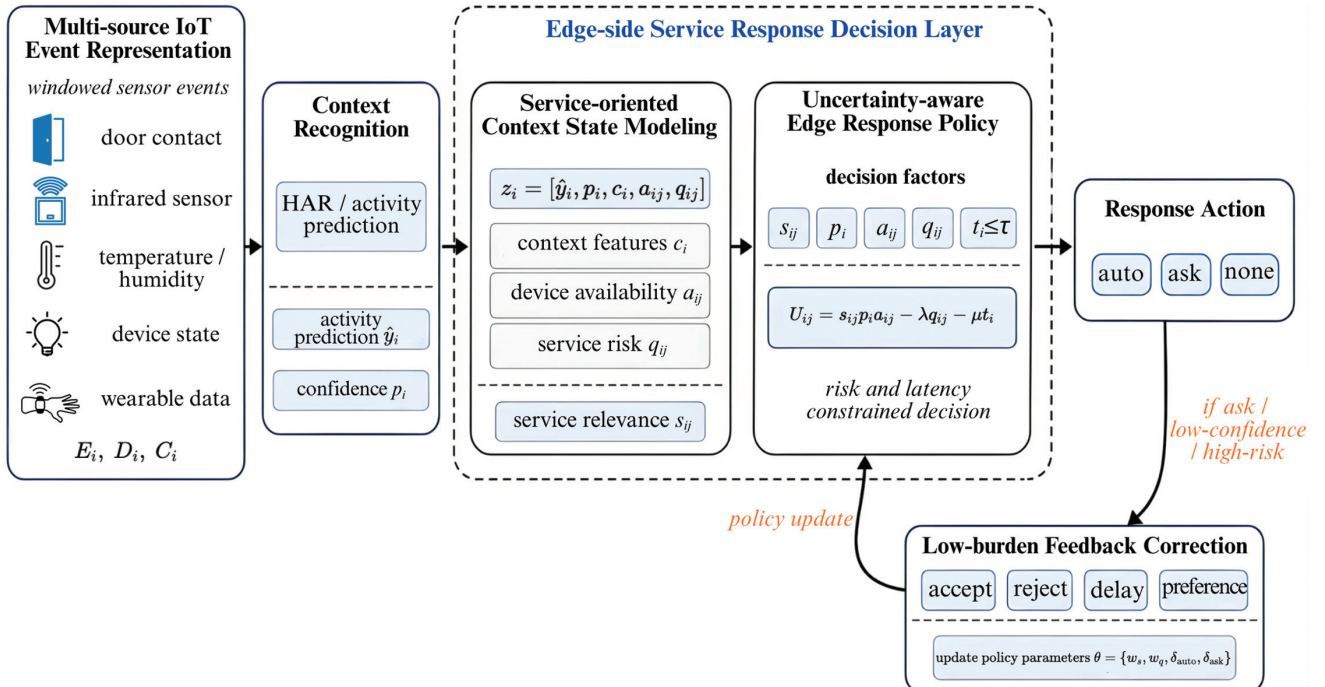


Fig. 3. Overall framework of the proposed edge-constrained context-aware service response method.

ambiguous. None suppresses responses that are irrelevant, unavailable, or unsuitable for the current context. The edge side therefore functions as a service-response decision layer rather than only as an inference deployment location.

Finally, the low-burden feedback correction module receives acceptance, rejection, delayed-reminder, or preference-adjustment feedback and assists in updating subsequent response policies. The feedback mechanism modifies policy-related parameters for later decisions rather than changing the current sensor observation. In the present experiments, it evaluates whether lightweight feedback can adjust subsequent response behavior and does not represent long-term personalized preference learning from real users.

The modules are linked through service relevance s_{ij} , service risk q_{ij} , response utility U_{ij} , and feedback parameter θ . The final outputs include activity prediction, recognition confidence, candidate service relevance, and service-response action. To further explain the internal mechanisms of Sections II.C, Fig. 4 presents the detailed structure of the service-oriented context state, uncertainty-aware response policy, and feedback correction mechanism.

C. SERVICE-ORIENTED CONTEXT STATE REPRESENTATION

To bridge the task gap between HAR and IoT service response, this paper constructs a service-oriented context state representation. An activity label alone cannot determine a service action, since the same activity may require different responses under different contexts and device states.

Given a sample $x_i = \{E_i, D_i, C_i\}$, the activity recognizer outputs the activity prediction $\hat{y}_i$ and recognition confidence p_i . This paper combines them with contextual information, device availability, and service risk to construct a service-oriented state:

$$z_i = [\hat{y}_i, p_i, c_i, a_{ij}, q_{ij}] \quad (4)$$

where c_i denotes contextual features, a_{ij} denotes the availability of candidate service j , and q_{ij} denotes service risk. In the current implementation, service risk is quantified through rule-based assignment rather than a learned estimator. The assignment considers four factors: consequences of incorrect execution, activity context,

associated device state, and potential safety or disturbance implications. Each candidate service is categorized as low, medium, or high risk and encoded as $q_{ij} = 0.2, 0.5$, or 0.8 , respectively. For example, ventilation during cooking is low risk, whereas nighttime lighting is medium risk. Automatic door-lock or security actions are high risk because incorrect execution may have greater consequences.

This rule-based design is adopted because CASAS-Aruba and ARAS do not provide native service-risk annotations. The assigned risk is therefore an operational variable rather than an observed user-specific preference. The framework does not require risk to remain rule-based; with sufficiently labeled service logs, q_{ij} can be replaced by a learnable estimator without changing the response-policy structure. The service relevance is further calculated as follows:

$$s_{ij} = \phi(\hat{y}_i, c_i, d_j) \cdot a_{ij} \cdot (1 - q_{ij}) \quad (5)$$

where $\phi(\cdot)$ represents the matching degree among the activity, context, and candidate service d_j , which can be implemented through service mapping rules or a learnable function. This module transforms activity recognition outputs into service-response candidates. If this module is removed, the system degenerates into an activity-label-only scheme, which may cause erroneous triggering under ambiguous activities, unavailable devices, or high-risk scenarios. This can be further verified through ablation settings such as w/o service relevance, w/o device state, and w/o service risk.

D. UNCERTAINTY-AWARE EDGE SERVICE-RESPONSE DECISION POLICY

As shown in Fig. 4, the uncertainty-aware edge service-response policy converts the context state into edge-side service actions. It jointly considers recognition confidence, service relevance, device availability, service risk, and latency budget.

For candidate service j , the edge response utility is defined as:

$$U_{ij} = s_{ij} \cdot p_i \cdot a_{ij} - \lambda q_{ij} - \mu t_i \quad (6)$$

where U_{ij} denotes response utility, t_i denotes end-to-end response time, and λ and μ denote the risk and latency weights, respectively. The response action is defined as:

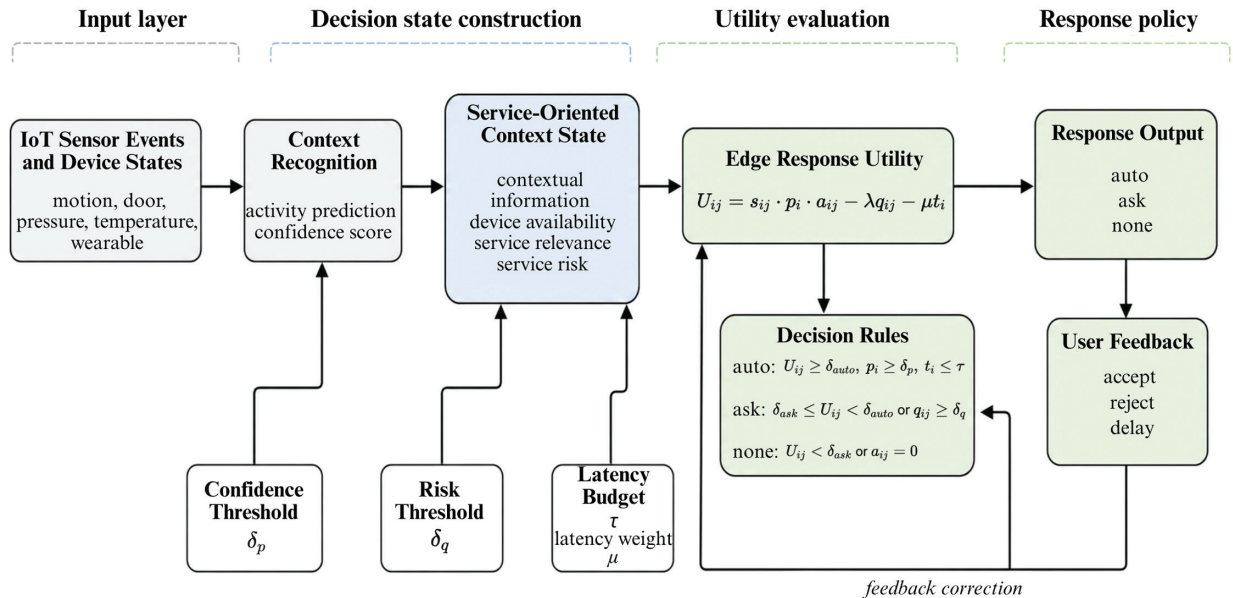


Fig. 4. Structure of the service-oriented context state and edge response policy.

$$\hat{r}_i = \begin{cases} \text{auto}, & U_{ij} \geq \delta_{\text{auto}}, p_i \geq \delta_p, t_i \leq \tau \\ \text{ask}, & \delta_{\text{ask}} \leq U_{ij} < \delta_{\text{auto}} \text{ or } q_{ij} \geq \delta_q \\ \text{none}, & U_{ij} < \delta_{\text{ask}} \text{ or } a_{ij} = 0 \end{cases} \quad (7)$$

Here, δ_{auto} is the minimum utility for automatic execution and δ_{ask} is the lower utility boundary for confirmation. The minimum recognition-confidence threshold for automatic execution is denoted by δ_p . The risk threshold above which confirmation is preferred is denoted by δ_q . These thresholds are selected on the 15% validation partition and fixed before testing. Their sensitivity is further evaluated in the response-threshold sensitivity experiment.

For multi-resident data such as ARAS, the framework handles identity ambiguity indirectly through recognition confidence and contextual uncertainty rather than explicit resident identification. When overlapping sensor events reduce activity reliability, the resulting decrease in p_i makes automatic execution less likely and increases the probability of an ask response. Context and device-state information provide additional constraints under overlapping activities. However, the framework does not assign sensor events to individual residents and should not be interpreted as a resident-identification model. Remaining ambiguous cases are examined in the failure analysis.

The policy therefore controls service actions under uncertainty, risk, and latency constraints rather than relying on activity predictions alone. Without the uncertainty-aware policy, the system reduces to static rules, a confidence-threshold strategy, or ordinary edge-side recognition. This is examined through Edge Response without Uncertainty, w/o uncertainty policy, and w/o latency constraint.

E. LOW-BURDEN FEEDBACK CORRECTION

To incorporate user feedback into the service-response correction process, this paper designs a low-burden feedback correction mechanism. This module does not perform emotion recognition. Instead, when the system outputs ask, it receives user feedback u_i , including acceptance, rejection, delayed reminder, or preference adjustment, and uses it to correct subsequent response policies. In this paper, the feedback correction experiment adopts rule-constructed feedback to verify the role of feedback signals as an auxiliary updating mechanism. Different feedback types make limited adjustments to service relevance weights, risk weights, and response thresholds. Real long-term user feedback is beyond the scope of the present experiment. Accordingly, the experiment evaluates whether feedback signals can alter subsequent response parameters rather than demonstrating personalized preference learning from real users.

The feedback update parameters are defined as:

$$\theta_i = \{w_s, w_q, \delta_{\text{auto}}, \delta_{\text{ask}}\}, \theta_{i+1} = \theta_i + \Delta(u_i) \quad (8)$$

where w_s denotes the service relevance weight, w_q denotes the risk weight, and δ_{auto} and δ_{ask} denote the automatic execution threshold and confirmation request threshold, respectively. $\Delta(u_i)$ denotes the feedback correction amount. Acceptance feedback increases service relevance or lowers the automatic-execution threshold under similar contexts. Rejection feedback increases the risk weight or the automatic-execution threshold. Delayed-reminder feedback reduces the tendency toward automatic execution in the current time period. Preference-adjustment feedback modifies response tendencies under specific contexts. If this module is removed, the system has difficulty adjusting subsequent responses according to feedback. This mechanism can be verified through the w/o feedback correction ablation setting.

F. OBJECTIVE FUNCTION AND TRAINING STRATEGY

This paper adopts a staged training strategy. First, the activity recognition module is trained to output activity prediction $\hat{y}_i$ and recognition confidence p_i from multi-source IoT events. Then, response labels r_i are constructed based on service mapping rules, and the service-response decision module is trained accordingly. Finally, the response policy parameters are updated in the feedback stage. This strategy avoids mixing activity recognition, response mapping, and feedback correction into a single training objective, thereby facilitating separate analysis of the role of each stage.

The overall objective function is defined as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{act}} + \alpha \mathcal{L}_{\text{res}} + \beta \mathcal{L}_{\text{risk}} + \gamma \mathcal{L}_{\text{lat}} \quad (9)$$

where $\mathcal{L}_{\text{act}} = \text{CE}(\hat{y}_i, y_i)$ is used to maintain basic context recognition capability, and $\mathcal{L}_{\text{res}} = \text{CE}(\hat{r}_i, r_i)$ is used to optimize the selection of auto, ask, and none response actions. These two terms correspond to the activity recognition module and the edge response decision module, respectively.

The risk constraint term is defined as:

$$\mathcal{L}_{\text{risk}} = \mathbb{I}(\hat{r}_i = \text{auto}) \cdot q_{ij} \quad (10)$$

where $\mathbb{I}(\cdot)$ is the indicator function, and q_{ij} denotes the triggering risk of candidate service j . This term is used to penalize automatic execution under high-risk contexts. The latency constraint term is defined as:

$$\mathcal{L}_{\text{lat}} = \max(0, t_i - \tau) \quad (11)$$

where t_i denotes the end-to-end response time, and τ denotes the latency budget. The hyperparameters were selected on the 15% validation partition of each dataset using grid search. The search spaces were $\alpha \in 0.5, 1.0, 1.5$, $\beta \in 0.1, 0.3, 0.5$, and $\gamma \in 0.01, 0.05, 0.10$. Validation Service Response Accuracy (SRA) was used as the primary selection criterion, with lower Auto Error used to resolve configurations with similar SRA while maintaining the latency budget. The resulting values were $\alpha = 1.0$, $\beta = 0.3$, and $\gamma = 0.05$, and the same values were fixed for all test-set evaluations. No test-set results were used for hyperparameter selection. A larger β makes the policy more conservative, whereas a larger γ increases the emphasis on response timeliness. The objective design is further examined through six ablation variants. These variants remove service relevance, device state, service risk, the uncertainty policy, the latency constraint, and feedback correction, respectively.

G. ALGORITHMIC PROCEDURE AND COMPLEXITY ANALYSIS

The proposed method consists of four stages: context recognition, service state construction, edge response decision-making, and feedback correction. The system first performs windowing on multi-source IoT events to obtain activity prediction and recognition confidence. It then constructs a service-oriented state by combining device state, environmental context, and service risk and calculates candidate service relevance. At the edge side, the system outputs the response action according to response utility, risk, and latency budget. If the output is a confirmation request, user feedback is received and used to correct subsequent response policies.

Let the time-window length be T , the number of sensors be M , and the number of candidate services be K . The complexity of the activity recognition module depends on the selected backbone and is denoted as $O(F_{\text{act}})$. The calculation of service relevance and

Algorithm 1. Edge-Aware Service Response Decision

Input: E_i , D_i , C_i , latency budget τ
Output: response action $\hat{r}_i$
 Predict activity $\hat{y}_i$ and confidence p_i from E_i .
 Construct the service-oriented state z_i using $\hat{y}_i$, p_i , D_i , and C_i .
 For each candidate service j , compute service relevance s_{ij} .
 Compute response utility U_{ij} under risk and latency constraints.
 If $U_{ij} \geq \delta_{\text{auto}}$, $p_i \geq \delta_p$, and $t_i \leq \tau$, output auto.
 If $\delta_{\text{ask}} \leq U_{ij} < \delta_{\text{auto}}$ or $q_{ij} \geq \delta_q$, output ask.
 If $U_{ij} < \delta_{\text{ask}}$ or $a_{ij} = 0$, output none.
 If $\hat{r}_i = \text{ask}$, collect feedback u_i and update θ .

response utility requires traversal over candidate services, with a complexity of $O(K)$. Therefore, the single-sample inference complexity is: $O(F_{\text{act}} + K)$, where F_{act} denotes the forward computation cost of the activity recognition model and K denotes the number of candidate services. This paper does not treat lightweight network design as an independent contribution; instead, it places service-response decision-making at the edge side. End-to-end response time, average latency, maximum latency, and whether the latency budget τ is satisfied are used to evaluate the feasibility of edge response.

IV. EXPERIMENTS AND RESULTS

A. EXPERIMENTAL SETUP

1). DATASETS. This paper uses two smart home datasets, CASAS-Aruba and ARAS. CASAS-Aruba represents a single-resident daily-living environment with 11 activity categories and events from 31 motion sensors, 4 door sensors, and 5 temperature sensors. ARAS contains two real multi-resident homes, each equipped with 20 ZigBee binary sensors, and provides 27 activity labels. The datasets therefore provide complementary single- and multi-resident

settings for evaluating context recognition and service-response decisions.

Neither dataset was treated as a native service-response dataset. Original sensor streams were used for context recognition, whereas response labels were constructed through service mapping rules. Events were chronologically divided into 70% training, 15% validation, and 15% testing partitions. Windows were generated separately within each partition to avoid leakage. A window length of 60 s with a 30-s step was used. The terminal event determined the activity label. The window length is 60 s with a 30-s step, and the terminal event determines the activity label.

Service mapping rules assign auto, ask, or none according to activity, context, device state, and service risk. Because the datasets lack real service-execution logs and preference labels, device availability and risk are constructed from activity context, service requirements, and safety or disturbance implications. Consistent with Section II.C, low-, medium-, and high-risk services are encoded as 0.2, 0.5, and 0.8. These labels support response-decision experiments and do not represent real user preferences. Table II provides representative examples.

The sensing configurations in Table I are directly associated with the experimental difficulty and the response mappings in Table II. CASAS-Aruba mainly evaluates single-resident context-to-service decisions, whereas ARAS introduces greater ambiguity through overlapping multi-resident activities. Sensor observations provide activity and contextual evidence. The rule fields in Table II combine this evidence with device availability and service risk to construct response labels.

2). BASELINES. This paper compares the proposed method with eight baselines, as summarized in Table III. Hidden Markov Model (HMM) [31] and Conditional Random Field (CRF) [32] represent traditional probabilistic and structured sequence models. Deep-ConvLSTM [33], Bootstrapped Smart Home HAR [34], and Tiny Transformer HAR [35] represent deep, recent, and lightweight edge HAR methods. Rule-based Service Mapping [36] directly converts activity labels into responses, while Context-aware Rule

Table I. Overview of datasets and experimental protocol

Dataset	Scenario	Sensor types and quantity	Sensor function	Activities	Response label source
CASAS-Aruba	Single-resident smart home	31 motion, 4 door, 5 temperature sensors	Motion: occupancy and transitions; door: entry/exit and open/close events; temperature: environmental context	11 daily-living categories, including meal preparation, resting, eating, working, sleeping, household routines, and entry/exit	Service mapping rules
ARAS	Two multi-resident real homes	20 ZigBee binary sensors per house, including contact, pressure/force, proximity, distance, light/temperature, and infrared-related sensing	Presence, object/furniture interaction, door/cabinet state, room use, and activity-related environmental events	27 multi-resident daily-living activities covering food preparation/eating, sleeping, hygiene, household tasks, leisure/communication, and leaving home	Service mapping rules

Table II. Examples of service mapping rules

Activity	Context	Device state	Risk level	Response
Night getting up	night, bedroom	light available	medium	ask
Leaving home	door open	lock available	high	ask
Cooking	kitchen	ventilation available	low	auto
Resting	living room	no required service	low	none

Table III. Summary of baseline methods

Category	Method	Role
Traditional HAR	HMM [31]	Probabilistic temporal baseline
Traditional HAR	CRF [32]	Structured sequence baseline
Deep HAR	DeepConvLSTM [33]	Deep sensor-sequence baseline
Smart home HAR	Bootstrapped HAR [34]	Recent smart home HAR baseline
Edge HAR	Tiny Transformer HAR [35]	Lightweight edge baseline
Rule response	Rule-based mapping [36]	Fixed response-rule baseline
Context reasoning	Context-aware Reasoner [37]	Context-enhanced rule baseline
Response policy	Edge Response w/o Uncertainty [38]	Internal uncertainty-removal baseline
Proposed	Proposed method	Full response-decision model

Reasoner [37] additionally considers time, location, and device state. Edge Response without Uncertainty [38] uses the same context input as the proposed method but removes uncertainty-aware decision-making. For HAR baselines, predicted activity labels are converted into auto, ask, and none using the same mapping protocol. This allows us to assess whether improved recognition directly benefits service-response decisions.

3). IMPLEMENTATION DETAILS. All methods used the same data split, input windows, and preprocessing. Sensor events were converted into fixed-length windows, and sensor IDs were represented using one-hot encoding or embeddings. Temporal information was encoded by hour, weekday, and day/night state. Device states were represented using binary variables. The activity recognition module was trained first. The service-response module was then trained using the constructed response labels, after which the response-policy parameters were updated in the feedback experiment.

Trainable models use Adam with an initial learning rate of 1×10^{-3} , batch size 64, a maximum of 100 epochs, and early stopping with patience 10. These settings were determined through preliminary validation-based tuning rather than test performance. The learning rate and batch size provided stable optimization. The 100-epoch limit allowed sufficient convergence. Early stopping terminated training when validation performance ceased improving, thereby reducing overfitting. The same configuration was applied consistently to trainable comparison methods.

Consistent with Section II.F, $\alpha = 1.0$, $\beta = 0.3$, and $\gamma = 0.05$ were selected on the 15% validation partition and fixed during testing. Response and risk thresholds were also determined from validation data without test-set adjustment. Edge-response experiments were conducted on a Raspberry Pi 4 or equivalent constrained gateway with fixed CPU-thread and memory settings. Each setting

was repeated 100 times. Average and maximum Edge Response Latency (ERL) were reported.

4). EVALUATION METRICS. This paper uses three evaluation metrics. Macro-F1 measures activity recognition performance. SRA measures the consistency between predicted response actions and rule-mapped labels. SRA is not equivalent to real user satisfaction or real preference prediction. ERL measures the end-to-end time from the input event window to the output response action. It covers event input, activity prediction, service relevance calculation, response utility calculation, and action output. For cloud-only inference, data uploading and response returning time are additionally included.

B. MAIN RESULTS

This section compares the proposed method with all baselines on CASAS-Aruba and ARAS. The evaluation includes context recognition performance, response decision consistency, confirmation-action performance, risky automatic execution, and ERL. For HAR baselines, predicted activity labels are converted into response actions using the same service mapping protocol. Therefore, the comparison examines not only recognition accuracy but also whether recognition outputs can be reliably transformed into auto, ask, and none decisions.

As shown in Tables IV and V, deep and lightweight HAR methods generally achieve stronger Macro-F1 than traditional temporal models. However, higher activity recognition accuracy does not always lead to better service-response decisions. Rule-based and context-aware baselines show competitive latency because they involve simpler decision procedures, but their response consistency is limited under ambiguous or high-risk contexts. Edge Response without Uncertainty performs efficiently, yet its confirmation-related performance is weaker because it lacks explicit uncertainty handling.

Table IV. Main results on CASAS-Aruba

Method	Macro-F1 $\uparrow$	SRA $\uparrow$	Ask-F1 $\uparrow$	Auto Error $\downarrow$	ERL $\downarrow$
HMM	82.6 $\pm$ 1.5	78.9 $\pm$ 1.6	72.4 $\pm$ 2.1	9.8 $\pm$ 0.9	58.4 $\pm$ 3.7 ms
CRF	84.1 $\pm$ 1.3	80.5 $\pm$ 1.5	74.1 $\pm$ 1.8	8.9 $\pm$ 0.8	63.7 $\pm$ 4.0 ms
DeepConvLSTM	88.1 $\pm$ 1.1	84.2 $\pm$ 1.3	77.8 $\pm$ 1.7	7.5 $\pm$ 0.8	92.6 $\pm$ 6.3 ms
Bootstrapped HAR	89.4 $\pm$ 0.7	85.7 $\pm$ 1.0	79.2 $\pm$ 1.4	6.9 $\pm$ 0.6	91.8 $\pm$ 5.6 ms
Tiny transformer HAR	88.6 $\pm$ 0.9	85.2 $\pm$ 1.1	78.8 $\pm$ 1.5	7.1 $\pm$ 0.7	69.4 $\pm$ 4.8 ms
Rule-based mapping	82.6 $\pm$ 1.5	81.6 $\pm$ 1.4	74.9 $\pm$ 1.9	8.2 $\pm$ 0.9	54.8 $\pm$ 3.6 ms
Context-aware Reasoner	84.1 $\pm$ 1.3	84.6 $\pm$ 1.2	78.1 $\pm$ 1.6	7.3 $\pm$ 0.7	67.5 $\pm$ 4.5 ms
Edge Response w/o Uncertainty	89.1 $\pm$ 0.8	86.8 $\pm$ 0.9	80.1 $\pm$ 1.3	6.6 $\pm$ 0.6	61.7 $\pm$ 4.2 ms
Proposed	89.1 $\pm$ 0.8	89.0 $\pm$ 0.8	83.3 $\pm$ 1.2	5.4 $\pm$ 0.6	74.6 $\pm$ 4.8 ms

Table V. Main results on ARAS

Method	Macro-F1 $\uparrow$	SRA $\uparrow$	Ask-F1 $\uparrow$	Auto Error $\downarrow$	ERL $\downarrow$
HMM	74.8 $\pm$ 1.9	70.6 $\pm$ 2.0	64.2 $\pm$ 2.5	13.7 $\pm$ 1.1	60.1 $\pm$ 4.1 ms
CRF	76.3 $\pm$ 1.6	72.4 $\pm$ 1.8	66.1 $\pm$ 2.2	12.8 $\pm$ 1.0	65.2 $\pm$ 4.4 ms
DeepConvLSTM	79.8 $\pm$ 1.5	77.1 $\pm$ 1.6	70.2 $\pm$ 2.0	10.8 $\pm$ 0.9	96.7 $\pm$ 6.8 ms
Bootstrapped HAR	80.9 $\pm$ 1.2	78.8 $\pm$ 1.4	72.5 $\pm$ 1.8	9.9 $\pm$ 0.8	94.3 $\pm$ 6.2 ms
Tiny transformer HAR	81.6 $\pm$ 1.4	79.5 $\pm$ 1.3	73.1 $\pm$ 1.7	9.4 $\pm$ 0.8	72.7 $\pm$ 5.1 ms
Rule-based mapping	74.8 $\pm$ 1.9	74.9 $\pm$ 1.8	67.5 $\pm$ 2.2	11.5 $\pm$ 1.0	56.9 $\pm$ 3.9 ms
Context-aware Reasoner	76.3 $\pm$ 1.6	78.4 $\pm$ 1.5	71.6 $\pm$ 1.9	10.2 $\pm$ 0.8	69.1 $\pm$ 4.7 ms
Edge response w/o uncertainty	82.4 $\pm$ 1.1	81.1 $\pm$ 1.3	75.2 $\pm$ 1.6	8.8 $\pm$ 0.7	64.9 $\pm$ 4.6 ms
Proposed	82.2 $\pm$ 1.2	83.3 $\pm$ 1.0	78.1 $\pm$ 1.5	7.3 $\pm$ 0.7	78.5 $\pm$ 5.2 ms

The proposed method achieves the best SRA and Ask-F1 on both datasets and reduces risky automatic execution. However, it does not obtain the lowest ERL or the highest Macro-F1 in every case. These results suggest that the main advantage of the proposed method lies in response decision consistency rather than pure recognition accuracy or minimal latency.

Overall, the proposed method is not uniformly superior across all metrics. Bootstrapped HAR achieves the highest Macro-F1 on CASAS-Aruba. On ARAS, Edge Response without Uncertainty achieves a slightly higher Macro-F1 than the proposed method. Rule-based and uncertainty-free response baselines also obtain lower ERL. However, the proposed method performs better on the metrics most directly related to the paper’s objective, namely SRA, Ask-F1, and Auto Error. This indicates that incorporating service relevance, device state, risk constraints, and uncertainty-aware response selection is more useful for service-response decision-making than optimizing activity recognition or latency alone.

C. STATISTICAL SIGNIFICANCE ANALYSIS

To further examine whether the improvement in service-response consistency is statistically reliable, paired t-tests were conducted on SRA over repeated runs. The proposed method was compared with three strong baselines: Bootstrapped HAR, Tiny Transformer HAR, and Edge Response without Uncertainty. These methods were selected because they achieved relatively competitive results in the main comparison. The significance level was set to 0.05, and 95% confidence intervals were reported for the SRA difference between the proposed method and each baseline.

As shown in Table VI, the proposed method achieves statistically significant SRA improvements over the three selected baselines on both datasets. The improvement is more evident on ARAS, where multi-resident activity overlap and contextual ambiguity make response selection more difficult. On CASAS-Aruba, the improvement over Edge Response without Uncertainty is smaller but remains statistically significant. This result indicates that

uncertainty-aware response selection contributes to service-response consistency even when the underlying context-recognition input is unchanged. These results support the conclusion that the proposed method improves rule-mapped response action consistency, rather than merely benefiting from random variation across runs.

Overall, the statistical results are consistent with the main experimental findings. The proposed method does not dominate all secondary indicators, but its advantage on SRA remains statistically significant against strong recognition-based and response-based baselines. Therefore, the observed improvement can be attributed mainly to service-oriented state modeling and uncertainty-aware response decision-making under the rule-mapped label setting.

D. REPRESENTATION AND FEATURE SPACE ANALYSIS

To examine whether the proposed service-oriented context state is more suitable for response action selection, this section analyzes the learned feature space on CASAS-Aruba and ARAS. The analysis includes t-distributed Stochastic Neighbor Embedding (t-SNE) visualization, feature distribution, and quantitative separation metrics. This experiment is used only to support the representation-level explanation of the main results, rather than to claim complete separability among response actions.

Figure 5 shows the t-SNE visualization of activity-only representations and the proposed service-oriented context representations. In the activity-only setting, auto and ask samples are mixed in several regions, especially for activities with similar sensor patterns. After adding confidence, device state, service relevance, and risk information, the three response categories become more distinguishable. However, the boundary between auto and ask remains partially overlapped, which is consistent with the later failure analysis under medium-risk and low-confidence scenarios.

Figure 6 presents the distribution of confidence- and risk-related features across response actions. Auto samples are mainly associated with higher confidence and lower risk, while ask samples appear

Table VI. Statistical significance analysis based on SRA

Dataset	Comparison	Δ SRA	95% CI	p-Value
CASAS-Aruba	Proposed vs Bootstrapped HAR	+3.3	[2.1, 4.5]	0.003
CASAS-Aruba	Proposed vs Tiny Transformer HAR	+3.8	[2.4, 5.0]	0.002
CASAS-Aruba	Proposed vs Edge Response w/o Uncertainty	+2.2	[0.7, 3.5]	0.018
ARAS	Proposed vs Bootstrapped HAR	+4.5	[2.8, 6.1]	0.002
ARAS	Proposed vs Tiny Transformer HAR	+3.8	[2.0, 5.4]	0.006
ARAS	Proposed vs Edge Response w/o Uncertainty	+2.2	[0.6, 3.7]	0.021

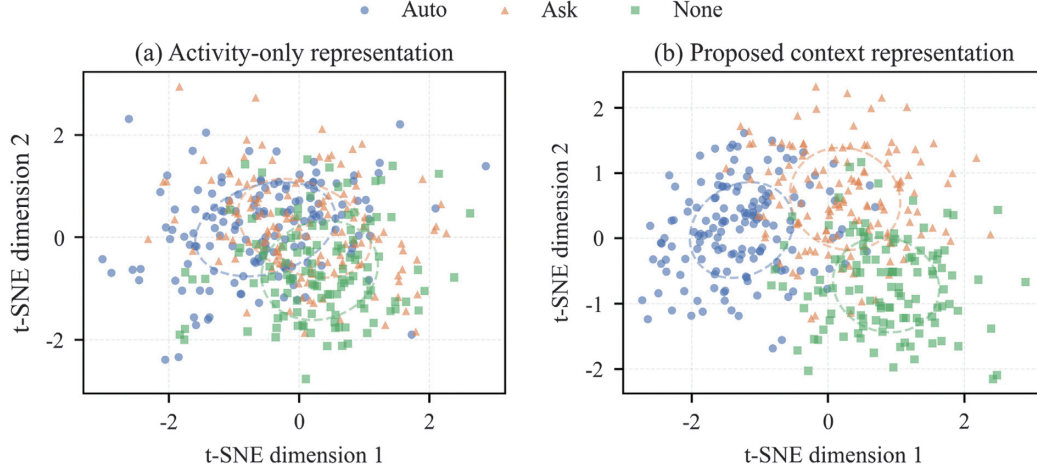


Fig. 5. t-SNE visualization of response-action embeddings. (a) Activity-only representation. (b) Proposed service-oriented context representation.

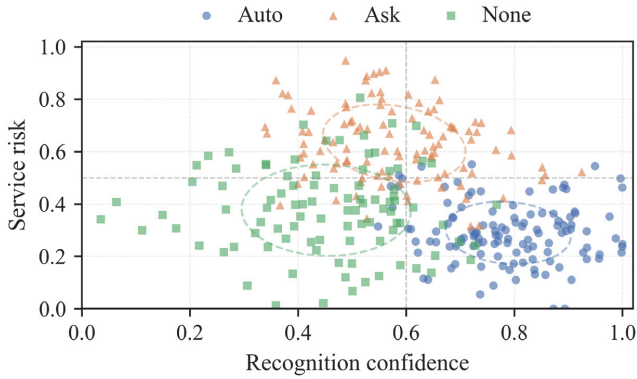


Fig. 6. Feature distribution of response actions under confidence and risk levels.

more frequently in medium-confidence or higher-risk regions. None samples are more related to low service relevance or unavailable device states. These results indicate that response actions are affected by multiple service-side factors, rather than by activity labels alone.

Table VII further reports the silhouette score and inter-class distance. Compared with activity-only features, the proposed context state improves feature-space separation on both datasets. The improvement is more visible on ARAS, suggesting that service-side factors are more useful when multi-resident overlap and contextual ambiguity are stronger. Nevertheless, the separation scores remain moderate, indicating that some ambiguous response cases are still difficult to distinguish.

Overall, the representation analysis supports the claim that the proposed method builds a more response-oriented feature space. It improves the separation of auto, ask, and none decisions but does

not remove all overlap. This also explains why the proposed method improves SRA and Ask-F1 in the main results while still producing some errors in ambiguous smart home scenarios.

E. INTERPRETABILITY AND DECISION-FACTOR ANALYSIS

This section examines the factors that influence response-action selection on the CASAS-Aruba and ARAS test sets. The analysis considers recognition confidence, service relevance, device availability, service risk, contextual information, and the latency constraint. Since no independent attention module is introduced, the analysis focuses on decision-factor interpretability.

Figure 7 shows factor contribution scores estimated by feature perturbation. Service relevance and service risk have the strongest

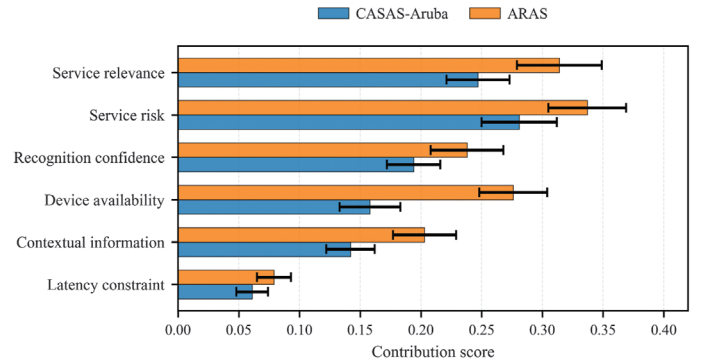


Fig. 7. Decision-factor contribution analysis for response action selection.

Table VII. Feature-space separation analysis

Representation	CASAS Silhouette $\uparrow$	CASAS Inter-class Distance $\uparrow$	ARAS Silhouette $\uparrow$	ARAS Inter-class Distance $\uparrow$
Activity-only representation	0.29 ± 0.04	1.36 ± 0.10	0.22 ± 0.05	1.14 ± 0.12
Activity + confidence	0.35 ± 0.03	1.49 ± 0.09	0.28 ± 0.04	1.27 ± 0.10
Activity + context + device state	0.39 ± 0.05	1.61 ± 0.13	0.33 ± 0.05	1.43 ± 0.14
Proposed context state	0.46 ± 0.04	1.77 ± 0.12	0.40 ± 0.06	1.59 ± 0.15

Table VIII. SRA change after decision-factor perturbation

Perturbed factor	CASAS Δ SRA ↓	ARAS Δ SRA ↓	Main affected response
Service relevance	2.3 ± 0.6	3.0 ± 0.8	auto/none
Service risk	2.7 ± 0.5	3.5 ± 0.7	auto/ask
Recognition confidence	1.8 ± 0.4	2.4 ± 0.7	ask
Device availability	1.5 ± 0.5	2.7 ± 0.6	auto/none
Contextual information	1.4 ± 0.4	2.0 ± 0.5	ask/none
Latency constraint	0.6 ± 0.3	0.8 ± 0.2	response timing

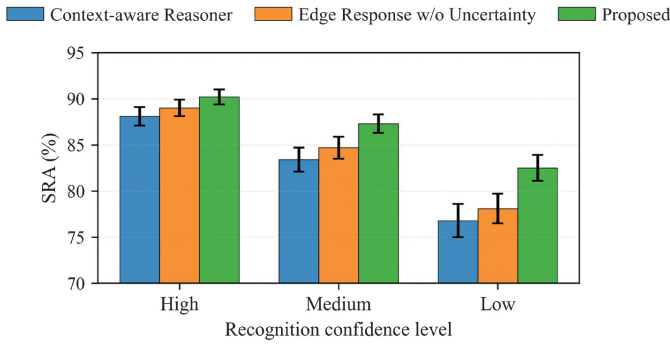
influence on response action selection, while confidence mainly affects whether the model chooses ask under uncertain contexts. Device availability has a larger effect on ARAS, reflecting the higher complexity of multi-resident scenarios. Latency has limited influence on the response category but remains important for controlling edge response time.

Table VIII reports the SRA decrease after perturbing each factor. Disturbing service relevance or service risk causes larger performance drops, indicating that activity labels alone are insufficient for response decisions. Confidence and device availability also

affect response consistency, especially for ask and auto decisions. The latency constraint has a smaller effect on SRA but is more related to ERL and latency-budget satisfaction.

Figure 8 further compares SRA under different confidence levels. Test samples were divided into high-, medium-, and low-confidence groups according to activity prediction confidence. The proposed method shows only a modest advantage under high-confidence conditions, but the gap becomes larger under medium- and low-confidence samples. This suggests that the uncertainty-aware policy mainly contributes to response consistency in ambiguous scenarios.

Overall, the interpretability results indicate that the proposed method does not rely only on activity recognition outputs. Response actions are jointly affected by service relevance, risk, confidence, device state, and contextual constraints. This supports the central claim that smart home IoT response should be modeled as a service-oriented decision process rather than a direct extension of activity classification.

**Fig. 8.** SRA under different recognition confidence levels.

F. ERROR AND FAILURE CASE ANALYSIS

Although the proposed method improves response consistency under rule-mapped labels, errors still occur in ambiguous or incomplete smart home contexts. Before analyzing these errors, Fig. 9 presents a representative six-panel response case. It illustrates the transformation from raw sensor events to windowed inputs, activity

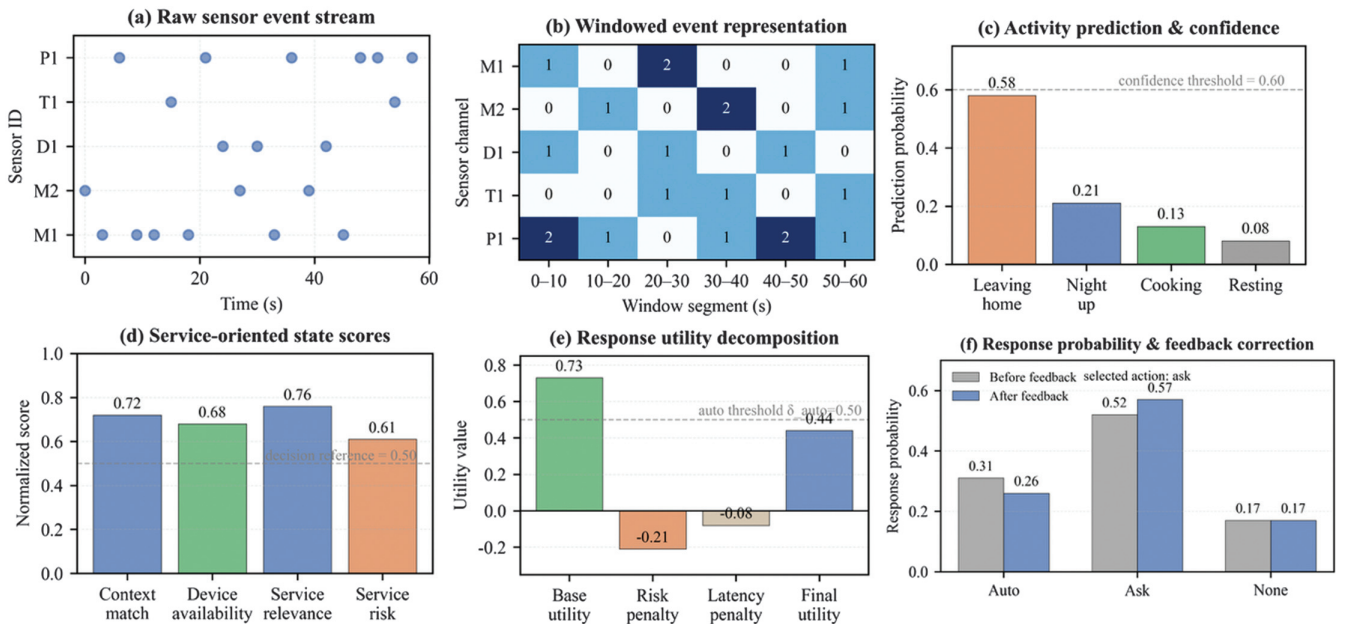
**Fig. 9.** Six-panel visualization of a representative service response case. (a) Raw sensor event stream. (b) Windowed event representation. (c) Activity prediction and confidence. (d) Service-oriented state scores. (e) Response utility decomposition. (f) Response probability and feedback correction)

Table IX. Representative failure cases and future directions

Case	Scenario	Triggering factor	Predicted	Expected	Main reason	Future direction
1	Multi-resident overlap	Kitchen and hallway sensors are triggered within the same window	auto	ask	Activity boundaries become ambiguous when events from different residents overlap.	Resident-aware context modeling
2	Sensor missingness	A key motion sensor is missing during a bedroom-to-hallway transition	none	ask	Incomplete sensing weakens service relevance estimation.	Missing-sensor robust inference
3	High-risk conflict	Door-open event occurs with low-confidence leaving-home prediction	auto	ask	Service risk is underestimated when confidence and device state conflict.	Adaptive risk calibration
4	Preference mismatch	The user repeatedly rejects evening reminders	ask	none	Rule-mapped labels cannot fully represent personal preference.	Real feedback logs and preference learning

prediction, service-oriented context state, response utility, response probability, and feedback correction. This visualization helps clarify where uncertainty, incomplete sensing, or response-rule limitations may affect the final response decision.

As shown in Table IX, errors mainly arise from multi-resident overlap, sensor missingness, high-risk conflicts, and preference mismatch. Overlapping activities increase uncertainty in activity boundaries, while missing sensors weaken service relevance estimation. High-risk conflicts may still cause overly aggressive automatic execution when risk is underestimated. Preference mismatch further indicates that rule-mapped labels cannot replace real user preferences.

These cases show that the proposed method is still limited by sensing completeness, risk-rule design, and the lack of real long-term feedback data. Future work should therefore incorporate resident-aware modeling, missing-sensor robust inference, adaptive risk calibration, and real user confirmation logs. This analysis also clarifies the boundary of the present experiments. The response labels are constructed from service mapping rules. Therefore, the current results mainly demonstrate rule-mapped response consistency rather than real user preference adaptation.

G. ABLATION STUDY

Ablation experiments were conducted on CASAS-Aruba and ARAS to evaluate the contribution of each component. All variants

used the same data split, preprocessing, activity recognition backbone, and training settings as the full model. Only one component was removed in each variant. SRA, Ask-F1, and ERL were used as the main metrics. All results were reported as mean $\pm$ standard deviation over five runs.

Figure 10 further shows how response-action distributions change after removing key components. Removing service risk increases the proportion of automatic execution, indicating more aggressive response behavior. Removing the uncertainty-aware policy reduces confirmation requests, while removing service relevance increases none decisions because service matching becomes less reliable. These distributional changes explain the SRA and Ask-F1 decreases reported in Table X.

Removing the latency constraint causes only a small change in SRA but clearly increases ERL. This result shows that latency modeling mainly supports edge-side response feasibility rather than response-category selection. Removing feedback correction leads to a moderate decrease in SRA and Ask-F1, suggesting that feedback provides auxiliary adjustment under repeated or preference-sensitive contexts. Since the feedback is rule-constructed, this result should be interpreted as preliminary evidence rather than proof of real preference adaptation.

Overall, the ablation results confirm that service relevance, device state, service risk, and uncertainty modeling are central to response consistency. The latency constraint mainly controls edge response time. Feedback correction provides auxiliary adjustment.

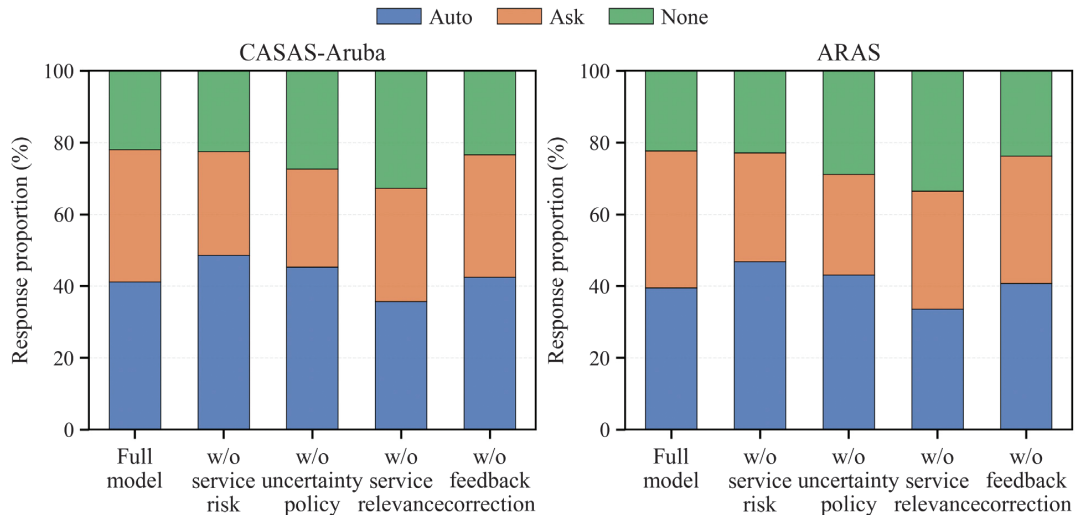
**Fig. 10.** Response-action distributions after removing key components.

Table X. Ablation study results

Variant	CASAS SRA $\uparrow$	ARAS SRA $\uparrow$	CASAS Ask-F1 $\uparrow$	ARAS Ask-F1 $\uparrow$	CASAS ERL $\downarrow$
Full model	89.0 ± 0.8	83.3 ± 1.0	83.3 ± 1.2	78.1 ± 1.5	74.6 ± 4.8 ms
w/o service relevance	86.4 ± 1.1	80.1 ± 1.5	80.2 ± 1.6	74.5 ± 1.8	73.8 ± 5.1 ms
w/o device state	87.0 ± 1.0	80.5 ± 1.7	81.1 ± 1.5	74.9 ± 2.0	74.1 ± 4.7 ms
w/o service risk	85.8 ± 1.3	79.6 ± 1.6	79.4 ± 1.8	73.6 ± 1.9	74.7 ± 5.2 ms
w/o uncertainty policy	86.1 ± 1.2	79.8 ± 1.8	78.7 ± 1.9	72.8 ± 2.1	72.9 ± 4.9 ms
w/o latency constraint	88.4 ± 1.0	82.5 ± 1.3	82.6 ± 1.4	77.4 ± 1.6	96.8 ± 6.7 ms
w/o feedback correction	88.1 ± 1.1	82.2 ± 1.4	82.0 ± 1.5	76.5 ± 1.7	74.3 ± 5.4 ms

This is consistent with the design motivation of the proposed service-oriented context state and edge response policy.

H. RESPONSE THRESHOLD SENSITIVITY ANALYSIS

A one-factor-at-a-time sensitivity analysis was conducted for the proposed method to examine the effect of response thresholds. Three threshold groups were evaluated: recognition-confidence threshold δ_p , service-risk threshold δ_q , and the auto/ask utility thresholds δ_{auto} and δ_{ask} . Each threshold was varied at 0.8 $\times$, 0.9 $\times$, 1.0 $\times$, 1.1 $\times$, and 1.2 $\times$ its validation-selected value, while the remaining settings were fixed. No retraining was performed.

SRA, Ask-F1, and Auto Error were used to evaluate threshold effects. Figure 11 summarizes the results on CASAS-Aruba and ARAS.

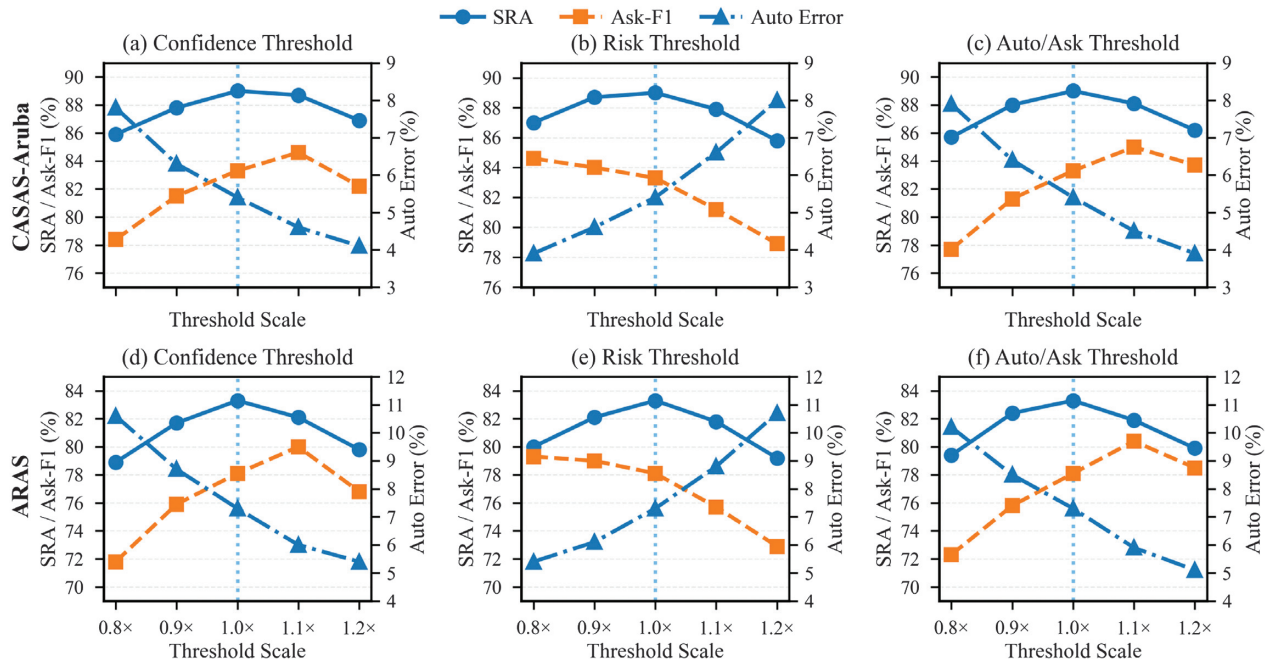
The results show that performance remains relatively stable within a moderate range around the selected thresholds. Increasing the confidence threshold or lowering the risk threshold makes the policy more conservative and generally reduces Auto Error. However, excessively strict settings may reduce SRA or Ask-F1. More permissive settings increase automatic execution and may lower confirmation burden, but Auto Error rises when uncertain or

higher-risk cases are executed directly. Similar trade-offs are observed when varying the auto/ask utility boundary.

Overall, the results indicate that threshold selection affects the balance among response consistency, confirmation behavior, and erroneous automatic execution. The validation-selected thresholds provide a balanced operating point, while overly conservative or permissive settings lead to less favorable response behavior.

V. DISCUSSION

The main results indicate that the benefit of the proposed framework comes primarily from transforming activity recognition outputs into service-oriented decisions rather than from improving HAR accuracy alone. On CASAS-Aruba, the proposed method achieves an SRA of $89.0 \pm 0.8\%$, compared with $86.8 \pm 0.9\%$ for Edge Response without Uncertainty. Both methods obtain a Macro-F1 of $89.1 \pm 0.8\%$. On ARAS, SRA increases from $81.1 \pm 1.3\%$ to $83.3 \pm 1.0\%$, although the proposed method does not achieve the highest Macro-F1. The SRA gains over the uncertainty-free baseline are statistically significant on both datasets ($p = 0.018$ for CASAS-Aruba and $p = 0.021$ for ARAS). These results support the role of the service-oriented context state in combining recognition confidence, service relevance, device availability, context, and risk for response

**Fig. 11.** Sensitivity of SRA, Ask-F1, and Auto Error to response-threshold variation.

selection. Representation analysis provides additional support. On CASAS-Aruba, the silhouette score increases from 0.29 to 0.46 when activity-only features are replaced by the proposed context state. On ARAS, the score increases from 0.22 to 0.40.

Uncertainty and risk control mainly affect whether ambiguous cases are executed automatically or redirected to confirmation. On CASAS-Aruba, the proposed method increases Ask-F1 from 80.1% to 83.3% and reduces Auto Error from 6.6% to 5.4% compared with Edge Response without Uncertainty. On ARAS, Ask-F1 increases from 75.2% to 78.1%, while Auto Error decreases from 8.8% to 7.3%. Figure 8 further shows that the advantage becomes more evident under medium- and low-confidence conditions. The ablation results are consistent with this interpretation. Removing service risk reduces SRA from 89.0% to 85.8% on CASAS-Aruba and from 83.3% to 79.6% on ARAS. Removing the uncertainty-aware policy reduces SRA to 86.1% and 79.8%, respectively. Removing feedback correction causes smaller decreases to 88.1% and 82.2%. This indicates that feedback acts as an auxiliary correction mechanism rather than the primary source of performance improvement. Figure 11 further shows that moderate threshold variations preserve broadly comparable performance. Overly permissive settings increase Auto Error, whereas overly conservative settings can reduce response consistency.

The practical value of this framework lies in supporting more cautious and context-sensitive edge-side responses rather than simply maximizing activity recognition accuracy. Such a decision structure may be useful for elderly assistance, nighttime lighting, security reminders, and routine smart home services. In these applications, inappropriate automatic execution can be more problematic than an additional confirmation request. In the current experiments, response decisions remain within the specified edge-latency budget, supporting the feasibility of local decision-making without implying real-home deployment.

Several limitations remain. First, CASAS-Aruba and ARAS provide activity labels rather than native service-response records, so SRA measures consistency with rule-mapped labels rather than actual user preference. Second, service risk is currently quantified through rule-based low-, medium-, and high-risk assignments and may not capture household-specific risk. Third, feedback correction uses rule-constructed feedback signals rather than long-term real user interaction. Fourth, multi-resident ambiguity is handled indirectly through confidence and contextual uncertainty; the framework does not perform explicit resident identification. Finally, threshold sensitivity was evaluated only within the tested range around the validation-selected operating point, and parameter size and energy consumption were not systematically examined. Future work should incorporate real service logs, resident-aware modeling, adaptive risk calibration, missing-sensor robustness, and longitudinal preference feedback.

VI. CONCLUSION

The findings supported three main contributions. First, the service-oriented context state provided a more suitable basis for response selection than activity labels alone by incorporating confidence, device availability, context, service relevance, and risk. Second, the uncertainty-aware edge response policy improved response consistency and confirmation quality while reducing inappropriate automatic execution. The proposed method achieved SRA values of $89.0 \pm 0.8\%$ on CASAS-Aruba and $83.3 \pm 1.0\%$ on ARAS. The corresponding Auto Error rates were $5.4 \pm 0.6\%$ and $7.3 \pm 0.7\%$, respectively. Third, low-burden feedback provided an auxiliary

mechanism for adjusting subsequent response parameters. However, the evidence was based on rule-constructed feedback and should not be interpreted as real personalized preference learning.

The practical significance of the framework was reflected in its support for more cautious and context-sensitive edge-side service responses rather than simply maximizing activity recognition accuracy. The auto, ask, and none structure provided a mechanism for balancing automatic execution with user confirmation. The examined applications included elderly assistance, nighttime lighting, security reminders, and routine home services. These findings suggested potential value for improving the traceability and controllability of service decisions under uncertainty, risk, device, and latency constraints.

The study was limited by rule-derived response labels, rule-based service-risk quantification, and the absence of real long-term user feedback. Multi-resident ambiguity was handled indirectly, and explicit resident identification was not addressed. Threshold sensitivity was evaluated only within the tested operating range. Future research should validate the framework using real home service and confirmation logs and develop resident-aware context modeling and adaptive risk calibration. It should also evaluate energy and resource costs on edge devices and investigate personalized response policies based on longitudinal user feedback.

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no conflicts of interest to report regarding the present study.

REFERENCES

- [1] Y. Chen, H. Zhang, and S. Zhong, "Design and implementation of smart home system based on IoT," *Results Eng.*, vol. 24, Article 103410, 2024.
- [2] Y. Lu *et al.*, "Research on designing context-aware interactive experiences for sustainable aging-friendly smart homes," *Electronics*, vol. 13, no. 17, Article 3507, 2024.
- [3] Y. Wang *et al.*, "Investigating the behavioral intention of smart home systems among older people in Linyi City," *Buildings*, vol. 14, no. 10, Article 3145, 2024.
- [4] J. Zhu, D. Wang, and Y. Zhao, "Design of smart home environment based on wireless sensor system and artificial speech recognition," *Meas.: Sens.*, vol. 33, Article 101090, 2024.
- [5] P. Srivatsa and T. Plötz, "Using graphs to perform effective sensor-based human activity recognition in smart homes," *Sensors*, vol. 24, no. 12, Article 3944, 2024.
- [6] M. J. C. S. Reis and C. Serôdio, "Edge AI for real-time anomaly detection in smart homes," *Future Internet*, vol. 17, no. 4, Article 179, 2025.
- [7] H. Xu *et al.*, "How to keep balance between interaction and automation? Toward user overall positive experience of IoT-based smart home design," *Electronics*, vol. 13, no. 7, Article 1375, 2024.
- [8] A. S. Schulz *et al.*, "User perceptions and experiences with smart homes - the smart home as an obedient guard dog, disinterested cat, ambitious octopus or busy beehive," in *Proceedings of Mensch und Computer*, 2024, pp. 171–183.
- [9] R. Kalaria *et al.*, "Adaptive context-aware access control for IoT environments leveraging fog computing," *Int. J. Inf. Secur.*, vol. 23, no. 4, pp. 3089–3107, 2024.

- [10] Y. Sone *et al.*, “Development of a control support system for smart homes using the analysis of user interests based on mixed reality,” *ROBOMECH J.*, vol. 12, no. 1, Article 2, 2025.
- [11] H. Chen *et al.*, “Enhancing human activity recognition in smart homes with self-supervised learning and self-attention,” *Sensors*, vol. 24, no. 3, Article 884, 2024.
- [12] M. Thukral *et al.*, “Layout-agnostic human activity recognition in smart homes through textual descriptions of sensor triggers (TDOST),” *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 9, no. 1, pp. 1–38, 2025.
- [13] T. S. Qureshi *et al.*, “A systematic literature review on human activity recognition using smart devices: advances, challenges, and future directions,” *Artif. Intell. Rev.*, vol. 58, no. 9, Article 276, 2025.
- [14] N. Sedaghati, S. Ardebili and A. Ghaffari, “Application of human activity/action recognition: a review,” *Multimedia Tools Appl.*, vol. 84, no. 28, pp. 33475–33504, 2025.
- [15] M. M. Gad *et al.*, “Personalized smart home automation using machine learning: predicting user activities,” *Sensors*, vol. 25, no. 19, Article 6082, 2025.
- [16] A. Bazán-Muñoz *et al.*, “Taxonomy and software architecture for real-time context-aware collaborative smart environments,” *Internet of Things*, vol. 26, Article 101160, 2024.
- [17] H. Zhou *et al.*, “Efficient human activity recognition on edge devices using DeepConv LSTM architectures,” *Sci. Rep.*, vol. 15, no. 1, Article 13830, 2025.
- [18] F. C. Andriulo *et al.*, “Edge computing and cloud computing for Internet of Things: A review,” *Informatics*, vol. 11, no. 4, Article 71, 2024.
- [19] T. Alam *et al.*, “Internet of Things-based home automation with network mapper and MQTT protocol,” *Comput. Electr. Eng.*, vol. 120, Article 109807, 2024.
- [20] M. Elkhodr, S. Khan and E. Gide, “A novel semantic IoT middleware for secure data management: Blockchain and AI-driven context awareness,” *Future Internet*, vol. 16, no. 1, Article 22, 2024.
- [21] N. Topalli and A. Badii, “A user-centric context-aware framework for real-time optimisation of multimedia data privacy protection, and information retention within multimodal AI systems,” *Sensors*, vol. 25, no. 19, Article 6105, 2025.
- [22] M. N. Halgamuge and D. Niyato, “Adaptive edge security framework for dynamic IoT security policies in diverse environments,” *Comput. Secur.*, vol. 148, Article 104128, 2025.
- [23] Y. Yu *et al.*, “Understanding user experience with smart home products,” *J. Comput. Infor. Syst.*, pp. 1–23, 2024.
- [24] Y. J. Tian *et al.*, “Benefits and barriers associated with the use of smart home health technologies in the care of older persons: a systematic review,” *BMC. Geriatr.*, vol. 24, no. 1, Article 152, 2024.
- [25] C. Zhou, Y. Qian and J. Kaner, “A study on smart home use intention of elderly consumers based on technology acceptance models,” *PLoS One*, vol. 19, no. 3, Article e0300574, 2024.
- [26] J. Luo *et al.*, “Positive strategies for enhancing elderly interaction experience in smart healthcare through optimized design methods: An INPD-based research approach,” *Sustainability*, vol. 16, no. 20, Article 8770, 2024.
- [27] N. A. Felber *et al.*, “The smart home, a true home? How new technologies disrupt the experience of home for older persons,” *Univers. Access Inf. Soc.*, vol. 24, no. 1, pp. 897–910, 2025.
- [28] N. A. Daruwala, “Behavioural intention and smart home technology adoption: the role of technophobia and psychological needs across cultures and genders,” *Int. J. Technol. Hum. Interact. (IJTHI)*, vol. 21, no. 1, pp. 1–24, 2025.
- [29] W. Li *et al.*, “Responsible smart home technology adoption: Exploring public perceptions and key adoption factors,” *Internet of Things*, vol. 32, Article 101622, 2025.
- [30] Y. Liu, R. Tamura, and L. Xiao, “Barriers to older adults adapting smart homes: perceived risk scale development,” *Buildings*, vol. 14, no. 5, Article 1226, 2024.
- [31] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [32] J. D. Lafferty, A. McCallum and F. C. N. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” in *Proceedings of the 18th International Conference on Machine Learning (ICML, 2001)*, pp. 282–289.
- [33] F. J. Ordóñez and D. Roggen, “Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition,” *Sensors*, vol. 16, no. 1, Article 115, 2016.
- [34] S. K. Hiremath and T. Plötz, “Maintenance required: Updating and extending bootstrapped human activity recognition systems for smart homes,” in *2024 International Conference on Activity and Behavior Computing (ABC)*, 2024, pp. 1–13.
- [35] E. Lattanzi, L. Calisti and C. Contoli, “Are transformers a useful tool for tiny devices in human activity recognition?” in *Proceedings of the 2024 8th International Conference on Advances in Artificial Intelligence (ICAAI '24)*, 2024, pp. 339–344, ACM.
- [36] Z. Meng and J. Lu, “A rule-based service customization strategy for smart home context-aware automation,” *IEEE Trans. Mob. Comput.*, vol. 15, no. 3, pp. 558–571, 2016.
- [37] M. Alirezaie *et al.*, “An ontology-based context-aware system for smart homes: E-care@home,” *Sensors*, vol. 17, no. 7, Article 1586, 2017.
- [38] Y. Gal and Z. Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning,” in *International Conference on Machine Learning*, 2016, pp. 1050–1059.