

Article

SRAM-PUF Response Recovery and Key Reconstruction for Intelligent Metering Terminals

Ping Meng ^{1,*}, Qing Wang ² , Xueming Qiao ¹, Pingxin Wang ¹, Xiangkun Zhang ¹ and Junmin Zhou ¹¹ Rural Electricity Department, State Grid Weihai Power Supply Company, Weihai 264200, China² Metrology Center, State Grid Shandong Electric Power Company, Jinan 250001, China

* Correspondence: mengping802@sina.com

Abstract

With the continued development of new-type power systems and intelligent metering infrastructures, field devices such as smart meters, power information acquisition terminals, and fusion terminals have become essential components for energy metering, state sensing, and data interaction. SRAM-based physical unclonable functions (SRAM-PUFs) exploit intrinsic device variations in on-chip SRAM to generate device-specific responses and provide a lightweight hardware basis for terminal identification and key generation. However, variations in operating conditions can cause SRAM response instability and reduce key reconstruction reliability. To address this issue, this paper proposes a SRAM-PUF response recovery and key reconstruction method for intelligent metering terminals. A Bidirectional Temperature-Task Continual Learning (BTCL) mechanism is developed to continuously learn response evolution patterns under different operating conditions. Local consistency constraints and SRAM-cell reliability priors are further incorporated to recover disturbed responses, while spatial interleaving and concatenated error correction are employed to improve key reconstruction stability. Experimental results show that the proposed method reduces the mean pre-ECC bit error rate (BER) to 2.14%, corresponding to reductions of 35.93% and 26.33% compared with the raw SRAM-PUF and B-Spline Continual Learning, respectively. After spatial interleaving and concatenated error correction, the mean post-ECC BER is further reduced to $8.45 \times 10^{-4}\%$, while the mean key error rate (KER) decreases to $2.70 \times 10^{-2}\%$. These results demonstrate that the proposed method effectively improves SRAM-PUF response consistency and key reconstruction reliability under varying operating conditions, providing a reliable hardware-response and key-reconstruction basis for subsequent terminal authentication applications.

Keywords: intelligent metering; SRAM-PUF; continual learning; response recovery; key reconstruction



Academic Editors: Junyu Chen,
Hongjian Lin, Dong Xie,
Xiangyu Meng and Weiyu Wang

Received: 11 September 2026

Revised: 10 October 2026

Accepted: 15 October 2026

Published: 18 October 2026

Copyright: © 2026 by the authors.
Licensee MDPI, Basel, Switzerland.
This article is an open access article
distributed under the terms and
conditions of the [Creative Commons
Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

With the continued development of new-type power systems and the Energy Internet of Things (EIoT) [1], a growing number of field devices, including smart meters, power information acquisition terminals, fusion terminals, and distributed renewable-energy metering terminals, are being connected to the power grid, forming an increasingly extensive intelligent metering infrastructure across the user, distribution, and distributed-energy sides. These terminals undertake essential functions such as energy metering, operating-state sensing, electricity-consumption data acquisition, and data transmission, and the

collected information has become an important basis for electricity billing, line-loss analysis, load management, and power-grid operation and decision-making [2]. With the increasing number of connected terminals and the growing demand for real-time and online metering services, trusted terminal access and secure data interaction have become increasingly important for the reliable operation of intelligent metering systems. However, field metering terminals are geographically distributed, large in number, and often deployed in unattended and complex operating environments. Conventional security mechanisms relying on software-based authentication or statically stored keys are therefore exposed to risks such as key leakage, terminal impersonation, and unauthorized device access [3]. From a broader cyber-physical power-system perspective, malicious attacks may interact with renewable-energy uncertainty and further affect system operation and decision-making. Recent studies have therefore investigated adaptive robust planning and operation strategies against multistage uncertain malicious attacks in power systems with volatile renewable generation [4]. These system-level security and resilience studies further highlight the importance of trustworthy field information and legitimate terminal identities as the foundation of reliable sensing, communication, and operational decision-making. Once terminal trustworthiness is compromised, not only the authenticity of metering-data sources but also the security and reliability of subsequent data acquisition, transmission, and processing may be affected. Therefore, establishing a low-cost, embedded, and reliable authentication mechanism for large-scale intelligent metering terminals is essential for ensuring secure metering-data transmission and reliable data-acquisition services in new-type power systems [5].

Physical unclonable functions (PUFs) exploit random process variations introduced during integrated-circuit fabrication to generate device-specific responses [6]. They enable device authentication and key generation without requiring long-term storage of fixed secret keys [7]. Compared with conventional approaches that permanently store cryptographic keys in nonvolatile memory, PUFs derive security parameters from intrinsic physical characteristics of the device, providing a lightweight security solution for resource-constrained terminals. These characteristics are well suited to smart meters, power information acquisition terminals, and distribution sensing nodes, which are typically deployed on a large scale, geographically distributed, resource-constrained, and operated for long periods without direct supervision [8]. Among different PUF implementations, static random-access memory physical unclonable functions (SRAM-PUFs) can directly exploit the power-up states of on-chip SRAM in microcontrollers to generate device-specific responses without requiring dedicated security chips or additional PUF circuitry, offering advantages in cost, resource consumption, and system integration [9]. Previous studies have demonstrated that SRAM-PUF reliability can be substantially affected by varying operating conditions and long-term device degradation [10]. Environmental variations may cause unstable SRAM cells to change their power-up states, thereby reducing response consistency across different operating conditions. Since SRAM-PUF responses serve as the physical source for device-specific authentication credentials, response instability directly affects whether a legitimate terminal can reproduce its enrolled hardware identity and reconstruct the associated cryptographic key. Excessive response errors may lead to key-reconstruction failure and consequently interrupt subsequent authentication or secure communication, whereas reliable response recovery improves the availability and consistency of the hardware-derived credential. Therefore, enhancing SRAM-PUF reliability under complex operating conditions is a prerequisite for maintaining stable trusted access and reliable key establishment in intelligent metering terminals.

Existing studies on SRAM-PUF reliability under complex operating conditions have mainly focused on reliable-response selection, error recovery, and circuit-level optimization.

Bit-selection approaches based on repeated power-up measurements can eliminate unstable SRAM cells and improve the reliability of responses used for authentication and key generation [11], although such approaches generally require a trade-off between response capacity and reliability. The feasibility of constructing device-specific PUF characteristics across different SRAM implementations and computing platforms has also been investigated [12,13]. Spatial relationships among neighboring SRAM cells have been exploited to improve stable-response extraction [14], while long-term response drift caused by device aging has been incorporated into reliability analysis [15].

For authentication-key recovery, helper-data schemes [16], fuzzy extractors [17], and error-correcting codes such as Reed–Solomon (RS) and Bose–Chaudhuri–Hocquenghem (BCH) codes [18] can improve the recovery of noisy PUF responses through redundancy and error correction. Studies on authentication for field power terminals have further demonstrated the importance of reliable identity information for trusted terminal access [19]. Secure and robust error-correction mechanisms for PUFs have also been investigated to balance response reconstruction capability and implementation overhead [20]. In addition, circuit-level PUF optimization [21], SRAM spatial-characteristic-based reliability enhancement [22], and dedicated subthreshold SRAM-PUF architectures [23] can improve response stability at the device or circuit level. In recent years, data-driven approaches have provided a new direction for improving SRAM-PUF authentication reliability in secure intelligent metering applications. By learning the relationship between environmental conditions and response errors, disturbed SRAM responses can be predicted and corrected without modifying the underlying hardware structure. Pratihari et al. established a machine-learning mapping between operating conditions and local window errors and incorporated cell reliability information for response recalibration [24]; however, the preservation of previously learned knowledge under continuously varying operating conditions and the conversion from window-level prediction to reliable response recovery were not fully addressed. Suragani et al. employed machine-learning techniques to identify disturbed PUF responses [25], but their approach mainly focused on specific disturbance conditions and lacked an explicit mechanism for continual adaptation to environmental changes. Najafi et al. used deep convolutional networks for memory-based PUF response classification and authentication [26], while limited attention was paid to knowledge preservation and stable response recovery under continuously changing operating conditions. Overall, existing data-driven SRAM-PUF reliability enhancement methods are predominantly designed for offline modeling or calibration under predefined operating conditions. Although sequentially observed samples can be incorporated into model updating, temperature variation is generally treated as a single environmental progression, without explicitly accounting for the asymmetric evolution of SRAM response errors on the two sides of the enrollment condition. In practical intelligent metering applications, however, terminals operate continuously over wide temperature ranges, and the response-error distribution may evolve differently during cooling and heating. A continual reliability-enhancement model should therefore not only adapt to newly encountered temperature conditions while retaining previously acquired knowledge, but also capture the direction-dependent characteristics of temperature-induced response drift. Furthermore, window-level error prediction alone does not directly provide the bit-level response required for hardware authentication and subsequent key reconstruction. These characteristics motivate a unified framework that couples direction-aware continual learning with reliable bit-level response recovery and error-tolerant key reconstruction.

To address these challenges, this paper proposes an SRAM-PUF response recovery and key reconstruction framework for intelligent metering terminals under varying operating conditions. The primary methodological innovation is a Bidirectional Temperature-Task

Continual Learning (BTCL) mechanism that models temperature-induced SRAM-PUF response evolution as two ordered continual-learning processes centered on the enrollment temperature. The cooling and heating branches are initialized from the same enrollment model while maintaining independent model parameters and optimizer states, allowing the two response-drift directions to evolve along separate update trajectories. For each newly encountered temperature task, a strict predict-before-update protocol is adopted: the current window-level response errors are first predicted without using the ground-truth information of the target temperature, and the corresponding sample is incorporated into the historical set only after prediction and evaluation. Historical replay and temperature-aware weighting are further integrated to preserve previously acquired response-error knowledge and strengthen adaptation to operating conditions farther from the enrollment temperature. On this basis, a reliability-prior-guided response recovery mechanism converts continuous window-level predictions into bit-level correction decisions by jointly exploiting overlapping-window consistency and SRAM-cell reliability information. Finally, deterministic fixed-stride spatial interleaving and Rep(3,1)-RM(2,5) concatenated error correction are employed to disperse locally correlated residual errors and improve the reliability of authentication-key reconstruction.

The main contributions of this work are summarized as follows:

1. A Bidirectional Temperature-Task Continual Learning (BTCL) mechanism is proposed as the core learning strategy for SRAM-PUF response-error prediction under continuously varying temperature conditions. Different from a conventional single-sequence continual-learning formulation, BTCL explicitly organizes cooling and heating conditions into two ordered task streams centered on the enrollment temperature. The two branches inherit a common enrollment model while maintaining independent parameter-update trajectories. A predict-before-update protocol, historical replay, and temperature-aware weighting are jointly employed to enable continual adaptation while preventing the current target-temperature response errors from participating in their own prediction and reducing the forgetting of previously learned temperature-response knowledge.
2. A reliability-prior-guided soft bit-pattern response recovery mechanism is developed to bridge window-level prediction and bit-level SRAM-PUF reconstruction. Continuous error predictions from overlapping windows and SRAM-cell reliability priors are jointly incorporated into local candidate-pattern evaluation, enabling potential bit-flip locations to be inferred without directly replacing the current response with the enrolled reference response.
3. A reliable authentication-key reconstruction scheme is established by combining deterministic fixed-stride spatial interleaving with Rep(3,1)-RM(2,5) concatenated error correction. The interleaving operation increases the physical separation among replicated bits and redistributes locally correlated residual errors across different coding positions, providing more favorable error patterns for subsequent decoding and improving the stability of key reconstruction.

2. Methods

2.1. Overall Framework of the Proposed Method

To address SRAM-PUF response instability and degraded key reconstruction reliability in smart meters, power information acquisition terminals, and fusion terminals under varying operating conditions, this paper proposes an SRAM-PUF response recovery and key reconstruction method for intelligent metering terminals. The overall procedure is illustrated in Figure 1.

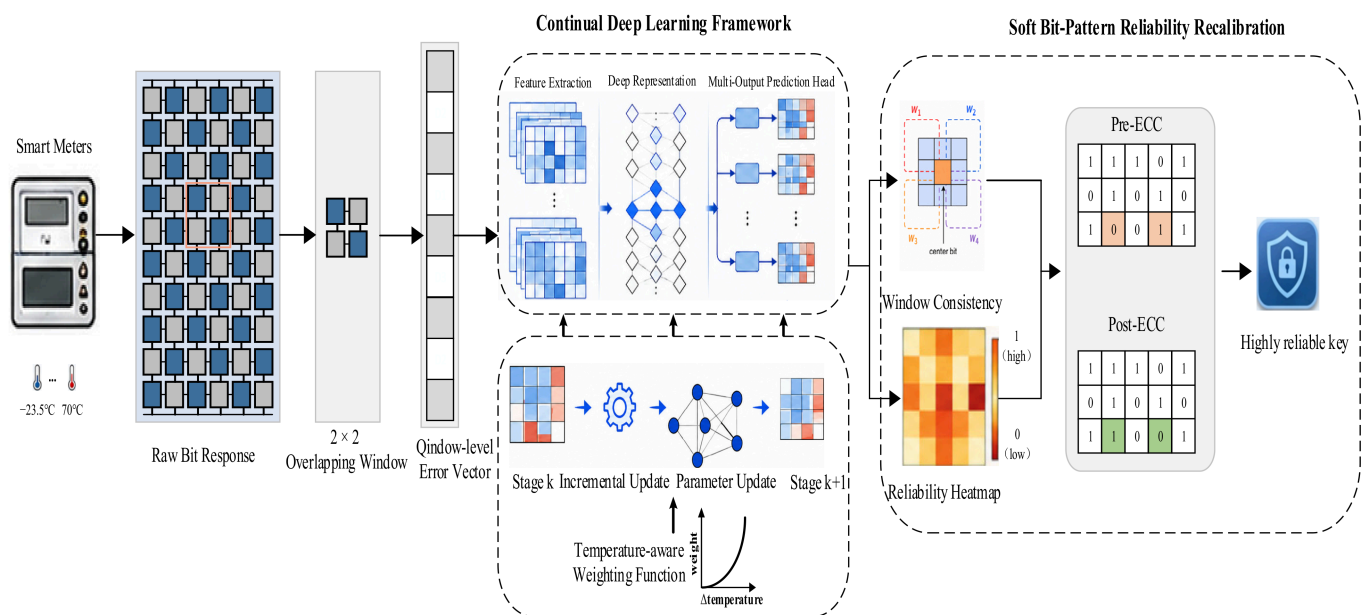


Figure 1. Overall framework of SRAM-PUF response recovery and key reconstruction.

The method takes the SRAM response obtained under the terminal enrollment condition as the authentication reference. First, overlapping sliding windows are employed to characterize local response variations under different temperature conditions. Then, ordered cooling and heating task sequences are constructed with the enrollment temperature as the center, and Bidirectional Temperature-Task Continual Learning (BTCL) is used to continuously learn the evolution of SRAM-PUF response errors under varying operating conditions, thereby improving the model's adaptability to complex environmental changes.

During authentication-information recovery, continuous window-level predictions, local consistency constraints, and SRAM-cell reliability priors are jointly incorporated to recover disturbed responses and improve the consistency of terminal authentication information. On this basis, highly reliable response cells are selected, and fixed-stride spatial interleaving together with Rep(3,1)-RM(2,5) concatenated error correction is applied to further suppress residual errors and achieve stable reconstruction of authentication keys. In the considered application scenario, the enrollment platform and protected reference information are assumed to be trustworthy, while an adversary may attempt to impersonate a legitimate terminal or replay previously transmitted authentication information. The reconstructed device key serves as a hardware-derived credential for subsequent standard challenge–response authentication and session-key establishment.

2.2. SRAM Response Bit-Flip Error Representation

Conventional SRAM-PUF modeling typically relies on single-bit stability analysis, in which individual memory cells are treated as independent random variables. However, in practical implementations, neighboring SRAM cells may exhibit local correlations arising from manufacturing processes, layout structures, and local noise conditions. As a result, independent single-bit modeling may fail to fully characterize such spatial dependencies. To address this issue, a window-level error representation is constructed in this study, transforming the original SRAM power-up responses from bit-level representations into window-level error representations containing local spatial information.

Let the power-up response of the b -th SRAM device at temperature t be denoted by $\mathbf{R}_b(t)$, and let the SRAM response $\mathbf{R}_b(t_0)$ at the enrollment temperature $t_0 = 24^\circ\text{C}$ serve as the enrolled reference response. To characterize the local error distribution induced by temperature variations, the window size should balance the use of neighborhood

information against the complexity of bit-flip localization. A 1×1 window cannot exploit spatial relationships among neighboring cells, whereas a larger window substantially expands the candidate error-pattern space and increases the ambiguity in mapping window-level error counts to specific bit-flip locations. In contrast, a 2×2 window preserves basic local neighborhood information while allowing each interior bit to be jointly constrained by four overlapping windows. Therefore, a 2×2 overlapping sliding window with a stride of 1 is adopted in this study.

For a local window whose upper-left corner is located at position (i, j) , the corresponding window-level error is defined as:

$$\mathbf{e}_b^{(i,j)}(t) = \sum_{u=0}^1 \sum_{v=0}^1 \left[r_b^{(i+u,j+v)}(t) \oplus r_b^{(i+u,j+v)}(t_0) \right] \quad (1)$$

where $\oplus$ denotes the XOR operation; r_b denotes the power-up response bit at the corresponding location of the b -th SRAM device at temperature t ; t_0 denotes the reference temperature; (u, v) denotes the relative coordinates within the window; and $\mathbf{e}_b^{(i,j)}(t)$ denotes the number of flipped bits in the window with respect to the enrolled reference response, taking values from 0 to 4.

After the entire SRAM response array is scanned using overlapping windows, the local window errors are arranged in a fixed spatial order to form the window-level error vector:

$$\mathbf{e}_b(t) = \left[e_b^{(1,1)}(t), e_b^{(1,2)}(t), \dots, e_b^{(H-1,W-1)}(t) \right]^T \quad (2)$$

where H and W denote the number of rows and columns of the SRAM response array, respectively, and the total number of overlapping windows is given by $N_W = (H - 1)(W - 1)$. For the 128×64 SRAM response array used in this study, a total of $127 \times 63 = 8001$ overlapping windows can be generated. Through this transformation, the original bit-level SRAM response is converted into a window-level error representation that incorporates local spatial structural information. Compared with conventional independent single-bit modeling, the proposed representation not only characterizes the degree of response degradation within each local region, but also exploits the overlap among neighboring windows to introduce spatial constraints. These constraints provide useful structural information for subsequent temperature-task learning and for inferring specific bit-flip locations from predicted window-level error counts. The window-level error modeling process is illustrated in Figure 2.

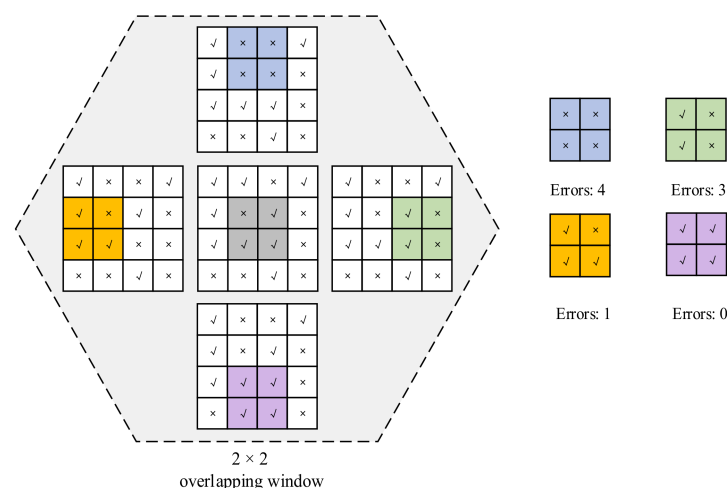


Figure 2. Illustration of SRAM response bit-flip error representation.

2.3. Continual Deep Learning Error Prediction Framework

Based on the window-level error representation, this paper constructs a Bidirectional Temperature-Task Continual Learning (BTCL) framework for window-level error prediction. Errors under different temperature conditions are modeled as correlated ordered tasks, and the model is continuously updated along the cooling and heating directions with the enrollment temperature as the reference point, enabling continual learning of window-error evolution patterns under changing environmental conditions. The overall architecture is illustrated in Figure 3.

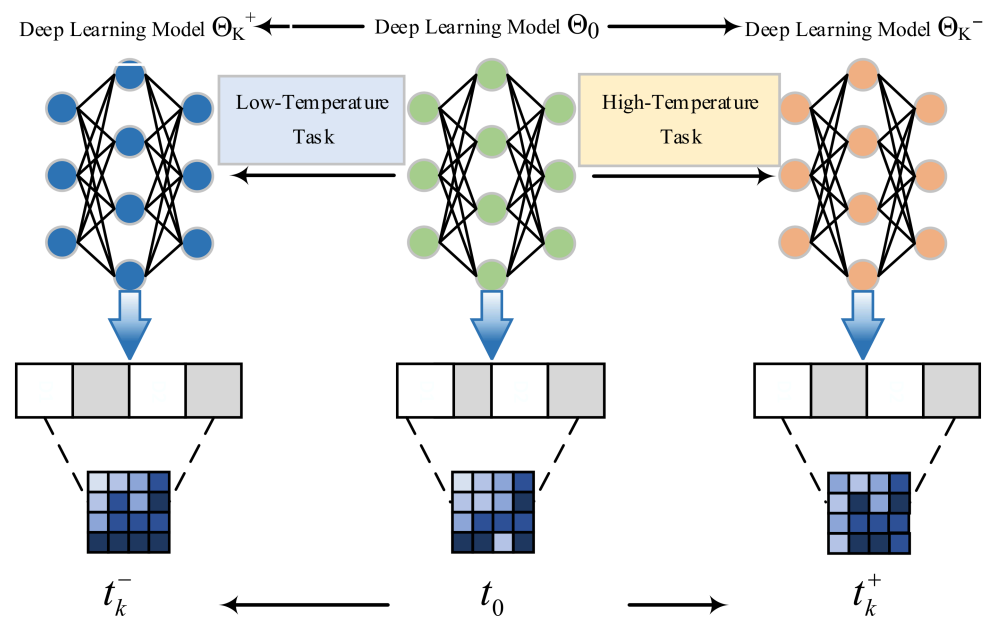


Figure 3. Bidirectional temperature-task continual learning framework for window-level error prediction.

The enrollment temperature $t_0 = 24\text{ }^{\circ}\text{C}$ is selected as the initial task. Considering that SRAM-PUF response errors exhibit different evolution trends on the two sides of the enrollment temperature, the cooling and heating task sequences are constructed in descending and ascending temperature order, respectively:

$$\begin{aligned}\mathcal{T}^- &= \{t_1^-, t_2^-, \dots, t_{K^-}^-\}, & t_0 > t_1^- > t_2^- > \dots > t_{K^-}^- \\ \mathcal{T}^+ &= \{t_1^+, t_2^+, \dots, t_{K^+}^+\}, & t_0 < t_1^+ < t_2^+ < \dots < t_{K^+}^+\end{aligned}\quad (3)$$

Both directions are initialized using the model trained at the enrollment temperature. Independent model parameters and optimizer states are then maintained for the cooling and heating branches, and updated sequentially along their respective temperature-task sequences. BTCL is instantiated independently for each enrolled SRAM device. For the b -th device, a device-specific base model is first established using its response data at the enrollment temperature, from which the cooling and heating branches are initialized and subsequently updated using only the temperature-task samples acquired from the same device. Model parameters and historical replay samples are not shared across different devices.

It should be emphasized that the ordered cooling and heating sequences are used to organize the continual-learning process rather than to assume a strictly monotonic temperature trajectory during practical operation. During field inference, branch selection is determined by the current operating temperature relative to the enrollment temperature t_0 . When $t < t_0$, the cooling branch is used; when $t > t_0$, the heating branch is used; and when the operating temperature returns to t_0 , the enrollment model is adopted.

Therefore, temperature reversals do not require model reinitialization. The corresponding branch retains its previously learned parameters and historical temperature-task information and can be reactivated when the operating condition returns to that side of the enrollment temperature.

For the k -th target temperature, the model first inherits the parameters from the previous task and performs prediction without using the ground-truth window errors at the current temperature. After prediction and performance evaluation, the current temperature sample is added to the historical set and jointly used with previously observed temperature samples for model updating. This process can be expressed as:

$$\hat{\mathbf{e}}_b(t_k) = f_{\theta_{b,k-1}}(\tilde{t}_k) \quad (4)$$

$$\theta_{b,k} = \operatorname{argmin}_{\theta} \sum_{t_i \in \mathcal{H}_k} \mathcal{L}_{\text{TA}}(f_{\theta}(\tilde{t}_i), \mathbf{e}_b(t_i)) \quad (5)$$

where $\hat{\mathbf{e}}_b(t_k)$ denotes the window-level error vector predicted at the target temperature, $\mathcal{H}_k$ denotes the set of historical temperature samples observed up to the current task, and $\mathcal{L}_{\text{TA}}$ denotes the temperature-aware weighted loss function. The “predict-before-update” learning protocol prevents the ground-truth error information at the current target temperature from being used for its own prediction. Parameter inheritance and historical-sample replay are employed to preserve previously acquired temperature knowledge and alleviate forgetting during continual task learning. The continual model-update procedure described above is conducted within a trusted enrollment or recalibration environment, where the enrolled reference response is available for constructing the window-level error targets required for model updating. During field deployment, the terminal performs only SRAM response acquisition, branch selection, and inference using the established model; no ground-truth bit-flip labels or plaintext enrollment reference response are required for local model optimization.

Considering that SRAM-PUF response errors generally increase as the operating temperature deviates from the enrollment condition, a temperature-aware weighting mechanism is further introduced. The contribution of each temperature task to model updating is adjusted according to its deviation from the enrollment temperature. The weight assigned to a temperature sample t is defined as:

$$w(t) = 1 + \alpha \left(\frac{|t - t_0|}{\Delta t_{\max}} \right)^{\gamma} \quad (6)$$

where $\Delta t_{\max}$ denotes the maximum temperature deviation from the enrollment temperature within the experimental temperature range, while α and γ control the weighting magnitude and growth rate, respectively. As the operating temperature moves farther away from the enrollment condition, $w(t)$ gradually increases, causing the model to place greater emphasis on temperature tasks associated with larger response variations during historical replay.

On this basis, the temperature-aware weighted loss is defined as:

$$\mathcal{L}_{\text{TA}} = \frac{w(t)}{N_W} \sum_{n=1}^{N_W} [\hat{e}_{b,n}(t) - e_{b,n}(t)]^2 \quad (7)$$

where N_W denotes the total number of windows, and $\hat{e}_{b,n}(t)$ and $e_{b,n}(t)$ denote the predicted and ground-truth errors of the n -th window, respectively. By assigning larger weights to samples farther from the enrollment temperature, the proposed loss improves the stability of window-level error prediction under complex operating conditions. Combined with bidirectional task partitioning and historical replay, BTCL can preserve previously learned

knowledge while continually adapting to new temperature tasks, thereby providing reliable window-level error estimates for subsequent SRAM-PUF response recovery.

2.4. Reliability-Prior-Guided Response Recovery Mechanism

If a fixed threshold is directly used for response correction, the continuous error information contained in the model predictions may be ignored, and originally stable SRAM cells may be over-corrected. Therefore, based on the window-level bit-flip error predictions, this paper incorporates SRAM-cell reliability priors and constructs a bit-level response recovery mechanism to achieve refined correction of hardware identity responses under wide-temperature conditions.

Let the model output a continuous window-level error prediction matrix at temperature t . For any center-bit position, a 3×3 local neighborhood is extracted, and all possible bit-flip patterns within this neighborhood are enumerated as:

$$\mathbf{z}^{(p)} = [z_1^{(p)}, z_2^{(p)}, \dots, z_9^{(p)}]^T, p = 1, 2, \dots, 512 \quad (8)$$

where $z_l^{(p)} = 1$ indicates that the l -th bit is identified as a potential flipped bit, where $z_l^{(p)} = 0$ indicates that the bit remains stable. Since the local neighborhood contains nine bits, there are $2^9 = 512$ possible candidate bit patterns. Each candidate bit pattern can be mapped to the number of flipped bits in the four overlapping 2×2 windows, denoted as:

$$\mathbf{c}^{(p)} = [c_1^{(p)}, c_2^{(p)}, c_3^{(p)}, c_4^{(p)}]^T \quad (9)$$

where $c_m^{(p)}$ denotes the number of flipped bits in the m -th local window under the p -th candidate bit pattern, with $m = 1, 2, 3, 4$. Meanwhile, the corresponding prior probability of bit flipping is constructed according to the reliability of each SRAM cell:

$$q_l = 1 - \rho_l \quad (10)$$

where ρ_l denotes the reliability of the l -th SRAM cell under the reference condition, and q_l represents the prior probability that a bit flip occurs at the corresponding location. A lower cell reliability corresponds to a higher prior probability of that bit being identified as a potential flipped bit.

On this basis, the local response recovery problem is formulated as the following soft bit-pattern reliability energy minimization problem:

$$E(\mathbf{z}^{(p)}) = \lambda_d \sum_{m=1}^4 (c_m^{(p)} - \hat{e}_m)^2 - \lambda_r \sum_{l=1}^9 [z_l^{(p)} \ln q_l + (1 - z_l^{(p)}) \ln(1 - q_l)] \quad (11)$$

$$\mathbf{z}^* = \underset{\mathbf{z}^{(p)}}{\operatorname{argmin}} E(\mathbf{z}^{(p)}) \quad (12)$$

where $\hat{e}_m$ denotes the continuous error prediction for the m -th local window output by the model, and $c_m^{(p)}$ denotes the number of flipped bits in the m -th window corresponding to the p -th candidate bit pattern. λ_d and λ_r are weighting coefficients for the window-error consistency term and the reliability-prior term, respectively. The first term constrains the window-level flip counts implied by the candidate bit pattern to remain consistent with the model predictions, whereas the second term incorporates SRAM-cell reliability priors, making low-reliability bits more likely to be identified as potential flipped bits.

From a probabilistic perspective, the above energy formulation can be interpreted as a maximum-a-posteriori (MAP) estimation of the local bit-flip pattern. Specifically, the

window-error consistency term reflects the likelihood that a candidate pattern produces the continuous window-level predictions, whereas the reliability term represents the prior probability of the corresponding bit-flip configuration derived from the enrolled SRAM-cell reliability. Consequently, minimizing the energy in (11) is equivalent to selecting the candidate pattern that achieves the best joint agreement with the model prediction and the cell-reliability prior. For the true local pattern p^* , correct identification is achieved when $E(p^*) < E(p), \forall p \neq p^*$. The ratio between λ_d and λ_r controls the relative confidence assigned to the model prediction and the reliability prior. A small λ_r makes the decision primarily dependent on the predicted window-error structure, whereas an excessively large λ_r may cause the prior information to dominate the candidate-pattern selection. Therefore, a moderate reliability-prior weight provides a balance between current response evidence and the enrolled cell-reliability information.

To avoid unnecessary bit-pattern searches when the predicted window errors are close to zero, a prediction gating threshold g is introduced. When the predictions of the four local windows satisfy $\sum_{m=1}^4 \hat{e}_m \geq g$, the energy values of all 512 candidate bit patterns are evaluated; otherwise, the current central bit is considered stable and no response correction is performed. After the optimal bit pattern is determined, the SRAM response at the current temperature is recovered according to the state of the central bit in the optimal pattern as follows:

$$\tilde{R}_b(i, j, t) = R_b(i, j, t) \oplus z_c^* \quad (13)$$

where $R_b(i, j, t)$ denotes the original SRAM response at the current temperature, and z_c^* denotes the flip state of the central bit in the optimal bit pattern. When $z_c^* = 1$, the current response bit is flipped; when $z_c^* = 0$, the original response bit is retained. Notably, this procedure does not directly replace the predicted position with the enrolled reference response. Instead, the error-correction decision is determined entirely by the window-level error predictions and the SRAM-cell reliability prior.

For boundary bits that cannot form a complete 3×3 neighborhood, the original response is retained without modification. Each interior bit is evaluated only once as the center of a local neighborhood, thereby avoiding repeated correction of the same bit. In this way, the proposed method further converts window-level error predictions into bit-level response recovery decisions, reducing the information loss and overcorrection associated with simple threshold-based replacement. The overall framework is illustrated in Figure 4.

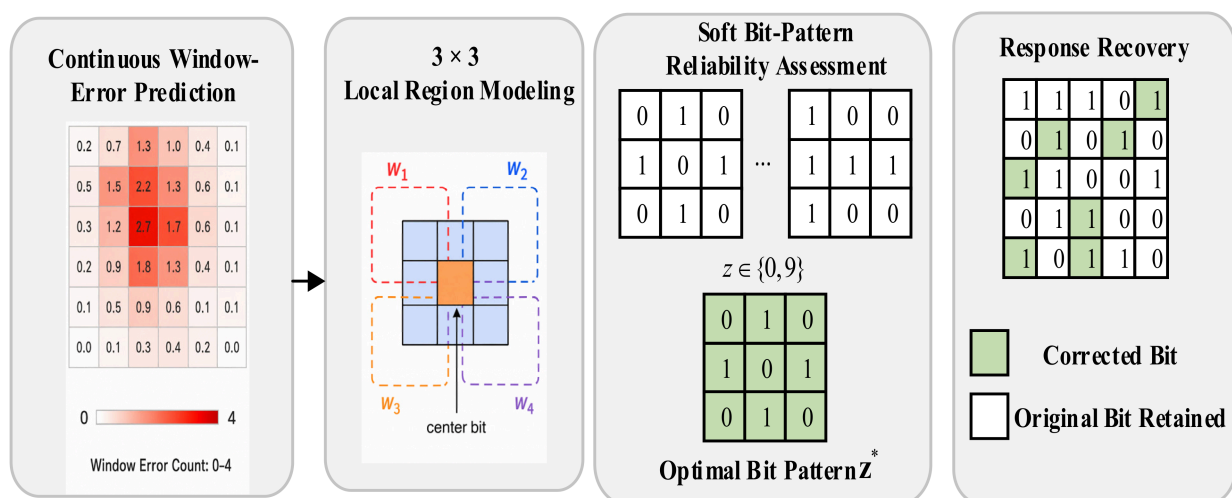


Figure 4. Illustration of reliability-prior-guided response recovery.

2.5. Fixed-Stride Spatial Interleaving and Concatenated Error Correction

After soft bit-pattern recalibration, the number of erroneous bits in the response is reduced; however, a small number of residual errors may still exhibit local spatial clustering. If multiple replicas used in repetition coding are selected from physically adjacent SRAM cells, local temperature disturbances may simultaneously affect several replicas, thereby weakening the error-correction capability of majority voting. To address this issue, a fixed-stride spatial interleaving scheme is introduced between reliable-bit selection and concatenated error correction. This scheme remaps adjacent reliable bits to different coding positions, thereby reducing the spatial correlation of residual errors.

Let the response of the b -th device at temperature t after soft bit-pattern recalibration be denoted by $\tilde{R}_b(t)$, and let the reliability of the SRAM cells be denoted by $\rho_b(i, j)$. Based on a reliability threshold τ , the reliable-bit set is constructed as follows:

$$\mathcal{M}_b = \{(i, j) \mid \rho_b(i, j) > \tau\} \quad (14)$$

The response bits in set $\mathcal{M}_b$ are extracted according to a fixed spatial order to obtain the reliable-bit sequence $r_b(t)$. To ensure a consistent block length for subsequent coding, the sequence length is truncated to an integer multiple of 768:

$$N_b = 768 \left\lfloor \frac{|\mathcal{M}_b|}{768} \right\rfloor \quad (15)$$

where N_b denotes the actual number of reliable bits from the b -th device participating in key recovery. Based on this sequence, a fixed-stride interleaving mapping is constructed. Let the interleaving stride be the smallest integer that is no less than $N_b/3$ the specified stride threshold and is coprime with N_b :

$$s_b = \min \left\{ s \geq \left\lfloor \frac{N_b}{3} \right\rfloor \mid \gcd(s, N_b) = 1 \right\} \quad (16)$$

The corresponding interleaved position mapping is defined as:

$$\pi_b(n) = (o_b + ns_b) \bmod N_b, n = 0, 1, \dots, N_b - 1 \quad (17)$$

where o_b denotes the fixed starting offset corresponding to the b -th device. Since s_b and N_b are coprime, the mapping $\pi_b(\cdot)$ constitutes a one-to-one permutation, ensuring that no reliable bits are lost or duplicated. The interleaved response sequence is expressed as:

$$\bar{r}_b(n, t) = r_b(\pi_b(n), t) \quad (18)$$

The same interleaving mapping is employed during both the enrollment and key reconstruction stages. The interleaving operation does not change the total number of erroneous bits in the response; instead, it redistributes the error locations across coding blocks and replicated codeword positions. In this manner, the three replicas in the Rep(3,1) code are selected, as far as possible, from SRAM cells that are physically separated, thereby reducing the probability that locally correlated errors simultaneously corrupt multiple replicas.

After interleaving, the response sequence is partitioned into coding blocks of length 768. For each 128-bit key block, the key is divided into eight 16-bit message groups. Each group is first encoded using an RM(2,5) code to generate a 32-bit codeword, and each bit of the resulting codeword is then repeated three times using Rep(3,1), producing a 96-bit

concatenated codeword. The eight encoded groups are subsequently concatenated to form a complete 768-bit codeword:

$$\mathbf{c}_{b,q} = \text{Rep}_3 \left[\text{RM}_{(2,5)} \left(\mathbf{k}_{b,q} \right) \right] \in \{0,1\}^{768} \quad (19)$$

where $\mathbf{k}_{b,q}$ denotes the q -th enrolled key block of the b -th device, and $\mathbf{c}_{b,q}$ denotes the corresponding concatenated coding result.

During enrollment, the interleaved response block $\bar{\mathbf{r}}_{b,q}(t_0)$ obtained at the enrollment temperature is XORed with the concatenated codeword to generate the helper data. During key reconstruction at temperature t , the current interleaved response block is XORed with the corresponding helper data:

$$\mathbf{h}_{b,q} = \bar{\mathbf{r}}_{b,q}(t_0) \oplus \mathbf{c}_{b,q} \quad (20)$$

$$\mathbf{y}_{b,q}(t) = \bar{\mathbf{r}}_{b,q}(t) \oplus \mathbf{h}_{b,q} \quad (21)$$

Subsequently, soft information is constructed using the cell reliability obtained during enrollment. The three repeated replicas are combined through reliability-weighted fusion, and the resulting soft decisions are then fed into the RM(2,5) decoder to recover the key:

$$\hat{\mathbf{k}}_{b,q}(t) = \text{Dec}_{\text{RM}} \left[\text{Dec}_{\text{Rep}} \left(\mathbf{y}_{b,q}(t) \right) \right] \quad (22)$$

3. Results

3.1. Experimental Platform and Data Acquisition

To verify the effectiveness of the proposed method in achieving stable recovery of authentication information and reliable key reconstruction for intelligent metering terminals under complex temperature conditions, Arduino UNO boards were selected as the experimental platform to collect SRAM power-up responses from multiple devices at different temperatures. Ten boards of the same model were used, and 1024 bytes of on-chip SRAM on each board were selected as the PUF response region, corresponding to 8192 bits arranged as a 128×64 two-dimensional array. The collected data include SRAM power-up responses under different temperature conditions, cell reliability information under the enrollment condition, and window-level error statistics, which are used for BTCL error prediction, response recovery, and authentication-key reconstruction experiments.

A total of 38 discrete temperature points were configured over the range of -23.5 to 69 °C, with 24 °C selected as the enrollment temperature. SRAM responses at all other temperatures were compared with the enrollment reference response. A 2×2 overlapping sliding window with a stride of 1 was used to calculate local response errors, producing 8001 window-level error values for each temperature sample as the prediction targets of the BTCL model. To analyze authentication-response stability across different operating temperature regions, the experimental temperature range was divided into a low-temperature region (below 0 °C), a medium-temperature region (0 to 45 °C), and a high-temperature region (above 45 °C). The enrollment-temperature point was excluded from the calculation of the full-range mean BER, standard deviation, and worst-case BER.

3.2. Experimental Setting and Evaluation Metrics

The experiments were implemented in Python 3.11 using PyCharm 2023 as the integrated development environment. The model was trained using the Adam optimizer with an initial learning rate of 1×10^{-3} , taking temperature as the input and window-level errors as the output. The model was first trained for 100 epochs using the enrollment-temperature samples. Continual learning was then performed separately along the cooling and heating

directions, with historical replay conducted for 20 epochs after each new temperature task. This training protocol ensures that the ground-truth response information at the target temperature is not used for its own prediction, while enabling the model to retain the error-distribution characteristics learned from previously observed temperature tasks during adaptation to new temperature conditions. Key reconstruction was performed using fixed-stride spatial interleaving and Rep(3,1)-RM(2,5) concatenated error correction.

The proposed method was evaluated from multiple perspectives, including window-level error prediction, erroneous-bit localization, response recalibration, historical knowledge retention, and key reconstruction. The main evaluation metrics include window-level prediction accuracy, bit-level Precision, Recall, and F1-score, the Bit Error Rate (BER) before error correction, average forgetting, BER after error correction, and empirical key error rate.

3.3. Experimental Results and Performance Analysis

To evaluate the capability of the proposed method to stably reconstruct SRAM-based hardware identities for field metering and data acquisition terminals under wide-temperature operating conditions, experiments were conducted from the perspectives of response bit errors, window-level error prediction, erroneous-bit localization, continual-learning stability, and key reconstruction. For a fair comparison, the Raw PUF baseline is evaluated using the same reliability-based bit-selection rule adopted in the key-reconstruction stage, where SRAM cells are retained according to the enrollment reliability threshold, but no learned response-recovery mechanism is applied. Therefore, the Raw PUF baseline represents the conventional reliability-screening strategy.

Figure 5 compares the pre-error-correction BERs of different methods over the temperature range from -23.5°C to 69°C . As the operating temperature deviates from the enrollment temperature of 24°C , the BER of the original SRAM-PUF gradually increases, with a more pronounced degradation in the low-temperature region. Owing to device-to-device variations introduced by manufacturing processes, B-spline transfer learning has difficulty transferring error patterns across different devices. Although B-spline continual learning reduces the BER to some extent, relatively large residual errors remain under extreme-temperature conditions.

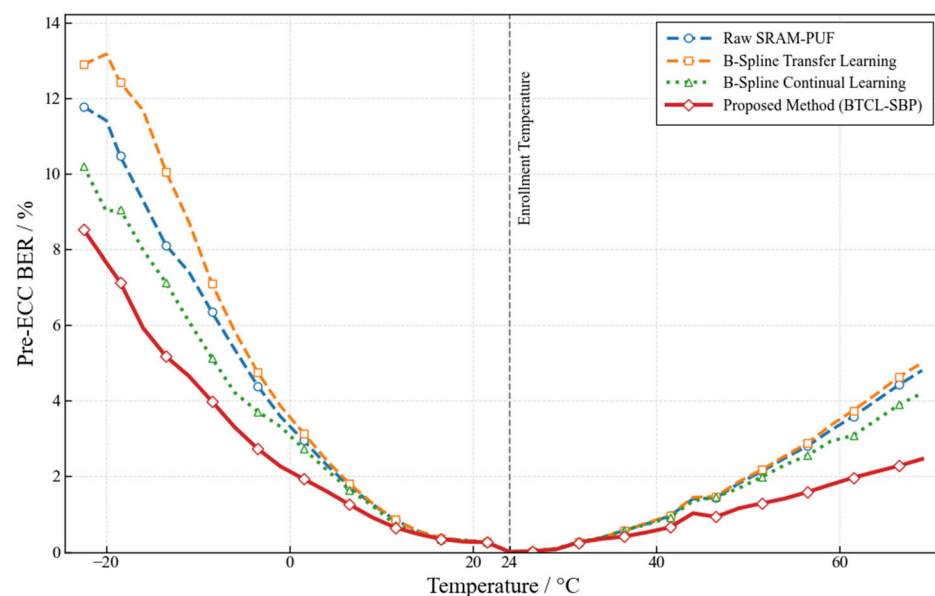


Figure 5. Pre-ECC bit error rates of different methods at different temperatures.

In comparison, the proposed method achieves lower BERs at most non-enrollment temperature points, demonstrating that the combination of BTCL and soft bit-pattern

recalibration can effectively improve response stability over a wide temperature range. From the perspective of field operation, the reduction in pre-error-correction BER indicates higher consistency between SRAM-based hardware identity responses obtained under different temperature conditions and the enrolled reference response, while also reducing the input error level that must be handled during subsequent key reconstruction.

Table 1 summarizes the mean BER and standard deviation of different methods across individual temperature regions and over the entire temperature range. The overall mean BER reflects the response consistency of each method across the full test range, the inter-temperature standard deviation characterizes the variation in recalibration performance with operating temperature, and the worst-case BER reflects the response recovery capability under adverse temperature conditions. The results show that the proposed BTCL method achieves the lowest BER in the low-, medium-, and high-temperature regions, with values of 5.13%, 0.65%, and 1.69%, respectively. Over the full temperature range, the mean BER is reduced from 3.35% for the original SRAM-PUF to 2.14%, outperforming both the transfer-learning and continual-learning methods. In addition, the inter-temperature standard deviation is reduced to 2.19%, representing a 33.45% decrease compared with the original SRAM-PUF. These results indicate that the proposed method can effectively suppress response errors under wide-temperature conditions and reduce the input error level for subsequent error correction and key reconstruction. Taken together, the three metrics demonstrate that the proposed method not only lowers the average error level but also improves response stability throughout wide-temperature operation.

Table 1. Comparison of Pre-ECC BERs (%) of Different Methods Across Temperature Ranges.

Method	Low-Temperature Region	Mid-Temperature Region	High-Temperature Region	Full-Range Mean	Standard Deviation
Raw PUF	7.81	0.88	3.07	3.35	3.29
Transfer Learning	9.05	0.91	3.18	3.72	3.87
Continual Learning	6.58	0.83	2.75	2.91	2.74
BTCL	5.13	0.65	1.69	2.14	2.19

To further assess the robustness of the performance comparison across different physical devices, the full-range mean pre-ECC BER was calculated independently for each of the ten SRAM-PUF boards, and the device-level statistical results are summarized in Table 2. The proposed method achieved a device-level mean BER of 2.14% with a standard deviation of 0.44% and a 95% confidence interval of 1.82–2.45%, compared with 3.33% [2.92%, 3.75%] for the Raw PUF, 3.71% [3.30%, 4.12%] for B-Spline Transfer Learning, and 2.90% [2.55%, 3.25%] for B-Spline Continual Learning. These results indicate that the performance advantage of the proposed method is consistently observed across the ten independently evaluated SRAM-PUF devices.

Table 2. Device-level statistical analysis of pre-ECC BER across ten SRAM-PUF devices.

Method	Mean Pre-ECC BER (%)	Device-Level SD (%)	95% CI (%)
Raw PUF	3.33	0.58	[2.92, 3.75]
B-Spline TL	3.71	0.58	[3.30, 4.12]
B-Spline CL	2.90	0.48	[2.55, 3.25]
BTCL + SBP	2.14	0.44	[1.82, 2.45]

To further investigate the capability of different methods to model local error distributions under temperature disturbances, Figure 6 compares the window-level error prediction

accuracy of B-spline transfer learning, B-spline continual learning, and the proposed BTCL method at different temperature points.

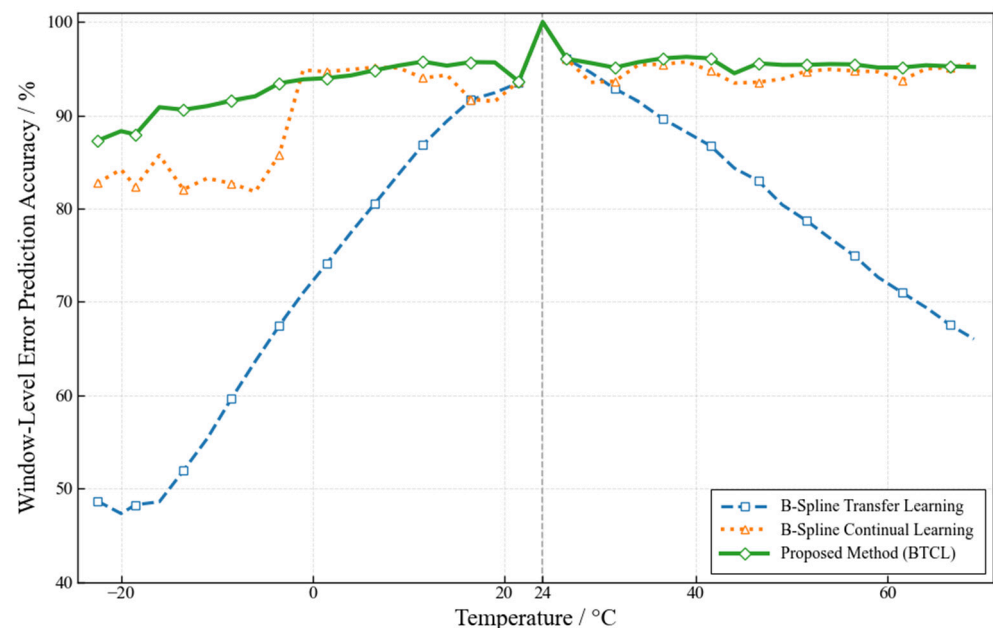


Figure 6. Exact window-error prediction accuracy.

Affected by cross-device differences in error distributions, the prediction accuracy of B-spline transfer learning decreases markedly as the operating temperature deviates from the enrollment temperature of 24 °C, with an average accuracy of only 75.62% over the full temperature range. By incorporating historical temperature samples from the target device, B-spline continual learning improves the average prediction accuracy to 91.76%, although certain fluctuations remain in the low-temperature region.

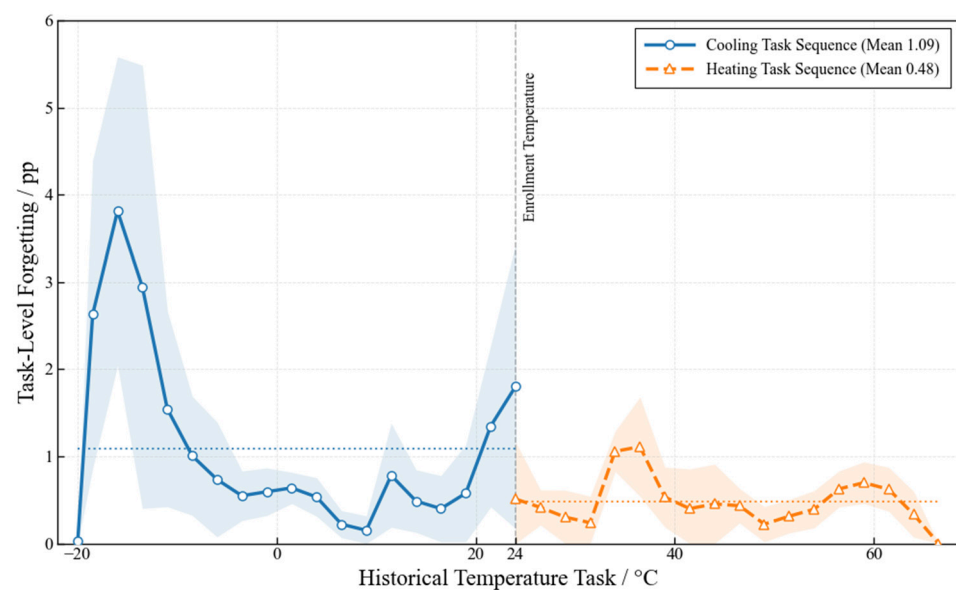
In comparison, the proposed BTCL method achieves higher prediction accuracy at most non-enrollment temperature points, with an average accuracy of 94.07% over the full temperature range. This represents improvements of 18.45 and 2.31 percentage points over B-spline transfer learning and B-spline continual learning, respectively. The proposed method also exhibits better stability under extreme-temperature conditions. These improvements are primarily attributed to bidirectional temperature-task modeling, historical sample replay, and the temperature-aware weighted loss, which enable the model to more accurately capture the evolution patterns of window-level errors.

To further evaluate the effectiveness of converting window-level predictions into bit-level recalibration decisions, Precision, Recall, and F1-score are employed to assess erroneous-bit localization performance, as summarized in Table 3. Since the original SRAM-PUF does not provide erroneous-bit prediction capability, it is not included in this comparison. B-spline continual learning achieves a Precision of 74.86%, but its Recall is only 19.60%, indicating a relatively conservative prediction strategy. Although it can accurately identify a subset of erroneous bits, a large number of actual bit-flip locations remain undetected. In comparison, the proposed method achieves a Precision, Recall, and F1-score of 79.57%, 65.23%, and 71.69%, respectively, corresponding to improvements of 4.71, 45.63, and 40.62 percentage points over B-spline continual learning.

Table 3. Comparison of erroneous-bit localization performance among different methods.

Method	Precision (%)	Recall (%)	F1 (%)
B-Spline Transfer Learning	10.90	0.93	1.71
B-Spline Continual Learning	74.86	19.60	31.07
BTCL	79.57	65.23	71.69

To further evaluate the capability of BTCL to retain previously acquired knowledge during sequential temperature-task learning, Figure 7 presents the forgetting values of historical temperature tasks along both the cooling and heating task sequences. Task forgetting is defined as the degradation in the final window-level prediction performance of a historical task, measured after completion of all temperature-task updates, relative to the best performance achieved for that task during the learning process. A smaller forgetting value indicates stronger retention of the knowledge associated with that temperature task.

**Figure 7.** Task forgetting under bidirectional temperature-task sequences. The horizontal dashed lines indicate the mean forgetting values of the corresponding task sequences.

As shown in Figure 7, the task-level forgetting of BTCL remains consistently low in both temperature directions and does not exhibit a cumulative increase as the number of tasks grows. The average forgetting values for the cooling and heating task sequences are 1.09 and 0.48 percentage points, respectively. These results indicate that parameter inheritance and historical sample replay enable the model to adapt to new temperature tasks while effectively preserving the learned characteristics of window-level error distributions from previous tasks. Compared with the heating direction, the cooling sequence exhibits slightly higher average forgetting, suggesting that window-level error distributions are more complex under low-temperature conditions and that the differences between adjacent tasks are more pronounced.

These results demonstrate that BTCL does not simply perform sequential fitting to the current temperature samples. Instead, through bidirectional task partitioning and historical sample replay, it continuously learns new temperature tasks while retaining knowledge acquired from previously observed temperatures. This mechanism provides effective support for maintaining stable window-level prediction performance over a wide temperature range.

To verify whether the improvement in response recalibration can be further translated into reliable key reconstruction, the mean pre-error-correction BER, mean post-error-correction BER, and empirical key error rate of different methods are compared under the same configuration of reliable-bit selection, fixed-stride spatial interleaving, and Rep(3,1)-RM(2,5) concatenated error correction, as shown in Figure 8. Because the post-error-correction BER and empirical key error rate differ substantially in magnitude, logarithmic scales are used for the vertical axes in Figure 8b,c.

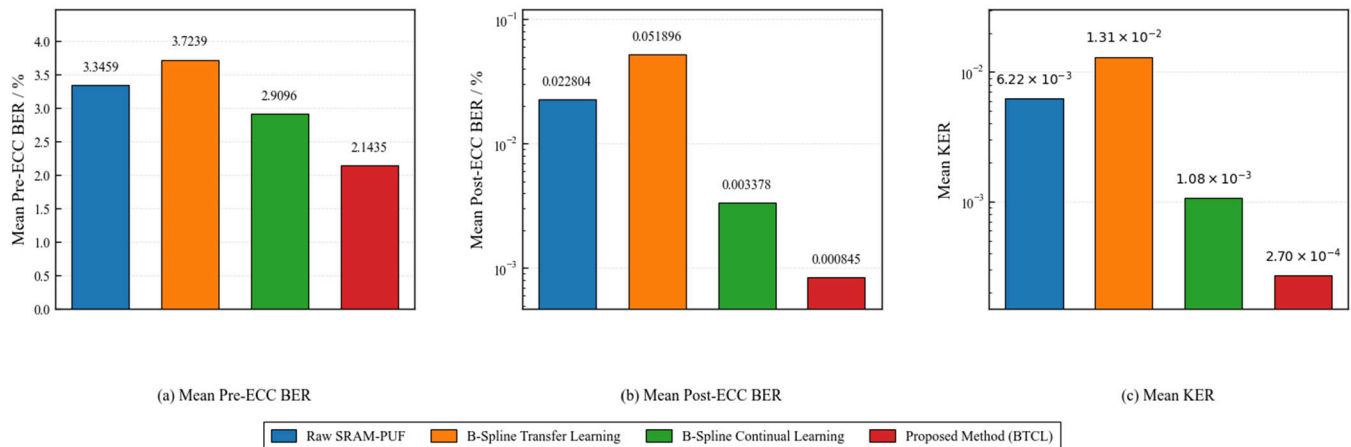


Figure 8. Comparison of average error-correction and key-reconstruction performance of different methods.

Figure 8 compares the performance of different methods from three perspectives: response recovery, error-correction decoding, and key reconstruction. As shown in Figure 8a, the proposed method achieves a mean pre-ECC BER of 2.14%, corresponding to reductions of 35.93% and 26.33% compared with the raw SRAM-PUF and B-Spline Continual Learning, respectively. This result indicates that BTCL can effectively learn the evolution of response errors under different temperature conditions, while the soft bit-pattern reliability recalibration further improves the recovery of disturbed responses. After fixed-stride spatial interleaving and concatenated error correction, the mean post-ECC BER is further reduced to $8.45 \times 10^{-4}\%$, as shown in Figure 8b, representing reductions of 96.29% and 74.99% relative to the raw SRAM-PUF and B-Spline Continual Learning, respectively. This demonstrates that effective suppression of response errors at the preceding stage can provide more reliable inputs for subsequent error-correction decoding. As shown in Figure 8c, the proposed method further reduces the mean key error rate to $2.70 \times 10^{-2}\%$, corresponding to decreases of approximately 95.65% and 75.00% compared with the above two methods, respectively, further verifying its effectiveness in stable authentication-key reconstruction. In contrast, B-Spline Transfer Learning does not achieve improvement in any of the three metrics, indicating that environmental response characteristics vary significantly across devices and that relying solely on cross-device transfer cannot ensure stable authentication and key recovery performance.

Overall, the proposed method enables the improvement in SRAM-PUF response reliability under complex operating conditions to be further propagated to the authentication-key reconstruction stage. The lower pre-ECC BER reduces the input error level of the concatenated error-correction process, while residual errors are further suppressed after spatial interleaving and decoding. Therefore, the proposed method can effectively improve SRAM-PUF response consistency and key reconstruction reliability for intelligent metering terminals, providing a reliable hardware-response basis for subsequent authentication and key-establishment procedures.

4. Discussions

To verify the necessity of modeling the heating and cooling processes using independent task sequences, the average BER of the original SRAM-PUF is calculated at different temperature deviations $|\Delta T|$ with the enrollment temperature of 24 °C as the reference, as shown in Figure 9. As the temperature gradually deviates from the enrollment temperature, the BERs in both the heating and cooling directions exhibit an overall increasing trend; however, their rates of increase differ markedly. Within a small temperature-deviation range, the BERs in both directions remain relatively low. As $|\Delta T|$ increases, the BER in the cooling direction rises more rapidly and gradually diverges from that in the heating direction. When the temperature deviation approaches 45 °C, the BER in the cooling direction is close to 12.00%, whereas that in the heating direction is approximately 5.00%, indicating that response degradation is substantially more pronounced on the low-temperature side under the same temperature deviation.

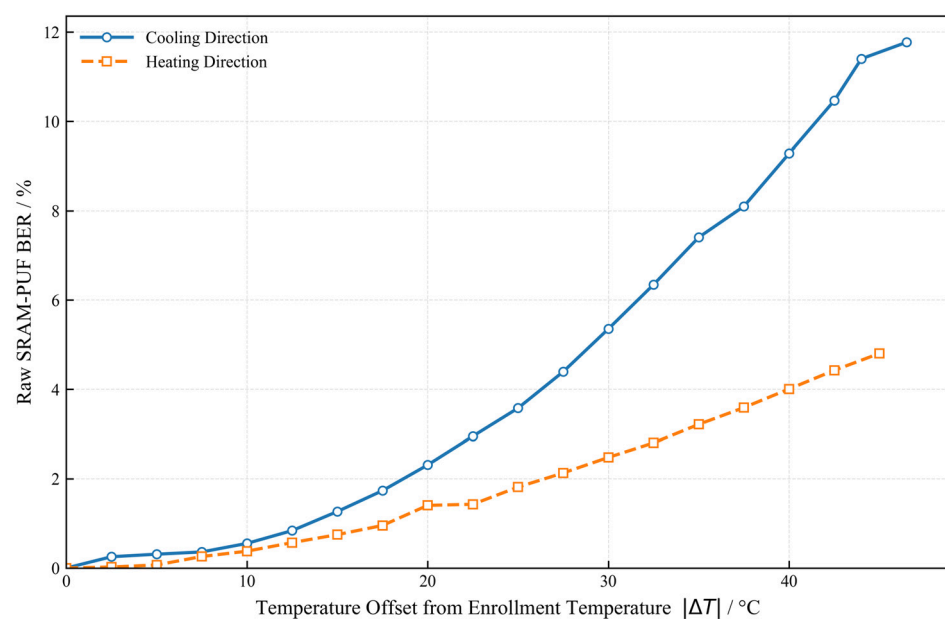


Figure 9. Error evolution trends of raw SRAM-PUF responses on both sides of the enrollment temperature.

These results demonstrate that SRAM-PUF errors are not determined solely by the absolute distance from the enrollment temperature, but also exhibit distinctly different evolution characteristics on the two sides of the enrollment point. If the heating and cooling tasks were incorporated into a single continual-learning sequence, the model would need to fit two different error-evolution patterns simultaneously, potentially increasing task interference during parameter updates. Therefore, with the enrollment temperature as the central reference, the heating and cooling processes are divided into two independent ordered temperature-task sequences and learned separately, providing empirical support for the proposed bidirectional temperature-task modeling strategy.

The soft bit-pattern energy function consists of a window-error consistency term and a reliability-prior term, whose relative contributions are controlled by λ_d and λ_r , respectively. Since the candidate bit pattern is determined by minimizing the energy function, dividing the entire objective by λ_d does not change the optimal solution when $\lambda_d > 0$. Therefore, the actual decision of the model is primarily governed by the ratio between the two weighting coefficients. Based on this observation, λ_d is fixed at 1 as the normalization reference, and only the sensitivity to the reliability-prior weight λ_r is analyzed in this study.

As shown in Table 4, when $\lambda_r = 0.25$, the average BER over the full temperature range and the worst-case BER are reduced to 2.14% and 8.52%, respectively, outperforming both the setting without the reliability prior and those with larger prior weights. This indicates that appropriately incorporating SRAM-cell reliability information can improve the discrimination of candidate bit patterns. As λ_r is further increased, an excessively strong prior constraint makes the bit-flip decision more conservative, causing some actual errors to remain uncorrected. Therefore, $\lambda_d = 1$ and $\lambda_r = 0.25$ are adopted in the subsequent experiments.

Table 4. Sensitivity analysis of the reliability-prior weight λ_r .

λ_r	Mean BER (%)	Standard Deviation (%)	Worst-Case BER (%)
0	2.38	2.32	8.67
0.10	2.21	2.25	8.56
0.25	2.14	2.19	8.52
0.50	2.62	2.55	9.19
1.00	2.97	3.03	10.90

This behavior is also consistent with the MAP interpretation of the recovery objective: when the reliability prior becomes overly dominant, mismatch between the enrolled reliability information and the current response condition can bias the candidate-pattern decision, whereas a moderate prior weight preserves the contribution of the current window-level prediction.

As shown in Table 5, the calibration performance over the full temperature range improves progressively as the individual functional modules are introduced. Taking B-spline continual learning as the baseline, with an average BER of 2.91% and a worst-case BER of 10.20%, bidirectional temperature-task partitioning enables the low- and high-temperature directions to be learned separately, thereby reducing parameter interference between different drift directions. Historical task replay alleviates catastrophic forgetting across sequential temperature tasks, while temperature-aware weighting assigns larger weights to more difficult tasks farther from the enrollment temperature, enhancing erroneous-bit localization under extreme-temperature conditions. After introducing SBP recalibration, the complete model achieves an average BER, inter-temperature standard deviation, and worst-case BER of 2.14%, 2.19%, and 8.52%, respectively. These results demonstrate that local bit-pattern constraints can effectively transform window-level error predictions into bit-level response corrections.

Table 5. Ablation results of different components.

Model	Bidirectional Task Split	Historical Replay	Temperature-Aware Weighting	SBP Recalibration	Mean BER (%)	Standard Deviation (%)	Worst-Case BER (%)
CL	×	×	×	×	2.91	2.74	10.20
Model A	✓	×	×	×	2.73	2.61	9.82
Model B	✓	✓	×	×	2.51	2.43	9.27
Model C	✓	✓	✓	×	2.32	2.29	8.91
BTCL	✓	✓	✓	✓	2.14	2.19	8.52

To further evaluate the contribution of the key-reconstruction stage, the effects of spatial interleaving and Rep(3,1)-RM(2,5) concatenated error correction are analyzed in Table 6. Without error correction, BTCL + SBP achieves a pre-ECC BER of 2.14%. Applying Rep(3,1)-RM(2,5) reduces the BER to $2.19 \times 10^{-2}\%$, demonstrating the effectiveness of the concatenated ECC in suppressing residual response errors. When fixed-stride spatial interleaving is further introduced, the post-ECC BER decreases to $8.45 \times 10^{-4}\%$. This

improvement indicates that dispersing locally correlated errors across different coding positions can enhance the decoding capability of the subsequent ECC stage.

Table 6. Ablation results of spatial interleaving and Rep(3,1)-RM(2,5) concatenated error correction.

Configuration	Interleaving	Rep(3,1)-RM(2,5)	Pre-ECC BER (%)	Post-ECC BER (%)
BTCL + SBP	×	×	2.14	—
BTCL + SBP + ECC	×	✓	2.14	2.19×10^{-2}
BTCL + SBP + Interleaving + ECC	✓	✓	2.14	8.45×10^{-4}

To evaluate the computational and storage overhead of the proposed framework, the complexity of its main components is summarized in Table 7. The dual-branch BTCL predictor contains 2,081,154 parameters and requires approximately 7.94 MB of FP32 parameter storage. Since only the branch corresponding to the current temperature direction is used for prediction, a single inference requires approximately 1.03 M MACs. Historical replay retains only the window-error vectors of previously observed temperature tasks, resulting in approximately 1.16 MB of replay data under the current experimental configuration, with the storage requirement increasing linearly with the number of observed tasks. The Soft BitPattern recovery module introduces no trainable parameters and requires only approximately 26.5 KB to store the 512 predefined candidate patterns. In addition, fixed-stride spatial interleaving performs only a linear permutation over the selected response bits, while Rep(3,1)-RM(2,5) concatenated error correction is implemented through deterministic encoding and decoding. Therefore, the main computational and storage overhead of the proposed framework is concentrated in the BTCL predictor, whereas the subsequent response-recovery, interleaving, and error-correction stages introduce relatively limited additional overhead.

Table 7. Computational complexity and storage overhead of the proposed framework.

Component	Parameters	Storage Overhead	Computational Cost
Dual-branch BTCL predictor	2,081,154	7.94 MB (FP32)	≈ 1.03 M MACs per prediction
Historical replay	0	≈ 1.16 MB for the complete temperature-task sequence	Increases linearly with the number of historical tasks
Soft BitPattern response recovery	0	≈ 26.5 KB	Up to 512 candidate-pattern evaluations per local position
Fixed-stride spatial interleaving	0	Negligible	$O(N)$
Rep(3,1)-RM(2,5) concatenated ECC	0	Negligible	Deterministic encoding and decoding

The present experiments were conducted on ten Arduino UNO boards of the same model; therefore, further validation across different hardware platforms, manufacturing batches, and larger device populations is still required to assess cross-platform and cross-device generalization. The present experimental evaluation focuses on temperature-induced response variation. Future validation will further extend the analysis to supply-voltage fluctuation, device aging, long-term response drift, and cross-batch variability to assess the robustness of the proposed method under a broader range of field operating conditions. Since the proposed learning and response-recovery procedures are driven by the measured SRAM error distributions and cell-reliability information, extension to different microcontroller/SRAM architectures can be achieved through platform-specific enrollment and adaptation.

5. Conclusions

This paper proposes an SRAM-PUF response recovery and key reconstruction method for intelligent metering terminals. The proposed method integrates overlapping-window modeling, historical replay, temperature-aware weighting, and SRAM-cell reliability priors to enable continual learning and reliable recovery of response errors under complex operating conditions. Spatial interleaving and concatenated error correction are further incorporated to improve the stability of key reconstruction. Experimental results show that the proposed method can effectively suppress SRAM-PUF response drift and improve response consistency and key reconstruction reliability under varying operating conditions, providing a reliable hardware-response and key-reconstruction basis for subsequent terminal authentication applications. Future work will focus on broader hardware validation using different device platforms and practical smart meters, power information acquisition terminals, and fusion terminals. The proposed method will also be further evaluated in practical intelligent metering systems in terms of authentication reliability, resource overhead, and engineering applicability through integration with terminal enrollment, authentication, key updating, and data interaction processes.

Author Contributions: Conceptualization, P.M., Q.W., X.Q., P.W., X.Z. and J.Z.; Methodology, P.M., Q.W., X.Q., P.W. and X.Z.; Software, P.M., Q.W., X.Q., P.W. and X.Z.; Validation, P.M., Q.W., X.Q. and P.W.; Writing—review & editing, X.Q.; Visualization, X.Z. and J.Z.; Supervision, X.Z. and J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Science and Technology Project of State Grid Weihai Power Supply Company under Grant No. 520613250001.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: Authors P.M., X.Q., P.W., X.Z. and J.Z. were employed by and Q.W. were employed by the Metrology Center, State Grid Shandong Electric Power Company, Jinan. The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Zhang, J.; Yu, S.; Xu, L. Research on the control method of distributed photovoltaic power generation output in novel power system considering the voltage over-limit risk. *Electr. Meas. Instrum.* **2026**, *63*, 11–23.
2. Li, G.; Wang, N.; Gao, R. Research on multi-source heterogeneous data integration technology for smart efficiency improvement in new-type power systems. *Electr. Meas. Instrum.* **2024**, *61*, 116–121.
3. Zhang, Y.; Song, M.; Ma, X. Research on distributed economic dispatching algorithm for information security and privacy protection in smart grid. *Electr. Meas. Instrum.* **2025**, *62*, 149–157.
4. Du, M.; Zhang, X.; Guerrero, J.M. Adaptive robust planner-attacker-operator model against multistage uncertain malicious attacks for power systems with volatile wind power. *IEEE Trans. Ind. Inform.* **2026**, *22*, 5378–5389. [[CrossRef](#)]
5. Si, X.; Jia, H.; Zeng, J.; Li, Y. PUF-based identity authentication for internet of things against machine learning attacks in zero-trust architecture. *J. Inf. Secur. Res.* **2026**, *12*, 33–42.
6. Gassend, B.; Clarke, D.; Van Dijk, M.; Devadas, S. Silicon physical random functions. In Proceedings of the 9th ACM Conference on Computer and Communications Security, Washington, DC, USA, 18–22 November 2002; pp. 148–160.
7. Guajardo, J.; Kumar, S.S.; Schrijen, G.J.; Tuyls, P. FPGA intrinsic PUFs and their use for IP protection. In Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems (CHES), Vienna, Austria, 10–13 September 2007; pp. 63–80.
8. Al-Meer, A.; Al-Kuwari, S. Physical unclonable functions (PUF) for IoT devices. *ACM Comput. Surv.* **2023**, *55*, 1–31. [[CrossRef](#)]
9. Holcomb, D.E.; Burleson, W.P.; Fu, K. Power-up SRAM state as an identifying fingerprint and source of true random numbers. *IEEE Trans. Comput.* **2009**, *58*, 1198–1210. [[CrossRef](#)]

10. Saraza-Canflanca, P.; Carrasco-Lopez, H.; Santana-Andreo, A.; Brox, P.; Castro-Lopez, R.; Roca, E.; Fernandez, F. Improving the reliability of SRAM-based PUFs under varying operation conditions and aging degradation. *Microelectron. Reliab.* **2021**, *118*, 114049. [\[CrossRef\]](#)
11. Wang, W.; Singh, A.D.; Guin, U. A systematic bit selection method for robust SRAM PUFs. *J. Electron. Test.* **2022**, *38*, 235–246. [\[CrossRef\]](#)
12. Van Aubel, P.; Bernstein, D.J.; Niederhagen, R. Investigating SRAM PUFs in large CPUs and GPUs. In *Proceedings of the International Conference on Security, Privacy, and Applied Cryptography Engineering*; Springer International Publishing: Cham, Switzerland, 2015; pp. 228–247.
13. Schrijen, G.J.; Van Der Leest, V. Comparative analysis of SRAM memories used as PUF primitives. In *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE)*; IEEE: Piscataway, NJ, USA, 2012; pp. 1319–1324.
14. Shi, T.; Chen, S.; Chen, H. SRAM PUF preselection method based on neighboring spatial majority voting. *J. Electron. Test.* **2026**, *42*, 117–128. [\[CrossRef\]](#)
15. Santana-Andreo, A.; Saraza-Canflanca, P.; Castro-Lopez, R.; Roca, E.; Fernandez, F. Reliability improvement of SRAM PUFs based on a detailed experimental study into the stochastic effects of aging. *AEU Int. J. Electron. Commun.* **2024**, *176*, 155147. [\[CrossRef\]](#)
16. Bösch, C.; Guajardo, J.; Sadeghi, A.R.; Shokrollahi, J.; Tuyls, P. Efficient helper data key extractor on FPGAs. In *Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 181–197.
17. Dodis, Y.; Ostrovsky, R.; Reyzin, L.; Smith, A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.* **2008**, *38*, 97–139. [\[CrossRef\]](#)
18. Zhou, Y.; Yu, Z. SRAM-PUF key extraction method and performance analysis based on RS and BCH codes. *Comput. Eng.* **2024**, *50*, 187–193.
19. Xue, J.; Wang, J.; Zhang, X.; Fan, Y.; Qu, J. A local HPLC+RF dual-mode communication ad hoc network identity authentication method in low-voltage station area. *Electr. Meas. Instrum.* **2024**, *61*, 219–224.
20. Yu, M.D.; Devadas, S. Secure and robust error correction for physical unclonable functions. *IEEE Des. Test Comput.* **2010**, *27*, 48–65. [\[CrossRef\]](#)
21. Wang, P.; Ren, M.; Chen, B.; Hu, S. Low-overhead and high-security PUF circuit based on cross-coupled structure. *J. Electron. Inf. Technol.* **2025**, *47*, 3093–3103.
22. Cao, R.; Mei, N.; Lian, Q. Method for improving the reliability of SRAM-based PUF using convolution operation. *Electronics* **2022**, *11*, 3493. [\[CrossRef\]](#)
23. Ni, L.; Zhang, J. S2RAM PUF: An ultra-low power subthreshold SRAM PUF with zero bit error rate. In *Proceedings of the 61st ACM/IEEE Design Automation Conference*, San Francisco, CA, USA, 23–27 June 2024; pp. 1–6.
24. Pratihari, K.; Chatterjee, S.; Chakraborty, R.S.; Mukhopadhyay, D. Enhancing SRAM-based PUF reliability through machine learning-aided calibration techniques. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2024**, *43*, 3491–3502. [\[CrossRef\]](#)
25. Suragani, R.; Nazarenko, E.; Anagnostopoulos, N.A.; Mexis, N.; Kavun, E.B. Identification and classification of corrupted PUF responses via machine learning. In *Proceedings of the 2022 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*; IEEE: Piscataway, NJ, USA, 2022; pp. 137–140.
26. Najafi, F.; Kaveh, M.; Martín, D.; Mosavi, M.R. Deep PUF: A highly reliable DRAM PUF-based authentication for IoT networks using deep convolutional neural networks. *Sensors* **2021**, *21*, 2009. [\[CrossRef\]](#) [\[PubMed\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.