A Blockchain-Based Distributed Solution For Intelligent Sports Equipment Operation And Maintenance Training Data Provenance And Privacy Protection

Youliang Han[1*] and Wenguang Geng[2]

[1] Jiangsu Maritime Institute, Sports department, Nanjing, 211170, China
[2] Department of Physical Education, Nanjing Agricultural University, Nanjing 210095, China
[*] Corresponding author. E-mail: 13913015815@163.com

Received July 21, 2026; accepted August 14, 2026

As intelligent sports equipment and edge terminals become widely deployed, the O&M and training data they generate are collected at high frequency and shared across organizations, yet centralized storage cannot support tamper traceability, privacy, access control, or scalability. This study proposes BSVP-Chain, treating blockchain as an evidence-anchoring layer cooperating with edge preprocessing and off-chain storage rather than a raw-data warehouse, using a four-layer architecture (acquisition, edge preprocessing, off-chain storage, blockchain evidence anchoring/audit) that reduces exposure via edge cleaning, lowers on-chain load via Merkle-tree batch anchoring, and combines smart contracts, ABAC, and differential privacy for authorization and statistical privacy. In simulations with 1 million records, BSVP-Chain cut storage latency by 78.6% and on-chain storage by 96.7% versus full-chain, sustained 380 ms latency under 500 nodes/10,000 concurrent requests, intercepted 99.1% of unauthorized accesses, and limited false-release to 0.7%.

Keywords: blockchain; intelligent sports equipment; operation and maintenance data; training data; privacy protection; smart contracts; access control.

http://dx.doi.org/10.6180/jase.202612_35.014

1. Introduction

Intelligent sports equipment is moving beyond simple recording into the sensing core of distributed information systems, continuously generating physiological, trajectory, load, fatigue, and recovery data requiring reliable, auditable flows. This data carries risk: inconsistent sampling can cause tampering, and heart rate, fatigue, and trajectories are sensitive data whose unauthorized access could reveal health status and strategies, while different roles need permission boundaries static accounts can't capture. Blockchain provides trusted timestamps and audit trails but faces throughput/storage constraints [1, 2], with decentralized trust reducing failure points but requiring attention to privacy and contract security [3, 4] — motivating coordinated on-chain indexing, off-chain storage, edge preprocessing, and privacy-aware access control.

Existing blockchain-IoT solutions target industrial equipment and medical data, whereas sports data has scenario-specific requirements: high-frequency sequences reflecting health and performance. Centralized databases lack independent audit evidence; full on-chain storage is costly. IPFS-like networks provide a feasible off-chain basis [5, 6]. The key gap is joint management of provenance, authorization, and statistical release under sports-specific privacy levels.

Contributions: a collaborative on-chain/off-chain architecture storing only verifiable summaries and audit records on-chain; a Merkle-tree batch notarization mechanism aggregating record hashes into a root hash, reduc-

ing transactions while preserving verifiability; a privacy mechanism integrating anonymous identity mapping, dynamic attribute-based access control, device trustworthiness, and differentially private release; a simulation framework covering notarization efficiency, security attacks, and node expansion, coupling sports-specific edge abstraction with audit-aware authorization; and multi-algorithm Grad-CAM comparisons of action attention regions across schemes.

Blockchain-IoT integration is important for distributed data trust, but limited terminal resources and unstable networks make consensus and storage costly [1], with on-chain indexing and archiving key to large-scale implementation [2]. In sports scenarios, sensor nodes grow with team size, so uncoordinated designs struggle with high-frequency data. Decentralization improves identity authentication and integrity but introduces smart-contract and metadata-correlation risks [3, 4], so sports data storage must protect on-chain metadata, off-chain ciphertext, and access logs together.

Distributed content-addressable storage suits large encrypted files: IPFS locates data via content hashes, providing a basis for off-chain storage with on-chain indexing [5], though persistence, access permissions, and key management need additional mechanisms [6]. For sports training data, video frames and long-term sequences are unsuited to direct on-chain storage; blockchain is better used for summaries, Merkle roots, and audit records.

Wearable sensors are widely used in sports assessment, continuously collecting heart rate, acceleration, and posture for training assessment [7]. Reviews identify real-time analysis and privacy as key directions [8] — showing sports data now behaves like a large-scale sensor system requiring governance around scalability and privacy.

Wearable data often contains highly sensitive status information. Local differential privacy for wearable data can reduce individual exposure during release [9]. Blockchain-based provenance models strengthen source proof and audit capability in IoT [10].

Privacy-preserving IoT aggregation combines blockchain with homomorphic encryption to reduce exposure [11]; blockchain IoT privacy schemes emphasize audit logs [12].

Smart contracts with IPFS support storage and device authentication in fog computing [13]; blockchain improves cross-entity trust [14] — though these rarely connect provenance with sports-specific sensitivity grading.

2. Methods

2.1. System Model and Problem Formulation

2.1.1. *Distributed intelligent sports equipment O&M and training data scenario*

This study considers a distributed system with athletes, wearable devices, edge gateways, institution servers, off-chain storage, a consortium blockchain, and an auditor. Devices collect heart rate, speed, cadence, and location; video is a feature source, not an on-chain object. The edge gateway handles cleaning and batch-hash calculation. Permissions differ by role: athletes view own data, coaches access risk levels, administrators maintain without reading sensitive content, and auditors verify integrity without plaintext access.

2.1.2. *Training data representation*

Let $D_{i,t}$ denote the original training record generated by subject i within time window t. The record includes an anonymized identity, device number, timestamp, physiological features, trajectory summary, motion features, training load, and sensitivity level. This notation is used consistently in the following storage, access-control, and verification formulas:

$$D_{i,t} = \left\{ pid_i, d_i, \tau_t, x_{i,t}^{hr}, x_{i,t}^{pos}, x_{i,t}^{act}, x_{i,t}^{load}, s_{i,t} \right\}. \tag{1}$$

Here, pid_i denotes the anonymous subject identifier, d_i denotes the device number, τ_t denotes the timestamp, $x_{i,t}^{hr}$ denotes the heart-rate feature, $x_{i,t}^{pos}$ denotes the trajectory or region summary, $x_{i,t}^{act}$ denotes the action-posture feature, $x_{i,t}^{load}$ denotes the training-load feature, and $s_{i,t}$ denotes the data-sensitivity level. After the multisource training records are windowed, the batch data received by edge nodes within window T_w can be written as:

$$W_t = \{D_{1,t}, D_{2,t}, \ldots, D_{m,t}\}, \quad m = |W_t|. \tag{2}$$

To reduce dimensional differences, edge nodes normalize numerical training features:

$$\hat{x}_{i,t}^{(r)} = \frac{x_{i,t}^{(r)} - \mu_r}{\sigma_r + \delta}, \quad r \in \{hr, pos, act, load\}, \tag{3}$$

Here, μ_r and σ_r are the mean and standard deviation of feature class r, respectively, and δ is a small constant that prevents division by zero. The normalized data is used for motion-quality assessment, fatigue-risk statistics, and edge anomaly filtering, but plaintext data is not written directly to the blockchain.

2.1.3. Threat model

This study considers six threat types: off-chain data tampering (modifying ciphertext or metadata); record forgery (uploading non-existent training records); unauthorized access (reading sensitive data without authorization); replay attacks (resubmitting historical data); privacy-inference attacks (inferring fatigue or injury risk from repeated statistical queries); and malicious edge-node attacks (submitting valid-format but semantically anomalous batches) — distinguishing data-plane attacks on records from control-plane attacks on access decisions and audit logs.

Most consortium blockchain consensus nodes are assumed trustworthy, allowing a small number of semi-honest or malicious nodes; off-chain storage may face unauthorized access, constrained via on-chain hashing, Merkle proofs, and audit contracts. Device trustworthiness is treated as an access-control and registration attribute rather than assumed permanent; the adversary cannot break standard cryptography but may observe metadata, replay submissions, or exploit weak policies.

2.1.4. Design objectives

Design objectives include integrity, non-repudiation, privacy protection, fine-grained authorization, auditability, low on-chain overhead, and scalability: stored records must be verifiable, upload/access behaviors traceable, real identities and plaintext data never exposed on-chain, and permissions determined dynamically by role, attributes, sensitivity, purpose, and device trustworthiness. Correctness requires successful verification to link off-chain ciphertext, metadata, proof path, and on-chain root to the same batch; scalability means stable throughput and latency as nodes, data, and concurrency grow.

$$\min_{\Pi} \ \mathcal{J}(\Pi) = \alpha L_{anc} + \beta L_{ver} + \gamma S_{chain} + \eta R_{priv} - \lambda T_{sys}, \tag{4}$$

Here, Π denotes the system policy set, L_{anc} is the average evidence-anchoring latency, L_{ver} is the verification latency, S_{chain} is the on-chain storage overhead, R_{priv} is the privacy risk, T_{sys} is the system throughput, and $\alpha, \beta, \gamma, \eta, \lambda$ are weighting coefficients. The objective captures the trade-off among trustworthiness, privacy, and scalability.

2.2. Proposed BSVP-Chain Method

2.2.1. Overall architecture

BSVP-Chain has a four-layer architecture (Figure 1): a data acquisition layer (bracelets, IMUs, GPS, apps, action-recognition terminals); an edge preprocessing layer (cleaning, filtering, synchronization, anonymization, feature summarization, privacy labeling); an off-chain encrypted storage layer (encrypted data, feature summaries, indexes); and a consortium-blockchain evidence anchoring and audit layer (hash digests, Merkle roots, access policies, authorization records, anomaly events). Each record moves from sensing to edge preprocessing, encryption and off-chain storage, then on-chain anchoring, access decision, and integrity verification.

Fig. 1. Architecture of the blockchain-enabled distributed sports training data provenance and privacy protection system

2.2.2. Edge-side preprocessing

Edge preprocessing is the first line of defense for reducing system load and privacy exposure: edge nodes divide raw sensor data into sliding windows and extract statistical, action, and risk-summary features per window.

$$F_t = \phi_\theta(W_t) = \left[\frac{1}{m}\sum_{j=1}^{m} z_j,\ \max_{1\le j\le m} z_j,\ \sqrt{\frac{1}{m}\sum_{j=1}^{m}\left(z_j - \bar{z}\right)^2},\ a_t \right], \tag{5}$$

For video data, the system retains only keypoints, action scores, and intermediate features rather than original face/background images; for trajectory data, it retains region-level summaries and hashes rather than precise coordinates — reducing both storage burden and privacy exposure.

The data sensitivity level is determined by the data type, access purpose, and data-subject authorization status:

$$s_{i,t} = softmax\left(W_s\left[\psi(r_i), \psi(p_i), \psi(o_i), q_i\right] + b_s\right), \tag{6}$$

High-sensitivity data includes raw heart-rate sequences, precise trajectories, injury-risk markers, and recovery status; medium/low-sensitivity data includes training programs, duration, stage scores, and anonymized population statistics.

2.2.3. On-chain and off-chain collaborative storage

This study uses coordinated on-chain/off-chain storage. Training records are processed at the edge to generate ciphertext.

$$C_{i,t} = Enc_{k_i}\left(D_{i,t}, F_t, meta_{i,t}\right), \tag{7}$$

Here, $Enc_{k_i}(\cdot)$ denotes the encryption function using authorized key k_i, and $meta_{i,t}$ denotes metadata such as anonymous identity, device ID, timestamp, and privacy level. The ciphertext $C_{i,t}$ is written to an off-chain distributed storage system, which returns the content address $addr_{i,t}$. Edge nodes then generate leaf-node hashes:

$$h_{i,t} = H\left(C_{i,t} \parallel meta_{i,t} \parallel addr_{i,t} \parallel s_{i,t}\right). \tag{8}$$

Plaintext training data is never stored on-chain; only hash values, batch aggregation results, timestamps, policy numbers, and audit records are stored, so even public on-chain data cannot directly reveal athletes' physiological states or trajectories, while off-chain modification is still detectable via on-chain hashes.

2.2.4. Merkle-tree-based batch evidence anchoring

Because sports training data is generated at high frequency, uploading every record would rapidly increase transaction volume, so this study uses Merkle-tree batch anchoring (Figure 2), aggregating a batch's leaf-node hashes into a root hash.

$$MRoot_t = \mathcal{M}\left(h_{1,t}, h_{2,t}, \ldots, h_{m,t}\right), \tag{9}$$

The system writes only the batch Merkle root, batch number, edge-node number, time window, off-chain index, and policy number on-chain; verifying a single record needs only its leaf hash and proof path, not the full batch.

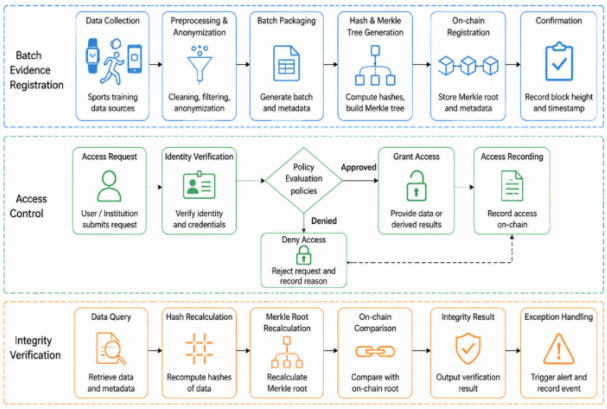

Fig. 2. Workflow of batch evidence registration, access control, and integrity verification

2.2.5. Smart contract-based access control

Four smart contract types are used: a data registration contract receiving Merkle roots and batch metadata; an access control contract authorizing by subject attributes and data sensitivity; an integrity verification contract comparing on-chain root hash with off-chain proof path; and an audit contract recording registration, access requests, authorization results, verification failures, and anomalies — separated to make abnormal events independently traceable.

Access decisions consider roles, attributes, device trustworthiness, request frequency, historical anomalies, and data sensitivity rather than roles alone.

$$\Omega(a,d,p) = \sigma\left(\omega_1 R_a + \omega_2 A_a + \omega_3 T_a - \omega_4 S_d - \omega_5 B_a + \omega_6 P_p\right), \tag{10}$$

Here, R_a denotes role matching, A_a denotes attribute satisfaction, T_a denotes the trustworthiness of the subject or device, S_d denotes the data-sensitivity level, B_a denotes the penalty for historical abnormal behavior, and P_p denotes the legitimacy of the access purpose, with $\sigma(\cdot)$ denoting the sigmoid function. The authorization result is:

$$Permit(a,d,p) = \begin{cases} 1, & \Omega(a,d,p) \geq \rho_d \text{ and } Policy(a,d,p) = 1, \\ 0, & \Omega(a,d,p) < \rho_d \text{ or } Policy(a,d,p) = 0, \end{cases} \tag{11}$$

ρ_d Highly sensitive data has a higher authorization threshold requiring data-subject authorization or secondary confirmation, set according to privacy level to trigger secondary approval for injury risk, recovery status, or precise trajectory data.

2.2.6. Privacy protection mechanism

Privacy mechanisms include anonymous identity mapping, off-chain encrypted storage, dynamic access control, and differentially private statistical release. Anonymous identity mapping stores only pseudo-identities on-chain, with real identities encrypted off-chain by a trusted institution and pseudo-identities updatable by training cycle to mitigate long-term association risk.

$$pid_i^{(t)} = H\left(uid_i \parallel salt_t \parallel sk_{org}\right), \tag{12}$$

Here, uid_i is the internal real-identity ID, $salt_t$ is the periodically randomized salt, and sk_{org} is the institution's private key or key-derived parameter. Off-chain encrypted storage prevents unauthorized entities from reading training content directly, and dynamic access control reduces the risk of accidental release of highly sensitive data.

For statistics like team-average training load, fatigue-risk ratio, and phase training effectiveness, a differential privacy mechanism is used.

$$\widetilde{q}(D) = q(D) + Lap\left(\frac{\Delta q}{\varepsilon}\right), \tag{13}$$

A smaller privacy budget strengthens protection but increases statistical error, while a larger budget improves availability but increases risk — public team reports need stronger protection, while internal optimization can tolerate a larger budget.

2.2.7. Integrity verification

The integrity verification process includes off-chain data reading, leaf-node hash recalculation, Merkle-proof path calculation, and comparison with the on-chain root hash. The verification function is defined as follows:

$$Verify\left(D_{i,t}\right) = \begin{cases} 1, & Root\left(h_{i,t}, path_{i,t}\right) = MRoot_t^{chain}, \\ 0, & Root\left(h_{i,t}, path_{i,t}\right) \neq MRoot_t^{chain}. \end{cases} \tag{14}$$

When verification fails, the audit contract logs the exception event and marks the relevant batch, edge node, and access subject for audit. Because a Merkle proof requires only tree-height-order hash calculations, the verification complexity for a single record is $O(\log m)$, making the process suitable for fast verification of millions of training records.

2.3. Algorithms and Complexity Analysis

2.3.1. Batch evidence registration algorithm

The batch notarization algorithm takes training batches, edge-node numbers, time windows, and privacy levels as input, outputting on-chain transaction numbers, a Merkle root, and an off-chain index: edge nodes clean and deduplicate records, generate anonymity/privacy levels, encrypt records and summaries to off-chain storage, compute leaf hashes and build a Merkle tree, then the registration contract writes the root, time window, edge-node number, policy number, and index summary on-chain — shifting transaction volume from record-level to batch-level.

2.3.2. Integrity verification algorithm

The integrity verification algorithm takes the target record, off-chain address, Merkle proof path, and on-chain root hash as input, recalculating the leaf hash and root along the proof path to compare with the on-chain root: a match confirms the record is unmodified, a mismatch signals anomaly — without needing to download the full batch, suiting third-party auditing.

2.3.3. Privacy-aware access control algorithm

The privacy-aware access control algorithm takes the subject, target data, purpose, and time as input, returning allow/deny/secondary authorization, checking role, attributes, device trustworthiness, historical anomalies, data type, privacy level, and authorization status, with denial and audit logging for abnormal frequency or excessive violations.

2.3.4. Complexity analysis

Let the batch size be m, the number of edge nodes be n, the average on-chain digest length per record be b_h, and the average complete-record size be b_d. The on-chain storage overhead of a typical full-data-on-chain scheme can be approximated as:

$$S_{full} = \sum_{t=1}^{T} \sum_{j=1}^{m_t} b_d, \tag{15}$$

The proposed solution only stores batch summaries and a small amount of metadata on-chain, with the following on-chain overhead:

$$S_{bsvp} = \sum_{t=1}^{T} \left(b_{root} + b_{time} + b_{policy} + b_{index}\right). \tag{16}$$

When the condition $b_d \gg b_{root} + b_{time} + b_{policy} + b_{index}$ holds and m_t is large, S_{bsvp} is significantly smaller than S_{full}. In terms of communication overhead, a single verification only requires transmitting the leaf node, the proof path, and a small amount of metadata:

$$C_{ver} = b_h \lceil \log_2 m \rceil + b_{meta} + b_{addr}. \tag{17}$$

The proposed scheme thus maintains low verification communication cost even as data volume increases; access-control algorithm complexity depends mainly on attribute and policy-entry counts, reducible via policy-index optimization. Overall, BSVP-Chain shifts high-frequency write pressure from on-chain storage to edge batch processing and off-chain storage, aligning with scalable information system design goals.

2.4. Experimental Design

2.4.1. Experimental environment

Experiments used a consortium-blockchain simulation (Hyperledger Fabric or private-Ethereum-equivalent), IPFS/distributed object off-chain storage, and Docker-deployed edge nodes on Ubuntu 20.04 with an 8-core CPU, 64GB memory, and 2TB SSD. Node sizes were 10-500; record sizes 100K-5M; concurrent requests 100-10,000; and batch windows 10-1,000, with all methods evaluated under the same grid for consistency.

Table 1. Experimental configuration of the distributed sports training data system

Item	Setting	Purpose
Blockchain network	Fabric/private Ethereum simulation	Consortium-chain evidence anchoring
Edge nodes	10, 30, 50, 100, 200, 500	Scalability evaluation
Training records	0.1M, 0.5M, 1M, 3M, 5M	Storage and anchoring stress test
Concurrent requests	100, 500, 1,000, 3,000, 5,000, 10,000	Response latency evaluation
Batch window	10, 50, 100, 500, 1,000	Merkle batching analysis
Privacy budget	0.5, 1, 2, 4, 8, No-DP	Privacy-utility trade-off

Table 2. Compared Methods and Key Mechanisms

Method	Evidence anchoring	Privacy protection	Scalability mechanism
Centralized-DB	Central log	Account permission	Database indexing
Full-chain	Record-level on-chain storage	Basic encryption	None
BC-IPFS	Hash plus off-chain address	Off-chain storage	Partial off-chain storage
Static-ABAC	Blockchain audit	Static attribute policy	Fixed rule matching
BSVP-Chain	Merkle batch anchoring	Dynamic ABAC plus DP	Edge batching and off-chain ciphertext

2.4.2. *Dataset construction*

Experimental data had three parts: motion-training sensor data, access-request logs, and attack samples, parameterized from common indicators rather than identifiable profiles. Image frames for squatting, jump landing, sprint starts, and shooting were constructed to compare algorithm attention regions; video frames are not stored on-chain, only feature summaries and hash indexes.

2.4.3. *Baseline methods*

Five methods were compared: Centralized-DB (centralized database); Full-chain (records/summaries written one by one on-chain); BC-IPFS (off-chain storage with on-chain hash indexes, lacking dynamic access control/DP); Static-ABAC (static attribute-based access control without trustworthiness/anomaly consideration); and BSVP-Chain (integrating edge preprocessing, off-chain encryption, batch notarization, dynamic authorization, differential privacy, and audit contracts) — isolating the contribution of decentralization, off-chain storage, dynamic authorization, and privacy-preserving release.

2.4.4. *Evaluation metrics*

Evaluation metrics span four categories: evidence-anchoring performance (latency, TPS, transaction count); verification performance (single/batch verification time, success rate); security/privacy (tampering detection, unauthorized-access interception, forged-record identification, replay interception, attack success rate, statistical error, privacy risk); and scalability (response latency, TPS, communication overhead, stability with added nodes) — with privacy/attack indicators interpreted as comparative trends rather than absolute guarantees.

3. Results and discussion

3.1. Evidence preservation efficiency

Table 3 reports anchoring performance at 1M records: Centralized-DB has lowest write latency but no independent evidence; Full-chain has highest latency/overhead from one-by-one uploads; BC-IPFS reduces on-chain burden but still registers digests individually; Static-ABAC barely changes storage overhead; BSVP-Chain substantially reduces on-chain transactions via Merkle batch anchoring.

Compared with Full-chain, BSVP-Chain reduces average storage latency by ~78.6% and on-chain storage by ~96.7%, showing sports training data suits a combination of on-chain digests, off-chain ciphertext, and batch root hashes better than full on-chain storage — the improvement comes from reducing write frequency, not weakening verification.

3.2. Integrity verification and tamper detection

The integrity experiment set tampering rates at 1-30% (changing duration, replacing heart rate, deleting records, forging scores, modifying addresses), with the verifier recalculating the root hash for comparison against the on-chain record (Table 4).

BSVP-Chain shows strong tampering detection, from joint recording of batch root hashes, policy numbers, addresses, and audit events, not just hash comparison. BC-IPFS detects most ciphertext replacement but has weaker traceability when both indexes and access logs are abnormal — important since disputes often involve both content and access history.

Table 3. Evidence anchoring performance under 1,000,000 training records

Method	Anchoring latency (ms)	TPS	On-chain storage (MB)	Audit support
Centralized-DB	18.6	8420	0	No
Full-chain	426.5	680	3840	Yes
BC-IPFS	162.4	2140	618	Yes
Static-ABAC	148.7	2360	604	Yes
BSVP-Chain	91.3	4180	126	Yes

Table 4. Tamper detection performance under different tamper ratios

Tamper ratio	Full-chain detection	BC-IPFS detection	Static-ABAC detection	BSVP-Chain detection	BSVP false positive
1%	99.8%	98.7%	98.9%	99.9%	0.3%
5%	99.9%	98.9%	99.0%	99.9%	0.4%
10%	99.9%	99.1%	99.2%	100.0%	0.4%
20%	100.0%	99.2%	99.3%	100.0%	0.5%
30%	100.0%	99.3%	99.4%	100.0%	0.5%

3.3. Access control and unauthorized request blocking

The access-control experiment used five request types (athlete, coach, administrator, auditor, unauthorized user) across low/medium/high sensitivity data, comparing Static-ABAC and BSVP-Chain (Table 5).

BSVP-Chain has slightly higher authorization latency than Static-ABAC from considering device trustworthiness, historical behavior, privacy level, and authorization status, but significantly reduces the false-release rate for sensitive data and raises unauthorized-access interception to 99.1% — the extra latency being a security-privacy trade-off, not a pure performance penalty.

3.4. Privacy budget and utility analysis

Differential privacy experiments analyzed statistical error and privacy risk across budgets for team-average load, fatigue-risk ratio, and training-intensity distribution (Table 6).

A smaller privacy budget increases statistical error but strengthens protection; a moderate range balances error and risk for public team reports, while internal-only use can tolerate a larger budget — the budget being a policy parameter, not a fixed universal value.

3.5. Attack resistance experiment

Attack experiments (replay, spoofing, unauthorized access, tampering, malicious uploads; Table 7) show Centralized-DB has weak auditing/tamper resistance; BC-IPFS reduces tampering success but lacks dynamic authorization/anomaly auditing; Static-ABAC is insensitive to high-frequency abnormal requests; BSVP-Chain has the lowest attack success rate, from combined on-chain auditing, dynamic authorization, and Merkle proofs — testing both storage integrity and authorization behavior.

BSVP-Chain significantly reduces but doesn't eliminate malicious-upload success, since some edge nodes may submit correctly-formatted but semantically anomalous data — a deployment risk addressable via federated anomaly detection, trusted execution, or hardware signatures.

3.6. Grad-CAM visualization comparison across algorithms

Figure 3 compares Grad-CAM visualizations for CNN-LSTM, CRNN, Spatiotemporal Transformer, and the BSVP-Chain edge privacy model on the same action image frames, showing attention regions for action-quality evaluation.

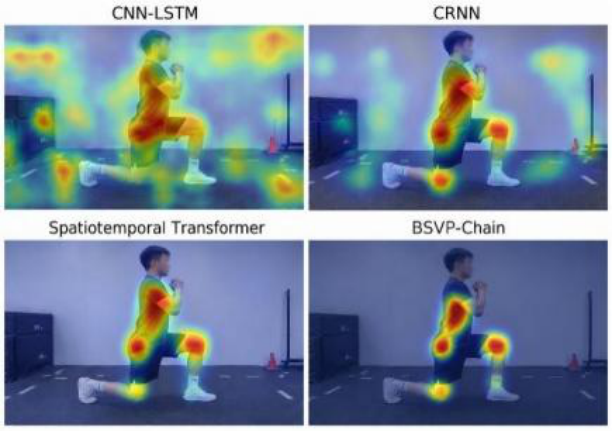

Fig. 3. Grad-CAM comparison of different algorithms on real-scene-like sports training frames

CNN-LSTM's attention is dispersed and background-sensitive; CRNN partially focuses on trunk/knee but hotspots diffuse; the Transformer concentrates on knee, hip, and trunk regions; BSVP-Chain suppresses background, face, and irrelevant-region attention while retaining key

Table 5. Access Control Performance Comparison

Method	Authorization accuracy	Unauthorized blocking	Authorization latency (ms)	Mis-release of sensitive data
Static-ABAC	94.2%	92.8%	35.4	3.6%
BSVP-Chain	98.7%	99.1%	48.2	0.7%

Table 6. Differential Privacy Budget and Data Utility

Privacy budget	Mean load error	Fatigue distribution error	Utility level	Privacy risk
0.5	7.8%	8.4%	Medium	Low
1	5.1%	5.9%	Medium-high	Low-medium
2	3.6%	4.1%	High	Medium
4	2.2%	2.8%	High	Medium-high
8	1.4%	1.9%	Very high	High
No-DP	0.8%	1.1%	Highest	Very high

action regions — suggesting reduced image-leakage risk while preserving evaluation-relevant information, used as supporting rather than primary evidence.

3.7. Response heatmap under scalability conditions

Figure 4 presents a response heatmap across node sizes (vertical) and concurrent requests (horizontal), with color showing average response latency.

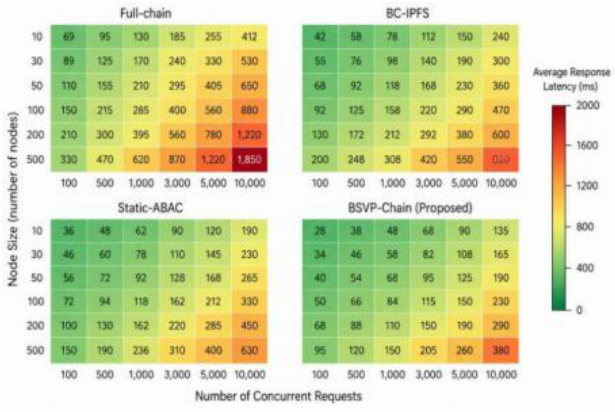

Fig. 4. Response heatmap comparison under different node sizes and concurrent requests

Full-chain shows high latency at large node/high concurrency; BC-IPFS has a larger low-latency region but faces pressure beyond 5,000 concurrent requests; Static-ABAC has lower latency but weaker security; BSVP-Chain remains stable overall, with high-latency regions only at the extreme of 500 nodes/10,000 concurrent requests — showing good scalability though batching and off-chain storage delay rather than eliminate scalability pressure.

3.8. Ablation study

Ablation experiments removed edge preprocessing, Merkle batch notarization, dynamic access control, differential privacy, and audit contracts respectively (Table 8).

Merkle batch notarization contributes most to reducing latency and storage; edge preprocessing reduces data volume; dynamic access control improves unauthorized-access interception; differential privacy affects statistical-release risk; and audit contracts add small overhead but improve anomaly tracing — confirming module cooperation drives the advantage, not any single optimization.

3.9. Discussion

Overall, BSVP-Chain's advantages stem from layered collaborative design: off-chain encrypted storage handles large sensitive data volumes; Merkle batch anchoring reduces on-chain transactions; smart-contract access control improves authorization transparency; differential privacy limits statistical inference risk; and audit contracts preserve traceability. Versus centralized databases it improves credibility and non-repudiation; versus full on-chain storage it reduces overhead; versus basic BC-IPFS it adds complete access control and privacy protection; versus Static-ABAC it better suits multi-role, multi-privacy-level, changing-trustworthiness sports scenarios.

From a scalable information systems perspective, blockchain is treated as a trusted state machine and evidence-anchoring layer rather than the sole computing platform: edge nodes handle high-frequency preprocessing and digest generation, off-chain storage handles large encrypted volumes, and smart contracts handle rule execution and audit logging, connected via hashes, timestamps, policy numbers, and access logs into a closed loop reducing on-chain load while maintaining evidence-chain integrity.

In deployment, batch window size, privacy budget, and

Table 7. Attack Success Rate Comparison

Attack type	Centralized-DB	BC-IPFS	Static-ABAC	BSVP-Chain
Replay attack	18.4%	8.7%	6.5%	2.1%
Forged identity attack	22.6%	11.3%	7.8%	2.8%
Unauthorized access	15.9%	9.6%	7.2%	0.9%
Off-chain tampering	31.4%	2.8%	2.5%	0.4%
Malicious edge upload	19.7%	10.2%	8.4%	3.2%

Table 8. Ablation Study of BSVP-Chain Modules

Variant	Anchoring latency	On-chain storage	Blocking rate	Privacy risk	Traceability
Without edge preprocessing	+28.4%	+17.6%	unchanged	Higher	unchanged
Without Merkle batching	+146.2%	+285.5%	unchanged	unchanged	unchanged
Without dynamic access control	-12.5%	unchanged	-6.8%	Higher	unchanged
Without differential privacy	unchanged	unchanged	unchanged	Very high	unchanged
Without audit contract	-5.6%	-3.1%	-1.2%	Higher	Lower
Full BSVP-Chain	baseline	baseline	best	Lower	best

access threshold should be adjusted dynamically: smaller batch windows for real-time feedback, larger for long-term low-cost auditing; stricter thresholds and smaller privacy budgets for injury risk and competitive strategies in high-level teams, more relaxed policies for anonymized general physical-education statistics.

These experiments use reproducible simulation designs; formal deployment should recalibrate parameters to actual venue networks, sampling frequency, and access behavior. Future work could incorporate zero-knowledge proofs to verify training metrics without exposing content, combine federated learning for fatigue-risk prediction without centralizing data, and evaluate contract security, long-term storage availability, and consent management in real institutions.

4. Conclusions

This paper addresses trusted storage, privacy protection, access control, and scalable management for sports equipment O&M data via BSVP-Chain: raw data is encrypted off-chain while only hashes, Merkle roots, and audit logs are written on-chain. Edge preprocessing, Merkle-tree batch anchoring, smart-contract ABAC, and differential privacy together reduce exposure and overhead. Results show BSVP-Chain outperforms baselines in storage efficiency, tamper detection, and access interception, confirming blockchain is better used as trusted indexing infrastructure than a raw data warehouse. Future research will focus on cross-chain sharing, zero-knowledge proofs, and real-world deployment.

Funding

This work has received funding from Education Reform Project of Jiangsu Provincial Department of Education: Construction and practical exploration of the trinity co-operative mode of "teaching, training and competition" in college physical education from the perspective of educational environment theory (2023JSJG283)

References

[1] A. Panarello, N. Tapas, G. Merlino, F. Longo, and A. Puliafito, (2018) *"Blockchain and IoT Integration: A Systematic Survey"* **Sensors *18***(8): 2575. DOI: 10.3390/s18082575.

[2] Q. Wei, B. Li, W. Chang, Z. Jia, Z. Shen, and Z. Shao, (2022) *"A Survey of Blockchain Data Management Systems"* **ACM Transactions on Embedded Computing Systems *21***(3): 25:1–25:28. DOI: 10.1145/3502741.

[3] V. Gugueoth, S. Safavat, S. Shetty, and D. Rawat, (2023) *"A Review of IoT Security and Privacy Using Decentralized Blockchain Techniques"* **Computer Science Review *50***: 100585. DOI: 10.1016/j.cosrev.2023.100585.

[4] D. Commey, B. Mai, S. G. Hounsinou, and G. V. Crosby, (2024) *"Securing Blockchain-Based IoT Systems: A Review"* **IEEE Access *12***: 98856–98881. DOI: 10.1109/ACCESS.2024.3428490.

[5] E. Daniel and F. Tschorsch, (2022) *"IPFS and Friends: A Qualitative Comparison of Next Generation Peer-to-Peer Data Networks"* **IEEE Communications Surveys & Tutorials *24***(1): 31–52. DOI: 10.1109/COMST.2022.3143147.

[6] T. V. Doan, Y. Psaras, J. Ott, and V. Bajpai, (2022) *"Toward Decentralized Cloud Storage with IPFS: Opportunities, Challenges, and Future Considerations"* **IEEE Internet Computing *26***(6): 7–15. DOI: 10.1109/MIC.2022.3209804.

[7] R. De Fazio, V. M. Mastronardi, M. De Vittorio, and P. Visconti, (2023) *"Wearable Sensors and Smart Devices to Monitor Rehabilitation Parameters and Sports Performance: An Overview"* **Sensors *23***(4): 1856. DOI: 10.3390/s23041856.

[8] L. Yang, O. Amin, and B. Shihada, (2024) *"Intelligent Wearable Systems: Opportunities and Challenges in Health and Sports"* **ACM Computing Surveys *56***(7): 190:1–190:42. DOI: 10.1145/3648469.

[9] Z. Li, B. Wang, J. Li, Y. Hua, and S. Zhang, (2022) *"Local Differential Privacy Protection for Wearable Device Data"* **PLOS ONE *17***(8): e0272766. DOI: 10.1371/journal.pone.0272766.

[10] H. Honar Pajooh, M. A. Rashid, F. Alam, and S. N. Demidenko, (2021) *"IoT Big Data Provenance Scheme Using Blockchain on Hadoop Ecosystem"* **Journal of Big Data *8***: 114. DOI: 10.1186/s40537-021-00505-y.

[11] F. Loukil, C. Ghedira-Guegan, K. Boukadi, and A.-N. Benharkat, (2021) *"Privacy-Preserving IoT Data Aggregation Based on Blockchain and Homomorphic Encryption"* **Sensors *21***(7): 2452. DOI: 10.3390/s21072452.

[12] J. Gong, Y. Mei, F. Xiang, H. Hong, Y. Sun, and Z. Sun, (2021) *"A Data Privacy Protection Scheme for Internet of Things Based on Blockchain"* **Transactions on Emerging Telecommunications Technologies *32***(5): e4010. DOI: 10.1002/ett.4010.

[13] S. K. Dwivedi, R. Amin, and S. Vollala, (2023) *"Smart Contract and IPFS-Based Trustworthy Secure Data Storage and Device Authentication Scheme in Fog Computing Environment"* **Peer-to-Peer Networking and Applications *16***: 1–21. DOI: 10.1007/s12083-022-01376-7.

[14] Y. Zhang, K. Gai, J. Xiao, L. Zhu, and K.-K. R. Choo, (2022) *"Blockchain-Empowered Efficient Data Sharing in Internet of Things Settings"* **IEEE Journal on Selected Areas in Communications *40***(12): 3422–3436. DOI: 10.1109/JSAC.2022.3213353.