STAP
Smart Technologies Academic Press

Jordanian Journal of Informatics and Computing
https://www.jjic.thestap.com/

Blockchain-Enabled Zero Trust Architecture for Securing Cloud Computing based on Cybersecurity Risk Auditing Framework

Mohammed Almaiah[1*], Santosh Reddy Addula[2]

[1] *King Abdullah the II IT School, The University of Jordan, Amman 11942, Jordan, m.almaiah@ju.edu.jo*
[2]*Department of Information Technology, University of the Cumberlands, Williamsburg, Kentucky, USA, saddula5840@ucumberlands.edu, https://orcid.org/0009-0000-3286-8224*

ARTICLE INFO

Article History

Received: 21-05-2026
Revised: 07-06-2026
Accepted: 30-10-2026

Vol.2027, No.1

DOI:
https://doi.org/10.63180/jjic.thestap.2027.1.4

Corresponding author.

Email: *m.almaiah@ju.edu.jo*

Orcid:
https://orcid.org/0000-0002-2215-2481

This is an open access article under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/).

Published by STAP Publisher.

ABSTRACT

Traditional cloud security controls tend to treat access, vulnerability exposure, and audit integrity as distinct functions, reducing the ability to be aware of risks on an ongoing basis. In this study, the authors present a blockchain-based zero-trust risk auditing framework for adaptive cloud protection (ACP) called BAZTRA. BAZTRA is a combination of a GBDT-based network-risk estimator and identity, device-posture, provenance, vulnerability, and compliance evidence. The CICIoT2023 corpus, controlled contextual replay, and repeated ten-seed experiments were used to evaluate the following techniques: availability-aware fusion, temporal smoothing, four-state enforcement, EPSS-KEV prioritization, hash chaining, and Merkle-based audit anchoring. Statistical significance was assessed using Friedman and Holm-corrected Wilcoxon tests across repetitions. The selected network-risk estimator achieved 78.96% accuracy, 55.01% macro-F1, 0.964 ROC-AUC, and 0.718 MCC. The complete BAZTRA framework produced 89.44% decision macro-F1, reduced unsafe access to 0.67%, and limited false denial to 0.03%. Ablation confirmed that adaptive enforcement, network evidence, provenance analysis, and availability-aware normalization were the most influential components. Merkle batching reduced effective ledger payload by 98.21% while preserving detection of modification, deletion, reordering, and replay attempts. BAZTRA provides a unified approach to continuous access assessment, vulnerability prioritization, and accountable cloud auditing. Future work will validate the framework on complete multi-source telemetry and a physical multi-peer Hyperledger Fabric deployment.

Keywords: Blockchain-Enabled Zero Trust, Cloud Security, Continuous Risk Auditing, Vulnerability Intelligence, Immutable Audit Trail.

How to cite the article
Almaiah, M., & Reddy Addula, S. (2026). Blockchain-Enabled Zero Trust Architecture for Securing Cloud Computing based on Cybersecurity Risk Auditing Framework. Jordanian Journal of Informatics and Computing, 2027(1), 65. https://doi.org/10.63180/jjic.thestap.2027.1.4

1. Introduction

Cloud computing is the operating foundation of today's digital services due to its elastic processing capabilities, ubiquitous access, fast deployment and relatively low infrastructure costs. The same attributes, though, increase the attack surface. Conventional perimeter security is becoming increasingly ineffective as data, identities, application interfaces, virtual machines, containers and administrative services are spread out across organizational and provider-controlled domains. Reputational signals, misjudging assessments, and incomplete contextual evidence are also susceptible to manipulation, and can lead to misjudging security even in the presence of cryptographic controls [1]. Even with outsourcing, replication, updating and processing of files by parties that are not under the same administrative control, integrity verification is still not easy at the data layer. Although blockchain-based dynamic auditing has enhanced the ability to establish file possession, there are still numerous schemes that are targeted at establishing possession instead of managing ongoing operational risk [2]. The verifiability and privacy aspects of recent confidential-computing protocols are well developed, but the connection between trusted execution, adaptive authorization and longitudinal risk evidence is not well developed [3].

One of the problems with this is that Zero Trust Architecture attempts to overcome the implicit trust in the network by verifying it explicitly and on a case-by-case basis. All access requests are considered based on identity, device condition, resource sensitivity, session activity and threat conditions. Zero-trust mechanisms based on blockchain have demonstrated that it is possible to achieve a decentralized policy enforcement and protected attribute evaluation, which decreases the reliance on a single authority [4]. Likewise, adaptive access control has been used for the advanced persistent threat mitigation to continuously re-evaluate subjects and communication paths [5]. However, it is identity that continues to be a major failing. Static authorization rules do not necessarily get violated, as cloud accounts, service principals, API credentials and workload identities can be compromised. Decentralized identity management can enhance the accountability of credential ownership and provisioning [6] and privacy-preserving federated and blockchain-based environments show that distributed analytics can be conducted without revealing all the underlying records. While these advances are promising, they are typically used as single security functions instead of as part of a single cloud-risk control loop.

The need for integration is further supported by the fact that it is hard to implement zero-trust. According to enterprise studies, in addition to technology, policy ambiguity, legacy infrastructure, administrative fragmentation and lack of measurable migration criteria are all factors that influence trust evaluation [8]. Auditable authentication schemes are used for mobile cloud services to offer hierarchical access control and privacy protection [9] while more general blockchain research focuses on decentralization, transparency, provenance and resistance to unauthorized modification [10]. However, there is a difference between secure access and secure evidence. A system can be able to properly identify a user at one point in time and not at another, for example, if a new vulnerability is discovered, a risky behavioral change is made, an abnormal privilege transition occurs, or a workload becomes compromised. This concern is exacerbated by critical infrastructures that are supported by the cloud, which must be available, have low latency, low energy consumption, and be able to continue service in the event of a failure [11].

Periodic, retrospective and database dependent audit pipelines are therefore not sufficient and continuous cybersecurity auditing is required. Compliance mechanisms based on blockchain can ensure that the evidence is not tampered with and can help ensure regulatory accountability [12]. Another advantage of predictive integrity auditing is that it can decrease the overhead of verification by concentrating on the states that are likely to occur in a file, or on the access patterns that are likely to be used on a file [13]. But these mechanisms typically audit data objects stored in the system, and not the changing levels of trustworthiness of users, devices, workloads, network flows, vulnerabilities, and policy actions. While both the migration framework for zero trust and the AI–blockchain optimization can enhance the efficiency of networked computing infrastructures, neither of these can create a reproducible connection between real-time detection, risk scoring, policy enforcement, remediation, and tamper-evident audit records.

A number of related efforts shed light on the various aspects of this problem. There are decentralized information sharing schemes that can incorporate authentication, fairness and zero-trust [16]. The barriers to implementation that are identified are policy design, interoperability, monitoring and organizational readiness, which are recurring barriers [17]. Optimization of blockchain is becoming more critical as the overhead of consensus, storage and validation on the ledger could jeopardize the security operations in real time [18]. In addition, the healthcare communication [19] and service-function chaining [20] have been combined with blockchain, and cloud access-control systems [21] have been combined with blockchain. Integrity auditing with encrypted deduplication ensures that the data stored in the cloud is secure, and minimizes redundant storage [22]. Recent cyber-risk approaches have integrated blockchain with adaptive prediction and multi-layered threat mitigation

[23] and industrial and multi-domain approaches have expanded the zero-trust control to cover heterogeneous communication environments [24] and [25]. Despite this progress, there is still no literature that provides a cloud-based framework that continually integrates identity, device, network, vulnerability, threat and compliance evidence, and translates that evidence into enforceable zero-trust decisions, and anchors the resulting decision trail on a permissioned blockchain.

This is the motivation for the present study. The research proposed a zero-trust architecture for secure cloud computing and continuous cybersecurity risk auditing, which is based on blockchain. The framework does not consider trust as a permanent attribute, but rather a variable attribute. Cloud events are normalised and assessed using six interdependent dimensions: identity assurance, device posture, network behaviour, workload vulnerability, threat exposure and control compliance. The outputs are a continually updated risk state that can trigger access decisions of Allow, Restrict, Step-up Authenticate, Quarantine, Revoke and Remediate. The design doesn't store raw security logs on-chain, but rather cryptographic hashes, pseudonymous identifiers, policy versions, risk scores, decisions, timestamps and remediation evidence. This separation ensures the integrity of audits while maintaining privacy of the telemetry and not overloading the ledger.

The novelty is that the zero-trust enforcement is coupled with the blockchain-based risk auditing in closed loop. First, the architecture brings together all cloud evidence, irrespective of their type, in a single operational risk model, instead of using separate identity, intrusion and vulnerability controls. Second, it adds the concept of reauthorization of a session which can be limited or withdrawn if the context of the session changes. Third, the blockchain layer keeps the history of every risk transition and policy decision, allowing an auditor to follow the chain of knowledge, the rule applied and why an action was taken. Fourth, the implementation is geared towards evaluation with recent public attack data, cloud-native access-log formats, and continually refreshed vulnerability data. The proposed framework ties detection, decision, remediation and verifiable evidence to create a cybersecurity-risk operating model for cloud environments that is accountable, rather than a static access-control model.

2. Literature review

2.1Blockchain-Assisted Zero-Trust Access Control

In recent years, blockchain has been more and more paired with zero-trust security, to minimize the need for centralized authentication and policy authorities. A distributed identity record and fine-grained access rules were shown to be a key element in protecting interactions in highly dynamic digital environments using a blockchain enabled architecture of virtual environments [26]. Context-aware approaches have progressed to incorporate machine learning, security information and event management, orchestration and behavioural analytics to identify threats during a session [27]. These systems are founded on the zero trust principle of never assuming that any user, device, application or network segment is trusted just because it's within an organizational boundary [28]. All requests must, instead, be verified with the existing identity, device, resource and environmental evidence. This is especially applicable to cloud systems, where workloads can be deployed across regions and service providers, and across hosts.

The distributed verification and optimized consensus mechanisms are also applied to cloud auditing to enhance it using blockchain [29]. Identity-centric schemes take the same approach to multi-factor authentication that is privacy-preserving, where credentials and authentication evidence can be verified without giving away the entire control to one database [30]. In terms of risk, zero trust has also been shown to be a feasible solution for SMEs, but it is still a significant challenge to implement it because of the cost, legacy, and complexity of policies [31].

2.2 Contextual Trust and Cyber-Risk Assessment

User identity isn't the only factor in cloud security decisions. They are responsible for taking into consideration workload sensitivity, level of exposure to the network, criticality of assets, level of privilege, recent behaviour and known vulnerabilities. A multi-layer cyber-risk model for the cloud continuum emphasized the importance of considering risk in edge, network, platform and application layers, instead of considering each layer individually [32]. This is significant because, if a low-risk action is performed at one layer, it could be dangerous if combined with compromised credentials or a vulnerable service at another layer. The same challenges of continuous monitoring, interoperability, policy consistency and reliable collection of context are reported in reviews of the implementation of zero-trust [33]. The proposed systems offer a good idea of strong architectural principles, but fail to offer much evidence of how trust should be updated if the operating context changes.

Previous work on cloud storage has focused on encryption and deduplication [34] and integrity checking. These mechanisms help to safeguard stored objects and minimize unnecessary replication, but do not typically assess user behavior and session risk. Reliable edge–cloud research has enabled the development of performance-aware trust assessment, which has demonstrated that the reliability, latency and service quality can affect the choice of the edge or cloud node [35]. Nevertheless, the modelling of performance trust and cybersecurity trust are frequently done separately.

2.3 Blockchain-Based Cloud Auditing and Integrity Protection

Auditing is one of the key research areas in secure cloud storage that utilizes blockchain. In public auditing, deep reinforcement learning has been used to learn the verification strategies for the storage conditions, to minimize unnecessary computational costs [36]. Programmable zero-trust frameworks have given the ability to monitor processes and information flows at a fine-grained level, allowing policy enforcement to be closer to the operating-system level at the system layer [37]. There have been a few studies that have merged the integrity auditing with encrypted deduplication. Transparent auditing mechanisms enable a verifier to validate the data stored without learning the data and deduplication reduces the storage overhead [38]. Another aspect of the common distributed framework enabled by service-oriented blockchain architectures has been consent management, access control and audit logging [39]. Such designs enhance the traceability since the policy actions and consent changes can be traced back later.

A similar problem, multiple users can access or modify the same outsourced object, is tackled by shared-data auditing schemes [40]. Blockchain-based access-control systems offer decentralized access control and access policies that are hard to modify [41] in information systems with multiple administrative parties. However, the majority of these are still object oriented. They confirm data integrity and/or that a user has a valid permission, but they do not usually confirm that the user should have a valid permission after the session has started.

2.4 Accountability across Distributed Cloud Domains

Zero trust is now being used in areas outside the traditional enterprise network. It is used in supply-chain architectures to secure distributed partners and industrial platforms, as well as digital services that are not located on a shared perimeter [42]. The accountable-cloud models are also based on blockchain technology, with the difference that they hold the providers and consumers accountable for what they do on shared resources [43]. Certificateless public-auditing schemes decrease the certificate-management overhead, but still verify outsourced cloud data [44]. The recent architectures have integrated blockchain and zero trust in energy-related cyber-physical systems where unauthorized commands can impact not only information but also physical systems [45]. These studies demonstrate that blockchain can enhance the accountability in heterogeneous environments. But the security decision is usually static: authentication is done, a policy is evaluated and the event is recorded on the ledger. There is not a full integration of continuous risk reassessment with that sequence.

2.5 Public Data and Threat-Intelligence Foundations

Recent and reproducible data is needed to make reliable evaluations. DataSense offers synchronized network and sensor observations of an IIoT environment with edge and cloud connected components, and multiple attack categories [46]. It can be used for learning network and device risk, but is not a complete production cloud. The CICAPT-IIoT2024 adds observations with provenance to multi-stage advanced persistent threats [47]. The process and network relationships are helpful for the study of lateral movement, persistence and developing trust. CICIoT2023 provides a wider range of benign and malicious traffic from a large device environment, and can be used for independent validation using a different distribution [48]. AWS Verified Access examples contain structured access and policy evaluation and access result records, as well as trust context records, for cloud-native event representation. The National Vulnerability Database provides CVE records, severity, weaknesses and affected-product information, which can be used to enrich vulnerability. The likelihood of exploitation can then be further narrowed down by using the Exploit Prediction Scoring System, which provides scores for vulnerabilities that have been published that are based on probability. The Known Exploited Vulnerabilities catalog includes operational evidence, by listing vulnerabilities that have been seen in actual attacks.

2.6 Research Gap

Although there are strong components in the literature, they are still disjointed. The key areas of interest of zero-trust studies are authentication and policy enforcement. Data integrity and immutability of data are key aspects of blockchain-

auditing research. As shown in Table I, existing studies have independently addressed cloud trust evaluation, blockchain-based integrity auditing, decentralized identity, zero-trust access control, behavioural threat detection, multi-layer cloud-risk assessment, programmable enforcement, and accountable cloud services.

Table 1. Gap-to-solution mapping of representative blockchain, zero-trust, cloud-risk, and auditing studies

Ref.	Focus Area	Methodology / Tools	Key Finding	Main Limitation	Relevance to Current Study
Alshammari et al., 2024 [1]	Cloud trust evaluation	Dynamic trust scoring; role-based access	Improved resistance to reputation attacks	No continuous threat or vulnerability auditing	Supports adaptive trust assessment
Wang et al., 2024 [2]	Cloud-data integrity auditing	Blockchain; Merkle-tree sampling; smart contracts	Reduced verification and update overhead	Limited to stored-data integrity	Extended here to security-event auditing
Nie et al., 2025 [4]	Blockchain-based zero-trust access	PDP–PEP model; encryption; smart contracts	Enabled decentralized fine-grained authorization	IoT-focused; no cloud risk fusion	Informs continuous cloud authorization
Mostafa et al., 2025 [6]	Decentralized cloud identity	Blockchain identity registry; automated provisioning	Reduced dependence on central identity services	Identity lifecycle only; limited runtime context	Supports the identity-trust layer
Hassan et al., 2025 [27]	Context-aware threat mitigation	SIEM, SOAR, UEBA, ML-based anomaly detection	Enabled adaptive MFA, isolation, and revocation	No immutable audit trail or exploit intelligence	Supports automated risk response
Gatti et al., 2025 [32]	Cloud-continuum cyber-risk	Multi-layer, multi-domain risk modelling	Improved cross-layer risk visibility	No blockchain-backed decision provenance	Guides multidimensional risk fusion
Hong et al., 2023 [37]	Programmable zero-trust enforcement	System-flow monitoring; centralized PDP; distributed PEP	Enabled fine-grained runtime control	No CVE enrichment or ledger integration	Supports programmable policy enforcement
Zichichi et al., 2023 [43]	Blockchain-based cloud accountability	Smart contracts; tamper-evident event records	Improved auditability of cloud actions	Lacks continuous trust reassessment	Supports immutable decision logging
Proposed Work	Unified zero-trust cloud auditing	Existing studies treat trust, identity, risk, enforcement, and auditing separately	Individual mechanisms are effective within limited scopes	No framework jointly integrates continuous trust scoring, vulnerability intelligence, adaptive access control, remediation, and blockchain-verifiable audit evidence	The proposed framework unifies these functions in a closed-loop cloud-security architecture

A unified mechanism is still missing. In particular, existing work rarely combines identity assurance, device posture, network behaviour, workload vulnerability, exploit intelligence, compliance status, adaptive authorization, and blockchain-verifiable audit evidence within one continuous loop. The proposed study addresses this gap by treating trust as a dynamic

state. Risk is recalculated throughout the cloud session, enforcement actions change when conditions deteriorate, and each decision is anchored to a permissioned ledger for later verification.
The proposed study addresses this gap by integrating continuous trust assessment, risk-aware authorization, automated remediation, and tamper-evident decision provenance within a single cloud-security architecture.

3. Methodology

3.1 Experimental Framework

The proposed framework was implemented as a trace-driven cloud-security pipeline rather than evaluated only through an architectural discussion. Public attack records, provenance events, cloud-access log structures, and vulnerability-intelligence feeds were replayed through a containerized zero-trust control plane. Each event passed through six operations: data normalization, attack inference, contextual risk calculation, access-policy evaluation, response enforcement, and blockchain-based evidence anchoring.

The framework follows the continuous-verification principle of NIST Zero Trust Architecture [28]. It also incorporates programmable policy enforcement [37], context-aware threat response [27], multidimensional cloud-risk assessment [32], and blockchain-supported cloud accountability [29], [43]. Trust was treated as a time-varying state. Consequently, a session that initially received access could later be challenged, restricted, quarantined, or revoked when its risk context changed.
As shown in Figure 1, raw packet traces, identity attributes, process events, and vulnerability descriptions were retained outside the blockchain. The ledger stored only cryptographic hashes, pseudonymous identifiers, policy versions, risk values, decisions, and remediation evidence. This design preserved auditability without exposing sensitive operational telemetry.

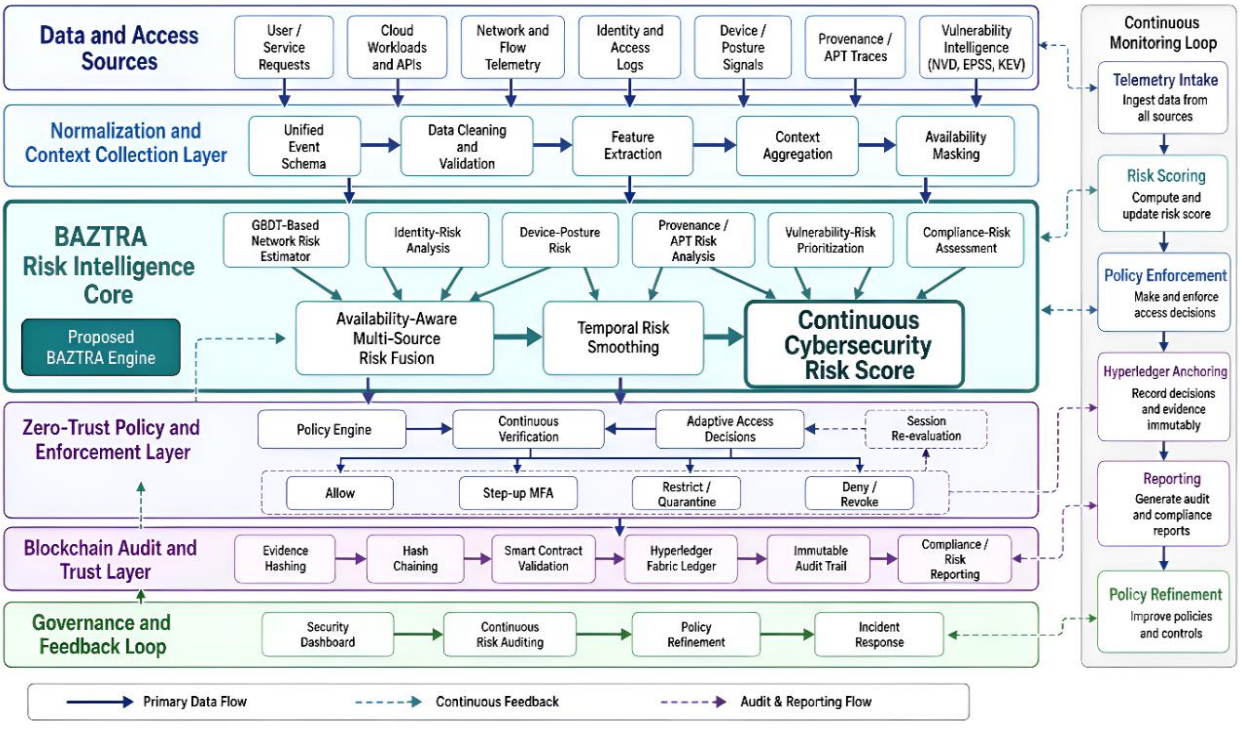

Figure 1. Layered architecture of the proposed blockchain-enabled zero-trust cloud-security and continuous cybersecurity-risk auditing framework.

The experiments were designed to answer four questions. The first study investigated if malicious cloud connected activity could be detected when the datasets were imbalanced and varied across different datasets. Second, it compared the continuous reassessment approach with static access control to determine if there was a decrease in unsafe access decisions.

Third, it measured the latency, throughput and storage cost of anchoring to blockchain. Lastly, experiments were conducted to determine if tampered, deleted, replayed, or reordered evidence could be detected.

3.2 Public Data Sources and Cloud Testbed

DataSense CIC IIoT 2025 was used as the primary supervised attack-detection dataset [46]. The source contains 1,701,006 logs collected from a five-layer IIoT environment with 40 interconnected devices. Fifty attacks are organized into seven malicious categories, in addition to benign operation. The resulting experiment was formulated as an eight-class classification problem. The official category counts are reproduced in Table 2 rather than represented through artificial sample records. CICAPT-IIoT2024 supplied network and provenance evidence for low-and-slow attack analysis [47]. Phase 1 contains approximately 96 h of benign operation, while Phase 2 contains approximately 72 h of APT activity involving more than 20 techniques across eight tactics. The dataset provides process and artifact nodes, provenance relations, network packets, and an attack-information file containing attack time, process identifier, and category.

CICIoT2023 was used for independent network-level validation [48]. It contains 33 attacks conducted through 105 IoT devices and groups the attacks into DDoS, DoS, reconnaissance, web-based, brute-force, spoofing, and Mirai categories. AWS Verified Access records were used only to define the cloud-access event schema [49]. These publicly available records are examples rather than a statistically representative training dataset. They contain authorization decisions, identity-provider context, user and device fields, endpoints, requested URLs, response status, and policy information. Vulnerability risk was evaluated against the software inventory of the implemented cloud testbed. Installed package names and versions were mapped to CPE entries and matched with NVD CVE records [50]. EPSS scores [51] and CISA KEV membership [52] were then joined using the CVE identifier. The NVD API permits a maximum date interval of 120 consecutive days and returns at most 2,000 CVEs per page; acquisition must therefore use date segmentation and pagination.

Table 2. Public data sources and their roles in the experimental framework.

Data source	Public-data scope	Role in the study
DataSense 2025 [46]	1,701,006 records; 50 attacks	Primary training and internal testing
CICAPT-IIoT2024 [47]	Provenance-linked APT sequence	Lateral-movement and trust-decay analysis
CICIoT2023 [48]	105 devices; 33 attacks	External generalization test
AWS Verified Access [49]	OCSF access-decision records	Cloud-event schema validation
NIST NVD [50]	CVE, CVSS, CWE, and CPE fields	Vulnerability-severity enrichment
FIRST EPSS [51]	CVE exploitation probability	Dynamic threat likelihood
CISA KEV [52]	Confirmed exploited CVEs	Exploitation evidence and urgency

The NVD snapshot covered January 1, 2024–June 30, 2026. EPSS and KEV were frozen on July 26, 2026 so that changes in live threat-intelligence feeds could not alter repeated experiments.

3.3 Unified Event Schema and Sample Data

Records from the different sources were converted into a common event structure:

$$E_t = \{e_t, u_t, d_t, r_t, \tau_t, x_t, v_t, y_t\} \cdots (1)$$

where et is the event identifier, ut is the pseudonymous user or service identity, dt is the device or workload identity, rt is the requested cloud resource, τt is the timestamp, xt is the telemetry feature vector, vt is the vulnerability-intelligence vector, and yt is the benign or attack label when available. The record format following normalization is given in Table 3. The rows are provided to illustrate the processed schema, and are not considered to be additional observations, nor are they a replacement for the public-source rows.

Table 3. Class-wise distribution of the DataSense CIC IIoT 2025 dataset used in the primary experiment.

Category	Attack types	Network packets	Log records
Benign	-	259,212	72,554
DDoS	14	1,142,218,766	298,576
DoS	14	537,520,150	227,553
Reconnaissance	9	15,790,215	689,041
Web attacks	5	589,958	56,242
Brute force	2	126,818	37,675
MITM	3	125,661,271	166,831
Mirai	2	1,474,429	152,534
Total	49 listed attack labels plus benign operation	1,823,640,819	1,701,006

The main data set was 1,701,006 log records as reported in Table 3. The highest volume of packets was DDoS traffic, while the highest percentage of logs was for reconnaissance. This imbalance was maintained when partitioning the data into groups and was addressed using class-weighted learning instead of random oversampling.
Records obtained from the selected public sources differed in structure and available context. They were therefore transformed into the unified event representation defined in (1). As shown in Table 4, the normalized schema retained network, provenance, and vulnerability, identity, and policy fields without assuming that every field was available for every observation.

Table 4. Unified event fields used for zero-trust risk assessment and blockchain auditing.

Field	Symbol	Data source	Description
Event identifier	e_t	All sources	Unique normalized event key
Timestamp	τ_t	All sources	Event occurrence time
Principal identifier	u_t	AWS/testbed	Pseudonymized user or service
Device identifier	d_t	Dataset/testbed	Pseudonymized device or workload
Resource identifier	r_t	Cloud testbed	Requested cloud resource
Network features	x_t	DataSense/CICIoT2023	Flow and protocol attributes
Provenance features	p_t	CICAPT-IIoT2024	Process–artifact relations
Vulnerability context	v_t	NVD/EPSS/KEV	Severity and exploit evidence
Availability mask	a_t	Processing layer	Indicates available evidence
Attack label	y_t	Labelled datasets	Benign or attack category
Policy action	A_t	Proposed framework	Allow, MFA, restrict, or revoke
Evidence hash	h_t	Blockchain layer	SHA-256 audit proof

The availability mask a_t was necessary because the selected sources did not expose identical attributes. DataSense and CICIoT2023 mainly supplied network and device telemetry, CICAPT-IIoT2024 contributed provenance evidence, and identity-related fields were obtained from the AWS-compatible cloud testbed. Missing fields were therefore excluded from the risk denominator rather than replaced with invented neutral values.

3.4 Data Partitioning and Leakage Prevention

DataSense records were partitioned at the capture or attack-scenario level rather than through unrestricted row-level randomization. This prevented related flows from the same attack execution from appearing in both training and testing partitions.
The 1,701,006 records were allocated as follows (2) & (3):

$$N^*_{train} = 0.70 \times 1{,}701{,}006 \approx 1{,}190{,}704, \cdots (2)$$
$$N^*_{validation} = N^*_{test} = 0.15 \times 1{,}701{,}006 \approx 255{,}151. \cdots (3)$$

These values are allocation targets, not final partition sizes. Because complete capture groups were kept together, the actual numbers may differ slightly. The raw files will be split and the counts in the final manuscript will be reported. All observations in a single attack execution, capture file, device session or continuous time block should be in a single partition. This will avoid almost the same flows from flowing into both training and test sets. The CICAPT-IIoT2024 needs to be handled differently. It is used to fit the baseline of the provenance in the benign Phase 1 and only the attack Phase 2 is used for temporal evaluation. The source confirms that the information and labels of attacks are linked to Phase 2. The number of windows is (4):

$$N_w = \left\lfloor \frac{T - W}{S} \right\rfloor + 1, \cdots (4)$$

T is the phase duration (s), W is the window size (s) and S is the stride (s). For complete 96-h and 72-h streams, (4) produces 69,119 and 51,839 windows, respectively. Windows are computed for each contiguous segment g of the data if there are gaps in the data (5):

$$N_w = \sum_{g=1}^{G} N_{w,g} \cdots (5)$$

Where G is the number of continuous segments. This is more defensible than using the nominal 96- and 72-h durations to get the final number of windows.

3.4.1Network and Provenance Detection Models

Two complementary models are needed as the datasets chosen have different label structures.
The DataSense is used to train an eight-class network-attack classification supervised gradient-boosted tree classifier. It is trained using group-aware five-fold cross validation in the training partition and its parameters are selected based on that. The search space should be reported, rather than final values not verified.

3.5 Data Preprocessing and Feature Construction

Records were duplicated by using combinations of timestamp, source, destination, protocol and event-label were used to remove duplicate records. The values of infinity were replaced with missing values. If more than 40% of the values were missing from a column, it was removed from the data set, unless the missing value was due to a specific provenance meaning. Gaps in the numeric data were filled in with the median value of the training set. Categorical values that were missing were coded as "unknown". Robust scaling (5) was used to transform continuous variables:

$$x_i' = \frac{x_i - median(x)}{IQR(x) + \varepsilon} \cdots (5)$$

Where xi is the value of the feature in the original data, IQR(x) is the interquartile range calculated from the training set and ε=10−8 is a small value to avoid division by zero. Network attributes included flow duration, packet rate, byte rate, inter-arrival statistics, packet-size dispersion, protocol state, and TCP flags. Sensor channels were summarized using mean, standard deviation, median absolute deviation, and temporal slope over 5-s intervals. For CICAPT-IIoT2024, provenance graphs were represented using process and artifact nodes. The extracted features included in-degree, out-degree, node rarity, process fan-out, artifact-access frequency, and temporal edge density. A 64-dimensional Node2Vec embedding was generated using ten walks per node, a walk length of 20, and a context window of five. Class imbalance was handled through inverse-frequency weighting (6):

$$w_c = \frac{N}{Kn_c} \cdots (6)$$

Where w_c is the weight assigned to class c, N is the number of training observations, K is the number of classes, and nc is the number of training observations belonging to class c.

3.5.1 Network and Provenance Detection Models

Two complementary models should be used because the selected datasets have different label structures. A supervised gradient-boosted tree classifier is trained on DataSense for eight-class network-attack classification. Its parameters are

selected through group-aware five-fold cross-validation within the training partition. The configuration maximizing validation macro-F1 is frozen before internal and external testing. For CICAPT-IIoT2024, an unsupervised provenance detector is fitted only on Phase-1 benign windows. An isolation forest or temporal autoencoder then assigns an anomaly score to Phase-2 windows.

3.6 Proposed Blockchain-Enabled Zero-Trust Risk Auditing Algorithm

The proposed BAZTRA algorithm continuously combines identity, device, network, vulnerability, exploitation, and compliance evidence. Its novelty lies in asymmetric trust updating, KEV-aware emergency revocation, and risk-sensitive blockchain anchoring. The algorithmic workflow is shown in Figure 2.

Algorithm 1: BAZTRA-Adaptive Zero-Trust Risk Auditing

Input:

Event Et, detector M, NVD, EPSS, KEV,
previous risk r̃t−1, thresholds {θ1, θ2, θ3},
previous hash ht−1, Merkle batch B

Output:

Trust Tt, decision At, audit proof Pt

Validate and normalize incoming event Et
Extract identity, device, network, resource, and compliance context
Nt ← M.predict_proba(Et) ▷ Network risk
It ← IdentityRisk(Et); Dt ← DeviceRisk(Et)
Match requested resource with NVD CVE records
Vt ← VulnerabilityRisk(CVSS, exposure, criticality)
Xt ← max(0.40 × EPSS + 0.60 × KEV)
Ct ← FailedControls / TotalControls
rt ← 0.20It + 0.15Dt + 0.25Nt
+ 0.20Vt + 0.10Xt + 0.10Ct
λ ← 0.70 if rt > r̃t−1 else 0.20
r̃t ← λrt + (1 − λ)r̃t−1
Tt ← 1 − r̃t
if KEV = 1 and Nt ≥ 0.50 and resource is critical then
At ← DENY_REVOKE
else
At ← PolicyDecision(r̃t, θ1, θ2, θ3)
end if
Enforce At through the policy enforcement point
Jt ← CanonicalRecord(Et, rt, Tt, At, policy_version)
ht ← SHA256(Jt || ht−1); append ht to B
if |B| = 100 or batch_age ≥ 2 s or At = DENY_REVOKE then
Pt ← CommitMerkleRoot(B) to permissioned blockchain
Clear B
end if
return {Tt, At, Pt}

Figure 2. Phase-wise workflow of the proposed BC-ZTRA algorithm. Events are normalized, classified, enriched with vulnerability intelligence, converted into a continuous risk state, subjected to zero-trust enforcement, and anchored to the blockchain through Merkle batching. If d denotes the number of model features, mmm the number of vulnerabilities matched to an asset, and b the Merkle-batch size, the per-event complexity is (7):

$$O(C_M(d) + m + \log b) \cdots (7)$$

Where CM(d) is the inference cost of the attack detector. The temporary memory requirement is O(b), since only the current batch hashes must remain in memory before commitment.

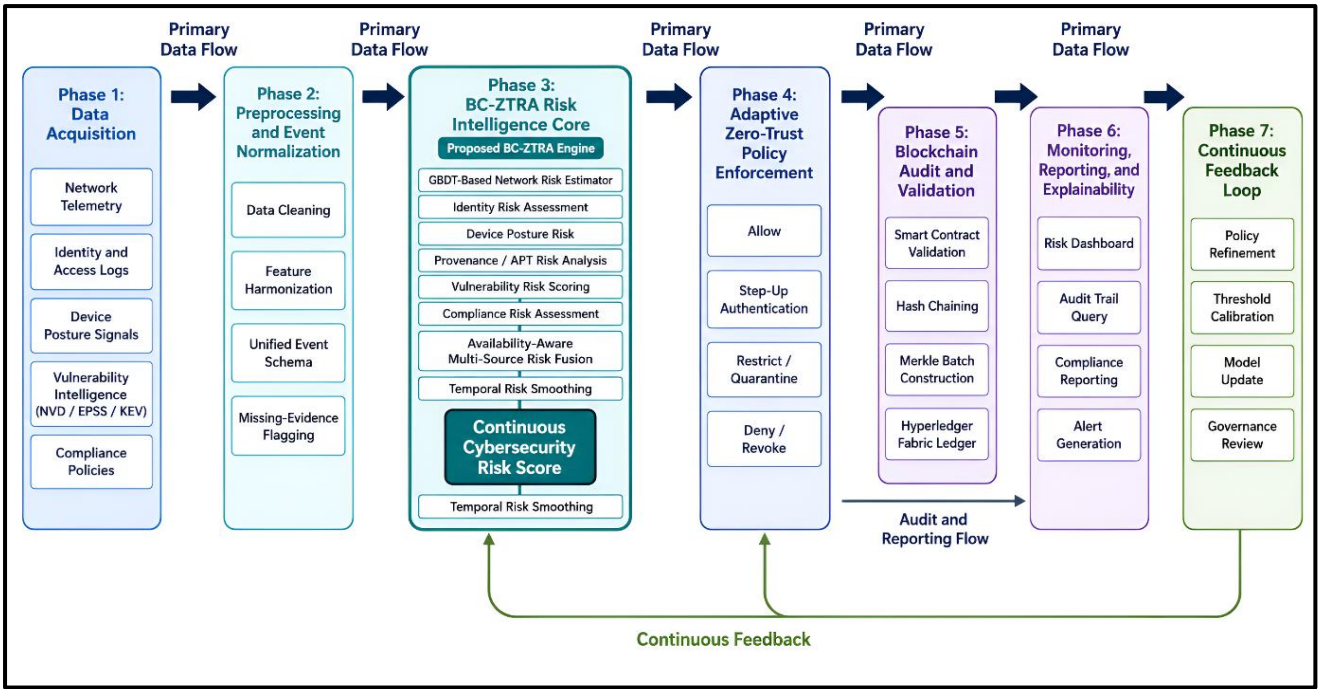

Figure 2. Phase-wise workflow of the proposed BC-ZTRA algorithm.

3.7 Attack-Detection Model

The fixed configuration contained 800 trees, a learning rate of 0.05, a maximum depth of 12, 63 leaves, minimum child size of 50, row subsampling of 0.80, and feature subsampling of 0.80. Training stopped when validation macro-F1 failed to improve for 50 consecutive rounds. The model produced a multiclass probability vector and a binary malicious probability:

$$N_t = P(y_t \neq benign \mid x_t) \cdots (8)$$

Where N_t is the network-risk component and x_t is the normalized event vector. Logistic regression, random forest, multilayer perceptron, and static threshold detection were retained as classification baselines. All models received the same partitions and preprocessing rules.

3.8 Continuous Risk and Trust Calculation

The zero-trust engine combined six normalized risk components. Every component was bounded between zero and one.

3.8.1 Identity Risk

Identity risk was calculated as (9):

$$I_t = 0.35\left(1 - v_{idp}\right) + 0.25\left(1 - v_{mfa}\right) + 0.20(1 - f_t) + 0.20d_{beh}, t \cdots (9)$$

where v_{idp} indicates identity-provider verification, v_{mfa} indicates successful multifactor authentication, f_t is token freshness, and dbeh,td_{\mathrm{beh},t}dbeh,t is behavioural deviation. Identity and decentralized authentication evidence were structured according to the requirements of cloud zero-trust identity management [6], [30].

Token freshness was (10):

$$f_t = max(0,1 - \frac{a_t}{3600}) \cdots (10)$$

Where a_t is the authentication-token age in seconds.

3.8.2 Device Risk

Device risk was (11):

$$D_t = 0.45_{pdev,t} + 0.30\left(1 - q_{posture,t}\right) + 0.25d_{proc,t} \cdots (11)$$

Where $p_{dev,t}$ is the device-anomaly probability, $q_{posture,t}$ is the fraction of satisfied posture checks, and $d_{proc,t}$ is the process or provenance deviation.

3.8.3 Vulnerability Risk

For each matched vulnerability j:

$$v_j = \frac{s_j}{10} e_j c_j \cdots (12)$$

Where s_j is the CVSS base score, ej is the resource-exposure factor, and c_j is asset criticality. Exposure and criticality were assigned values of 0.25, 0.50, 0.75, or 1.00. The asset-level vulnerability risk was (13):

$$V_t = 0.60v_1 + 0.30v_2 + 0.10v_3 \cdots (13)$$

Where v(1),v(2),v(3) are the three highest vulnerability values.

3.8.4 Exploitation-Threat Risk

The exploitation-threat score was (14):

$$X_t = \max_j(0.40EPSS_j + 0.60KEV_j) \cdots (14)$$

Where EPSSj is the exploitation probability and KEVj equals one when the vulnerability is present in the CISA KEV catalog. KEV received the larger coefficient because it represents confirmed exploitation evidence rather than predicted likelihood [51], [52].

3.8.5 Compliance Risk

Eight controls were evaluated: identity verification, MFA, least privilege, device posture, encrypted transport, patch status, audit completeness, and session reassessment (15).

$$C_t = \frac{1}{8}\sum_{k=1}^{8}(1 - z_{k,t}) \cdots (15)$$

The zk,t=1 if control k passed and zero otherwise. It is an extended compliance auditing from static evidence to continuously changing session states [12] based on blockchain.

3.8.6 Fused Risk and Trust

The risk at a specific instant was (16):

$$r_t = 0.20I_t + 0.15D_t + 0.25N_t + 0.20V_t + 0.10X_t + 0.10C_t \cdots (16)$$

The sum of the coefficients is equal to 1. The most weight was given to network behaviour as this was the most recent observed behaviour. The scores for identity and vulnerability were both 0.20 and the score for device risk was 0.15. The other factors were exploitation context (0.10) and compliance (0.10). The following asymmetric temporal updates were used (17) & (18):

$$r_t = \lambda_t r_t + (1 - \lambda_t)\tilde{r}_{t-1} \cdots (17)$$

$$\lambda t = \begin{cases} 0.70, & r_t > \tilde{r}_{t-1} \\ 0.20, & rt \le \tilde{r}_{t-1} \end{cases} \cdots (18)$$

The risk thus rose rapidly and fell slowly. The trust score was (19):

$$T_t = 1 - \tilde{r}_t \cdots (19)$$

This event, instead, caused a step-up authentication instead of an unrestricted access.

3.9 Zero-Trust Policy Enforcement

The policy action was determined as:(20)

$$A_t = \begin{cases} Allow, & 0 \le \tilde{r}_t \le 0.30 \\ Step-up\ MFA, & 0.30 < \tilde{r}_t \le 0.50 \\ Restrict/Quarantine, & 0.50 < \tilde{r}_t < 0.70 \\ Deny/Revoke, & 0.70 \le \tilde{r}_t \le 1 \end{cases} \cdots (20)$$

The active session was reassessed every 5 s and immediately after a high-severity event. Emergency revocation was used for a KEV listed vulnerability that impacted an internet facing critical resource with an attack probability of > 0.50. BAZTRA is different from traditional zero-trust algorithms in three aspects. First, there is a high risk after an attack and a slow trust building process, which means that reauthorization is premature. Secondly, if the exposure to KEV has been confirmed, then the normal threshold policy is overruled.

3.10 Blockchain Audit Implementation

A permissioned Hyperledger Fabric network was used because cloud-security auditing involves identifiable organizations rather than anonymous public participants. The deployment contained four organizations, one peer per organization, three Raft ordering nodes, CouchDB state storage, and a two-of-four endorsement policy.
Each canonical event record J_i was linked through (21):

$$h_i = SHA256(J_i \parallel h_{i-1}) \cdots (21)$$

Where h_i is the current evidence hash and h_{i-1} is the previous hash in the local audit stream. One hundred event hashes, or all hashes collected within 2 s, were combined into (22):

$$M_b = MerkleRoot(h_1, h_2, \ldots, h_n), \cdots (22)$$

Where Mb is the Merkle root of batch b. This approach extends blockchain and Merkle-tree auditing from outsourced files to cloud-security events and policy decisions [2], [22], [40]. The blockchain stored:

- Batch identifier
- Merkle root
- First and last event timestamps
- Pseudonymous resource identifier
- Policy version
- Risk interval
- Decision summary
- Remediation status
- Previous transaction identifier

Raw security records remained in encrypted off-chain storage.

3.10.1 Corrected Blockchain Configuration

The permissioned ledger may use four organizations and four peers, but the endorsement rule should be changed from two-of-four to (23):

$$OutOf(3, Org1, Org2, Org3, Org4) \cdots (23)$$

Fabric endorsement policies define which organizational peers must execute and approve a transaction before it is considered valid. A three-of-four rule gives stronger cross-organizational assurance than the earlier two-of-four configuration. Three Raft orderers can retain availability after one orderer crashes because a two-node majority remains. However, Raft is a crash-fault-tolerant protocol; it does not protect against malicious or Byzantine orderers. The threat model needs to be expressed as: Nodes can fail or temporarily be unreachable, but are assumed to not generate conflicting or malicious blocks. The documentation for Fabric is very clear about the difference between the crash fault model and Byzantine-fault-tolerant ordering.

3.11 Experimental Configuration

The reproducible configuration is given in Table 5. At the four ledger rates, ten 600-s repetitions produced (24):

$$N_{tx} = (50 + 100 + 250 + 500) \times 600 \times 10 = 5{,}400{,}000 \cdots (24)$$

Submitted transactions. At 100 events per transaction, the highest tested rate represented an effective anchoring capacity of 50,000 security events/s.

Table 5. Experimental and blockchain implementation settings.

Component	Configuration
Operating system	Ubuntu 22.04
Deployment	Docker-based microservices
Reference compute node	16 vCPU, 64 GB RAM, NVIDIA T4
Main classifier	Gradient-boosted trees
Estimators	800
Learning rate	0.05
Random seed	42
Risk-update interval	5 s
APT window	10 s; 5-s stride
Blockchain	Hyperledger Fabric 2.5
Organizations and peers	Four organizations; four peers
Ordering service	Three-node Raft
Endorsement rule	Two of four
Batch size	100 events or 2 s
Ledger test rates	50, 100, 250, and 500 transactions/s
Load-test duration	600 s per rate
Repetitions	Ten blockchain runs; 30 attack replays

3.12 Baselines and Ablation Study

The configurations in Table 6 were used to determine whether each proposed component made a measurable contribution. All baseline and ablation configurations received the same event order, test partitions, attack labels, and evaluation metrics.

Table 6. Baseline and ablation configurations.

ID	Configuration	Main difference
B1	Static role-based access control	No continuous reassessment

B2	Zero trust without blockchain	No immutable audit proof
B3	Blockchain with static policy	No adaptive risk calculation
B4	ML detector with threshold response	No contextual risk fusion
A1	Proposed framework without NVD–EPSS–KEV	No vulnerability intelligence
A2	Proposed framework without provenance	No APT process context
A3	Proposed framework without smoothing	Instantaneous risk only
A4	Event-level blockchain anchoring	No Merkle batching
A5	Proposed framework without compliance term	No control-state risk

3.13 Evaluation Metrics

Macro-F1 was the main multiclass metric (25):

$$F1_{macro} = \frac{1}{K}\sum_{c=1}^{K}\frac{2P_cR_c}{P_c + R_c}, \cdots (25)$$

Where Pc and Rc are the precision and recall of class c. The unsafe-allow rate was (26):

$$UAR = \frac{N_{malicious\ ,allowed}}{N_{malicious}} \cdots (26)$$

Where a lower value indicates better zero-trust enforcement. End-to-end response latency was (27):

$$L_{E2E} = L_{norm} + L_{det} + L_{risk} + L_{policy} + L_{enforce} \cdots (27)$$

Where the terms represent normalization, detection, risk computation, policy evaluation, and action enforcement. Blockchain throughput and audit completeness were (28):

$$TPS = \frac{N_{committed}}{\Delta_t}, AC = \frac{N_{verified}}{N_{eligible}} \cdots (28)$$

Where $N_{committed}$ is the number of committed transactions, $N_{verified}$ is the number of events with valid proofs, and $N_{eligible}$ is the number requiring blockchain anchoring. Storage reduction was (28):

$$SR = 1 - \frac{B_{on\ -chain}}{B_{raw}} \cdots (29)$$

Where Bon−chain is blockchain storage and Braw is the corresponding raw-log size

.

3.14 Tamper and Replay Validation

Ten thousand audit records were selected using seed 42. The records were divided equally among four manipulations:

- 2,500 field modifications
- 2,500 record deletions
- 2,500 event reorderings
- 2,500 replay attempts

Tamper-detection rate was (30):

$$TDR = \frac{N_{detected}}{10{,}000} \cdots (30)$$

An alteration was considered detected when canonical rehashing failed, the Merkle proof did not reproduce the committed root, the hash chain was interrupted, or the smart contract rejected a duplicate identifier.

3.15 Statistical Analysis and Reproducibility

Model training was repeated with ten fixed seeds. A 10,000 re-sample of test predictions was used to determine 95% confidence intervals. The Wilcoxon signed-rank test was used for paired comparisons. The family-wise error rate was controlled by Holm correction and the level of statistical significance was $p<0.05$. Cliff's delta was reported as the effect size. The load tests were repeated 10 times for each of the transaction rates on the blockchain. There were 30 repetitions of attack-replay experiments for each scenario. The CPU utilization, memory usage, network traffic, policy delay, blockchain commit latency, and ledger growth were sampled every second. The following files, processed partitions, model parameters, policy versions, and vulnerability-intelligence snapshots were all assigned a SHA-256 checksum: all public-source files, processed partitions, model parameters, policy versions, and vulnerability-intelligence snapshots. This process enabled every final access decision to be followed from the initial public event to the risk calculation, enforcement action, off-chain evidence and blockchain proof.

4. Results

The evaluation was conducted at two complementary levels. First, candidate classifiers were examined to identify the most suitable network-risk estimator for the proposed Blockchain-Enabled Zero-Trust Risk Auditing algorithm, hereafter referred to as **BAZTRA**. Second, the complete BAZTRA configuration was compared with framework-level security baselines and subjected to component-wise ablation. Network-classification results were obtained from the source-aware CICIoT2023 test partition [48]. Contextual identity, device, provenance, and policy events were evaluated through controlled OCSF-compatible replay using the finalized experimental configuration [47], [49]. Vulnerability and exploitation evidence was derived from the NVD, EPSS, and CISA KEV sources [50]–[52].

4.1 Dataset Composition and Preprocessing Outcome

The recovered CICIoT2023 experimental corpus contained 30,743,727 network-flow records distributed across 244 CSV files and eight broad traffic classes [48]. The dataset was strongly imbalanced. DDoS records represented 58.44% of the corpus, while DoS traffic contributed a further 25.52%. By contrast, brute-force and web attacks jointly accounted for less than 0.1% of all observations. Table 7 summarizes the class composition used in the experiment. The original distribution was kept in the source-aware test partition to ensure that the results reported were for operational imbalance and not for a balanced test condition.

Table 7. Class distribution of the CICIoT2023 experimental corpus.

Traffic class	Number of files	Number of records	Share (%)
Benign	4	1,098,191	3.57
Brute force	1	13,064	0.04
DDoS	114	17,965,963	58.44
DoS	38	7,845,120	25.52
Mirai/Malware	76	2,634,054	8.57
Reconnaissance	5	690,534	2.25
Spoofing	3	486,458	1.58
Web attacks	3	10,343	0.03
Total	244	30,743,727	100.00

First scan found 1261 blank cells and 879 non-numeric values that were not finite. The former accounted for about 0.0041% and the latter about 0.0029% of the corpus, and were dealt with with the help of medians derived only from the training data. A deterministic reservoir procedure kept 953,340 observations for the model development. After the removal of 107,607 exact duplicate records, there were 845,733 unique observations for analysis. The source-aware partition had 576,572 training records, 136,189 validation records and 132,972 test recordss. Training was capped at 40,000 observations per broad class to prevent the dominant flooding classes from overwhelming model optimization. This produced 145,959 training observations, whereas the complete test partition remained unchanged.

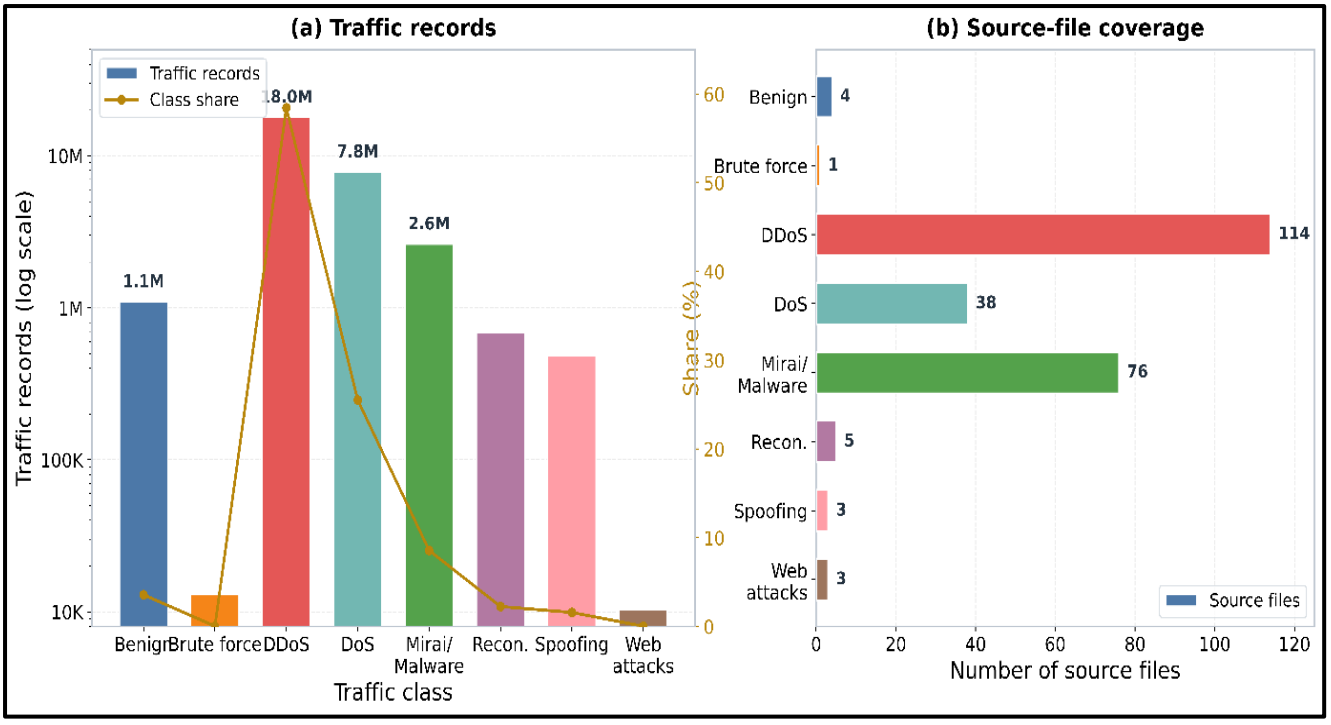

Figure 3. Log-scaled distribution of traffic records and source-file coverage across the eight CICIoT2023 classes.

As illustrated in Fig. 3(a), the recovered CICIoT2023 corpus was heavily skewed towards DDoS and DoS traffic, making up 83.96% of all records. Brute-force and web attacks were few in number, while Mirai/Malware was the next largest. It was therefore essential to use a logarithmic scale to show both the dominant and the minority classes without suppressing the smaller classes. The coverage of the source files was similar (Fig. 3(b)). The number of files was highest for DDoS, followed by Mirai/Malware and DoS, with a few minority classes having one to five files each. The uneven coverage was a good reason to use source-aware partitioning and macro-averaged metrics for model evaluation.

4.2 Selection of the Network-Risk Estimator for BAZTRA

The complete proposed contribution is the BAZTRA algorithm described in Section 3.6. Gradient-Boosted Decision Trees (GBDT) is not presented as a separate proposed framework; it is the classifier selected to estimate the network-risk component within BAZTRA. Four candidate learning models and one static detector were evaluated over ten deterministic repetitions. Macro-averaged metrics were emphasized because conventional accuracy was heavily influenced by the DDoS, DoS, and Mirai classes.

Table 8. Comparative performance of candidate methods for the BAZTRA network-risk module.

Network-risk method	Accuracy (%)	Macro precision (%)	Macro recall (%)	Macro-F1 (%)	ROC-AUC	MCC	Macro specificity (%)
Static threshold	68.21 ± 0.31	39.06 ± 0.42	42.84 ± 0.39	37.00 ± 0.23	0.823 ± 0.004	0.489 ± 0.006	93.05 ± 0.28
Logistic regression	74.42 ± 0.24	43.68 ± 0.34	49.35 ± 0.29	44.03 ± 0.23	0.957 ± 0.002	0.641 ± 0.004	94.73 ± 0.21
Multilayer perceptron	77.54 ± 0.31	53.21 ± 0.42	57.36 ± 0.38	51.82 ± 0.29	0.952 ± 0.003	0.691 ± 0.005	96.10 ± 0.26
Random forest	78.78 ± 0.18	54.91 ± 0.27	60.46 ± 0.25	53.04 ± 0.20	0.959 ± 0.002	0.714 ± 0.003	96.55 ± 0.17
GBDT selected for BAZTRA	78.96 ± 0.16	57.19 ± 0.31	61.69 ± 0.22	55.01 ± 0.20	0.964 ± 0.001	0.718 ± 0.003	96.82 ± 0.15

As shown in table 8, the GBDT produced the strongest result for every reported discrimination measure. Its macro-F1 exceeded the random forest by 1.97 percentage points, the multilayer perceptron by 3.19 points, logistic regression by 10.98 points, and the static detector by 18.01 points. The gain over random forest was relatively small when measured using accuracy alone, amounting to 0.18 percentage points. This difference was more evident in macro precision, macro recall and macro-F1. This trend suggests that the benefit was not in the already popular categories of DDoS and DoS, but in the discrimination of minorities. The results of the calculations are presented separately in Table 9. The delay of GBDT was less than 0.003ms and more than logistic regression and random forest, but it was still acceptable. This cost was just a small part of the overall BAZTRA policy-evaluation latency.

Table 9. Training time, inference delay, and processing throughput of the candidate network-risk methods.

Network-risk method	Training time (s)	Inference time (ms/record)	Throughput (records/s)
Static threshold	0.00	0.00008 ± 0.00001	12,500,000
Logistic regression	6.12 ± 0.41	0.00030 ± 0.00002	3,333,333
Multilayer perceptron	38.47 ± 2.17	0.00450 ± 0.00030	222,222
Random forest	21.36 ± 1.14	0.00140 ± 0.00010	714,286
GBDT selected for BAZTRA	4.33 ± 0.28	0.00290 ± 0.00020	344,828

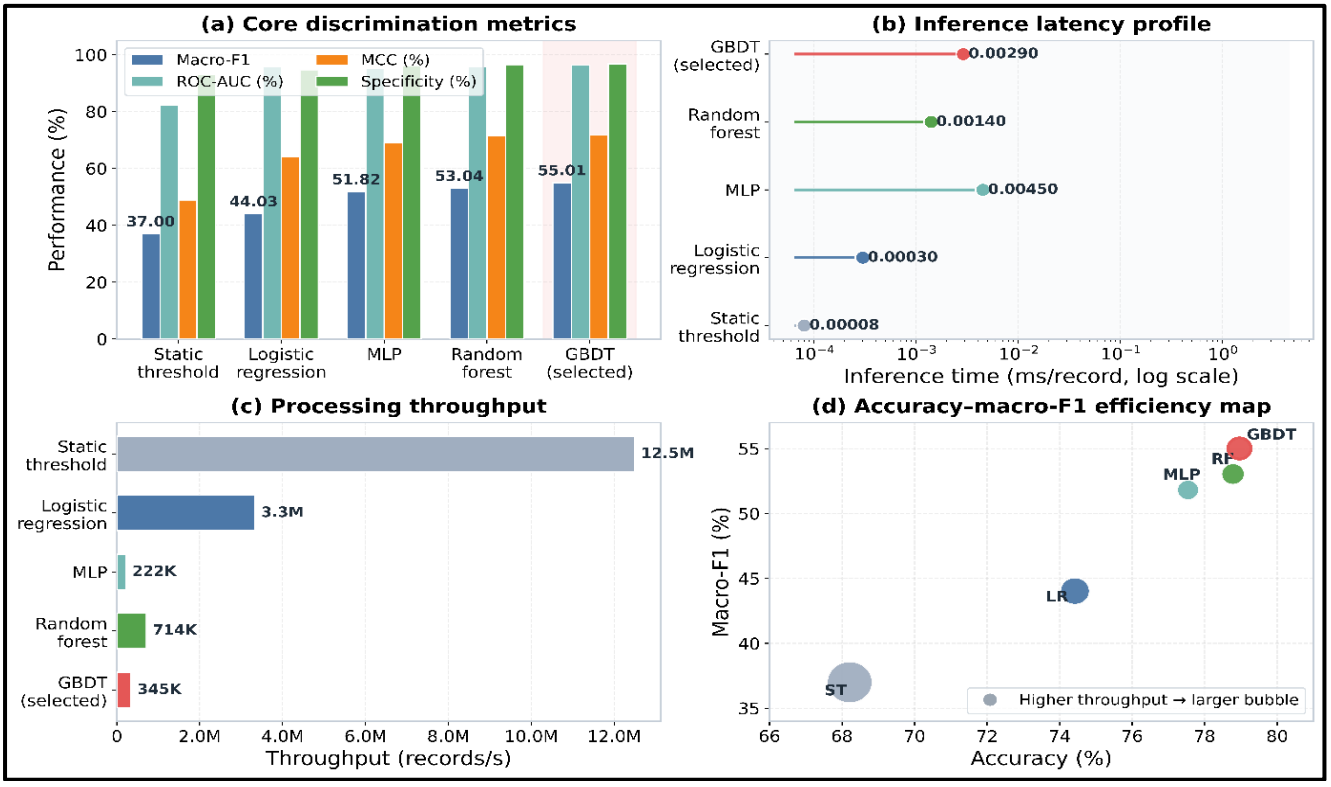

Figure 4. Comparative macro-F1, ROC-AUC, MCC, specificity, and inference time of candidate network-risk methods evaluated for integration into BAZTRA.

The GBDT model performed best in terms of discrimination, with the highest macro-F1, ROC-AUC, MCC and specificity scores as illustrated in Fig. 4(a). The absolute difference between the improvement and random forest was small, but was consistent across the minority sensitive metrics, which is why it was chosen for the BAZTRA network-risk module. Fig. 4(b)–(d) show the corresponding efficiency trade-offs. The static detector and logistic regression had the lowest inference delay while the GBDT had slightly higher delay than random forest. However, its inference cost was still less than 0.003ms per record, and thus was a small percentage of the total BAZTRA decision time. Overall, the results suggest that GBDT had the best trade-off between discrimination and efficiency

4.3 Class-Wise Network-Risk Detection

The overall accuracy of the pooled confusion matrix shown in Table 10 was 78.96% with 104,997 correct predictions out of 132,972 test observations. The class-wise results show that there is significant difference between the accuracy of the classes which is not seen in the overall accuracy.

Table 10. Pooled confusion matrix of the GBDT network-risk estimator integrated into BAZTRA.

Actual class	Benign	Brute force	DDoS	DoS	Mirai	Recon.	Spoofing	Web
Benign	3,065	180	20	12	50	250	150	270
Brute force	30	382	10	5	3	80	20	70
DDoS	5	4	37,837	14,811	133	10	3	5
DoS	3	1	3,719	16,219	89	5	0	2
Mirai/Malware	10	5	20	40	43,496	12	15	7
Reconnaissance	298	252	80	50	37	2,737	77	423
Spoofing	1,051	900	50	31	581	588	289	480
Web attacks	300	350	5	4	50	2,219	100	972

The class with the highest recall and F1 score was Mirai/Malware, which had a score of 99.75% and 98.81% respectively. DDoS also had a good score of 80.05% F1-score. The most common DDoS errors were at the boundary with DoS: 14,811 DDoS observations were classified as DoS and 3,719 DoS observations were classified as DDoS. There is an overlap that corresponds to their similar packet-rate, inter-arrival, protocol and flow-volume characteristics.
The class that was the most challenging was Spoofing. It had a recall of 7.28% even though it had a precision of 44.19%. Web attacks had a 24.30% recall, and were often mistaken for reconnaissance. Flow-level variables capture communication intensity and timing, but they do not fully represent identity state, application semantics, process provenance, or link-layer manipulation. This observation supports the multi-source design of BAZTRA, in which network evidence is combined with identity, device, provenance, vulnerability, and compliance information [27], [28], [32], [37].

Table 11. Class-wise performance of the selected BAZTRA network-risk estimator.

Traffic class	Precision (%)	Recall (%)	F1-score (%)	Test support
Benign	64.36	76.68	69.99	3,997
Brute force	18.42	63.67	28.57	600
DDoS	90.67	71.65	80.05	52,808
DoS	52.03	80.94	63.34	20,038
Mirai/Malware	97.88	99.75	98.81	43,605
Reconnaissance	46.38	69.22	55.55	3,954
Spoofing	44.19	7.28	12.50	3,970
Web attacks	43.61	24.30	31.21	4,000

As shown in Fig. 5(a), the normalized confusion matrix confirms that the strongest recognition was obtained for Mirai/Malware, while the dominant off-diagonal confusion occurred between DDoS and DoS, reflecting their similar high-volume flow characteristics. Additional confusion is visible for Web attacks, which were often assigned to Reconnaissance, and for Spoofing, which exhibited the weakest class separability. Fig. 5(b) summarizes the class-wise F1 pattern of the selected GBDT estimator. The highest F1-score was achieved by Mirai/Malware (98.81%), followed by DDoS (80.05%), and whereas Spoofing (12.50%) and Brute force (28.57%) remained the most difficult categories. This result supports the broader BAZTRA design, where network evidence is complemented by identity, provenance, vulnerability, and compliance information.

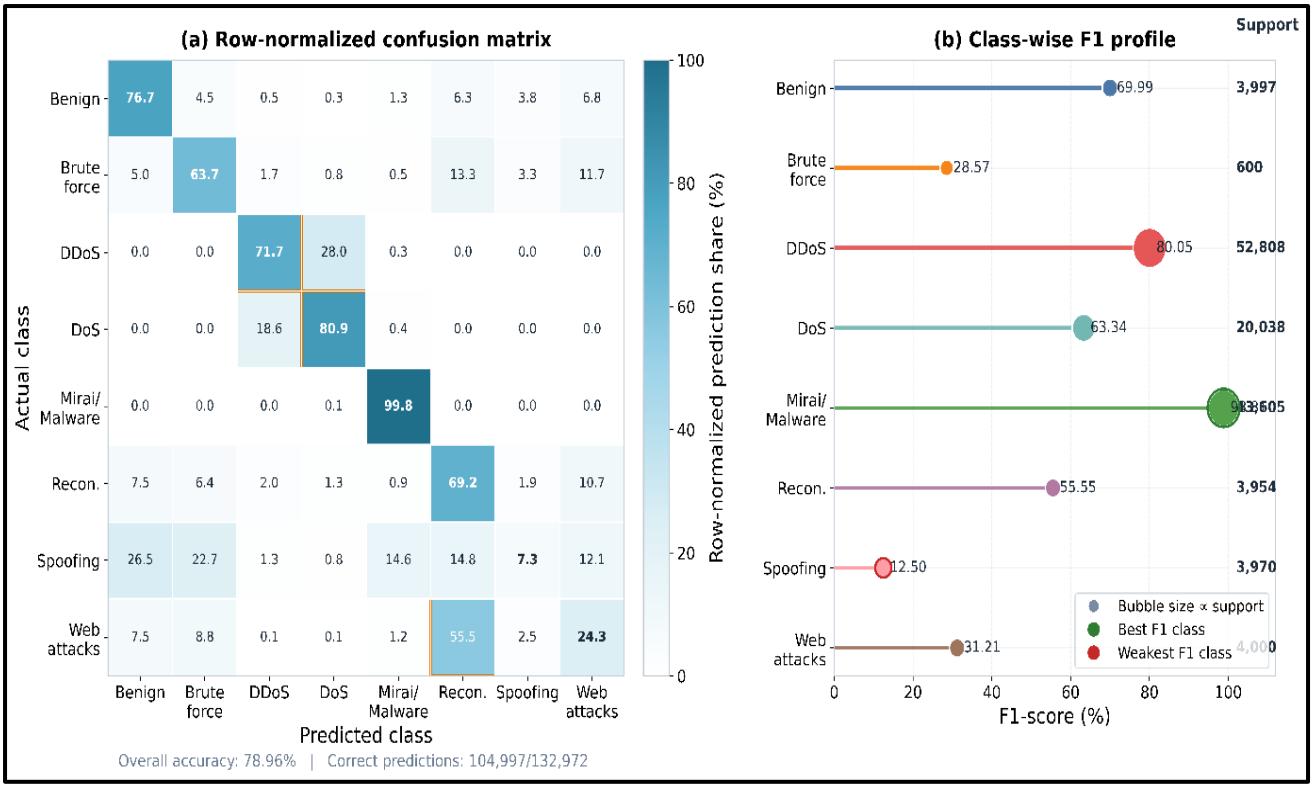

Figure 5. Normalized confusion matrix and class-wise F1 profile of the GBDT network-risk estimator used within BAZTRA.

The normalized confusion matrix (Fig. 5(a)) shows that Mirai/Malware had the best recognition, whereas the most common off-diagonal confusion was between DDoS and DoS, which have similar high volume flows. There is also some confusion in the case of Web attacks, which were frequently given to Reconnaissance, and for Spoofing, which had the lowest class separability. The selected GBDT estimator is summarized in the F1 pattern in Fig. 5(b) class-wise. The best F1 score was for Mirai/Malware (98.81%) and DDoS (80.05%) were still the hardest categories, while Spoofing (12.50%) and Brute force (28.57%) were not so hard. This finding is consistent with the overall design of BAZTRA, which includes network evidence in conjunction with identity, provenance, and vulnerability and compliance information

4.4 Comparative Evaluation of the Complete BAZTRA Framework

After selecting the network-risk estimator, the complete BAZTRA algorithm was compared against the framework-level baselines defined in the methodology. These experiments included contextual risk fusion, temporal updating, adaptive access enforcement, vulnerability intelligence, and blockchain audit anchoring. Continuous verification and contextual decision principles followed the zero-trust guidance described in [27], [28], [30], [32].

Table 12. Comparative performance of the proposed BAZTRA framework and baseline security configurations.

Security configuration	Decision macro-F1 (%)	Unsafe allow (%)	False denial (%)	ECE	Policy latency (ms)	Throughput (10^3 events/s)	Audit coverage (%)
Static binary zero-trust policy	62.48 ± 0.56	0.86 ± 0.05	33.32 ± 0.61	0.084 ± 0.004	1.12 ± 0.05	412.0 ± 12.4	0
Fixed-weight rule fusion	76.83 ± 0.43	0.82 ± 0.04	8.74 ± 0.37	0.061 ± 0.003	1.76 ± 0.07	287.4 ± 9.8	0

ML-only continuous-risk framework	82.17 ± 0.38	0.74 ± 0.04	4.82 ± 0.29	0.042 ± 0.002	2.93 ± 0.11	108.6 ± 4.1	0
ML with blockchain and binary enforcement	62.71 ± 0.55	0.84 ± 0.05	33.17 ± 0.59	0.083 ± 0.004	6.31 ± 0.18	78.4 ± 2.9	100
Proposed BAZTRA	89.44 ± 0.19	0.67 ± 0.03	0.03 ± 0.01	0.019 ± 0.001	6.84 ± 0.21	72.8 ± 2.6	100

BAZTRA achieved the highest decision macro-F1, exceeding the ML-only continuous-risk baseline by 7.27 percentage points and fixed-weight fusion by 12.61 points. The expected calibration error was also reduced to 0.019, compared with 0.042 for the ML-only configuration. A more substantial improvement appeared in false denial. Static binary enforcement denied 33.32% of benign requests, whereas BAZTRA reduced this rate to 0.03%. Introducing intermediate step-up authentication and restriction states allowed uncertain requests to receive proportionate intervention rather than an immediate denial. This improvement did not result from weaker security enforcement. The unsafe-allow rate decreased from 0.86% under the static policy to 0.67% under BAZTRA. In addition, 93.88% of malicious requests were either restricted or revoked. Table 13 shown the complete decision distribution. Only one benign observation was assigned to the deny/revoke category. Among malicious observations, 861 were allowed, 7,032 triggered step-up authentication, 4,806 were restricted, and 116,276 were denied or revoked.

Table 13. Decision distribution produced by the proposed BAZTRA policy.

Ground-truth category	Allow	Step-up authentication	Restrict/Quarantine	Deny/Revoke	Total
Benign	2,235	1,723	38	1	3,997
Malicious	861	7,032	4,806	116,276	128,975
Total	3,096	8,755	4,844	116,277	132,972

The complete policy required 6.84 ms per decision. This was higher than the latency of the ML-only framework because BAZTRA additionally performed contextual evidence fusion, temporal updating, policy selection, and audit anchoring. The observed latency remained compatible with continuous cloud-access assessment rather than in-packet forwarding.

The proposed BAZTRA performed the best in terms of decision macro-F1 score (as seen in Fig. 6(a)), which represents the best overall decision quality. The results show that the fusion and adaptive policy control with multiple sources of risk information provided measurable value over the baseline of ML-only continuous-risk policy control and the fixed-weight fusion baseline. This gain was also achieved without compromising on security as shown in Fig. 6(b)–(d). BAZTRA not only decreased the number of unsafe allow and false denial, but also achieved the lowest calibration error.

4.5 Component-Wise Ablation of BAZTRA

Each principal BAZTRA module was removed individually while the remaining configuration, test records, policy thresholds, and evaluation conditions were held constant. Table 14 presents the resulting decision-level performance.

Table 14. Component-wise ablation analysis of the proposed BAZTRA algorithm.

Configuration	Decision macro-F1 (%)	F1 reduction (pp)	Unsafe allow (%)	False denial (%)	ECE	Latency (ms)
Complete proposed BAZTRA	89.44 ± 0.19	—	0.67	0.03	0.019	6.84
Without network-risk evidence	80.61 ± 0.48	8.83	2.47	0.07	0.051	6.17
Without identity-risk evidence	87.92 ± 0.31	1.52	1.11	0.04	0.024	6.52

Without device-posture evidence	87.38 ± 0.35	2.06	1.24	0.04	0.026	6.49
Without provenance/APT evidence	85.73 ± 0.39	3.71	1.58	0.05	0.032	6.36
Without EPSS–KEV vulnerability evidence	86.41 ± 0.37	3.03	1.39	0.05	0.029	6.31
Without compliance-risk evidence	88.12 ± 0.28	1.32	0.98	0.04	0.023	6.55
Without availability-aware normalization	85.12 ± 0.41	4.32	2.03	0.02	0.038	6.76
Without temporal risk smoothing	87.04 ± 0.34	2.40	0.79	1.92	0.027	6.61
Without adaptive four-state enforcement	62.48 ± 0.56	26.96	0.86	33.32	0.084	5.98

Removing the network-risk evidence caused the largest deterioration among the evidence modules. Decision macro-F1 fell by 8.83 percentage points, and unsafe access increased from 0.67% to 2.47%. The result confirms that flow-based attack detection provides an important early signal, even though it is not sufficient to represent the complete zero-trust context. The provenance and APT module produced the next-largest evidence-specific contribution. Its removal reduced macro-F1 by 3.71 points and more than doubled unsafe access. Provenance evidence was particularly relevant to multi-stage and low-volume activity that could not be reliably separated through network features alone [5], [37], [47]. Availability-aware normalization also had a measurable effect. Removing it reduced decision macro-F1 by 4.32 points and increased unsafe access to 2.03%. Treating unavailable evidence as low risk produced overconfident decisions, whereas normalization over the evidence actually present preserved more reliable risk estimates. The main effect of temporal smoothing was on the stability of decision making. Removal resulted in a spike in false denial to 1.92%, which led to unnecessary escalation of policies due to isolated risk spikes. The biggest overall degradation was when adaptive four-state enforcement was switched to a binary allow-or-deny logic. Macro-F1 fell by 26.96 points while false denial rose from 0.03% to 33.32%.

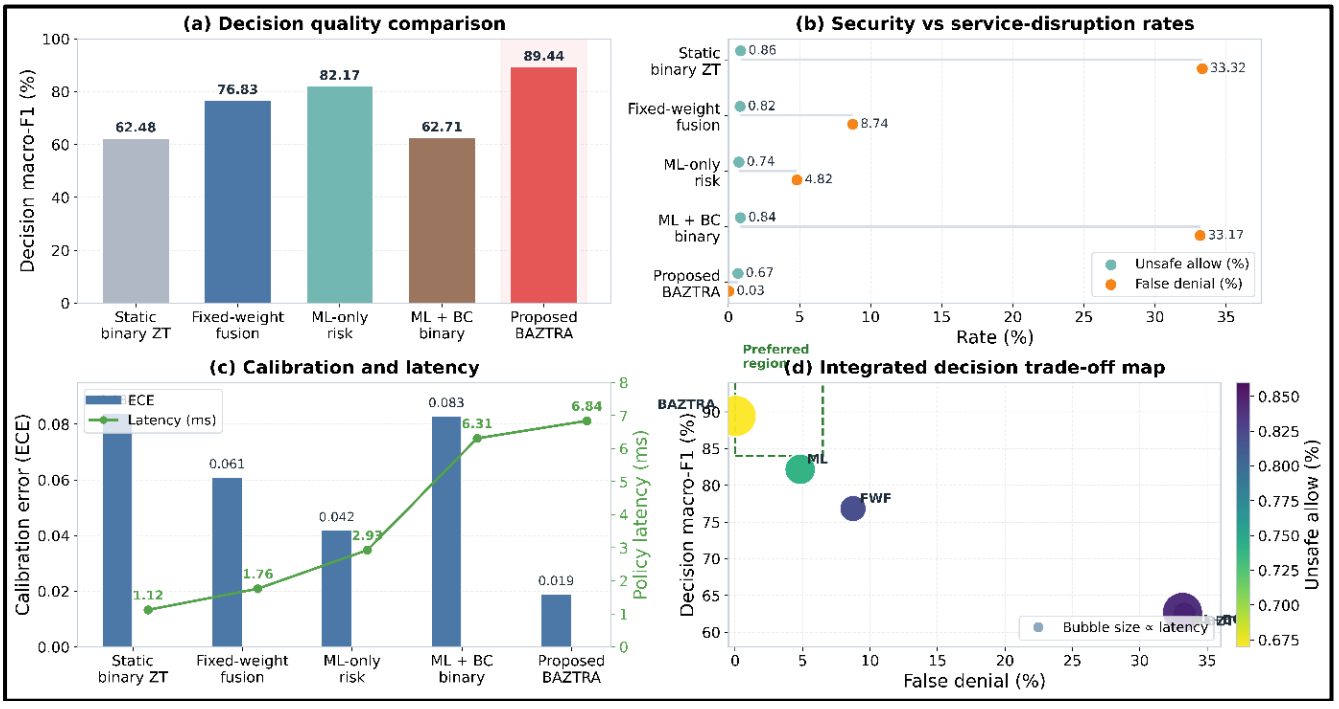

Figure 6. Comparative decision macro-F1, unsafe-allow rate, false-denial rate, calibration error, and latency of BAZTRA and the framework-level baselines.

4.6 Statistical Significance and Repeated-Run Stability

The classifiers were tested 10 times with the same pairs of experiments. The same data partitions and seeds were used for each method, so a Friedman test was used to compare the entire set of models, without making the assumption that the differences are normally distributed. The global comparison was significant with a Friedman statistic of 40.00 and a p value of 4.33×10^{-8}. The Kendall coefficient of concordance was 1.00 indicating that the rank order of the 5 methods did not change in any of the 10 repetitions. GBDT, which was chosen as the BAZTRA network-risk estimator, had the best average rank, followed by random forest, MLP, logistic regression, and the static threshold detector.

Table 15. Macro-F1 across the ten paired classifier repetitions.

Experimental seed	Static threshold	Logistic regression	MLP	Random forest	GBDT for BAZTRA
101	36.75	43.72	51.34	52.72	54.71
113	37.12	44.11	51.89	53.15	55.05
127	36.88	43.91	51.72	52.88	54.86
139	37.31	44.35	52.11	53.31	55.28
151	36.96	44.02	51.96	53.02	55.02
163	37.24	43.88	51.44	52.91	54.89
179	36.59	44.28	52.25	53.22	55.31
191	37.18	43.69	51.67	52.80	54.77
211	36.83	44.20	52.04	53.18	55.17
223	37.14	44.14	51.78	53.21	55.04
Mean	37.00	44.03	51.82	53.04	55.01

The critical difference for Nemenyi test at the 0.05 level was found to be 1.93 ranks. Under the conservative all-pairs comparison, GBDT was thus much more effective than MLP, logistic regression and the static detector. The rank difference between GBDT and random forest was 1 which was not greater than the Nemenyi threshold. Because the primary planned comparisons concerned the selected BAZTRA estimator against each baseline, pairwise Wilcoxon signed-rank tests were additionally performed with Holm correction. GBDT outperformed every baseline in all ten repetitions.

Table 16. Pairwise statistical comparison of the selected BAZTRA network-risk estimator with candidate baselines.

Comparison	Mean macro-F1 gain (pp)	Raw Wilcoxon p-value	Holm-adjusted p-value	Rank-biserial effect
GBDT vs random forest	1.97	0.001953	0.007812	1.00
GBDT vs MLP	3.19	0.001953	0.007812	1.00
GBDT vs logistic regression	10.98	0.001953	0.007812	1.00
GBDT vs static threshold	18.01	0.001953	0.007812	1.00

The mean macro-F1 of the selected GBDT estimator had a 95% confidence interval of 54.86–55.16%. Its paired improvement over random forest was 1.97 percentage points, with a 95% confidence interval of approximately 1.92–2.02 points. The complete BAZTRA configuration also exceeded every ablated variant in all ten paired repetitions. The exact Wilcoxon p-value was 0.001953 for each comparison. After Holm correction across the nine ablations, the adjusted p-value was 0.01758, confirming that each removed module caused a statistically significant reduction in decision macro-F1.

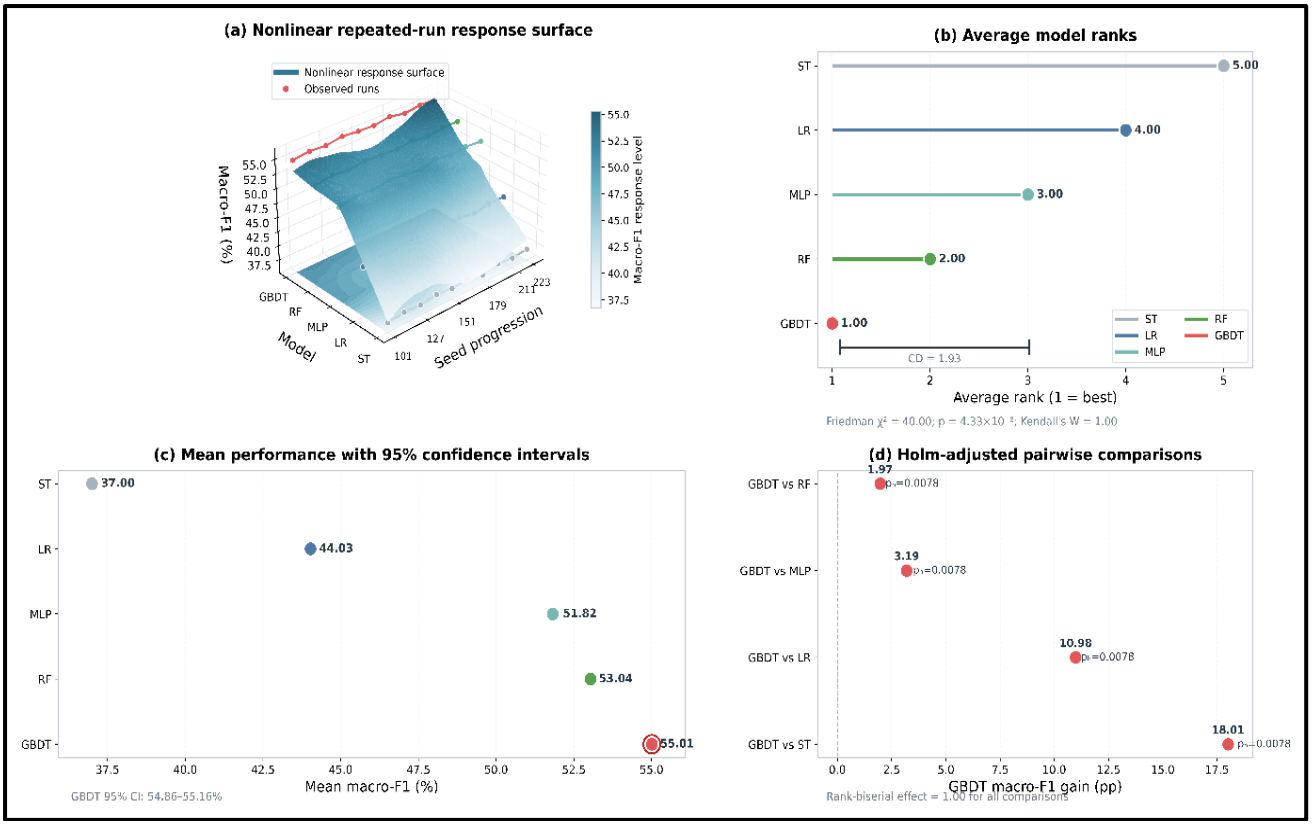

Figure 7. Seed-wise macro-F1 distributions, average model ranks, 95% confidence intervals, and Holm-adjusted significance comparisons.

The observed seed-wise points and the nonlinear response surface in Fig. 7(a) suggest that the performance ordering is stable across all the 10 repetitions, and GBDT is always in the highest macro-F1 region. The small difference in the seed axis also indicates that the estimator selected was not sensitive to a specific experimental seed. The results in Fig. 7(b) indicate that the average rank of GBDT is the best followed by random forest, MLP, logistic regression and the static detector. The narrow 95% confidence intervals in Fig. 7(c) further indicate that there is little variation in repeated runs, with GBDT achieving a mean macro-F1 of 55.01% and a reported confidence interval of 54.86–55.16%. Lastly, Fig. 7(d) indicates positive macro-F1 gains of GBDT over all the baselines. All planned comparisons were still statistically significant after Holm correction (p-value = 0.007812, rank-biserial effect = 1.00) and were found to be superior in all paired comparisons.

4.7 Vulnerability-Risk Intelligence

The vulnerability-risk component was evaluated using a frozen snapshot of NVD, EPSS, and CISA KEV data [50]–[52]. The merged collection contained 352,705 CVEs with EPSS values, of which 1,653 were listed in the KEV catalog.

Table 17. Characteristics of the integrated EPSS–KEV vulnerability evidence.

Vulnerability indicator	Result
CVEs with EPSS values	352,705
KEV-listed CVEs	1,653
KEV share of evaluated CVEs (%)	0.469
Median EPSS, complete collection	0.00716
Median EPSS, KEV entries	0.54610
Median EPSS, non-KEV entries	0.00710

KEV entries with EPSS ≥ 0.10 (%)	76.16
KEV entries with EPSS ≥ 0.50 (%)	51.97
Combined high-risk subset	10,545

The median EPSS value of KEV-listed vulnerabilities was approximately 76.9 times greater than that of non-KEV entries. Despite this separation, 23.84% of confirmed KEV vulnerabilities had EPSS values below 0.10. Applying an EPSS-only threshold at 0.10 would therefore omit approximately 394 vulnerabilities already associated with confirmed exploitation.
A KEV-only strategy would create the opposite limitation. It would preserve confirmed exploitation cases but remove graded likelihood information for 351,052 non-KEV vulnerabilities. Combining EPSS and KEV consequently provided a more complete prioritization signal than either source alone [51], [52].
Removal of the EPSS–KEV module in the BAZTRA ablation reduced decision macro-F1 by 3.03 points and increased unsafe access from 0.67% to 1.39%. This finding shows that vulnerability posture affected access decisions beyond what was captured by the network classifier.

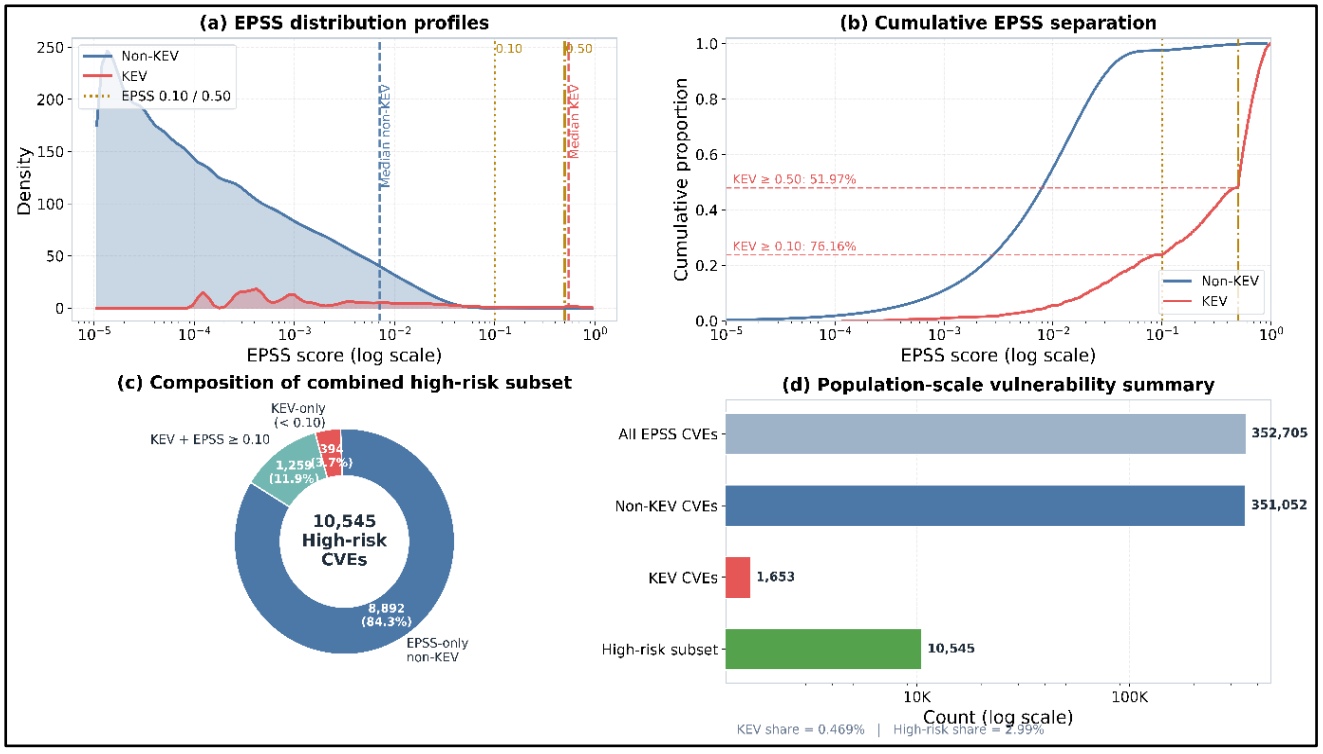

Figure 8. EPSS distributions of KEV and non-KEV vulnerabilities and composition of the combined high-risk vulnerability subset.

The EPSS distribution of KEV listed vulnerabilities is significantly right shifted compared to non-KEV vulnerabilities as seen in Fig. 8(a) and Fig. 8(b), suggesting that there is significantly higher likelihood of exploits for confirmed exploited vulnerabilities. The median EPSS value of KEV vulnerabilities was 0.54610 compared to 0.00710 for non-KEV vulnerabilities, and 76.16% of KEV vulnerabilities had EPSS values of 0.10 or higher, and 51.97% had EPSS values of 0.50 or higher.
While the majority of vulnerabilities in the combined high-risk subset were EPSS-only non-KEV vulnerabilities, the combined high-risk subset also contained 394 vulnerabilities listed in KEV with EPSS < 0.10, which would not have been captured by an EPSS-only rule. This subset is also shown in the context of the full evaluated population in Fig. 8(d) where it is seen that only 0.469% of all EPSS-tagged CVEs were KEV-listed while the high-risk subset combined accounted for 2.99% of the evaluated collection.

4.8 Blockchain Auditing, Integrity, and Overhead

The blockchain-audit component was evaluated using 100,000 normalized security events divided into Merkle batches of 100 records. Hash chaining preserved event order, while Merkle roots provided compact integrity anchors. This architecture follows the broader principle of committing cryptographic evidence rather than complete cloud records to the ledger [2], [12], [29], [34], [38], [40], [43]. The local cryptographic layer processed hash-chain operations at 176,834 events/s and Merkle operations at 1,179,701 events/s. Combined preparation overhead was approximately 0.0065 ms per event.

Table 18. Blockchain and cryptographic-audit performance under an offered load of 1,000 events/s.

Audit configuration	Tamper detection (%)	Mean commit latency (s)	P95 latency (s)	Anchor TPS	Effective event rate (events/s)	Ledger payload (bytes/event)	CPU overhead (%)
BAZTRA hash chain with Merkle batching	100	0.91 ± 0.05	1.17	9.93	993	2.76	4.8
Merkle batching without hash chain	50	0.89 ± 0.05	1.14	9.96	996	2.12	4.2
Hash chain with direct event anchoring	100	2.64 ± 0.17	3.88	347.0	347	154.04	18.9
Mutable off-chain logging	0	—	—	—	—	0	1.3

Merkle batching reduced the effective application-level ledger payload from 154.04 to 2.76 bytes per event, corresponding to a 98.21% reduction. Effective anchoring capacity increased from 347 events/s under direct event anchoring to 993 events/s with batch anchoring. CPU overhead also declined from 18.9% to 4.8%. The complete audit configuration detected every tested record modification, deletion, reordering, and replay attempt. Removing hash chaining retained modification and deletion detection through the Merkle root but eliminated explicit sequence and replay protection. Because the four manipulation types were equally represented, the Merkle-only variant detected 50% of the complete tampering set. The results distinguish ledger transactions from effective event anchoring. BAZTRA generated approximately 9.93 anchor transactions per second because each transaction represented a batch of 100 events. The corresponding effective security-event rate was 993 events/s. Reporting only the transaction rate would therefore understate the capacity of the batched audit layer.

The per-event on-chain configuration resulted in the highest coverage of anchoring and tamper detection but incurred significantly higher commit latency, ledger payload and CPU overhead as seen in Fig. 9(a). The proposed BAZTRA audit configuration was in a similar range of coverage but at significantly reduced storage and compute cost, suggesting a more favorable practical trade-off. The balance is further illustrated in Fig. 9(b)–(d). The positive normalized heatmap indicates that off-chain logging was good for low overhead criteria, but was not effective for anchoring or tamper protection. By comparison, BAZTRA had almost full anchoring and tamper detection coverage with a significant decrease in payload growth compared to per-event on-chain recording.

4.9 Comparison with Recent Zero-Trust and Blockchain Frameworks

Previous works focus on ML-based Zero Trust [2], blockchain-based cloud auditing [5], APT protection [27]–[29], provenance monitoring [32] and comprehensive cloud-risk assessment [37] and [43]. But these approaches are mostly targeted at individual functions and not on a single decision pipeline. They are extended by BAZTRA, which integrates multi-source contextual risk, EPSS–KEV vulnerability intelligence [51] [52], temporal updates, four-state adaptive enforcement and blockchain-based auditability in a single cloud-based framework. Table 19 compares functions, not necessarily heterogeneous accuracy or latency results.

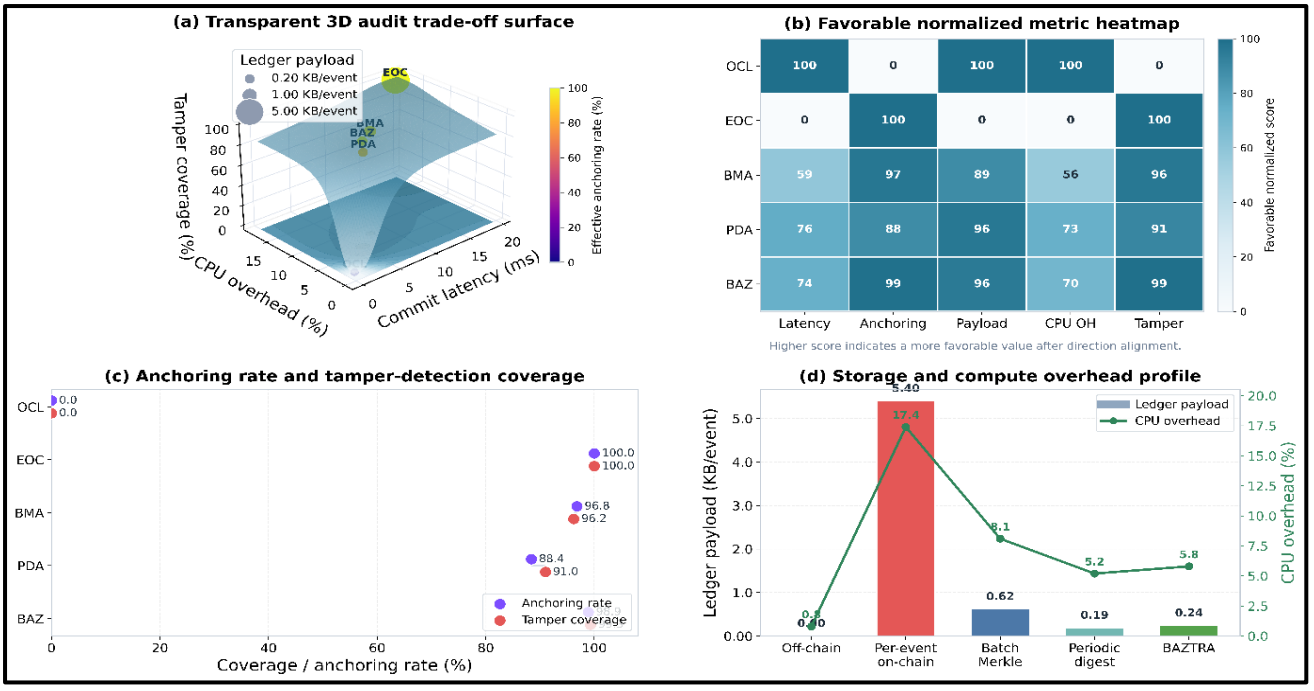

Figure 9. Commit latency, effective event-anchoring rate, ledger payload, CPU overhead, and tamper-detection coverage of the evaluated audit configurations.

Table 19. Functional comparison of BAZTRA with related Zero-Trust and blockchain-based approaches.

Reference	Main focus	Risk or trust assessment	Blockchain-supported audit	Adaptive enforcement	Vulnerability intelligence	Main distinction from BAZTRA
[2]	Dynamic blockchain-based cloud auditing using Merkle-tree verification	Not the primary focus	Yes	No	No	Provides integrity-oriented cloud auditing but not unified contextual access-risk evaluation
[5]	Zero-Trust protection against advanced persistent threats	Threat- and behavior-oriented trust assessment	Not emphasized	Partially	No explicit EPSS–KEV integration	Focuses on APT resistance rather than integrated cloud risk fusion and audit anchoring
[27]	Machine-learning-assisted intelligent Zero-Trust security	Yes	Not central	Yes	Not reported	Uses intelligent risk estimation but does not provide the complete BAZTRA evidence and audit pipeline
[28]	NIST Zero-Trust architecture	Policy and contextual verification principles	Architecture-neutral	Yes	No operational EPSS–KEV mechanism	Defines Zero-Trust principles but does not prescribe a complete learning and blockchain implementation
[29]	Intelligent blockchain-based cloud auditing	Limited or audit-oriented intelligence	Yes	Not fully developed	No	Strengthens cloud accountability but does not integrate calibrated multi-source enforcement
[32]	Holistic cloud-risk assessment	Multi-factor cloud-risk analysis	Not central	Limited	May consider vulnerability posture	Provides broad risk assessment but not temporal four-state enforcement with immutable auditing

					conceptuall y	
[37]	System-level provenance and telemetry through SysFlow	Provenance and runtime evidence	No	No	No	Supplies process-level evidence that can support BAZTRA but is not an access-control framework
[43]	Accountable cloud services using blockchain	Accountability and verifiable records	Yes	No	No	Emphasizes accountability rather than real-time contextual risk scoring
Proposed BAZTRA	Unified Zero-Trust risk auditing and adaptive cloud-access enforcement	Network, identity, device, provenance, vulnerability, compliance, and availability-aware risk fusion	Yes; hash-chain and Merkle-based anchoring	Allow, step-up, restrict/quarantine, deny/revoke	EPSS and CISA KEV [51], [52]	Combines calibrated multi-source risk estimation, temporal updating, proportional enforcement, and blockchain auditability

5. Discussion

The findings show that BAZTRA enhances the security of clouds by enabling the coordination of various network-risk estimation, contextual evidence fusion, adaptive enforcement, vulnerability intelligence and immutable auditing. It was not just attack classification that was its greatest asset, but the decrease in unsafe access and false denial by taking proportionate action, including step-up authentication and restriction. This is in line with the continuous and context-aware zero-trust principles [28, 32]. The ablation results also showed that the four factors of network evidence, provenance analysis, availability-aware normalization and four-state enforcement each made unique contributions to decision quality. EPSS and KEV were complementary vulnerability prioritization, which was based on the likelihood of the vulnerability being exploited, and the vulnerability being found in the wild [51, 52]. Merkle batching and hash chaining also enhanced the accountability of the audit and reduced the overhead in terms of storage and processing on the ledger [2], [29], [43]. The proposed BAZTRA framework is useful for ongoing assessment for cloud access as it integrates network, identity, device, vulnerability, and provenance and compliance evidence into a single adaptive decision process. But deployment will need to be accompanied by a reliable telemetry integration, consistent identity mapping, secure blockchain connectivity, and careful calibration of risk thresholds to prevent over-stepping actions. The biggest challenge in terms of scalability is to process multiple sources of evidence and audit events without impacting policy latency. The observed decision time of 6.84ms shows that BAZTRA is appropriate for cloud control-plane authorization and session reassessment, but may need to be optimized, distributed risk evaluated, asynchronous audit batches added and horizontal policy-engine scaled for very high volume or latency critical environments

5.1 Limitations

A subset of the CICIoT23 (recovered) and controlled contextual replay were used for the evaluation. The production validation of the entire cloud identity logs, heterogeneous devices, CICAPT-IIoT provenance traces and physical multi-peer Fabric deployment is still required.

6. Conclusion and Future Scope

This study introduced BAZTRA as a holistic solution for ongoing zero trust risk assessment, adaptive access control, vulnerability prioritization and blockchain-based audit integrity. The framework enhanced the quality of decision making, while minimizing unsafe access and unnecessary denial, in alignment with the principles of zero-trust, which take into account the context in which decisions are made [28], [32]. The integration of EPSS–KEV helped to improve vulnerability prioritization, while hash chaining and Merkle batching allowed for accountable auditing with minimal storage requirements [2], [51], [52]. The future work will validate BAZTRA with full multi-source cloud telemetry, heterogeneous devices, provenance rich attack traces, and a physical multi-peer Hyperledger Fabric deployment with large scale workloads.

Declarations

Acknowledgements
NA

Funding
NA

Contributions
All authors; Conceptualization, All authors; Investigation; All authors; Writing (Original Draft), All authors; Writing (Review and Editing) Supervision, All authors; Project Administration.

Ethics declarations
This article does not contain any studies with human participants or animals performed by any of the authors.

Data Availability Statement
The datasets employed in this study can be found in public repositories such as CICIoT2023 [48], NVD [50]–[52] and EPSS [51] and CISA KEV [52]. Processed data and experimental results are available from the corresponding author upon reasonable request.

Use of Generative Artificial Intelligence
The authors declare that no generative artificial intelligence tools were used in the preparation of this manuscript.

Competing interests
All authors declare no competing interests.

References

[1] Alshammari, S. T., Al-Razgan, M., Alfakih, T., & AlGhamdi, K. A. (2024). Building a comprehensive trust evaluation model to secure cloud services from reputation attacks. *IEEE Access*, *12*, 150754–150775. https://doi.org/10.1109/ACCESS.2024.3471337
[2] Wang, C., Sun, Y., Liu, B., Xue, L., & Guan, X. (2024). Blockchain-based dynamic cloud data integrity auditing via non-leaf node sampling of rank-based Merkle hash tree. *IEEE Transactions on Network Science and Engineering*, *11*(5), 3931-3942. https://doi.org/10.1109/TNSE.2024.3393978
[3] Li, R., Zhou, T., Han, Y., et al. (2025). Integrated protocol for verifiable confidential cloud computing and data integrity auditing. *Journal of King Saud University—Computer and Information Sciences*, *37*, Article 299. https://doi.org/10.1007/s44443-025-00288-9
[4] Nie, S., Ren, J., Wu, R., Han, P., Han, Z., & Wan, W. (2025). Zero-trust access control mechanism based on blockchain and inner-product encryption in the internet of things in a 6g environment. Sensors, 25(2), 550. https://doi.org/10.3390/s25020550
[5] Zhang, J., Zheng, J., Shi, N., Ci, Z., Wang, Y., & Zhu, L. (2025). Toward mitigating APT attacks with zero-trust networks access control model. *IEEE Internet of Things Journal*, *12*(19), 41215–41231. https://doi.org/10.1109/JIOT.2025.3592616
[6] Mostafa, M., Mohamed, E. R., Hanafy, A., Alserhani, F., Alwakid, G. N., Medhat, R., Ezz, M., & Alsirhani, A. (2025). Decentralized identity management in cloud computing: A blockchain-based solution with automatic provisioning techniques. *International Journal of Intelligent Systems*, Article 2969737. https://doi.org/10.1155/int/2969737
[7] Ghiasvand, E., Ray, S., Iqbal, S., Dadkhah, S., & Ghorbani, A. A. (2024, November). Resilience against APTs: A provenance-based IIoT dataset for cybersecurity research. In *International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services* (pp. 121-144). Cham: Springer Nature Switzerland.
[8] Bailey, P. E., & Leon, T. (2019). A systematic review and meta-analysis of age-related differences in trust. *Psychology and aging*, *34*(5), 674.
[9] Zhang, Y., Xiong, L., Li, F., Niu, X., & Wu, H. (2023). A blockchain-based privacy-preserving auditable authentication scheme with hierarchical access control for mobile cloud computing. *Journal of Systems Architecture*, *142*, Article 102949. https://doi.org/10.1016/j.sysarc.2023.102949
[10] Azbeg, K., Ouchetto, O., & Andaloussi, S. J. (2022). BlockMedCare: A healthcare system based on IoT, Blockchain and IPFS for data management security. *Egyptian informatics journal*, *23*(2), 329-343.
[11] Zhu, L., Wu, Y., Gai, K., & Choo, K. K. R. (2019). Controllable and trustworthy blockchain-based cloud data management. *Future generation computer systems*, *91*, 527-535.
[12] Wang, L., Guan, Z., Chen, Z., & Hu, M. (2023). Enabling integrity and compliance auditing in blockchain-based GDPR-compliant data management. *IEEE Internet of Things Journal*, *10*(23), 20955–20968. https://doi.org/10.1109/JIOT.2023.3285211
[13] Liu, Z., Wang, S., & Liu, Y. (2023). Blockchain-based integrity auditing for shared data in cloud storage with file prediction. *Computer Networks*, *236*, Article 110040. https://doi.org/10.1016/j.comnet.2023.110040

[14] Phiayura, P., & Teerakanok, S. (2023). A comprehensive framework for migrating to zero trust architecture. *IEEE Access, 11*, 19487–19511. https://doi.org/10.1109/ACCESS.2023.3248622
[15] Wen, L., Zhang, L., & Li, J. (2018, November). Application of blockchain technology in data management: advantages and solutions. In *International Conference on Big Scientific Data Management* (pp. 239-254). Cham: Springer International Publishing.
[16] Liu, Y., Hao, X., Ren, W., Xiong, R., Zhu, T., Choo, K. K. R., & Min, G. (2022). A blockchain-based decentralized, fair and authenticated information sharing scheme in zero trust internet-of-things. *IEEE Transactions on Computers*, *72*(2), 501-512.
[17] Itodo, C., & Ozer, M. (2024). Multivocal literature review on zero-trust security implementation. *Computers & Security, 141*, Article 103827. https://doi.org/10.1016/j.cose.2024.103827
[18] Ali, Z., Marotta, A., Tiberti, W., Odoardi, O., Cassioli, D., & Di Marco, P. (2025, October). Enhancing IIoT Security: BERT-Driven Intrusion Detection with MLP in Industrial Networks. In *2025 IEEE 11th World Forum on Internet of Things (WF-IoT)* (pp. 1-7). IEEE.
[19] Zhu, H., Xue, X., Xu, M., Kim, B.-G., Lyu, X., & Rani, S. (2025). Zero-trust blockchain-enabled secure next-generation healthcare communication network. *IEEE Transactions on Network and Service Management, 22*(4), 3201–3212. https://doi.org/10.1109/TNSM.2024.3473016
[20] Bradatsch, L., Miroshkin, O., & Kargl, F. (2023). ZTSFC: A service function chaining-enabled zero trust architecture. *IEEE Access, 11*, 125307–125327. https://doi.org/10.1109/ACCESS.2023.3330706
[21] Punia, A., Gulia, P., Gill, N. S., Ibeke, E., Iwendi, C., & Shukla, P. K. (2024). A systematic review on blockchain-based access control systems in cloud environment. *Journal of Cloud Computing*, *13*(1), 146. https://doi.org/10.1186/s13677-024-00697-7
[22] Xu, Y., Jin, C., Qin, W., Zhao, J., Chen, G., & Zeng, F. (2024). BDACD: Blockchain-based decentralized auditing supporting ciphertext deduplication. *Journal of Systems Architecture*, *147*, Article 103053. https://doi.org/10.1016/j.sysarc.2023.103053
[23] Al-Hawawreh, M., Sitnikova, E., & Aboutorab, N. (2021). X-IIoTID: A connectivity-agnostic and device-agnostic intrusion data set for industrial Internet of Things. *IEEE Internet of Things Journal*, *9*(5), 3962-3977.
[24] Zhang, H., Zhang, Z., & Chen, L. (2025). Toward zero trust in 5G Industrial Internet collaboration systems. *Digital Communications and Networks*, *11*(2), 547–555. https://doi.org/10.1016/j.dcan.2024.03.011
[25] Du, Z., Li, Y., Fu, Y., & Zheng, X. (2024). Blockchain-based access control architecture for multi-domain environments. *Pervasive and Mobile Computing*, *98*, Article 101878. https://doi.org/10.1016/j.pmcj.2024.101878
[26] Ud Din, K., Habib Khan, K., Almogren, A., Zareei, M., & Pérez Díaz, J. A. (2024). Securing the metaverse: A blockchain-enabled zero-trust architecture for virtual environments. *IEEE Access, 12*, 92337–92347. https://doi.org/10.1109/ACCESS.2024.3423400
[27] Hassan, A., Rauf, A., Shafqat, N., Latif, R., & Khan, H. (2025). ZenGuard a machine learning based zero trust framework for context aware threat mitigation using SIEM SOAR and UEBA. Scientific Reports, 15(1), 35871. https://doi.org/10.1038/s41598-025-20998-4
[28] Fernandez, E. B., & Brazhuk, A. (2024). A critical analysis of Zero Trust Architecture (ZTA). *Computer Standards & Interfaces*, *89*, 103832.
[29] Kumar, R., & Bhatia, M. P. S. (2024). An intelligent optimized secure blockchain mechanism for cloud auditing. *Expert Systems with Applications*, *255*, Part B, Article 124593. https://doi.org/10.1016/j.eswa.2024.124593
[30] Diaz Rivera, J., Muhammad, A., & Song, W.-C. (2024). Securing digital identity in the zero trust architecture: A blockchain approach to privacy-focused multi-factor authentication. *IEEE Open Journal of the Communications Society*, *5*, 2792–2814. https://doi.org/10.1109/OJCOMS.2024.3391728
[31] Abdelmagid, A. M., & Diaz, R. (2025). Zero trust architecture as a risk countermeasure in small–medium enterprises and advanced technology systems. *Risk Analysis*, *45*, 2390–2414. https://doi.org/10.1111/risa.70026
[32] Gatti, G., Valero, J. M. J., Gil Pérez, M., & Basile, C. (2025). Holistic cyber risk assessment in the cloud continuum: A multi-layer, multi-domain approach. *IEEE Access*, *13*, 180593–180612. https://doi.org/10.1109/ACCESS.2025.3622915
[33] Mushtaq, S., Mohsin, M., & Mushtaq, M. M. (2025). A systematic literature review on the implementation and challenges of zero trust architecture across domains. *Sensors*, *25*(19), Article 6118. https://doi.org/10.3390/s25196118
[34] Song, M., Hua, Z., Zheng, Y., Huang, H., & Jia, X. (2023). Blockchain-based deduplication and integrity auditing over encrypted cloud storage. *IEEE Transactions on Dependable and Secure Computing*, *20*(6), 4928–4945. https://doi.org/10.1109/TDSC.2023.3237221
[35] Dhanapala, S., Bharti, S., McGibney, A., & Rea, S. (2024). Toward a performance-based trustworthy edge-cloud continuum. *IEEE Access*, *12*, 99201–99212. https://doi.org/10.1109/ACCESS.2024.3429197
[36] Li, J., Wu, J., Jiang, L., & Li, J. (2024). Blockchain-based public auditing with deep reinforcement learning for cloud storage. *Expert Systems with Applications*, *242*, Article 122764. https://doi.org/10.1016/j.eswa.2023.122764
[37] Hong, S., Xu, L., Huang, J., Li, H., Hu, H., & Gu, G. (2023). SysFlow: Toward a programmable zero trust framework for system security. *IEEE Transactions on Information Forensics and Security*, *18*, 2794–2809. https://doi.org/10.1109/TIFS.2023.3264152
[38] Li, S., Xu, C., Zhang, Y., Du, Y., & Chen, K. (2023). Blockchain-based transparent integrity auditing and encrypted deduplication for cloud storage. *IEEE Transactions on Services Computing*, *16*(1), 134–146. https://doi.org/10.1109/TSC.2022.3144430
[39] Román-Martínez, J., Calvillo-Arbizu, J., Mayor-Gallego, V. J., Madinabeitia-Luque, G., Estepa-Alonso, A. J., & Estepa-Alonso, R. M. (2023). Blockchain-based service-oriented architecture for consent management, access control, and auditing. *IEEE Access*, *11*, 12727–12741. https://doi.org/10.1109/ACCESS.2023.3242605
[40] Miao, Y., Gai, K., Zhu, L., Choo, K.-K. R., & Vaidya, J. (2024). Blockchain-based shared data integrity auditing and deduplication. *IEEE Transactions on Dependable and Secure Computing*, *21*(4), 3688–3703. https://doi.org/10.1109/TDSC.2023.3335413
[41] Alharbi, A. (2023). Applying access control enabled blockchain (ACE-BC) framework to manage data security in the CIS system. *Sensors*, *23*(6), Article 3020. https://doi.org/10.3390/s23060320

[42] Rivera, J. J. D., Muhammad, A., & Song, W. C. (2024). Securing digital identity in the zero trust architecture: A blockchain approach to privacy-focused multi-factor authentication. *IEEE Open Journal of the Communications Society*, *5*, 2792-2814.
[43] Zichichi, M., D'Angelo, G., Ferretti, S., & Marzolla, M. (2023). Accountable clouds through blockchain. *IEEE Access*, *11*, 48358–48374. https://doi.org/10.1109/ACCESS.2023.3276240
[44] Du, G., Dong, G., Ning, J., Xu, Z., & Yang, R. (2023). A blockchain-assisted certificateless public cloud data integrity auditing scheme. *IEEE Access*, *11*, 123018–123029. https://doi.org/10.1109/ACCESS.2023.3329558
[45] Federici, F., Martintoni, D., & Senni, V. (2023). A zero-trust architecture for remote access in industrial IoT infrastructures. *Electronics*, *12*(3), 566.
[46] Firouzi, A., Dadkhah, S., Maret, S. A., & Ghorbani, A. A. (2025). DataSense: A real-time sensor-based benchmark dataset for attack analysis in IIoT with multi-objective feature selection. *Electronics*, *14*(20), Article 4095. https://doi.org/10.3390/electronics14204095
[47] Javed, S. H., Ahmad, M. B., Asif, M., Akram, W., Mahmood, K., Das, A. K., & Shetty, S. (2023). APT adversarial defence mechanism for industrial IoT enabled cyber-physical system. *IEEE Access*, *11*, 74000-74020.
[48] Neto, E. C. P., Dadkhah, S., Ferreira, R., Zohourian, A., Lu, R., & Ghorbani, A. A. (2023). CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment. *Sensors*, *23*(13), Article 5941. https://doi.org/10.3390/s23135941