Proof of Evaluation: ML-Driven Consensus for Trustworthy Permissioned Blockchains

Rashmi Ratnayake*, Madhusanka Liyanage†, Liam Murphy‡
*†Network Softwarization and Security Labs (NetsLab), School of Computer Science, University College Dublin, Ireland
‡School of Computer Science, University College Dublin, Ireland
Email: *rashmi.ratnayake@ucdconnect.ie, †madhusanka@ucd.ie, ‡liam.murphy@ucd.ie

***Abstract*—Blockchains are widely adopted as distributed storage systems to eliminate single points of failure and ensure data immutability. However, while they guarantee that data cannot be altered once recorded, they do not ensure that the data is accurate or trustworthy at the time of inclusion. Existing consensus mechanisms primarily validate agreement among nodes rather than the correctness of the data itself, enabling erroneous or malicious inputs to be permanently embedded in the ledger. This paper proposes Proof of Evaluation (PoEval), a data-centric, trust-by-verification consensus mechanism designed for permissioned blockchains. In PoEval, incoming data is treated as untrusted and must be validated prior to block inclusion. An Evaluation Coordination Committee (ECC) manages evaluator selection and aggregates trust assessments. Selected evaluators independently apply certified machine learning (ML) models to assess data trustworthiness. A quorum-based majority voting mechanism across heterogeneous models determines data eligibility, while node reputation governs block proposal rights. By embedding ML-driven validation within the consensus workflow, PoEval ensures that only verified, high-integrity data is recorded on the blockchain. Experiments using IoT datasets demonstrate that PoEval significantly reduces false acceptance rates and processing latency, and achieves higher data reliability compared to conventional and outlier-aware consensus mechanisms, offering a scalable path toward trust-by-design distributed ledgers.**

***Index Terms*—Blockchain, Consensus, Machine Learning, Data Trust, IoT**

I. Introduction

Blockchains have evolved from financial ledgers to universal platforms for secure data exchange. Their core strengths—immutability, decentralization, and auditability—make them attractive to critical infrastructures such as logistics, healthcare, and industrial IoT. A blockchain network maintains a distributed ledger where each participant, or node, holds a copy of the same ledger, and updates are applied through a consensus mechanism. This ensures synchronization among all nodes and consensus on transaction validity, preserving the integrity and security of the system. Blocks of data added to the ledger are cryptographically linked, ensuring that the information on the blockchain remains immutable and resistant to tampering or alteration. Due to these strengths, blockchain is often described as a "trust machine." However, this trust applies only to the integrity and immutability of the recorded data, not to its truthfulness or accuracy. Therefore, the reliability of a blockchain ultimately depends on the quality of the input data. While this limitation has been recognized in several studies [1], [2], limited progress has been made toward developing comprehensive solutions.

Previous approaches attempt to secure the data origin through oracle services or APIs that connect blockchains to off-chain sources [3]. However, most of these traditional oracles act as external data providers rather than internal validators. They ensure that external data are available to the blockchain, but the correctness of that data remains unverified. Such dependence on external sources reintroduces centralized trust assumptions. Other works integrate rule-based validation [4] into consensus or adopt redactable blockchains [5] that allow modifying incorrect records after storage. Others focus on adding new entries to flag incorrect or untrustworthy data after it has been detected [6]. While these methods help mitigate data quality issues, they either fail to evaluate the intrinsic reliability of the data content or compromise immutability and system performance.

To address these limitations, we propose Proof of Evaluation (PoEval), a trust-aware consensus mechanism for permissioned blockchains that verifies data trustworthiness before ledger inclusion. In PoEval, participating nodes act as independent evaluators, executing validated machine learning (ML) models to assess incoming data before consensus. An Evaluation Coordination Committee (ECC), formed from permissioned nodes, provides decentralized coordination of off-chain evaluation by assigning verification tasks and aggregating cryptographically signed evaluator decisions without embedding complex orchestration logic into the consensus layer. The final trust verdict is derived through quorum-based majority voting, ensuring that only data independently validated by multiple evaluators can proceed to block proposal and consensus.

The main contributions of this paper are outlined as follows:

- We introduce a novel trust-aware consensus mechanism, *PoEval*, for permissioned blockchains, which ensures that only trustworthy data are added to the ledger, preserving immutability while preventing the inclusion of inaccurate or unreliable data.
- We propose a data-centric trust-by-verification approach that treats incoming data as untrusted by default and employs ML models to evaluate data trustworthiness prior to blockchain storage, eliminating reliance on external data source reputations.
- We integrate an ML-based pre-consensus trust evaluation layer, enabling nodes to independently and collaboratively verify data reliability before block inclusion.
- We validate the proposed approach experimentally, demon-

ISBN 978-3-903176-82-9

strating a substantial reduction in the amount of untrustworthy data recorded on the blockchain.

The rest of this paper is organized as follows. Section II summarizes relevant prior research, Section III presents the proposed PoEval methodology, and Section IV describes the threat model and threat mitigation analysis. Section V discusses the experimental results. Section VI concludes the paper and outlines future research avenues.

II. Related Work

This section provides an overview of prior work related to data trust and validation in blockchain systems. Existing studies can be broadly classified based on the timing of validation: prior to data entry, at the time of recording, or after storage.

Before addition: Most approaches rely on oracles and APIs to source reliable data from external services. Oracles act as bridges between off-chain and on-chain environments, ensuring data integrity via cryptographic proofs and aggregation from multiple sources, as exemplified by Chainlink [3]. Similarly, APIs provide standardized access to data from institutional, weather, or supply-chain sources. However, these mechanisms are less effective for data generated in decentralized contexts such as the IoT, where sensor faults, tampering, and cyberattacks are common [6]. As a result, ensuring the accuracy and trustworthiness of IoT data prior to blockchain inclusion remains a significant challenge.

During addition: Several blockchain-based frameworks aim to enhance data quality and trust at the time of recording. Works such as [7]–[9] propose multi-layer data quality control mechanisms, lightweight IoT trust models, and adaptive validation techniques. DeTRM [1] tracks supply chain operations using reputation systems, while [4] enforces syntactic and semantic validation rules through smart contracts. Although these approaches improve data governance, they largely depend on external device characteristics, predefined rules, or static validation logic, and do not perform direct, fine-grained evaluation of the intrinsic trustworthiness of incoming data.

Consensus-based solutions extend this line of work by embedding trust or reputation mechanisms into the block validation process. Reputation-driven consensus protocols such as Proof-of-X-Repute [10] and blockchain reputation-based consensus mechanisms [11] dynamically adjust node influence based on historical behavior, thereby discouraging malicious participation. Other examples include Proof of Trust [12] for crowdsensing environments and ConSenseIoT [13], which integrates decentralized identities into consensus decisions. In parallel, outlier-aware consensus protocols have been proposed for Hyperledger Fabric [14], where low-rank statistical modeling techniques are employed to identify anomalous updates prior to consensus.

While these approaches enhance resilience against faulty or malicious nodes, their primary focus remains on *who* participates in consensus rather than *what data* is admitted to the ledger. ML, when employed, is typically used to infer long-term node reputation or detect statistical anomalies, and data validation is often indirect, coarse-grained, or performed after block inclusion. Consequently, incorrect or untrustworthy data may still be written on-chain before any correction occurs.

TABLE I: Comparative Analysis of our Contribution

Characteristic	Ref. **[20]**	Ref. **[5]**	Ref. **[18]**	Ref. **[14]**	Ref. **[15], [16]**	Ref. **[6], [19]**	**Our work**
Data trust assessment	–	–	–	✓	–	✓	✓
Reduces untrustworthy data on blockchain	–	✓	✓	✓	–	✓	✓
Integrated with consensus workflow	✓	–	–	✓	✓	–	✓
Uses ML models	–	–	–	–	✓	✓	✓
Prevents storing untrustworthy data on blockchain	–	–	–	–	–	–	✓
Preserves blockchain immutability	✓	–	✓	✓	✓	✓	✓

More recent studies have explored the use of ML to enhance the adaptability and efficiency of blockchain consensus mechanisms. Venkatesan and Rahayu [15] propose hybrid consensus designs that use ML for combining components of different consensus mechanisms such as Proof of Work (PoW), Proof of Stake (PoS), Delegated Proof of Stake (DPoS), and Practical Byzantine Fault Tolerance (PBFT). Similarly, BFTBrain [16] leverages reinforcement learning to adaptively control Byzantine fault tolerant (BFT) consensus behavior by switching among protocol configurations in response to workload variations and adversarial conditions. While these approaches demonstrate effective ML-driven consensus optimization, they primarily focus on protocol efficiency, assuming transaction or data correctness rather than explicitly verifying it.

After addition: Post-hoc strategies such as rollback and overturn mechanisms [17], as well as redactable [5] or correctable [18] blockchains, aim to mitigate the impact of harmful or incorrect data after it has already been stored. Although these methods provide recovery capabilities, they weaken immutability guarantees, introduce additional computational overhead, and fail to prevent the initial inclusion of untrustworthy data.

Collectively, these studies highlight the need for data-centric, decentralized validation mechanisms that are tightly integrated with blockchain consensus, ensuring that only trustworthy data are recorded on-chain. Existing ML-based trust evaluation approaches typically operate either before or after storage and are often combined with reputation mechanisms to gradually reduce the influence of untrustworthy sources [6]. Subsequent extensions have explored real-time trust assessment augmented with historical reputation to generate initial trust labels, which are then refined through validator-side ML checks [19]. While effective in improving post-hoc trust assessment, these approaches still permit untrustworthy data to be written to the blockchain prior to corrective action.

This work advances the state of the art by introducing an ML-integrated consensus mechanism that evaluates data trustworthiness *prior* to block inclusion, thereby admitting only data that have been independently verified by multiple evaluators using certified ML models. By embedding ML-based data trust evaluation as a first-class, pre-consensus primitive—while decoupling data trust from node reputation and still leveraging reputation for evaluator selection and block proposal

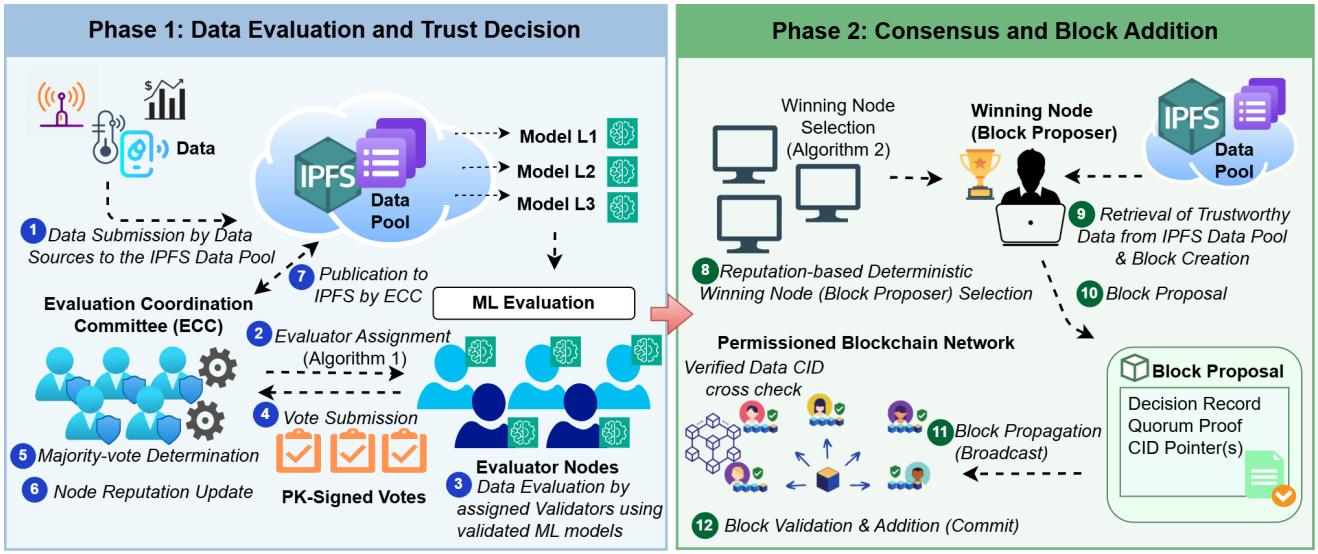

Fig. 1: PoEval Workflow

eligibility—PoEval bridges the gap between data-centric trust assessment and consensus-layer enforcement. A comparative summary in Table I shows that existing approaches neither directly assess data integrity within consensus nor integrate ML-based evaluation as a core consensus component, distinguishing PoEval from prior work.

III. Proposed Approach

This work presents a novel method for incorporating ML-based data validation into the blockchain consensus mechanism. By exploiting ML techniques such as anomaly detection and predictive modeling, the proposed system evaluates data quality and reliability prior to storage. An ECC coordinates this process by assigning evaluation tasks among blockchain nodes, aggregating their ML-based assessment outcomes, and updating node reputations based on historical evaluation accuracy, thereby enabling transparent and auditable operation across the network. Embedding this verification process within the consensus workflow ensures that only trustworthy data are recorded on-chain, effectively transforming the blockchain from a static immutable ledger into an adaptive, self-verifying system. The InterPlanetary File System (IPFS) is used as a decentralized, content-addressed off-chain storage layer for data, models, and evaluation artifacts, reducing on-chain storage overhead while preserving integrity, traceability, and auditability. PoEval is well suited for permissioned networks, where authenticated nodes, consortium governance, and controlled committee assignment enable effective reputation management, evaluation coordination, and oversight of ML models, datasets and data types.

The overall workflow, illustrated in Fig. 1, is organized into two main phases: (i) Data Evaluation and Trust Decision, coordinated by the ECC, and (ii) Consensus and Block Addition, led by the reputation-based winning node. As a prerequisite to these operations, ML models intended for use within the system must first undergo a dedicated Model Validation Process, conceptually illustrated in Fig. 2. Although depicted in a simplified form, this process follows a mechanism analogous to the data evaluation phase described later in this section.

A. Model Validation Process

In PoEval, a collection of standardized training and testing datasets is stored on IPFS, providing all participating nodes with consistent and verifiable resources for model training. To ensure that only reliable models are used for data validation, each trained model must undergo a certification phase comprising the following steps (Fig. 2). Approved models are published to an IPFS-based certified model registry and made available to evaluator nodes for subsequent data assessment. Evaluator nodes may use different certified model types, independently validated model instances, or different feature views from the certified model registry, provided that they satisfy the common certification criteria. This controlled heterogeneity improves evaluator independence and reduces the risk of correlated errors, while preserving common quality and verification standards.

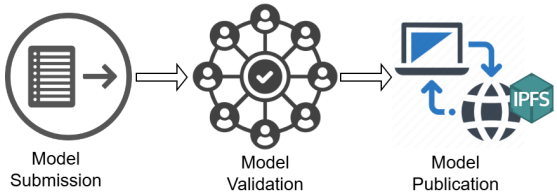

Fig. 2: Model Validation Process

1) Model Submission: Model providers upload the trained ML models and corresponding dataset hashes to IPFS, and submit the resulting content identifiers (CIDs) to the ECC, which initiates the evaluator assignment process.

2) Model Validation: The submitted models are distributed to evaluator nodes for independent testing. Each evaluator verifies the model's accuracy and consistency against predefined thresholds (e.g., 90%) using the reference datasets. Similar to the data evaluation process, evaluator assignments and voting are coordinated by the ECC to ensure fairness and transparency.

The selection of evaluator nodes follows the hybrid reputation-weighted and random sampling strategy described in Algorithm 1, ensuring that all eligible permissioned nodes retain a non-zero probability of participation, while favoring high-reputation and newly admitted nodes in a manner consistent with authenticated and governed network membership.

3) Model Publication: Only models that meet or exceed the performance threshold are approved and stored on IPFS. These validated models become part of the certified model registry accessible to evaluator nodes during the subsequent data evaluation phase.

B. Phase 1: Data Evaluation and Trust Decision

1) Data Submission: Entities such as users, organizations, and IoT sensor devices submit data to the network together with associated metadata (e.g., origin, timestamp, location, and a digital signature). The submitted data are stored in a distributed off-chain repository such as IPFS to support scalable evaluation without incurring on-chain storage overhead, while remaining content-addressed and retrievable for evaluation and audit purposes. The data submitter transmits the corresponding IPFS CID, along with the associated metadata and digital signature, to the ECC. Upon verifying the signature, the ECC collectively initiates the evaluator assignment process.

2) Evaluator Assignment: The ECC coordinates the assignment of evaluator nodes for both data and model validation tasks, as outlined in Algorithm 1. To balance reliability with fairness, PoEval employs a hybrid selection strategy that combines reputation-weighted and random exploration components. This ensures that high-reputation nodes are more likely to be chosen based on their proven accuracy, while lower-reputation or newly admitted nodes retain a non-zero probability of participation. Such hybrid sampling prevents dominance by a small subset of nodes, promotes diversity, and enhances resistance to collusion.

- **Reputation-weighted selection:** Nodes are selected with probability proportional to $w(m) = \sqrt{R(m)} + \varepsilon$. The square-root function is monotonic, ensuring that higher-reputation nodes retain higher selection likelihood and incentives for honest behavior, while its concavity introduces diminishing returns for very high $R(m)$, preventing evaluator dominance and reducing concentration or collusion risk. The ε term guarantees a non-zero selection probability for low-reputation or newly joined nodes.
- **Exploration-based selection:** A fraction η of slots is randomly assigned, ensuring opportunities for new or lower-reputation nodes and promoting long-term fairness.
- **Diversity guard (optional):** To minimize collusion, evaluator sets may exclude multiple nodes from the same organization per task.

3) Data Evaluation: Each selected evaluator independently assesses the trustworthiness of its assigned data using validated ML models retrieved from IPFS. If a node prefers to use a custom model, it must first submit the model through the validation process described earlier. Evaluators record detailed logs including data hash, model hash, and outputs for transparency.

Algorithm 1 Data and Model Evaluator Assignment

Require:
M: set of candidate nodes, each with reputation score $R(m) \in [0, 1]$ for $m \in M$
D: set of data (or model) items/subsets to be evaluated
k: number of evaluators to assign per $d \in D$
η: exploration ratio $(0 \leq \eta \leq 1)$
ε: small constant ensuring non-zero selection probability
L: maximum number of assignments per node per round
$\mathrm{count}(m)$: number of assignments already given to node m in the current round

Ensure: Assignment list A mapping each $d \in D$ to its evaluator set S_d

$A \leftarrow \emptyset$; $selected \leftarrow \emptyset$
Compute weights $w(m) \leftarrow \sqrt{R(m)} + \varepsilon \quad \forall m \in M$ ▷ sub-linear weighting to avoid dominance
$E \leftarrow \{m \in M \mid \mathrm{count}(m) < L\}$ ▷ eligible nodes under cooldown limit
for all $d \in D$ **do**
  $S_d \leftarrow \emptyset$
  $k_{\mathrm{rep}} \leftarrow \lfloor (1 - \eta)\, k \rfloor$ ▷ reputation-weighted quota
  $k_{\mathrm{exp}} \leftarrow k - k_{\mathrm{rep}}$ ▷ random exploration quota
  /* Reputation-weighted selection (exploitation) */
  $E_{\mathrm{rep}} \leftarrow E \setminus selected$
  Select k_{rep} nodes from E_{rep} with probability $\propto w(m)$
  Add selected nodes to S_d
  /* Random selection (exploration) */
  $E_{\mathrm{rnd}} \leftarrow E \setminus (selected \cup S_d)$
  Uniformly select k_{exp} nodes from E_{rnd}
  Add selected nodes to S_d
  /* Diversity guard (optional) */
  if multiple nodes from the same organization exist in S_d **then**
    Replace surplus nodes with next best candidates from $E \setminus S_d$
  end if
  Append (d, S_d) to A
  $selected \leftarrow selected \cup S_d$
end for
return A

4) Vote Submission: After evaluation, each node submits its classification (e.g., trustworthy or untrustworthy) as a vote signed using its private key to the ECC, together with the corresponding evaluation log stored on IPFS.

5) Majority-vote Determination: The ECC aggregates evaluator votes using a quorum-based majority-voting scheme to determine the trust level of each data item. A decision is accepted if at least a quorum of q out of k evaluator votes agree, where k is the number of assigned evaluators for the data item and q is the governance-defined quorum threshold (e.g., $q = \lceil 0.5k \rceil$). If consensus is inconclusive, additional evaluation rounds are triggered. This step produces the final trust decision for the evaluated data item.

6) Node Reputation Update: After each round of data evaluation or ML model certification, evaluator performance is assessed and reputation scores are updated accordingly. For evaluator node m in the node set M, the updated reputation after round t is computed as:

$$R(m, t+1) = (1-\gamma)\, R(m,t) + \gamma\, S(m,t), \tag{1}$$

where $S(m,t)$ denotes the accuracy ratio of node m—defined as the proportion of its evaluation outcomes that agree with the final consensus or quorum-based majority decision—and γ $(0 < \gamma < 1)$ controls the influence of the most recent round. All reputation values are then normalized to the interval $[0, 1]$ to maintain comparability across nodes. This update mechanism rewards sustained accuracy while penalizing inconsistent behavior, ensuring that long-term trustworthy nodes gradually accumulate higher reputation scores.

7) Publication to IPFS: After aggregation and normalization, the ECC publishes the final consensus outcomes for each evaluated data item, together with the updated normalized reputation values, to IPFS. This ensures network-wide visibility, traceability, and auditability of both trust decisions and evaluator performance over time.

In the permissioned PoEval setting, node participation is governed by a consortium-managed membership policy. New nodes may join the network only after successful authentication and approval by the governing authority. Upon admission, each node is assigned a neutral baseline reputation, typically set to a mid-range value (e.g., $R(m, 0) = 0.5$) or to the current network-wide average reputation, thereby ensuring equitable inclusion in subsequent evaluation rounds and avoiding bias against newly admitted participants. Evaluator selection follows a reputation-weighted probabilistic approach, where nodes with higher reputations have greater chances of being assigned validation tasks, while lower-reputation or newly joined nodes retain a non-zero probability of selection. This mechanism maintains fairness, prevents reputation lock-in, and mitigates Sybil attacks or collusion attempts by linking influence to authenticated identity and long-term performance. All authenticated participants can gradually improve their reputation through consistent and accurate performance across both model and data validation tasks, while inactive nodes experience gradual reputation decay until re-engagement, ensuring that only active and reliable participants retain high influence within the consensus process.

C. Off-Chain Storage and IPFS-Based Data Retrieval

PoEval adopts IPFS as an off-chain storage layer to avoid placing raw data or model artifacts directly on the blockchain and to support scalable, content-addressed data retrieval. For each submitted data item or model update x, the submitter uploads the payload (or an encrypted version) of it to IPFS and obtains a content identifier $\mathrm{CID}(x)$, which is a cryptographic hash uniquely bound to the content. Only the CID, together with minimal metadata such as a timestamp, submitter identity, and digital signature, is disseminated to the network. Designated evaluator nodes retrieve the payload by resolving $\mathrm{CID}(x)$ through the IPFS network and verify data integrity by recomputing the content hash and matching it against the CID. Each evaluator then applies a pre-validated ML model locally to assess the trustworthiness of the retrieved data or model update. In the case of encrypted payloads, evaluation is performed over authorized metadata or privacy-preserving representations, ensuring that raw data confidentiality is maintained. The evaluation output, such as a trust label or score, together with the corresponding data CID, model identifier, and evaluator signature, is stored off-chain on IPFS as an evaluation record.

Following evaluation, signed evaluation records are aggregated to derive a final trust decision for each data item. This decision is represented as a compact decision record containing the data CID, the final trust label, and a quorum proof derived from evaluator signatures. During block creation, only references to these decision records, and optionally a Merkle root computed over multiple decision records, are committed on-chain. Raw data and intermediate evaluation artifacts remain off-chain, while their integrity and auditability are preserved through content-addressed references.

D. Evaluation Coordination Committee (ECC) Architecture and Verification

In PoEval, the ECC is a distinct coordination layer separate from blockchain nodes that perform data evaluation and block consensus. It does not execute ML models, propose blocks, or validate blocks. Instead, it assigns evaluators, collects signed results, aggregates votes, and publishes verifiable coordination metadata. Blockchain nodes retrieve data and certified models from IPFS, execute ML inference locally, and submit signed outputs; thus, trust decisions are determined by evaluator votes, while the ECC provides coordination only.

PoEval targets permissioned consortium blockchains, where authenticated membership and governance are explicit parts of the system model. ECC members must therefore be authenticated consortium entities, and any addition, replacement, or removal of members requires consortium quorum approval. ECC composition may also enforce organizational diversity, including an optional maximum number of members per organization. For robustness, the ECC operates as a replicated committee of consortium-run coordination services executing identical logic. Deterministic leader rotation orders coordination actions only and does not affect evaluator judgments or final trust outcomes. If an ECC member becomes unavailable, other replicas continue operation; if it misbehaves, it can be removed through quorum-approved reconfiguration. All ECC operations are cryptographically verifiable. Evaluator assignments, aggregated voting outcomes, and reputation updates are signed and published with hashes and related metadata, enabling independent verification by blockchain nodes and peer ECC members.

E. Phase 2: Consensus and Block Addition

8) Winning Node Selection: The node responsible for proposing the next block (block proposer) is selected based on reputation, as formalized in Algorithm 2. Nodes with higher reputations are prioritized during selection, aligning consensus leadership with consistent evaluation accuracy. The selection process is fully deterministic—each node can independently

Algorithm 2 Winning Node Selection

Require:
M: set of candidate nodes, each with reputation score $R(m)$ for $m \in M$
h_{t-1}: previous block hash
k_p: leader candidate set size (top-k_p nodes by reputation)
Ensure: Selected winning node $m^\star$
1: $sorted \leftarrow \mathrm{sort}(M, \text{by } R(m) \text{ in descending order})$
2: $eligible \leftarrow \{sorted[1], \ldots, sorted[k_p]\}$
3: $R_{\text{scores}} \leftarrow \mathrm{concat}(R(m)\ \forall m \in eligible)$
4: $combined\_val \leftarrow \mathrm{hash}\big(\mathrm{encode}(R_{\text{scores}}) \parallel h_{t-1}\big)$
5: $winner\_index \leftarrow combined\_val \bmod k_p$
6: **return** $m^\star \leftarrow eligible[winner\_index]$

compute the same outcome using the shared reputation scores and the previous block hash, ensuring verifiable and tamper-resistant leader election without requiring additional coordination or randomness beacons. This mechanism is termed *Proof of Evaluation (PoEval)* because block proposal eligibility depends on reputation gained through participation in data and model evaluation tasks. Fairness is maintained since any authenticated node can be selected as an evaluator, allowing it to improve its reputation over time and eventually qualify for block proposal.

9) Retrieval of Trustworthy Data and Block Creation: Once selected, the block proposer retrieves all data items that have been verified as trustworthy through the ECC's consensus results stored on IPFS. Only those data entries that reached a final trust consensus are included in the block assembly. The proposer compiles the corresponding data hashes, reputation updates, and consensus metadata to construct a new block that represents an immutable record of validated information. This ensures that each block contains only verified, high-quality data, preventing unvalidated or malicious content from entering the ledger.

10) Block Proposal: The block proposer constructs the block including verified data references, decision records, and metadata such as the proposer's identifier, timestamp, and reference to the previous block hash.

11) Block Propagation (Broadcast): The proposed block is disseminated to all validator nodes in the blockchain network.

12) Block Validation and Addition: Validator nodes then independently verify the proposed block to ensure that: (i) all included data items correspond to entries confirmed as trustworthy by the data evaluation process, (ii) the block structure and cryptographic hashes are valid, and (iii) the proposer's identity and reputation satisfy the eligibility criteria. If the majority of validators confirm the block's validity, it is appended to the blockchain ledger, making the verified data immutable and globally auditable. The block proposer and participating evaluators may receive rewards (e.g., tokens or transaction fees) based on their reputation and contributions, although the specific reward mechanism lies outside the scope of this paper.

To preserve liveness, if the selected proposer fails to broadcast a valid block within a predefined timeout interval, or if multiple conflicting proposals are detected, the system triggers a re-election based on the next eligible node's reputation score. Timeout thresholds are parameterized with respect to block generation intervals to ensure responsiveness under variable network delays. In the event of temporary forks, PoEval resolves conflicts deterministically: the chain containing the highest number of valid blocks proposed by correctly elected nodes within their allowed time intervals is recognized as the canonical chain. This ensures that all honest nodes converge on a single consistent state while maintaining both safety and progress guarantees.

F. Consortium Governance and Threshold Management

In PoEval, trust evaluation thresholds and model validation criteria are governed by the consortium operating the permissioned blockchain. These parameters are not set by individual nodes but are defined through consortium-level governance policies agreed upon by authorized stakeholders (e.g., infrastructure operators, data owners, or regulatory representatives).

Model validation thresholds—such as minimum accuracy, robustness, or false-positive rate on reference datasets—are specified during system initialization and stored as configuration parameters accessible to all nodes. Updates to these thresholds follow a governance-driven process, requiring approval by a quorum of consortium members or designated governance committee, and are versioned to ensure auditability and reproducibility. Threshold updates are applied only at predefined epochs or block heights, preventing mid-round inconsistencies and ensuring that all evaluator nodes apply the same values in subsequent evaluation rounds, thereby maintaining consistent behavior across the network.

To ensure robustness against misconfiguration or delayed updates, PoEval defines fallback policies. If no active model satisfies the current validation thresholds, the system continues to operate using the most recently certified set of models, or temporarily applies conservative acceptance criteria. This prevents denial-of-service conditions arising from overly strict thresholds while preserving safety guarantees.

IV. Security Analysis

A. Threat Model

This section details the attack strategies that the system may face. PoEval is based on the following assumptions:

1) The system assumes that a quorum (i.e., a majority) of evaluator nodes behaves honestly for each evaluation task.
2) The ECC operates as a distributed committee composed of independently managed nodes, ensuring no single point of control or bias.
3) Adversarial attacks against the ML models are not explicitly considered in this work.

The threat model includes the following attack strategies:

- *Simple Attack*: A malicious node persistently submits incorrect or misleading data trustworthiness evaluations, irrespective of the true data quality, with the goal of degrading the overall evaluation accuracy or increasing the likelihood of untrustworthy data being accepted.
- *Camouflage Attack*: A malicious node provides correct data trustworthiness evaluations pretending to be honest,

and only attacks the system when it has reached a high reputation level and is selected as the block proposer.

- *Eclipse Attack*: A malicious node tries to isolate another node by controlling all the connections to and from that node. The attacker then feeds the isolated node false or manipulated data, which can lead to the propagation of invalid blocks within the network.
- *Sybil Attack*: A malicious node creates multiple fake identities (Sybil nodes) to gain a disproportionately large influence over the network. This could allow the attacker to disrupt network operations, manipulate consensus, or control the blockchain.
- *Free-riding or false-reporting*: Refers to nodes that avoid performing the actual work required for accurate data assessment, undermining the integrity of the evaluation mechanism by providing random or consistently identical votes, rather than contributing meaningful evaluations.

B. Threat Mitigation Analysis

This subsection analyzes how PoEval mitigates the potential threats outlined in the threat model.

1) Simple Attack: The system assumes that a quorum of evaluator nodes behaves honestly for each evaluation task. Accordingly, even if a single node consistently provides incorrect evaluations, its influence is limited by the requirement for a quorum-based majority to determine data trustworthiness. Final decisions are based on aggregated evaluations from multiple independent evaluators, not any single node.

2) Camouflage Attack: Although a node may accumulate a high reputation by behaving honestly and be selected as a block proposer, the data it proposes is independently verified by other nodes to ensure it was determined as trustworthy before being added to the blockchain. Honest nodes will identify and reject any block containing untrustworthy data. This means that even if a node that has gained a high reputation starts behaving maliciously, it cannot unilaterally compromise the system without the consensus of other nodes.

3) Eclipse Attack: In PoEval, block validity is determined by the inclusion of signed decision records and quorum proofs produced during the pre-consensus evaluation phase. As a result, even if a malicious node attempts to isolate honest participants and propagate invalid blocks, a temporarily isolated node can still verify the authenticity and integrity of a proposed block by validating evaluator signatures and content-addressed commitments, rather than relying solely on peer connectivity. Consequently, an adversary cannot successfully introduce blocks containing untrustworthy data without controlling a quorum of evaluators, thereby limiting the effectiveness of eclipse attacks.

4) Sybil Attack: Nodes in the network gain the opportunity to become a block proposer based on their reputation, which is built over time by acting honestly and in accordance with network rules. This reputation-based selection makes it costly and time-consuming for an attacker to build enough high-reputation Sybil nodes to significantly affect the network. Even if a malicious actor somehow manages to gain high reputation for multiple nodes, the quorum-based majority vote requirement for data evaluation acts as a second layer of defense.

5) Free-riding or false-reporting: Nodes are incentivized to utilize certified ML models to increase their chances of achieving higher alignment in quorum-based majority-voting. Submitting random or consistently identical votes will likely result in fewer matches, leading to significant reductions in reputation. This, in turn, decreases their likelihood of being selected as data evaluators or block proposers in the future.

V. Experiments

This section presents results from experiments conducted using prototype implementations. All implementations were done on a Windows 11 Pro (64-bit) machine with an Intel i7-1265U processor (1.80 GHz) and 32 GB of RAM.

A. Data Trust Classification Using Multiple ML Models

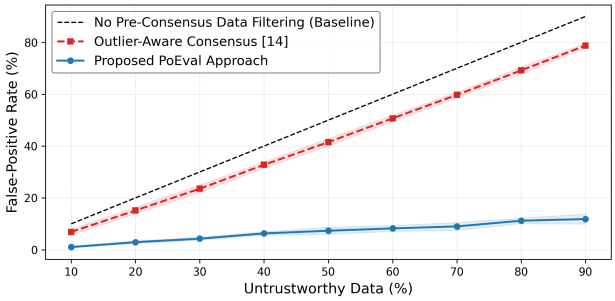

Fig. 3: FPR Evaluation Results on the Intel Lab Dataset

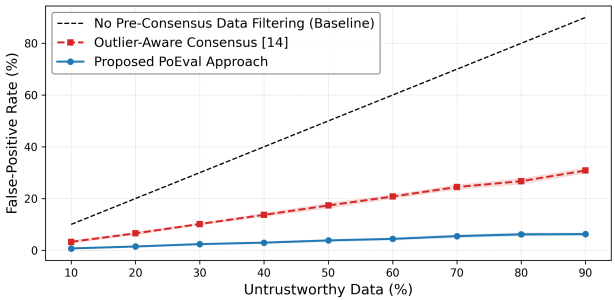

Fig. 4: FPR Evaluation Results on the Financial Dataset

For the ML-based trust evaluation, three datasets were employed to assess both domain-specific performance and cross-domain generalizability: two IoT sensor datasets, namely the Intel Lab dataset [21] and a modified version of the Labelled Wireless Sensor Network Data Repository (LWSNDR) [22], as well as a financial transaction dataset [23]. These datasets were selected for their relevance to data-driven blockchain applications, where ensuring data trustworthiness prior to on-chain inclusion is critical, and to evaluate the robustness of the proposed approach beyond the IoT domain.

The Intel Lab dataset consists of real-world sensor readings collected from a wireless sensor network (WSN) deployment at the Intel Berkeley Research Lab. For our experiments, temperature measurements were extracted, and additional untrustworthy samples were synthetically generated using the Random Walk Infilling (RWI) algorithm [24] to emulate realistic anomaly patterns observed in practical sensing environments. The modified LWSNDR dataset contains temperature and humidity readings from a WSN, in which sensor faults—including Offset, Gain,

TABLE II: FPR Evaluation Results on the Modified LWSNDR Dataset

Fault Type	Fault Rate 10% (FPR, %)		Fault Rate 20%		Fault Rate 30%		Fault Rate 40%		Fault Rate 50%	
	[14]	Ours	[14]	Ours	[14]	Ours	[14]	Ours	[14]	Ours
Offset β=2	7.93 ± 0.44	**0.10 ± 0.04**	17.95 ± 0.44	**0.11 ± 0.04**	27.50 ± 0.72	**0.10 ± 0.04**	37.70 ± 0.70	**0.09 ± 0.03**	47.80 ± 0.66	**0.09 ± 0.03**
β=4	7.51 ± 0.50	**0.10 ± 0.04**	17.42 ± 0.48	**0.09 ± 0.03**	27.22 ± 0.70	**0.09 ± 0.03**	37.33 ± 0.69	**0.09 ± 0.03**	47.35 ± 0.65	**0.09 ± 0.03**
β=6	7.15 ± 0.54	**0.10 ± 0.04**	16.92 ± 0.54	**0.09 ± 0.03**	26.33 ± 0.85	**0.09 ± 0.03**	36.20 ± 0.68	**0.09 ± 0.03**	46.19 ± 0.62	**0.09 ± 0.03**
β=8	6.23 ± 0.47	**0.10 ± 0.04**	16.14 ± 0.49	**0.08 ± 0.04**	25.81 ± 0.73	**0.08 ± 0.04**	35.97 ± 0.65	**0.08 ± 0.04**	45.98 ± 0.61	**0.08 ± 0.04**
β=10	6.16 ± 0.48	**0.09 ± 0.04**	15.96 ± 0.51	**0.09 ± 0.04**	25.71 ± 0.74	**0.08 ± 0.04**	35.90 ± 0.67	**0.08 ± 0.04**	45.92 ± 0.61	**0.08 ± 0.04**
Gain β=2	5.46 ± 0.46	**0.10 ± 0.03**	15.31 ± 0.47	**0.09 ± 0.03**	24.62 ± 0.75	**0.09 ± 0.03**	35.04 ± 0.75	**0.09 ± 0.03**	44.96 ± 0.60	**0.09 ± 0.03**
β=4	5.53 ± 0.46	**0.08 ± 0.03**	15.49 ± 0.46	**0.09 ± 0.03**	25.16 ± 0.76	**0.09 ± 0.03**	35.30 ± 0.72	**0.09 ± 0.03**	45.24 ± 0.61	**0.09 ± 0.03**
β=6	5.66 ± 0.49	**0.06 ± 0.03**	15.59 ± 0.49	**0.09 ± 0.03**	25.30 ± 0.77	**0.09 ± 0.03**	35.47 ± 0.76	**0.09 ± 0.03**	45.43 ± 0.65	**0.09 ± 0.03**
β=8	5.66 ± 0.49	**0.06 ± 0.03**	15.64 ± 0.50	**0.08 ± 0.03**	25.41 ± 0.75	**0.09 ± 0.03**	35.57 ± 0.77	**0.09 ± 0.03**	45.59 ± 0.61	**0.09 ± 0.03**
β=10	5.66 ± 0.48	**0.06 ± 0.03**	15.73 ± 0.51	**0.08 ± 0.03**	25.46 ± 0.77	**0.09 ± 0.03**	35.63 ± 0.79	**0.09 ± 0.03**	45.72 ± 0.66	**0.09 ± 0.03**
Stuck-at	6.38 ± 0.48	**0.09 ± 0.03**	16.56 ± 0.50	**0.09 ± 0.03**	26.51 ± 0.70	**0.09 ± 0.03**	37.12 ± 0.75	**0.09 ± 0.03**	46.84 ± 0.66	**0.09 ± 0.03**
Out-of-Bounds	6.00 ± 0.51	**0.09 ± 0.03**	16.10 ± 0.52	**0.09 ± 0.03**	25.89 ± 0.80	**0.09 ± 0.03**	36.01 ± 0.78	**0.09 ± 0.03**	46.03 ± 0.64	**0.09 ± 0.03**

Stuck-at, and Out-of-Bounds—were artificially injected at rates ranging from 10% to 50%, following the methodology described in [22]. To evaluate cross-domain generalizability, a financial transaction dataset containing fraud labels was also employed. The dataset, created by Caixabank Tech for the 2024 AI Hackathon, combines transaction records, customer attributes, and card-related information collected over an extended temporal span. Its heterogeneous feature composition and real-world fraud annotations make it well suited for assessing data trustworthiness in non-sensor domains, where data integrity directly impacts downstream decision-making. From this dataset, a balanced subset of 26,664 records was extracted for supervised model training and evaluation.

Five ML models were employed for binary trust classification: (i) a Support Vector Machine with an RBF kernel ($C = 1$), (ii) a Multi-Layer Perceptron using the Adam optimizer with a learning rate of 0.001, ReLU activation, and a single hidden layer containing 30 neurons, (iii) a Random Forest classifier with 300 trees and a maximum depth of 30, (iv) a distance-weighted K-Nearest Neighbours classifier with $k = 3$, and (v) an eXtreme Gradient Boosting (XGBoost) model with 100 estimators, a learning rate of 0.1, and a maximum depth of 3. For each data sample, predictions from the individual models were aggregated using a majority-voting scheme requiring agreement from at least three out of five models, thereby reducing the impact of individual model bias and variance. All models were implemented in Python.

To benchmark the proposed ML-driven trust evaluation approach, we compare it against the Outlier-Aware Consensus Protocol [14], which employs a low-rank, learning-based outlier detection mechanism to identify anomalous data prior to PBFT execution, without performing explicit ML–based data trust evaluation. Both approaches were evaluated on all three datasets under identical experimental conditions to ensure a fair comparison. Each experiment was repeated ten times, and performance was assessed using the False Positive Rate (FPR). In our experimental setup, trustworthy samples are treated as the positive class; consequently, the FPR corresponds to the false acceptance rate, i.e., the proportion of untrustworthy samples incorrectly accepted as trustworthy out of all evaluated samples. This reflects the admission of malicious or faulty data into the blockchain. Mean FPR values were reported together with 95% confidence intervals.

The results presented in Fig. 3, Fig. 4, and Table II show that the proposed ML-based ensemble consistently achieves lower false positive rates (FPRs) and higher overall accuracy across all datasets and fault injection scenarios. By leveraging complementary decision boundaries from multiple classifiers through majority voting, the proposed approach mitigates individual model limitations, resulting in more reliable trust assessment and ensuring that only trustworthy data are committed to the blockchain. In Fig. 3 and Fig. 4, in addition to the outlier-aware consensus mechanism, we include a reference baseline representing conventional consensus protocols that perform no pre-consensus data filtering. In such systems, all submitted data are recorded on-chain, causing the false positive rate to increase proportionally with the fraction of untrustworthy data. This reference baseline highlights the necessity of pre-consensus data trust evaluation and contextualizes the performance gains achieved by PoEval.

B. Processing Time Evaluation

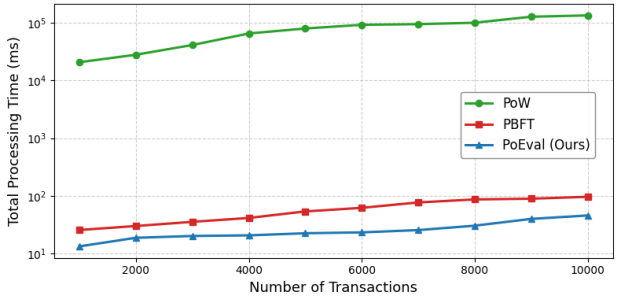

Fig. 5: Total Processing Time Across Consensus Mechanisms

For processing time evaluation, we developed a blockchain prototype in Python and integrated the proposed PoEval consensus mechanism. For comparative analysis, the prototype was also implemented with alternative consensus mechanisms, including PoW and PBFT, as used in the outlier-aware consensus protocol in [14]. The system exposes RESTful APIs to support core blockchain operations and inter-node interactions. To emulate a multi-node deployment, multiple blockchain instances were executed concurrently within a controlled experimental environment, with a total of 10 nodes participating in the consensus process. PoW difficulty and PBFT timeout parameters were

fixed across all experiments to ensure a consistent comparison. For each experiment, we measured the total processing time required to validate, reach consensus on, and commit a batch of transactions, starting from the submission of the first transaction in the batch until final ledger confirmation. This metric captures both computational costs and inter-node coordination overhead introduced by the consensus mechanism.

As shown in Fig. 5, PoEval consistently achieves lower total processing times than PBFT and PoW as transaction volume increases. This processing time captures the cumulative computational and coordination overhead of each consensus mechanism. PBFT exhibits substantially higher overhead due to message-intensive coordination, while PoW is dominated by cryptographic puzzle solving. By contrast, PoEval avoids both excessive coordination and costly cryptographic operations, resulting in more efficient transaction processing under increasing workloads. From a complexity standpoint, PoW remains computationally expensive because it relies on repeated hash-based puzzle solving, while PBFT incurs $O(n^2)$ communication overhead from all-to-all coordination among n participating nodes. PoEval, in contrast, requires only lightweight evaluator coordination and vote aggregation, scaling approximately as $O(k)$ per evaluated item for k selected evaluators, with proposer selection costing up to $O(n \log n)$ when reputation rankings over n participating nodes are recomputed each round.

VI. Conclusions and Future Work

This paper introduced PoEval, a data-centric, trust-by-verification consensus mechanism for permissioned blockchains. PoEval validates incoming data prior to consensus by assigning evaluator nodes that independently assess trustworthiness using certified ML models. An Evaluation Coordination Committee (ECC) orchestrates evaluator selection, aggregates model outputs, and updates node reputations within an authenticated and auditable framework. Experimental results show that PoEval achieves lower processing times than PBFT and PoW while significantly reducing false acceptance rates compared to conventional and outlier-aware consensus mechanisms. By embedding ML-driven validation into the consensus workflow, PoEval transforms the blockchain from a passive immutable ledger into an intelligent, self-validating system suitable for consortium-based applications such as supply chains, IoT data assurance, and compliance networks.

Future work will extend PoEval to large-scale and multi-domain settings, incorporate adaptive model retraining to address concept drift, and introduce robustness mechanisms against adversarial attacks on ML models. Exploring adaptations beyond strictly permissioned environments remains an additional research direction.

Acknowledgement

This work was partly supported by European Union in the ENSURE-6G project (Grant ID. 101182933).

References

[1] G. Dharma Putra, C. Kang, S. S. Kanhere, and J. Won-Ki Hong, "DeTRM: Decentralised Trust and Reputation Management for Blockchain-based Supply Chains," in *2022 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2022.

[2] R. Ratnayake, M. Liyanage, and L. Murphy, "Can We Trust Blockchain-IoT Data?" *IEEE Internet of Things Magazine*, 2025.

[3] L. Breidenbach, C. Cachin, B. Chan, A. Coventry, S. Ellis, A. Juels, F. Koushanfar, A. Miller, B. Magauran, D. Moroz *et al.*, "Chainlink 2.0: Next steps in the evolution of decentralized oracle networks," *Chainlink Labs*, vol. 1, pp. 1–136, 2021.

[4] C. A. Ardagna, R. Asal, E. Damiani, N. E. Ioini, M. Elahi, and C. Pahl, "From trustworthy data to trustworthy IoT: A data collection methodology based on blockchain," *ACM Transactions on Cyber-Physical Systems*, vol. 5, no. 1, pp. 1–26, 2020.

[5] Y. Hou, J. Zou, L. Wang, X. Lu, X. Lu, and M. Wang, "PRBCP: Publicly Redactable Blockchain With Off-Chain Reputation-Based Consensus Protocol," *IEEE Transactions on Network and Service Management*, vol. 22, no. 5, pp. 4603–4618, 2025.

[6] R. Ratnayake, M. Liyanage, and L. Murphy, "Trust management and bad data reduction in internet of vehicles using blockchain and AI," in *IEEE 97th Vehicular Technology Conference*. IEEE, 2023.

[7] C. Cappiello, M. Comuzzi, F. Daniel, and G. Meroni, "Data quality control in blockchain applications," in *Business Process Management: Blockchain and Central and Eastern Europe Forum*. Springer, 2019.

[8] S. Rouhani and R. Deters, "Data trust framework using blockchain technology and adaptive transaction validation," *IEEE Access*, vol. 9, pp. 90 379–90 391, 2021.

[9] V. Dedeoglu, R. Jurdak, G. D. Putra, A. Dorri, and S. S. Kanhere, "A trust architecture for blockchain in IoT," in *Proceedings of the 16th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. Association for Computing Machinery, 2020.

[10] E. K. Wang, R. Sun, C.-M. Chen, Z. Liang, S. Kumari, and M. Khurram Khan, "Proof of X-repute blockchain consensus protocol for IoT systems," *Computers & Security*, 2020.

[11] M. T. de Oliveira, L. H. Reis, D. S. Medeiros, R. C. Carrano, S. D. Olabarriaga, and D. M. Mattos, "Blockchain reputation-based consensus: A scalable and resilient mechanism for distributed mistrusting applications," *Computer Networks*, 2020.

[12] J. Zou, B. Ye, L. Qu, Y. Wang, M. A. Orgun, and L. Li, "A proof-of-trust consensus protocol for enhancing accountability in crowdsourcing services," *IEEE Transactions on Services Computing*, vol. 12, no. 3, pp. 429–445, 2018.

[13] H. Niavis and K. Loupos, "ConSenseIoT: a consensus algorithm for secure and scalable blockchain in the IoT context," in *Proceedings of the 17th International Conference on Availability, Reliability and Security*. Association for Computing Machinery, 2022.

[14] M. Salimitari, M. Joneidi, and M. Chatterjee, "AI-enabled blockchain: An outlier-aware consensus protocol for blockchain-based IoT networks," in *IEEE Global Communications Conference*. IEEE, 2019.

[15] K. Venkatesan and S. B. Rahayu, "Blockchain security enhancement: an approach towards hybrid consensus algorithms and machine learning techniques," *Scientific Reports*, vol. 14, no. 1, p. 1149, 2024.

[16] C. Wu, H. Qin, M. J. Amiri, B. T. Loo, D. Malkhi, and R. Marcus, "{BFTBrain}: Adaptive {BFT} consensus with reinforcement learning," in *22nd USENIX Symposium on Networked Systems Design and Implementation (NSDI 25)*, 2025.

[17] A. Carvalho, J. W. Merhout, Y. Kadiyala, and J. Bentley II, "When good blocks go bad: Managing unwanted blockchain data," *International Journal of Information Management*, vol. 57, p. 102263, 2021.

[18] A. Marsalek and T. Zefferer, "A correctable public blockchain," in *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications (TrustCom)*, 2019.

[19] R. Ratnayake, M. Liyanage, and L. Murphy, "Evaluating data trust in blockchain-based IoT systems using machine learning techniques," in *IEEE 22nd Consumer Communications & Networking Conference*, 2025.

[20] O. Aluko and A. Kolonin, "Proof-of-reputation: An alternative consensus mechanism for blockchain systems," *International Journal of Network Security & Its Applications*, vol. 13, 2021.

[21] P. Bodik, W. Hong, C. Guestrin, S. Madden, M. Paskin, and R. Thibaux, "Intel lab data," 2004.

[22] S. Zidi, T. Moulahi, and B. Alaya, "Fault detection in WSNs through SVM classifier," *IEEE Sensors Journal*, vol. 18, no. 1, 2018.

[23] CaixaBank Tech, "Financial transactions dataset: Analytics," Kaggle dataset, AI Hackathon, 2024, Accessed: 2026-02-04. [Online]. Available: https://www.kaggle.com/datasets/computingvictor/transactions-fraud-datasets

[24] T. Tadj, R. Arablouei, and V. Dedeoglu, "On evaluating IoT data trust via machine learning," *Future Internet*, vol. 15, no. 9, 2023.