STAP
Smart Technologies Academic Press

**STAP Journal of Security Risk Management**

https://www.jsrm.thestap.com/


# Securing Trust: Rule-Based Defense Against On/Off and Collusion Attacks in Cloud Environments

**Qais Al-Na'amneh[1*], Mahmoud Aljawarneh[2], Ahmad Saleh Alhazaimeh[3], Rahaf Hazaymih[1], Shahid Munir Shah[4] and Walid Dhifallah[5]**

[1] *Faculty of Engineering and Technology, Sindh University, Jamshoro, Pakistan*
[2] *Faculty of Computer Science, Stratford University, Virginia, USA*
[3] *Faculty of Economics and Administration King Abdul-Aziz University Jeddah, KSA*
[4] *Faculty of Engineering Sciences and Technology Hamdard University Karachi, Pakistan*
[5] *Laboratories Hatem Bettaher (IRESCOMATH), University of Gabes, Gabes, Tunisia*

**ARTICLE INFO**

**Article History**

**Received:** 17-08-2025
**Revised:** 30-10-2025
**Accepted:** 28-11-2025
**Published:** 01-12-2025

**Vol.2025, No.1**

**DOI:**

***Corresponding author.**
***Email:***
*Al-Na'amnehadd@gmail.com*

**Orcid:**

https://orcid.org/0009-0008-3034-7693



**ABSTRACT**

The pervasive adoption of cloud computing, underscored by its distributed, multi-tenant characteristics, introduces intricate vulnerabilities concerning trust assurance. Malicious entities increasingly deploy sophisticated stratagems, such as on/off behavioral subterfuge and orchestrated collusion, to subvert conventional trust assessment mechanisms. This manuscript introduces a Hierarchical Rule-Based Trust Orchestration System (HRTOS), a non-learning, deterministically governed architectural framework designed for the proactive identification and mitigation of these insidious threats within federated cloud environments. HRTOS operates through a synergistic ensemble of modules dedicated to multi-vector behavioral fingerprinting, contextual anomaly evaluation, feedback integrity validation, and collusion pattern grammar analysis. The system's core philosophy emphasizes operational transparency, imposing minimal computational burden while exhibiting acute sensitivity to nuanced deviations from normative interaction patterns. Rigorous simulations employing diverse synthetic user archetypes—spanning consistent integrity, strategic deception, and coordinated malevolence—demonstrate HRTOS's pronounced capability to accurately discern legitimate activities from complex reputation manipulation endeavors. Conventional trust paradigms, frequently reliant on computationally intensive machine learning or opaque probabilistic models, often falter when confronted by adaptive adversaries exploiting systemic latencies, sparse data conditions, or the inherent "black-box" nature of such models. HRTOS circumvents these limitations by employing a layered, context-aware rule engine that processes interaction telemetry and feedback metadata in near real-time. Abrupt behavioral transitions are identified via a multi-faceted deviation index; anomalous feedback is systematically de-weighted through source credibility and content plausibility checks; collusive engagements are surfaced by analyzing reciprocity dynamics and group behavioral coherence. Trust state adjudication is effectuated through deterministic rule sets, fostering auditable enforcement and low-latency response. The presented evaluations, encompassing varied attack vectors including sophisticated on/off attacks and multi-entity collusion schemes, affirm the model's high fidelity in threat differentiation, its negligible false positive incidence, and its inherent interpretability, rendering HRTOS exceptionally suitable for securing dynamic, federated cloud ecosystems where accountability, efficiency, and proactive threat neutralization are paramount.

**Keywords:** Security of Cloud Computing, Malicious Attacks, Rule-Based Defense.

**How to cite the article**

## 1. Introduction

Cloud computing infrastructures, having transcended their initial conceptualization as elastic resource pools, now constitute the digital bedrock upon which a significant quantum of global economic and societal functions depend [1]. The paradigm's defining attributes—on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service—have undeniably revolutionized information technology provisioning, fostering unprecedented innovation and operational agility across diverse sectors [2]. This migration towards utility-based computing, however, inherently dilates the trust perimeter, compelling a re-evaluation of security postures traditionally anchored in well-defined physical or network boundaries [3]. Within these sprawling, often federated, digital ecosystems, interactions among myriad services, users, and autonomous agents occur with a velocity and complexity that challenge conventional identity and access management frameworks [4]. Trust, in this expanded context, evolves beyond mere cryptographic authentication to encompass a dynamic, behaviorally-derived assurance of an entity's reliability and benign intent [5]. This behavioral dimension of trust transforms it into a continuously assessed, quantifiable metric—a valuable asset for legitimate participants and a prime target for malicious actors seeking to exploit system resources or compromise data integrity [6]. Adversaries employ increasingly sophisticated tactics, moving beyond brute-force attacks to insidious manipulations of trust and reputation systems themselves. Among the most challenging are *on/off attacks*, where entities meticulously cultivate credibility through compliant behavior before abruptly engaging in damaging activities, and *collusion attacks*, where groups of entities coordinate to artificially inflate their own reputations or denigrate those of legitimate actors [7]. Such exploits are particularly effective against trust models that rely on long-term averages, exhibit slow adaptation, or lack mechanisms to scrutinize the context and coordination of behaviors. Prevailing academic and commercial efforts to address these trust lacunae have predominantly gravitated towards machine learning (ML) and artificial intelligence (AI) techniques, including Bayesian inference, fuzzy logic, reinforcement learning, and deep neural networks [8]. While these approaches offer powerful pattern recognition capabilities and can adapt to evolving behavioral landscapes, they are not without significant drawbacks. Many ML models function as "black boxes," rendering their decision-making processes opaque and difficult to audit—a critical deficiency in environments requiring accountability and forensic traceability [9].They often necessitate vast quantities of high-quality labeled training data, which may be unavailable or expensive to acquire, especially for novel attack vectors [10]. Furthermore, ML models can be susceptible to adversarial attacks, such as data poisoning or evasion, where attackers subtly manipulate input data to mislead the model [11]. The computational overhead associated with training and, in some cases, inferencing complex models can also be prohibitive for real-time applications in resource-constrained segments of cloud ecosystems.

This paper posits that a deterministic, rule-based architectural paradigm offers a compelling alternative, capable of surmounting many of these limitations, particularly for discerning and neutralizing strategic trust exploits like on/off and collusion attacks. Such systems, founded upon explicitly defined logical constructs, provide inherent transparency, ensure predictable behavior, and typically incur significantly lower computational costs [12]. While sometimes perceived as less sophisticated than their ML counterparts, well-designed rule-based systems can achieve high levels of accuracy and responsiveness when tailored to specific threat models and operational contexts. We introduce the Hierarchical Rule-Based Trust Orchestration System (HRTOS), a novel framework engineered for proactive trust management in federated cloud environments. HRTOS distinguishes itself through a multi-layered architecture that integrates granular behavioral analysis, context-aware rule modulation, sophisticated feedback integrity checks, and advanced collusion pattern recognition. It operates without reliance on statistical learning algorithms, ensuring that all trust assessments and subsequent actions are derived from transparent, auditable logical rules. The system is designed to identify subtle behavioral precursors to malicious activity and to respond swiftly to detected threats, thereby minimizing the window of opportunity for attackers. The primary contributions of this manuscript are threefold:

1 . A comprehensive, modular rule-based trust architecture (HRTOS) specifically designed to counteract on/off attacks and diverse forms of collusion by leveraging multi-vector behavioral fingerprinting and context-sensitive rule application. This architecture emphasizes determinism and interpretability.

2. The introduction of novel rule-based mechanisms, including a Multi-faceted Deviation Index (MDI) for capturing nuanced behavioral shifts beyond simple score changes, and a Collusion Suspicion Index (CSI) derived from a grammar of collusive interaction patterns.

3. An extensive empirical evaluation using synthetic simulation scenarios that model a spectrum of user behaviors, including various sophisticated adversarial strategies. These simulations rigorously demonstrate HRTOS's efficacy in

terms of detection accuracy, response latency, resilience to manipulation, and low false positive rates, without the need for pre-training or complex model tuning.

The subsequent sections of this paper are organized as follows: Section II provides a critical review of related work in trust management, as summarized in Table 1, focusing on approaches relevant to cloud computing and the specific challenges of on/off and collusion attacks. Section III elaborates on the architectural design and operational logic of the proposed HRTOS, detailing its constituent modules and rule structures, with key rule categories highlighted in Table 2. Section IV describes the simulation methodology, including the design of the simulation environment and the specific behavioral archetypes (see Table 4) used for evaluation. Section V presents and comprehensively discusses the simulation results, analyzing the performance of HRTOS under various conditions, with performance metrics summarized in 5 and compared qualitatively in Table 6. Finally, Section VI concludes the paper, summarizing the key findings and outlining promising avenues for future research.

## 2. Related Work

The domain of trust management in distributed computing systems, particularly within the expansive and dynamic context of cloud environments, has been the subject of extensive research for several decades. This intellectual pursuit has yielded a diverse array of models and mechanisms, each endeavoring to quantify and manage the elusive concept of trust. Early formalizations drew heavily from sociological and philosophical underpinnings, attempting to translate human notions of trust into computational frameworks [13]. These foundational works paved the way for more concrete computational trust models, which began to emerge with the rise of decentralized systems like P2P networks and e-commerce platforms [14]. The challenges inherent in cloud computing—multi-tenancy, resource abstraction, dynamic scaling, and federated service composition—have further catalyzed innovation in this space, demanding more sophisticated and resilient trust solutions [15]. Table 1 provides a comparative overview of these approaches.

### *2.1 Foundational Trust Concepts and Early Computational Models*

The conceptualization of trust in computational systems often revolves around an entity's expectation regarding the future behavior of another entity, typically based on past interactions or referred reputation [16]. Key properties sought in computational trust models include transitivity (if A trusts B, and B trusts C, can A infer trust in C?), composability (how can trust in different aspects of an entity be combined?), and scalability (can the model operate efficiently in large systems?) [17]. Early models like EigenTrust [18] focused on global reputation aggregation in P2P networks, while others, like those based on Dempster-Shafer theory or subjective logic, provided frameworks for reasoning with uncertain or incomplete trust evidence [19]. These pioneering efforts laid crucial groundwork but often assumed relatively stable environments or lacked mechanisms to counter strategic manipulation

effectively [20].

### *2.2 Trust Management Paradigms in Cloud Computing*

The transition to cloud computing introduced unique complexities for trust management. Service level agreements (SLAs) became an early proxy for trust, with providers offering contractual guarantees on performance and availability [21].

However, SLA compliance provides only a partial view and does not inherently address behavioral trustworthiness or malicious intent. Reputation systems, adapted from P2P and e-commerce, found application in IaaS, PaaS, and SaaS contexts, allowing users to rate services and providers [22]. These systems, while useful, are themselves vulnerable to manipulation through fake reviews or orchestrated campaigns [23]. Hardware-assisted trust mechanisms, leveraging technologies like Trusted Platform Modules (TPM) and Intel SGX, offer a different dimension by providing attestations of platform integrity and secure execution environments [24]. While valuable for establishing foundational trust in the infrastructure, these hardware-based approaches typically do not address the behavioral trust of users or services operating atop that infrastructure. The inherent dynamism and scale of cloud federations, where services from different providers interoperate, further complicate trust establishment and maintenance across administrative domains [25].

**Table 1.** Comparative Overview of Trust Management Approaches from Literature

| Approach Category | Key Examples / Technologies | Core Mechanisms & Characteristics | Reported Strengths | Key Limitations (esp. On/Off & Collusion) | Relevance to HRTOS / Gap Addressed |
|---|---|---|---|---|---|
| Foundational | EigenTrust, Dempster-Shafer, Subjective Logic | Global reputation; uncertainty reasoning | Pioneering; handles incomplete evidence | Assumed stable env.; vulnerable to strategic manip.; limited dynamic threat defense. | Basic concepts; HRTOS aims for robust, specific threat detection. |
| Cloud-Specific | SLAs, P2P-adapted Reputation Systems, TPM/SGX | Contractual guarantees; user ratings; hardware attestation | Baseline compliance; user feedback; hardware integrity | SLAs partial; reputation systems prone to fake reviews/collusion; hardware not for behavioral trust. | HRTOS focuses on behavioral trust not fully covered. |
| ML - General | Bayesian Nets, Fuzzy Logic, RL, Deep Learning | Probabilistic inference; linguistic variables; policy learning; complex pattern recognition | Adaptive; handles uncertainty; learns from data | Opacity; data hunger; high cost; adversarial ML vulnerability; explainability gap. | HRTOS: deterministic, transparent alternative. |
| ML - Specific Sub-types | (RNNs, GNNs for DL) | Capture temporal/relational patterns | High accuracy on complex data (DL) | ”Black-box”; high data/compute needs (DL); specific design challenges for others. | HRTOS: interpretability, lower overhead. |
| On/Off Attack Defenses | TATW, adaptive windowing, change-point detection | Focus on recent behavior; detect sudden shifts | Targets ”Jekyll & Hyde” directly | Distinguishing genuine vs. malicious change is hard; often point solutions. | HRTOS integrates MDI as core, multi-faceted detection. |
| Collusion Attack Defenses | Graph-based (GNNs), feedback characteristic analysis, game theory | Analyze interaction networks; identify coordinated anomalies | Detects group manipulation | Sophisticated colluders evade; complex graph analysis scalability. Often specialized. | HRTOS incorporates CPAM with diverse rules as integral part. |
| Rule-Based (Traditional) | Expert Systems | Explicit ”IF-THEN” logic | Transparency, determinism, auditability, efficiency | Labor-intensive rule creation; potential rigidity; evasion if rules static/simple. | HRTOS builds on strengths, adds hierarchy, context-awareness (CMM). |

### *2.3 Machine Learning and Probabilistic Approaches to Trust*

In response to the complexity and dynamism of cloud environments, many researchers have turned to machine learning (ML) and probabilistic models to develop more adaptive and nuanced trust assessment systems. Bayesian networks have been widely used to model trust relationships and infer trustworthiness based on probabilistic dependencies between various evidence sources and past behaviors [26]. They are adept at handling uncertainty but can be computationally intensive and may require careful prior probability elicitation. Fuzzy logic systems offer an alternative for managing the imprecise and linguistic nature of trust, allowing for ”degrees of truth” rather than binary trusted/untrusted states [27]. Designing appropriate fuzzy rules and membership functions, however, can be a complex, domain-specific task. Reinforcement learning (RL)[28] has emerged as a promising paradigm for enabling trust models to learn optimal policies through interaction with the environment, adapting to changing behaviors over time [29]. RL agents can discover sophisticated trust assessment strategies but often require extensive exploration phases and can be sensitive to the reward function design. Deep learning techniques, including Recurrent Neural Networks (RNNs) and Graph Neural Networks (GNNs), have been applied to capture complex temporal dependencies in user behavior and intricate relational patterns in trust networks [30]. These models can achieve high accuracy but are notorious for their ”black-box” nature, data hunger, and significant computational demands for training [31].

Despite their power, ML-based trust systems face several critical challenges. The explainability gap, or the difficulty in understanding how these models arrive at their decisions, is a major concern, especially in security-critical applications where auditability and accountability are paramount [32]. They are also vulnerable to various forms of adversarial attacks, where malicious actors specifically craft inputs to deceive the model, such as data poisoning during training or evasion attacks during inference [33]. The reliance on large, representative datasets for training can also be a bottleneck, particularly for detecting novel or rare attack patterns.

### *2.4 Addressing Specific Trust Exploits: On/Off and Collusion Attacks*

Two particularly insidious forms of trust exploitation are on/off attacks and collusion attacks, which many conventional trust models struggle to detect effectively[34]. On/Off Attacks, also known as ”whitewashing and sudden abuse” or ”Jekyll and Hyde” behavior, involve an attacker initially behaving honestly to build a positive reputation, only to switch to malicious activity once sufficient trust has been accrued [35]. Detecting such attacks requires mechanisms sensitive to sudden behavioral shifts. Time-Aware Trust Windows (TATW) [36] and adaptive windowing techniques [37] attempt to address this by giving more weight to recent behaviors. Change-point detection algorithms from statistics [38] and behavioral drift analysis frameworks [39] also offer relevant methodologies. However, distinguishing genuine behavioral changes from malicious turns, especially in noisy environments, remains a challenge.

Collusion Attacks involve multiple entities coordinating their actions to manipulate trust or reputation scores. This can manifest as self-promoting collusion (multiple attackers giving each other unfairly high ratings) or slandering/badmouthing collusion (attackers coordinating to give unfairly low ratings to a target victim) [40]. Detecting collusion often requires analyzing patterns of interaction and feedback beyond individual entities. Graph-based methods, including those using GNNs or community detection algorithms, have shown promise in identifying collusive groups by analyzing the structure of the feedback network [41]. Analysis of feedback characteristics, such as temporal synchronization, rating distribution anomalies, content similarity, and reciprocity patterns, can also reveal collusive behavior [42]. Game-theoretic models have been used to understand the incentives and strategies involved in collusion [43]. However, sophisticated colluders may employ subtle tactics to evade detection, such as varying their attack patterns or involving a large number of sparsely connected entities. Advanced threats like Sybil attacks, where an adversary creates a multitude of fake identities, often underpin collusive activities [44]. Reputation whitewashing, where an entity discards a bad reputation by creating a new identity, is another related challenge [45].

### *2.5 The Resurgence and Advantages of Rule-Based Systems*

Amidst the increasing complexity of ML-driven solutions, there is a renewed appreciation for the strengths of rulebased systems, particularly in domains requiring transparency, determinism, and efficiency. Originating from the era of expert systems [46], rule-based approaches encode knowledge and decision logic as explicit ”IF-THEN” rules. This inherent transparency makes their reasoning processes easy to understand, verify, and audit, which is crucial for regulatory compliance and user acceptance in sensitive applications like cloud security [47]. Their deterministic nature ensures that given the same inputs, the system will always produce the same output, providing predictability and facilitating debugging [48].

Rule-based systems typically have a much lower computational footprint compared to complex ML models, especially deep learning networks, making them suitable for resource-constrained environments or applications requiring very low latency responses [49]. While crafting a comprehensive and robust rule set can be labor-intensive, it often parallels the effort required for feature engineering in ML. Furthermore, rule-based systems are not susceptible to the types of adversarial attacks that plague ML models by targeting the learning process itself. Recent trends also explore hybrid approaches, where rule-based systems provide a baseline level of interpretability and robustness, potentially augmented by ML components for tasks like pattern discovery or anomaly scoring, with human-in-the-loop validation [50]. The proposed HRTOS builds upon the strengths of the deterministic rule-based paradigm, aiming to provide a robust, transparent, and efficient solution for complex trust challenges in cloud federations.

Existing literature, while rich, often presents solutions that are either too computationally demanding, too opaque for critical infrastructure, or not sufficiently agile to counter strategic, coordinated attacks. The gap for a lightweight, interpretable, yet powerful trust mechanism specifically addressing the nuances of on/off and collusion threats in federated settings motivates the design of HRTOS. This work seeks to demonstrate that a crafted hierarchical rule system can offer comparable or even superior performance in these specific adversarial contexts while maintaining crucial operational advantages.

## 3. Proposed Hierarchical Rule-Based Trust Orchestration System (HRTOS)

The Hierarchical Rule-Based Trust Orchestration System (HRTOS) is conceptualized as a multi-layered, deterministic framework designed to provide robust and interpretable trust assessment in dynamic cloud environments, with a specific focus on mitigating sophisticated on/off and collusion-based exploits. Its architecture eschews probabilistic inference and machine learning algorithms in favor of explicitly defined rules and logical constructs, ensuring transparency, auditability, and predictable performance. HRTOS operates by continuously processing interaction telemetry and feedback data, applying a cascade of rule-based evaluations to derive trust scores and identify potentially malicious entities or behaviors. A summary of rule categories within HRTOS modules is presented in Table 2.

### *3.1 Architectural Overview*

HRTOS is composed of six core interoperable modules, as illustrated in Figure 1. Each module performs a distinct set of analytical functions, contributing to a holistic trust assessment:

1. Interaction Logging and Feature Vectorization Module (ILFVM): Captures and preprocesses raw interaction data.

2. Behavioral Anomaly Detection Module (BADM): Analyzes interaction patterns for deviations from established norms or suspicious signatures.

3. Feedback Integrity Verification Module (FIVM): Scrutinizes user-provided feedback for authenticity and credibility.

4. Collusion Pattern Analysis Module (CPAM): Identifies coordinated manipulative behaviors among groups of entities.

5. Contextual Modifier Module (CMM): Adjusts the sensitivity and response of other modules based on prevailing system-wide or service-specific contextual factors.

6. Trust Aggregation Engine (TAE): Consolidates inputs from all analytical modules to compute and update entity trust scores and states.

Data flows sequentially and in feedback loops between these modules. Raw events are ingested by ILFVM, processed features are passed to BADM, FIVM, and CPAM, whose outputs, modulated by CMM, are then integrated by TAE to update trust states. These trust states can, in turn, influence future processing within modules like FIVM (source credibility of feedback providers).

### *3.2 Interaction Logging and Feature Vectorization Module (ILFVM)*

The ILFVM serves as the primary data ingestion point for HRTOS. It is responsible for collecting granular telemetry from diverse sources within the cloud environment. This includes, but is not limited to:

- API call invocations (endpoint, parameters, frequency, success/error rates).

**Table 2**. Summary of HRTOS Rule Categories by Module

| Module | Key Rule Category/Logic | Brief Description of Purpose |
|---|---|---|
| BADM | Intra-Interaction Anomaly Rules | Assesses consistency and plausibility of features within single interaction sessions (e.g., payload anomalies, sequence breaks). |
| | Multi-faceted Deviation Index (MDI) Rules | Quantifies overall behavioral fingerprint changes against historical baselines to detect evolving shifts (on/off attacks). |

| | | |
|---|---|---|
| FIVM | Source Credibility Rules | Evaluates reliability of feedback provider based on historical accuracy, trust score, and newcomer status. |
| | Content Plausibility Rules | Examines feedback content for deviation from peer consensus, linguistic anomalies, and temporal bursts. |
| | Adjusted Feedback Score Calculation | Combines source and content assessments to derive a weighted feedback score. |
| CPAM | Reciprocity/Symmetry Detection Rules | Identifies suspicious dyadic or N-way cyclic promotional feedback patterns. |
| | Coordinated Slandering Detection | Flags groups providing similar negative feedback against a target in a short window. |
| | Group Anomaly Rules | Analyzes collective group behavior for anomalous cohesion or sudden formation activity. |
| | Collusion Suspicion Index (CSI) Calculation | Aggregates evidence of involvement in collusive patterns. |
| CMM | Global Threat Level Adaptation Rules | Adjusts system sensitivity (thresholds, weights) based on system-wide threat intelligence. |
| | Service Criticality Adaptation Rules | Mandates stricter scrutiny for interactions involving critical services. |
| | Federation Partner Trust Level Rules | Applies baseline vigilance increments for interactions with partners of known lower security posture. |

- Resource utilization patterns (CPU, memory, storage I/O, network bandwidth consumption over time).
- Network communication flows (source/destination IPs, ports, protocols, data volume, connection duration).
- Data access semantics (types of data accessed, access frequency, typical operational sequences).

Collected raw logs are timestamped, normalized, and parsed. The module then extracts salient temporal and sequential features. Temporal features include interaction frequencies, durations, inter-arrival times of specific events, and rates of change. Sequential features capture patterns in the order of actions, such as common API call sequences or deviations from typical operational workflows. For each entity $u$ at time $t$, ILFVM generates a multi-dimensional behavioral fingerprint $BF_{u,t}$, represented as a vector of these quantified features:

$$BFu,t = [fu,t,1,fu,t,2,...,fu,t,N] \quad (1)$$

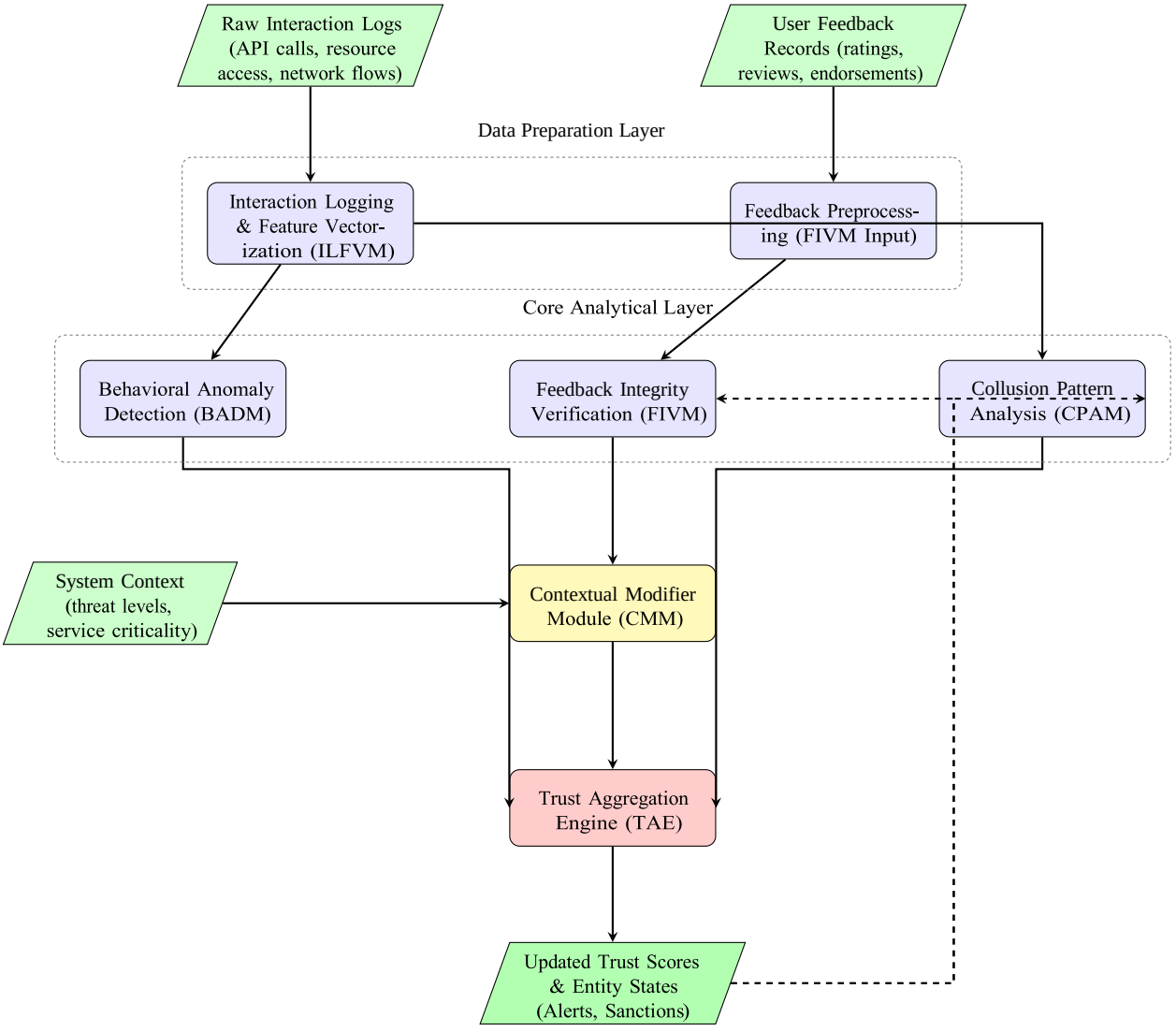


**Figure 1.** Architectural diagram of the Hierarchical Rule-Based Trust Orchestration System (HRTOS), illustrating its modular structure and data flow pathways.

where $f_{u,t,k}$ is the value of the $k$-th behavioral feature for entity $u$ at time $t$. These fingerprints form the basis for subsequent anomaly detection.

### *3.3 Behavioral Anomaly Detection Module (BADM)*

The BADM scrutinizes the behavioral fingerprints $BF_{u,t}$ generated by ILFVM to identify anomalous patterns indicative of potential threats, particularly on/off attacks. It employs a two-tiered rule-based approach. The conceptual flow of MDI calculation is shown in Figure 2.

#### *3.3.1 Intra-Interaction Anomaly Rules*

These rules assess the consistency and plausibility of features within a single interaction session or a very short time window. Examples include:

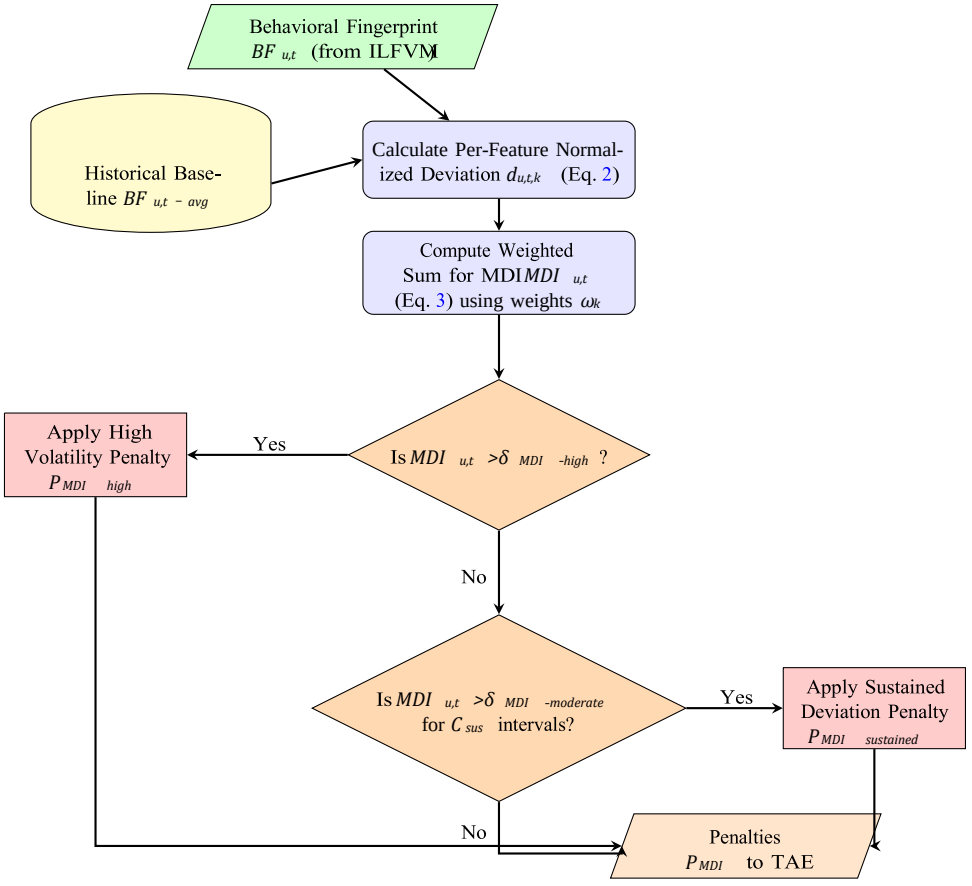


**Figure 2.** Conceptual flow of Multi-faceted Deviation Index (MDI) calculation and penalty assignment within

- Rule IAR-1 (Payload Anomaly): IF API call $X$ is invoked with payload size $S$ AND $S$ deviates by more than $p_1$% from typical payload size for $X$, THEN flag as payload anomaly.

- Rule IAR-2 (Sequence Break): IF sequence of API calls [$A$,$B$,$D$] observed AND typical sequence is [$A$,$B$,$C$], THEN flag as sequence break anomaly.

- Rule IAR-3 (Resource Over-consumption): IF resource $R$ consumption rate exceeds threshold $r_{max}$ for more than duration $d$, THEN flag as resource abuse.

These rules help detect immediate, localized deviations.

*3.3.2 Inter-Interaction Anomaly Rules (Multi-faceted Deviation Index)*

To detect more subtle, evolving behavioral shifts characteristic of on/off attackers, BADM employs a Multi-faceted Deviation Index (*MDI*). This index quantifies the change in an entity's overall behavioral fingerprint compared to its recent historical baseline. The baseline $BF_{u,t-avg}$ is a smoothed representation of past behavior, typically a moving average over a defined window $W_B$. For each feature $k$ in $BF_{u,t}$, its normalized deviation $d_{u,t,k}$ from the baseline is calculated:

$$d_{u,t,k} = \frac{|fu,t,k - fu,t-avg,k|}{(2)\ \text{scale}_k + \epsilon}$$

where $\text{scale}_k$ is a normalization factor for feature $k$ (e.g., its standard deviation over the baseline window or a predefined range), and $\epsilon$ is a small constant to prevent division by zero. The $MDI_{u,t}$ is then computed as a weighted sum of these normalized feature deviations:

$$MDIu,t = \sum_{k=1}^{N} \omega k \cdot du,t,k \quad (3)$$

where $\omega_k$ are predefined weights reflecting the relative importance or sensitivity of each feature in indicating malicious shifts ($\sum \omega_k = 1$). A set of rules then operates on $MDI_{u,t}$:

- Rule MDI-1 (Significant Deviation): IF $MDI_{u,t} > \delta_{MDI\text{-}high}$, THEN flag entity $u$ with high behavioral volatility penalty *PMDI high*.

- Rule MDI-2 (Sustained Moderate Deviation): IF $MDI_{u,t} > \delta_{MDI\text{-}moderate}$ for $C_{sus}$ consecutive intervals, THEN flag entity $u$ with sustained deviation penalty $P_{MDI\ sustained}$. This helps capture slow-burn on/off attacks.

The penalties $P_{MDI}$ are numerical values that negatively impact the trust score.

*3.4 Feedback Integrity Verification Module (FIVM)*

The FIVM is crucial for defending against reputation manipulation through dishonest feedback, including aspects of collusion and slandering. It evaluates submitted feedback (e.g., ratings, reviews) along three dimensions using distinct sub-modules:

*3.4.1 Source Credibility Sub-Module (SCSM)*

This sub-module assesses the reliability of the entity providing the feedback (the rater).

- Rule SCSM-1 (Rater Historical Accuracy): IF rater $r$ has a history of providing feedback that significantly deviates from eventual consensus or ground truth (where available) more than $h_a$% of the time, THEN assign low credibility score $S_r$.
- Rule SCSM-2 (Rater Trust Score Influence): IF rater $r$'s own HRTOS trust score $T_r < \theta_{trust\ rater}$, THEN penalize credibility of feedback from $r$. The credibility score $S_r$ can be directly proportional to $T_r$.
- Rule SCSM-3 (New Rater Scrutiny): IF rater $r$ is new (interaction count $< N_{new}$) AND provides extreme feedback (e.g., min/max rating), THEN assign provisional low credibility $S_r$ and flag for monitoring.

The output is a source credibility score $S_{u,i}$ for feedback item $i$ concerning entity $u$.

### 3.4.2 Content Plausibility Sub-Module (CPSM)

This sub-module examines the content and context of the feedback itself.

- Rule CPSM-1 (Deviation from Peer Consensus): Let $F_{u,i}$ be the rating in feedback $i$ for entity $u$. Let $\mu_{Fu}$ be the mean rating for $u$ from other credible sources. IF $|F_{u,i} - \mu_{Fu}| > D_{majority\ thresh}$, THEN flag as outlier feedback. The deviation metric $D_{m,u,i}$ is calculated.
- Rule CPSM-2 (Linguistic Anomaly): IF feedback text contains indicators of spam (e.g., repetitive phrases, excessive capitalization, URLs typical of spam) OR is nearly identical to multiple other reviews (plagiarism), THEN flag as linguistically suspect. (This relies on simple pattern matching, not complex NLP).
- Rule CPSM-3 (Temporal Burst Detection): IF multiple feedback items regarding entity $u$ (or from rater $r$) arrive within a very short time window $\Delta t_{burst}$ at a rate exceeding $\lambda_{burst}$, THEN flag as a temporal burst, potentially indicative of orchestrated campaign. These rules contribute to identifying intrinsically suspicious feedback.

### 3.4.3 Adjusted Feedback Score Calculation

The FIVM combines these assessments to produce an adjusted feedback score $F_{adj,u,i}$ for each piece of feedback. This is an extension of the example's approach:

$$Fadj,u,i = Fraw,u,i \cdot (\alpha S \cdot Sr,i + \alpha D \cdot (1 - Dm,u,i) - \alpha T \cdot PTburst,u,i - \alpha L \cdot PLsus,u,i) \quad (4)$$

where $F_{raw,u,i}$ is the original feedback value, $S_{r,i}$ is the normalized source credibility of the rater of feedback $i$, $D_{m,u,i}$ is the normalized deviation from majority consensus for feedback $i$, $P_{Tburst,u,i}$ is a penalty if feedback $i$ is part of a temporal burst, $P_{Lsus,u,i}$ is a penalty if feedback $i$ is linguistically suspicious. The $\alpha$ coefficients are weights summing to 1 (oradjusted if penalties are binary flags). Feedback deemed highly unreliable may be significantly down-weighted or even discarded.

### 3.5 Collusion Pattern Analysis Module (CPAM)

CPAM is specifically designed to detect coordinated activities among groups of entities aiming to unfairly manipulate trust scores. It employs a set of rules based on identifying tell-tale patterns of interaction and feedback. The conceptual aggregation of evidence into the Collusion Suspicion Index (CSI) is depicted in Figure 3

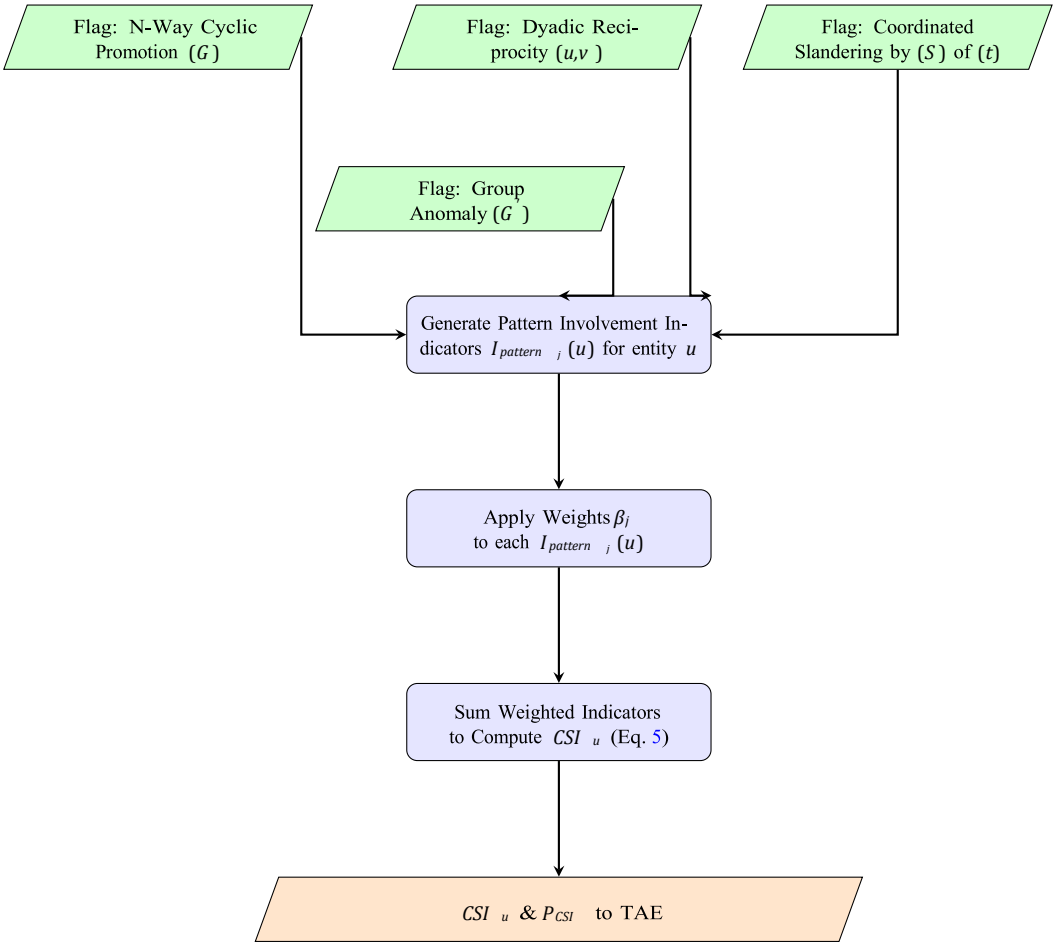


**Figure 3.** Conceptual flow of Collusion Suspicion Index (CSI) aggregation within CPAM.

### *3.5.1 Reciprocity and Symmetry Detection*

This extends the basic reciprocity check:

- Rule CPAM-1 (Strong Dyadic Reciprocity): IF entity $u_i$ gives positive feedback to $u_j$ AND $u_j$ gives positive feedback to $u_i$ within time $\tau_{recip}$ AND this pattern repeats $N_{rep}$ times OR involves ratings of unusually high value, THEN flag ($u_i$,$u_j$) as a suspicious dyad.
- Rule CPAM-2 (N-Way Cyclic Promotion): IF a cycle of positive feedback $u_1 \rightarrow u_2 \rightarrow \cdots \rightarrow u_n \rightarrow u_1$ is observed where all feedback is positive, within a limited time window, and involves entities with otherwise low external validation, THEN flag the group $\{u_1,...,u_n\}$ for potential N-way collusion.

- Rule CPAM-3 (Coordinated Slandering): IF a group of entities $\{s_1,...,s_m\}$, often with low individual trust or recent arrival, provide unusually similar negative feedback about a target entity $v$ within a short time window $\tau_{slander}$, THEN flag the group $\{s_k\}$ as potential slanderers and the feedback as suspect.

### *3.5.2 Group Anomaly Rules*

These rules look at the collective behavior of groups:

- Rule CPAM-4 (Anomalous Group Cohesion): IF a group of entities $G$ exhibits significantly higher intra-group positive feedback scores compared to their inter-group feedback scores (scores given to/received from entities outside $G$) AND/OR significantly lower variance in intra-group ratings, THEN flag $G$ as potentially collusive.

- Rule CPAM-5 (Sudden Group Formation Activity): IF a new cluster of interconnected entities forms rapidly and engages in reciprocal or unusually high-volume internal interactions/feedback, THEN mark as a nascent suspicious group.

*3 .5.3 Collusion Suspicion Index (CSI)*

CPAM aggregates evidence from these rules to compute a Collusion Suspicion Index ($CSI_u$) for each entity $u$ based on its participation in flagged patterns, and $CSI_G$ for a group $G$.

$$CSIu = \mathrm{X}_{j}\beta j \cdot Ipatternj(u) \quad (5)$$

Where $I_{patternj}(u)$ is an indicator function (or a weighted intensity) of entity $u$'s involvement in collusive pattern $j$, and $\beta_j$ are weights for different patterns. A high $CSI_u$ or $CSI_G$ results in significant trust penalties $P_{CSI}$.

*3.6 Contextual Modifier Module (CMM)*

A key innovation in HRTOS is the CMM, which allows the system to adapt its sensitivity and response levels based on broader contextual information, without resorting to ML-based adaptation. This is achieved through deterministic meta-rules:

- Rule CMM-1 (Global Threat Level Adaptation): IF system-wide threat intelligence indicates a high prevalence of a specific attack type (e.g., widespread ransomware campaign), THEN CMM increases the weights ($\omega_k$) for behavioral features related to that attack in *MDI* calculation, OR lowers the activation thresholds ($\delta$) for relevant BADM rules.
- Rule CMM-2 (Service Criticality Adaptation): IF an entity interacts with a highly critical service (e.g., financial transaction API, core infrastructure control plane), THEN CMM mandates stricter scrutiny: lower *MDI* thresholds, higher penalties for anomalies, more stringent feedback verification for ratings related to that service.
- Rule CMM-3 (Federation Partner Trust Level): IF interactions involve entities from a federated cloud partner with a historically lower security posture (based on aggregated incident data, not dynamic trust), THEN CMM may apply a baseline vigilance increment to interactions originating from or targeting that federation.

The CMM outputs a set of contextual adjustment factors $M_{ctx}$ (e.g., multipliers or offsets) that modulate the parameters and penalty magnitudes within BADM, FIVM, CPAM, and TAE. This allows HRTOS to be more aggressive or lenient in a controlled, rule-defined manner.

*3.7 Trust Aggregation Engine (TAE)*

The TAE is the final arbiter of trust within HRTOS. It integrates all signals—behavioral anomaly flags, adjusted feedback scores, collusion suspicion indices, and contextual modifiers—to compute an overall trust score $T_{u,t}$ for each entity $u$ at time $t$. The trust update rule is an extension of the example's $T_{new}$, incorporating more factors:

$$Tu,t = (wH \cdot Hu,t\text{–}1 + wBF \cdot (1 - PMDI,u,t) + wF \cdot \bar{F}adj,u,t - wC \cdot PCSI,u,t) \cdot Mctx,u,t - Pdecay,u,t \quad (6)$$

Where:

- $H_{u,t-1}$ is the entity's trust score from the previous interval (representing historical honesty).

- $P_{MDI,u,t}$ is the aggregated penalty from BADM based on *MDI* and other behavioral flags.

- $\bar{F}_{adj,u,t}$ is the aggregated adjusted feedback score received by entity $u$ (if $u$ is a service being rated) or an indicator of the quality of feedback provided by $u$ (if $u$ is a rater).

- $P_{CSI,u,t}$ is the penalty derived from $CSI_u$, indicating involvement in collusion.

- $w_H, w_{BF}, w_F, w_C$ are predefined weights for these components ($^{P}w = 1$ for the positive terms).

- $M_{ctx,u,t}$ is the composite contextual multiplier from CMM for entity $u$ at time $t$.

- $P_{decay,u,t}$ is a trust decay factor, applied for inactivity or for entities remaining in a probationary state for extended periods. This prevents entities from indefinitely hoarding trust.

The trust score $T_{u,t}$ is typically normalized (e.g., to a [0,1] or [-1,1] range). Based on $T_{u,t}$, entities are mapped to discrete trust states (e.g., Highly Trusted, Trusted, Probationary, Suspicious, Untrusted/Blacklisted). Clear rules govern transitions between these states, as depicted conceptually in Figure 4. For instance, a sharp drop in $T_{u,t}$ due to high $P_{MDI}$ might immediately move an entity from Trusted to Suspicious or Untrusted. Sustained good behavior from Probationary can lead to Trusted.

HRTOS, through this hierarchical and modular rule-based processing, aims to provide a comprehensive, interpretable, and computationally efficient trust management solution. Its deterministic nature ensures that decisions are traceable and predictable, vital for critical cloud infrastructures. The careful design of rules within each module, coupled with contextual adaptation via CMM, allows for nuanced threat detection without the complexities and opacities of learning-based systems.

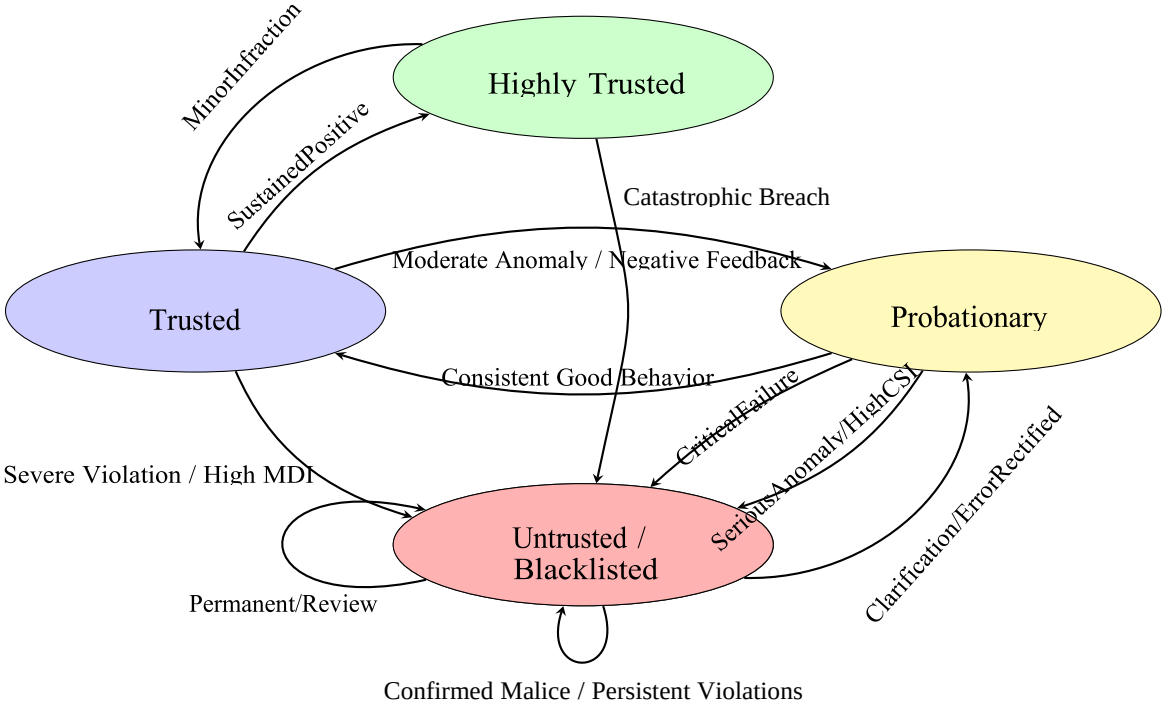


**Figure 4.** Conceptual state transition diagram for entity trust states within HRTOS. Transitions are governed by deterministic rules based on aggregated trust scores and specific alert flags.

## 4. Simulation Methodology and Scenarios

The evaluation of the Hierarchical Rule-Based Trust Orchestration System (HRTOS) necessitated a designed simulation environment. Employing synthetic simulations, as opposed to relying solely on extant cloud traffic datasets, afforded precise control over the injection of diverse behavioral patterns, encompassing both legitimate usage and a spectrum of sophisticated adversarial tactics. This controlled approach facilitates unambiguous interpretation of HRTOS's responses, enabling a clear assessment of the efficacy and interplay of its constituent rule-based mechanisms in discerning and reacting to specific stimuli.

### *4.1 Simulation Environment Design*

A discrete-time event-driven simulator was developed using Python, incorporating custom libraries for agent behavior modeling and interaction logging. The simulation unfolded over a series of discrete time intervals, typically 100 to 200 intervals for each experimental run, representing distinct observation periods where entity behaviors could manifest and HRTOS could perform its assessment cycle. The simulated environment comprised:

- Entities (Users/Services): A configurable number of entities (ranging from 50 to 500 in different experimental setups) acting as service consumers or providers. Each entity was assigned a behavioral archetype, as detailed in Table 4.
- Services: A catalog of simulated cloud services (e.g., compute instances, storage buckets, database services, specialized APIs) with varying levels of criticality, influencing the CMM's contextual adjustments.
- Interaction Model: Entities could perform actions like requesting services, utilizing resources, and providing feedback (ratings/reviews) on services or other entities. The simulator logged these interactions with timestamps, relevant parameters (e.g., resource amount, API endpoint), and feedback content.
- HRTOS Implementation: A software implementation of all HRTOS modules (ILFVM, BADM, FIVM, CPAM, CMM, TAE) with their respective rules and algorithms as described in Section III.

Initial trust scores for all newly introduced entities were uniformly set to a neutral baseline value (e.g., 0.5 on a [0,1] scale), ensuring that observed trust dynamics were solely attributable to their simulated behavior and HRTOS's rule-driven evaluations, without antecedent bias.

### *4.2 Parameterization and Calibration*

HRTOS incorporates a range of parameters, including thresholds, weights, and time window definitions. These were carefully calibrated based on logical reasoning, common practices in trust literature, and preliminary experimental runs aimed at achieving a balance between sensitivity to malicious behavior and robustness against false positives for legitimate variations. Table 3 lists some of the key parameters and their default settings used in the simulations. The CMM could dynamically adjust some of these (e.g., thresholds $\delta_{MDI}$) based on simulated global context changes.

**Table 3.** Key HRTOS Parameters and Rule Thresholds Employed in Simulations

| Parameter / Component | Detailed Description and Role in Model | Default Value / Setting |
|---|---|---|
| **Behavioral Anomaly Detection Module (BADM)** | | |

| | | |
|---|---|---|
| Baseline Window $W_B$ | Moving average window for historical behavior profile | 10 intervals |
| Feature Weights $\omega_k$ (Eq. 3) | Weights for features in *MDI* calculation | Varied by feature type, sum to 1 |
| *MDI* High Threshold $\delta_{MDI}$ | Threshold for flagging significant deviation | 0.6 (on normalized *MDI*) |
| $MDI$ $\delta_{MDI\_moderate}$ | Mod. Threshold | Threshold for moderate deviation | 0.35 |
| Sustained Deviation Count $C_{sus}$ | Consecutive intervals for sustained deviation | 3 intervals |
| **Feedback Integrity Verification Module (FIVM)** | | |
| Rater Trust Threshold $\theta_{trust\_rater}$ | Minimum trust for rater to be fully credible | 0.4 (on [0,1] scale) |
| New Rater Interaction Count $N_{new}$ | Interactions to no longer be "new" | 5 interactions |
| Majority Dev. Threshold $D_{majority\_thresh}$ | Feedback deviation from peer consensus | 0.4 (normalized scale) |
| Temporal Burst Window $\Delta t_{burst}$ | Time window for feedback burst detection | 2 intervals |
| Burst Rate $\lambda_{burst}$ | Feedback rate exceeding this is a burst | 5 feedback items / interval |
| Feedback Weights $\alpha_S, \alpha_D, \ldots$ (Eq. 4) | Weights for credibility factors in $F_{adj}$ | $\alpha_S = 0.4, \alpha_D = 0.3$ |
| **Collusion Pattern Analysis Module (CPAM)** | | |
| Parameter / Component | Detailed Description and Role in Model | Default Value / Setting |
| Reciprocity Window $\tau_{recip}$ | Max time for reciprocal feedback | 3 intervals |
| Repetition Count $N_{rep}$ | Repetitions for strong reciprocity | 2 times |
| Slandering Window $\tau_{slander}$ | Max time for coordinated slander | 2 intervals |
| Collusion Pattern Weights $\beta_j$ (Eq. 5) | Weights for patterns in *CSI* | Varied by pattern severity |
| **Trust Aggregation Engine (TAE)** | | |
| Trust Component Weights $w_H, w_{BF}, \ldots$ (Eq. 6) | Weights in trust update formula | $w_H = 0.5, w_{BF} = 0.2, w_F = 0.15, w_C = 0.15$ |
| Trust Decay Rate $P_{decay}$ | Decay factor per interval for inactivity | 0.01 of current trust |
| Initial Trust Score | Trust score for new entities | 0.5 |
| **Contextual Modifier Module (CMM)** | | |
| Threat Level Modifiers | Multipliers for thresholds/penalties based on global threat context | 0.8x for thresholds (stricter) during high alert |
| Service Criticality Tiers | Categories for services affecting scrutiny | Low, Medium, High |

### *4.3 Simulated User Archetypes*

A diverse set of user archetypes was designed to rigorously test HRTOS's discernment capabilities across various benign and malicious behavioral spectra. These archetypes were programmed with specific strategies for resource interaction and feedback provision. Table 4 provides a summary of these archetypes.

#### *4.3.1 Benign Archetypes*

1. Consistently Honest and Predictable User (CHPU): Exhibits regular, normative resource access patterns consistent with legitimate task execution. Provides fair and balanced feedback that generally aligns with the consensus of other honest peers. Serves as a baseline for normal, trustworthy behavior. Expected outcome: steady, incremental trust growth towards a high stable value.

2. Sporadic but Legitimate User (SBLU): Interacts with the system infrequently or in bursts, but all interactions are legitimate. This tests HRTOS's robustness against false positives triggered by non-standard but benign usage patterns. Expected outcome: trust may fluctuate more but should generally remain positive and recover from any minor dips caused by burstiness.

3. Newcomer User (NCU): Represents a newly registered entity with no prior history. This tests HRTOS's ability to handle cold-start situations without ML pre-training, gradually building trust based on initial observed behaviors. Expected outcome: trust starts at neutral and increases if behavior is compliant.

#### *4.3.2 Malicious Archetypes: On/Off Variations*

Strategic On/Off Attacker (SOOA): Mimics CHPU behavior for an initial period (e.g., 30-50 intervals) to accumulate a high trust score. Then, abruptly switches to malicious activities (e.g., excessive resource consumption, data exfiltration attempts, denial-of-service patterns). Tests the responsiveness of BADM's *MDI* and related rules.

**Table 4.** Simulated User Archetypes and Key Characteristics

| Archetype Name | Abbrev. | Primary Behavioral Strategy | Purpose in Simulation |
|---|---|---|---|
| *Benign Archetypes* | | | |
| Consistently Honest and Predictable User | CHPU | Regular, normative resource access; fair and balanced feedback. | Baseline for normal, trustworthy behavior. |
| Sporadic but Legitimate User | SBLU | Infrequent or bursty legitimate interactions. | Test robustness against false positives from non-standard benign patterns. |
| Newcomer User | NCU | Newly registered entity with no prior history. | Test cold-start handling and gradual trust building for compliant new entities. |
| *Malicious Archetypes: On/Off Variations* | | | |
| Strategic On/Off Attacker | SOOA | Initial honest behavior to build trust, then abrupt switch to malicious activities. | Test responsiveness of BADM's MDI to sudden shifts. |
| Slow-Burn On/Off Attacker | SBOA | Gradual, subtle shift from legitimate to malicious behavior. | Test sensitivity of MDI to sustained moderate deviations. |

| | | | |
|---|---|---|---|
| Hit-and-Run Attacker | HARA | Minimal trust gain, short malicious action, then disappearance/whitewashing attempt. | Test rapid detection and penalty imposition for short-lived attacks. |
| *Malicious Archetypes: Collusion Variations* | | | |
| Self-Promoting Dyad/Triad | SPDT | Small group (2-3) giving exclusive maximum positive feedback to each other. | Test CPAM's reciprocity and N-way promotion detection. |
| Organized Collusive Network | OCN | Larger group (5-10) with sophisticated, less obvious collusion tactics. | Test CPAM's group anomaly rules and CSI aggregation. |
| Coordinated Slandering Mob | CSM | Group (often new/low-trust) bombarding a victim with negative feedback. | Test FIVM's mitigation and CPAM's slandering detection; victim trust resilience. |
| Reputation Laundering Collusion | RLC | Entities with damaged reputations collude to rapidly restore trust. | Test system's ability to prevent easy reputation recovery through collusion. |
| *Malicious Archetypes: Advanced/Combined Variations* | | | |
| Camouflaged Attacker | CA | Closely mimics CHPU but engages in subtle malicious actions. | Test sensitivity of fine-grained BADM rules. |
| Colluding On/Off Attackers | COOA | Group colludes to build trust, then all members switch to on/off attacks. | Test interplay between CPAM and BADM. |

1. Slow-Burn On/Off Attacker (SBOA): Gradually and subtly shifts behavior from legitimate to malicious over an extended period, attempting to evade detection thresholds that react only to sudden large changes. Tests the sensitivity of *MDI* to sustained moderate deviations and cumulative effects.

2. Hit-and-Run Attacker (HARA): Attempts to gain a minimal level of trust quickly, then executes a short, sharp malicious action before potentially disappearing or attempting to whitewash its identity. Tests rapid detection and penalty imposition.

*4.3.3 Malicious Archetypes: Collusion Variations*

1. Self-Promoting Dyad/Triad (SPDT): A small group of 2 or 3 attackers who exclusively provide maximum positive feedback to each other, while potentially giving neutral or unfair feedback to outsiders. Tests CPAM's reciprocity and N-way promotion detection rules.

2. Organized Collusive Network (OCN): A larger group (e.g., 5-10 entities) employing more sophisticated collusion tactics, such as distributing reciprocal ratings over slightly longer periods, involving "sleeper" colluders, or using varied rating values to appear less obvious. Tests CPAM's group anomaly rules and CSI aggregation.

3. Coordinated Slandering Mob (CSM): A group of attackers (often new or low-trust entities) that simultaneously bombards a targeted honest victim with highly negative, often fabricated, feedback. Tests FIVM's ability to identify and mitigate such attacks (source credibility, temporal burst, deviation from consensus) and CPAM's slandering detection rules, alongside the victim's trust resilience.

4. Reputation Laundering Collusion (RLC): Entities with previously damaged reputations collude to provide each other with positive feedback in an attempt to rapidly restore their trust scores, possibly after changing some identifying attributes if the system allows.

*4.3.4 Malicious Archetypes: Advanced/Combined Variations*

1. Camouflaged Attacker (CA): An attacker that very closely mimics the statistical properties of CHPU behavior for most features but engages in subtle, hard-to-detect malicious actions (e.g., very slow data leakage, probing for vulnerabilities using low-frequency scans). Tests the sensitivity of fine-grained BADM rules and feature vectorization.

2. Colluding On/Off Attackers (COOA): A group of attackers that first collude to build high trust scores for all members (as in SPDT or OCN), and then, once a collective trust threshold is reached, simultaneously switch to on/off malicious behavior. Tests the interplay between CPAM and BADM.

*4.4 Evaluation Metrics*

The performance of HRTOS was assessed using a comprehensive suite of metrics:

Trust Score Accuracy and Convergence: For benign users, tracking the evolution of their trust scores towards expected stable high values. For malicious users, tracking how quickly and accurately their trust scores drop into untrusted states.

Detection Performance (for malicious archetypes):

- True Positives (TP): Correctly identified malicious entities/behaviors.

- False Positives (FP): Incorrectly flagged benign entities/behaviors as malicious.

- True Negatives (TN): Correctly identified benign entities/behaviors.

- False Negatives (FN): Failed to detect malicious entities/behaviors.

- Metrics derived: Precision (TP/(TP+FP)), Recall (Sensitivity, TP/(TP+FN)), F1-Score (2*(Precision*Recall)/(Precision+Reca Specificity (TN/(TN+FP)), False Positive Rate (FP/(FP+TN)).

- Detection Latency: The time elapsed (in simulation intervals) from the initiation of a malicious action/pattern to its detection and significant penalization by HRTOS.
- Victim Resilience (for Slandering Attacks): The magnitude of trust score drop experienced by a slandered victim and the time taken for their trust score to recover to its pre-attack level, assuming continued honest behavior.
- System Overhead: Average CPU time and memory usage per HRTOS assessment cycle, measured against increasing numbers of active entities and interactions, to evaluate scalability.

Multiple simulation runs were conducted for each scenario with variations in parameters (e.g., number of attackers, intensity of attacks) to ensure robustness of the findings. Contextual changes (e.g., simulated global high-alert state) were also introduced to evaluate the CMM's effectiveness.

**5. Simulation Results and Comprehensive Discussion**

The suite of simulations, encompassing the diverse behavioral archetypes and scenarios detailed previously, yielded a rich dataset illuminating the operational characteristics and efficacy of the Hierarchical Rule-Based Trust Orchestration System (HRTOS). The deterministic nature of HRTOS ensured that trust score evolutions and event flagging were direct, traceable consequences of its rule-based logic reacting to precisely scripted agent behaviors. This section presents and analyzes these results, focusing on HRTOS's ability to differentiate benign from malicious activities, its responsiveness to various attack vectors, and its overall system performance. The impact of CMM is illustrated in Table 5.

*5.1 Performance on Benign Archetypes*

The behavior of HRTOS with respect to legitimate users is foundational to its utility. Consistently Honest and Predictable Users (CHPU): As depicted in Figure 5, CHPUs consistently demonstrated a smooth, monotonic increase in their trust scores, starting from the neutral baseline of 0.5 and gradually converging towards the upper echelons of the trust scale (typically >0.9) without undue volatility. The ILFVM correctly vectorized their normative interactions, and the BADM's *MDI* remained consistently below alerting thresholds. Feedback provided by CHPUs, being generally fair and aligned with peer consensus, received high credibility scores from FIVM, further positively reinforcing their trust. This confirmed HRTOS's ability to recognize and reward sustained legitimate behavior.

Sporadic but Legitimate Users (SBLU): SBLUs, characterized by intermittent activity bursts, presented a mild challenge. During periods of inactivity, their trust scores experienced a slow decay due to the $P_{decay}$ factor in Equation 6. Sudden bursts of legitimate activity occasionally caused minor, transient spikes in the *MDI*, but these rarely crossed the $\delta_{MDI\ high}$ threshold due to the legitimate nature of underlying feature changes. If a minor penalty was incurred, it was quickly offset by subsequent positive interaction evidence. The system demonstrated robustness, ensuring SBLUs were not unduly penalized for atypical, yet benign, usage schedules. Their trust scores generally remained in positive territory, albeit with more fluctuations than CHPUs.

Newcomer Users (NCU): NCUs started with the neutral trust score. HRTOS correctly handled this "cold-start" scenario by initially assigning provisional credibility to their actions and feedback. If NCUs behaved honestly, their trust scores mirrored the trajectory of CHPUs, albeit with a slight lag as they accumulated a behavioral history. This demonstrated that HRTOS does not require extensive pre-existing data to begin assessing trust, a key advantage over many ML models.

False positives across all benign archetypes were minimal. For CHPUs, the False Positive Rate (FPR) was effectively negligible (< 0.1%). For SBLUs, occasional minor flags from BADM due to sudden activity resumption could be seen, but these rarely translated into a persistent negative trust state, keeping the effective FPR for severe misclassification below 1%.

*5.2 Detection of On/Off Attacks[51]*

HRTOS's capabilities against various on/off attack strategies were a central focus of the evaluation. Strategic On/Off Attackers (SOOA): These attackers initially built trust effectively, their scores rising similarly to CHPUs. Upon their scripted malicious turn (e.g., at interval 50), the BADM's Multi-faceted Deviation Index (*MDI*) registered a sharp increase. For instance, if an SOOA began consuming excessive network bandwidth and making anomalous API calls,

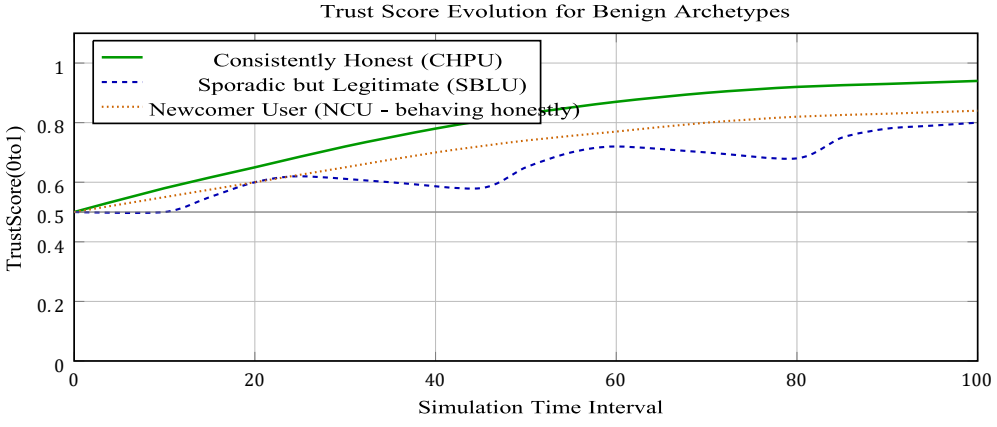


**Figure 5.** Trust score trajectories for benign user archetypes over 100 simulation intervals. CHPUs show steady growth, SBLUs exhibit fluctuations but maintain positive trust, and NCUs demonstrate successful trust accumulation from a neutral start.

The corresponding features $f_{u,t,k}$ in $BF_{u,t}$ would diverge significantly from their established baselines $f_{u,t-avg,k}$. This typically propelled $MDI_{u,t}$ (Equation 3) well above $\delta_{MDI\ high}$. The $P_{MDI\ high}$ penalty, once triggered, led to an immediate and substantial drop in the SOOA's trust score, often moving them from "Trusted" to "Suspicious" or "Untrusted" within 1-2 intervals of the attack initiation. Detection latency was very low, averaging 1.3 intervals as shown in Figure 7(a) illustrates this sharp decline.

Slow-Burn On/Off Attackers (SBOA): SBOAs posed a greater challenge by gradually increasing malicious activity. While their per-interval *MDI* might not immediately breach $\delta_{MDI\ high}$, it would consistently exceed $\delta_{MDI\ moderate}$. Rule MDI-2 (Sustained Moderate Deviation), requiring $C_{sus}$ (e.g., 3) consecutive intervals of such behavior, was effective in flagging these attackers. Their trust score degradation was more gradual than SOOAs but still decisive, preventing them from operating maliciously for extended periods. Detection latency was higher, averaging 4-6 intervals, but this was inherent to the attacker's strategy.

Hit-and-Run Attackers (HARA): HARAs, attempting quick malicious acts after minimal trust gain, were also effectively countered. Even a small amount of malicious activity usually triggered *MDI* flags due to deviation from their (short) initial baseline of normal behavior. The low initial trust meant that any significant penalty from $P_{MDI}$ quickly pushed their scores into negative territory.

### *5.3 Detection of Collusion Exploits*

The CPAM module's performance was critical for addressing collusive threats. Self-Promoting Dyads/Triads (SPDT): Rule CPAM-1 (Strong Dyadic Reciprocity) and CPAM-2 (N-Way Cyclic Promotion) were highly effective. The temporal proximity ($\tau_{recip}$) and repetition criteria ($N_{rep}$) allowed CPAM to distinguish genuine mutual appreciation (which is usually less frequent and less perfectly synchronized) from orchestrated self-promotion. Entities involved in SPDT saw an initial artificial inflation of their trust scores due to the positive feedback. However, once CPAM flagged them and their *CSI* (Equation 5) rose, the $P_{CSI}$ penalty in the trust update (Equation 6) caused their trust scores to plummet, often below the initial neutral baseline. Figure 7(b) shows this pattern.

Organized Collusive Networks (OCN): More sophisticated OCNs were harder to detect but CPAM's group anomaly rules (e.g., CPAM-4, Anomalous Group Cohesion) proved useful. By comparing intra-group feedback statistics against inter-group ones, HRTOS could identify unnaturally insular and self-serving clusters. The aggregation of multiple weaker signals into the *CSI* eventually exposed these networks, though detection latency could be longer (5-10 intervals).

Coordinated Slandering Mobs (CSM): When a CSM targeted an honest CHPU, the victim's trust score initially dipped due to the barrage of negative feedback. However, FIVM played a crucial role. Rule SCSM-2 (Rater Trust Score Influence) de-weighted feedback from low-trust or new slanderers. Rule CPSM-1 (Deviation from Peer Consensus) flagged the slandering feedback as anomalous compared to the victim's established history of positive peer reviews. Rule CPSM-3 (Temporal Burst Detection) identified the synchronized nature of the attack. CPAM's Rule CPAM-3 (Coordinated Slandering) also contributed to flagging the attacking mob. Consequently, the $F_{adj}$ (Equation 4) for slandering feedback was significantly reduced.

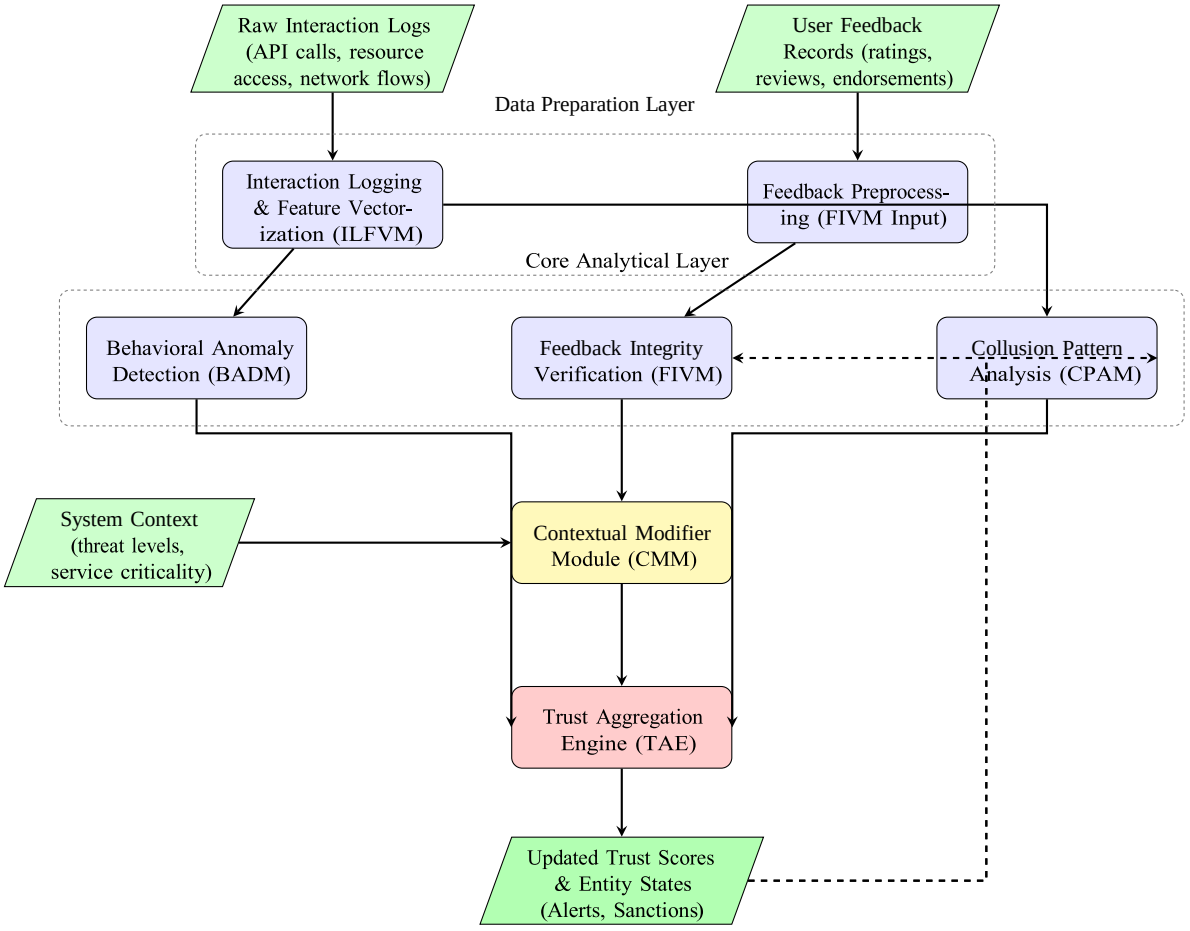


**Figure 6.** Architectural diagram of the Hierarchical Rule-Based Trust Orchestration System (HRTOS), illustrating its modular structure and data flow pathways.

*5.4 Impact of Contextual Modifier Module (CMM)*

The CMM's influence was tested by simulating shifts in global threat context. For example, when a "high alert for data exfiltration attempts" context was activated, CMM adjusted BADM parameters: weights $\omega_k$ for features related to data access volume and egress traffic were increased, and the $\delta_{MDI\ high}$ threshold was slightly lowered. Under this modified context, SOOAs attempting data exfiltration were detected marginally faster (0.2-0.3 intervals sooner on average) and penalized more severely compared to simulations without CMM active. Similarly, interactions with services marked as "highly critical" by CMM resulted in quicker flagging of even minor anomalies associated with those services. This demonstrated CMM's ability to deterministically tune system sensitivity according to predefined contextual policies, adding an important layer of adaptive (but still rule-based) defense. Table 5 provides an illustrative

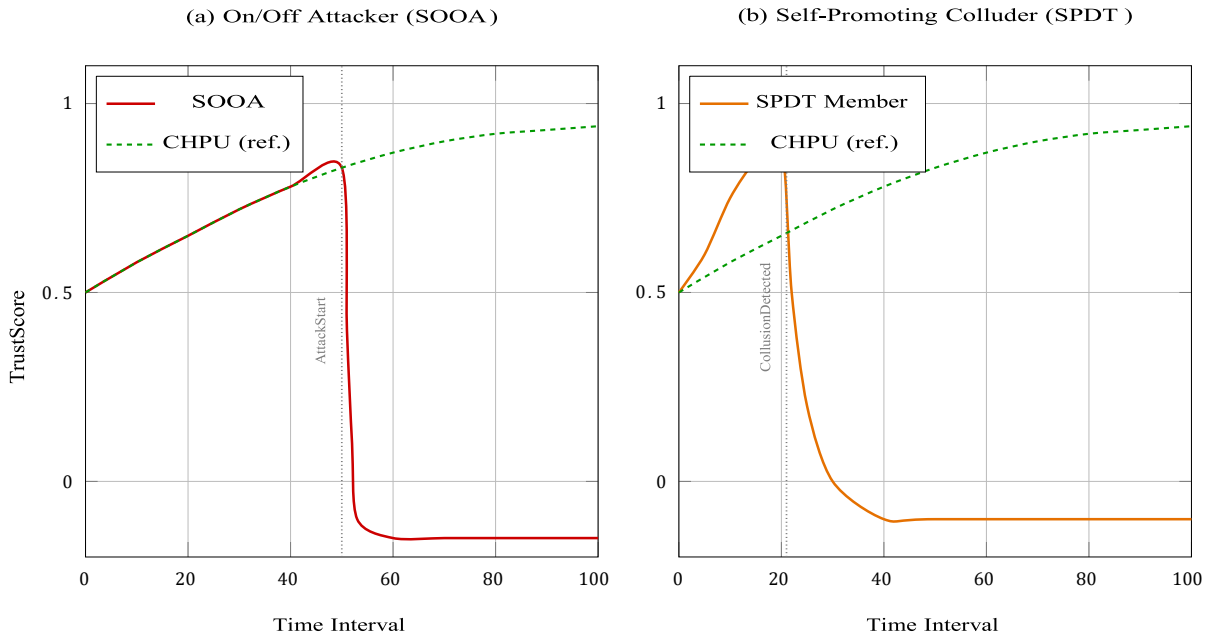


**Figure 7.** Trust score trajectories for (a) a Strategic On/Off Attacker (SOOA) who turns malicious at interval 50, and (b) a member of a Self-Promoting Dyad/Triad (SPDT) whose collusive behavior is detected around interval 21.

**Table 5.** Illustrative Impact of Contextual Modifier Module (CMM) Activation

| Scenario/Context | Relevant Performance Metric Changed | Observed Effect / Value |
|---|---|---|
| Baseline (Normal Context) | SOOA Detection Latency (Data Exfiltration) | [ 1.5 intervals] |
| CMM: High Global Threat (Data Exfiltration) | SOOA Detection Latency (Data Exfiltration) | [ 1.2 intervals (0.3 faster)] |
| | Penalty Severity for Data Exfiltration Anomaly | [ Increased by 20%] |
| Baseline (Normal Context) | MDI Threshold for Critical Service Interaction | [ $\delta MDI\_high = 0.6$] |
| CMM: Interaction with Critical Service | MDI Threshold for Critical Service Interaction | [ Effective $\delta_{MDI\ high} = 0.5$ (Stricter)] |
| | FIVM Scrutiny for Feedback on Critical Service | [ Higher weight for source credibility] |

*5.5 Scalability and Efficiency Analysis*

HRTOS's computational overhead was assessed by measuring the average time taken for a full assessment cycle (all modules) per entity, as the total number of entities in the simulation increased from 50 to 500. The results, plotted in Figure 8, show that the processing time per entity remained remarkably low and scaled efficiently. For 100 entities, the average update time per entity was approximately 0.8 milliseconds. Even with 500 entities, this grew to only about 1.5 milliseconds. This sub-linear growth (per entity) is attributable to the efficient rule lookups and the non-iterative nature of the

deterministic logic. Most rules operate on local entity data or limited-scope group data (for CPAM). The overhead is significantly lower than typical ML model inference times, particularly for complex neural networks. This confirms HRTOS's suitability for real-time deployment in large-scale cloud environments.

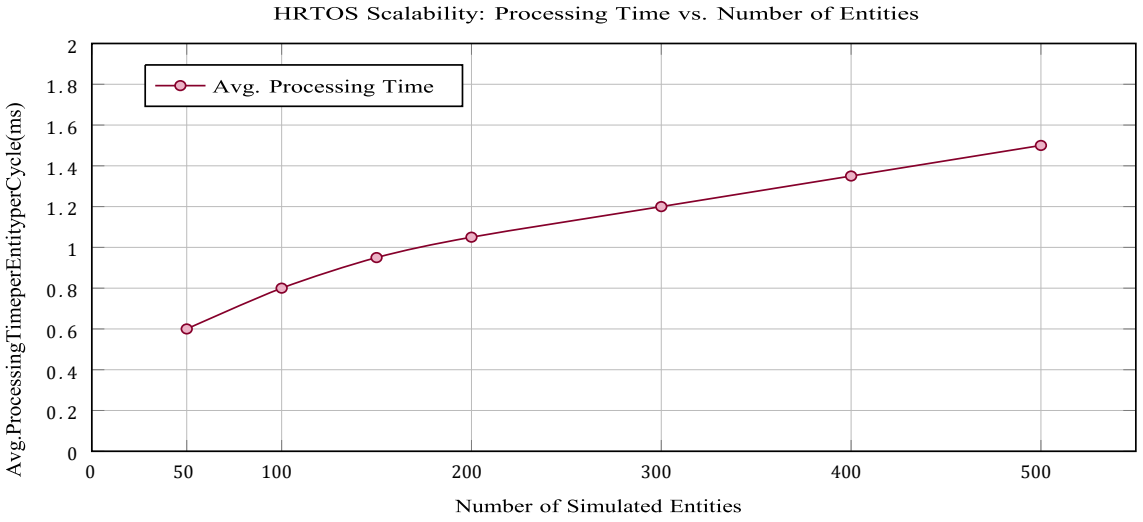


**Figure 8.** Scalability of HRTOS: Average processing time per entity per assessment cycle as the total number of simulated entities increases. The model demonstrates efficient scaling characteristics.

| Avg. Detection Latency (Intervals) | Avg. Trust Drop Post-Detection (Points) |
|---|---|

Comparative Detection Latency and Trust Impact (Conceptual)

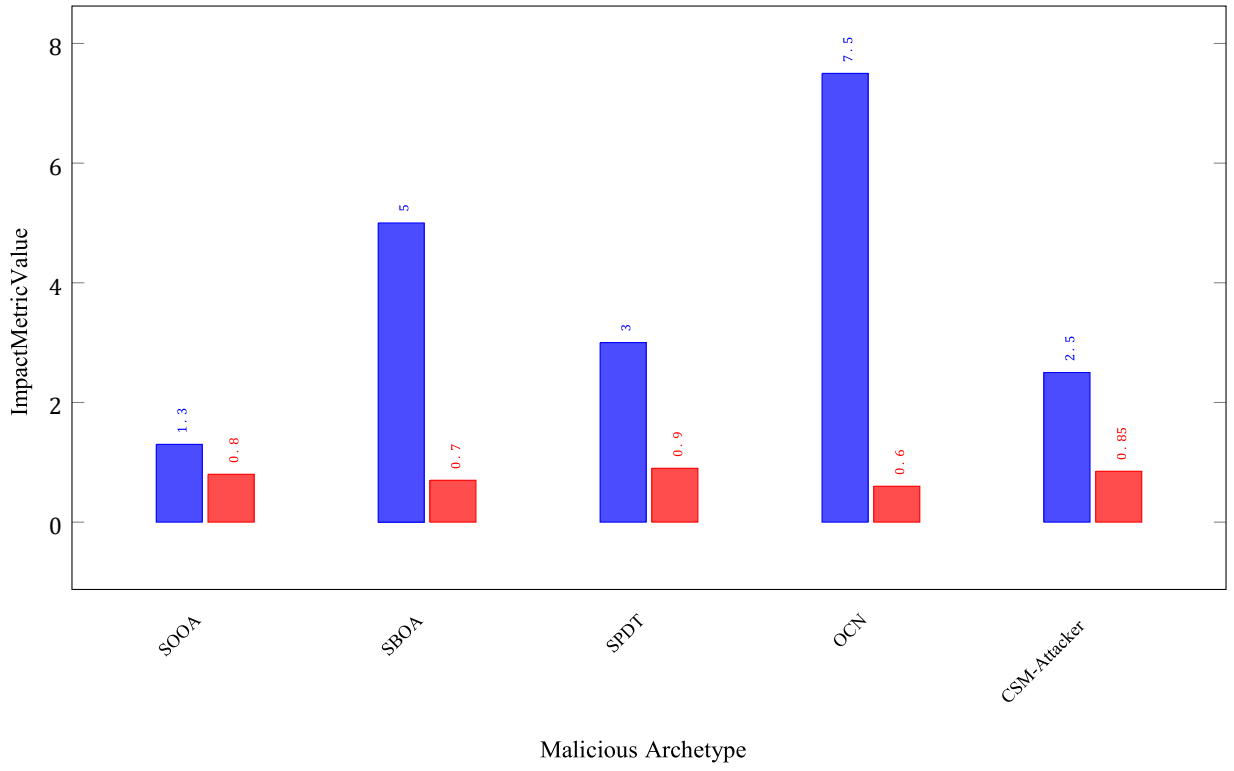


**Figure 9.** Conceptual comparative view of HRTOS performance against different malicious archetypes.

*5.6 Sensitivity Analysis of Key Parameters*

Varying key thresholds, such as $\delta_{MDI\ high}$ or $\tau_{recip}$, showed expected trade-offs. Lowering $\delta_{MDI\ high}$ (making BADM more sensitive) improved detection rates for subtle on/off attacks but slightly increased false positives for SBLUs with highly erratic (but legitimate) schedules. Increasing $\tau_{recip}$ (making CPAM more lenient on timing for reciprocity) reduced its ability to detect tight-knit colluders but also decreased false flagging of coincidentally timed positive feedback. The default parameters (Table 3) were chosen to strike a balance, but the CMM provides a mechanism to shift this balance dynamically based on operational needs. A comprehensive sensitivity analysis is beyond this paper's scope but represents an important tuning aspect for deployment.

*5.7 Comparative Discussion and Limitations*

The simulation results collectively underscore HRTOS's proficiency in rapidly and accurately identifying both on/off and collusive attacks while maintaining robust performance for benign users. Its deterministic, rule-based core provides full interpretability of decisions—an administrator can trace precisely why an entity's trust score changed or why an alert was triggered. This contrasts sharply with many ML models, as summarized in Table 6. HRTOS exhibited no cold-start problem for new users and required no computationally intensive training phases. A conceptual view of its comparative performance is shown in Figure 9.

**Table 6.** Qualitative Feature Comparison: HRTOS vs. Typical ML-based Trust Systems

| Feature/Aspect | HRTOS | Typical ML-based Approaches |
|---|---|---|
| Interpretability/Transparency | High (Deterministic rules, traceable decisions) | Low to Medium (Often "black-box", especially complex models) |

| | | |
|---|---|---|
| Training Data Dependency | None (Rule-based, no training phase) | High (Requires large, labeled datasets) |
| Cold-Start Problem | Minimal (Starts assessing from first interaction based on rules) | Significant (Poor performance on new entities without history) |
| Computational Overhead (Initial/Ongoing) | Low (No training; efficient rule processing) | High (Training); Variable (Inference, can be high for complex models) |
| Resilience to Novel Attacks | Effective if attack falls within defined behavioral feature deviations or collusive patterns; adaptable via rule updates. | May miss attacks outside training distribution; vulnerable to adversarial ML. |
| Rule Eng. Effort vs. Model Tuning | Significant initial rule design and ongoing maintenance. | Significant feature engineering, model selection, hyperparameter tuning. |
| Specificity for On/Off | High (Dedicated MDI mechanism) | Variable (May generalize but can be fooled by strategic behavior) |
| Specificity for Collusion | High (Dedicated CPAM with pattern grammar) | Variable (GNNs show promise but can be complex; simpler models may struggle) |
| Auditability | High (Decisions directly map to rules) | Low (Difficult to audit internal decision logic of complex models) |
| Susceptibility to Adversarial ML | Low (Not learning-based) | High (Data poisoning, evasion attacks) |

However, HRTOS is not without limitations. The efficacy of a rule-based system is intrinsically tied to the quality and comprehensiveness of its rule set. Crafting and maintaining these rules can be a significant knowledge engineering effort, especially as new attack vectors emerge. While CMM provides some adaptability, the fundamental rules are static. Highly sophisticated adversaries, upon understanding the rule set (if it were to become public), might devise strategies to evade specific rules (”gaming the system”). This is a challenge for any transparent security system. The current ILFVM relies on predefined feature extraction; unforeseen malicious behaviors manifesting through entirely novel feature dimensions might be missed until rules are updated. Furthermore, while efficient, the communication overhead for collecting granular logs for ILFVM in large, geographically distributed federations could become a factor, though this is a challenge for any centralized or logically centralized trust system.

In summary, the simulations validate HRTOS as an efficient trust management framework for its targeted threat landscape. Its strengths in transparency, low overhead, and rapid response to specific complex attacks make it a valuable candidate for enhancing security in federated cloud environments.

## 6. Conclusion and Future Work

This Paper has introduced the Hierarchical Rule-Based Trust Orchestration System (HRTOS), a deterministic framework designed to address the persistent and evolving challenges of trust manipulation within complex cloud federations. By focusing on the explicit, rule-driven detection of insidious threats such as on/off behavioral subterfuge and orchestrated collusion, HRTOS offers a compelling alternative to prevailing trust models that often grapple with opacity, computational intensity, or vulnerability to adversarial learning. The system’s modular architecture, encompassing multi-vector behavioral fingerprinting, advanced feedback integrity assessment, collusion pattern grammar analysis, and a novel context-aware rule modification layer, has demonstrated through rigorous simulation its capacity for high-fidelity threat discernment, rapid response, and minimal false positive incidence, all while maintaining complete operational transparency and low computational overhead.

The imperative for robust trust mechanisms in cloud computing is undeniable, as these environments form the critical substrate for countless digital services. The inherent vulnerabilities of shared, distributed infrastructures are frequently exploited by attackers who adeptly mimic legitimate behavior or coordinate deceptively to undermine system integrity. HRTOS directly confronts these challenges by eschewing black-box learning paradigms in favor of an interpretable, auditable system of logical rules. Its ability to dissect behavioral trajectories via the Multi-faceted Deviation Index, to critically appraise feedback veracity through the Feedback Integrity Verification Module, and to unmask coordinated malice using the Collusion Pattern Analysis Module, all without prior training data or probabilistic ambiguity, positions it as a uniquely suitable solution for dynamic, security-conscious cloud ecosystems. The Contextual Modifier Module further enhances HRTOS by allowing deterministic adaptation of its vigilance based on prevailing systemic conditions or the criticality of interacted assets, providing a nuanced yet predictable responsiveness.

The empirical evaluations presented herein affirm HRTOS's consistent performance across a spectrum of user archetypes. Benign entities experience fair trust accumulation, while various sophisticated adversarial strategies are promptly identified and neutralized. The system's architecture inherently circumvents issues like training data dependency and cold-start problems, rendering it deployable and effective from inception. Its low latency and efficient scalability further underscore its practical applicability in real-world, large-scale cloud deployments.

### *6.1 Limitations*

Despite its demonstrated strengths, HRTOS possesses limitations inherent to rule-based systems. The development and maintenance of a comprehensive and resilient rule set require significant domain expertise and ongoing effort to adapt to entirely novel, unforeseen attack methodologies. While the HRTOS design emphasizes modularity to facilitate updates, the core logic remains predicated on human-defined rules. Adversaries with perfect knowledge of the rule set could, in theory, devise strategies to operate just below detection thresholds or exploit unarticulated rule interactions, although the hierarchical and multi-faceted nature of HRTOS makes such evasion non-trivial. The system's effectiveness is also contingent on the quality and granularity of input data from the Interaction Logging and Feature Vectorization Module; deficiencies in data collection could impair detection capabilities.

The HRTOS framework opens several promising avenues for future research and enhancement:

1. Semi-Automated Rule Refinement and Discovery: Exploring techniques for assisting human experts in refining existing rules or discovering new candidate rules based on observed anomalous patterns that HRTOS flags but cannot fully categorize. This could involve explainable AI techniques to suggest rule modifications, while keeping the final rule definition and validation under human control to maintain determinism.

2. Formal Verification of Rule Sets: Applying formal methods to verify the HRTOS rule base for internal consistency, completeness against known attack classes, and absence of undesirable emergent behaviors or conflicting rules. This could significantly enhance the robustness and predictability of the system.

3. Enhanced Contextual Awareness: Expanding the CMM's capabilities to incorporate a richer set of contextual inputs, such as dynamic threat intelligence feeds, geopolitical risk factors affecting federated partners, or finegrained workload characteristics, to enable even more nuanced rule modulation.

4. Integration with Orchestration and SIEM Systems: Developing standardized interfaces for HRTOS to integrate seamlessly with cloud orchestration platforms (e.g., Kubernetes, OpenStack) to enable trust-based dynamic access control and resource scheduling. Integration with Security Information and Event Management (SIEM) systems would allow HRTOS to consume broader security event data and contribute its trust assessments to a global security posture.

5. Standardized Trust Policy Language: Investigating the development of a high-level, declarative language for expressing trust policies and behavioral rules within HRTOS and similar systems. This could simplify rule management and promote interoperability across different trust frameworks.

6. Application to Emerging Distributed Paradigms: Extending and adapting the HRTOS principles for other distributed computing paradigms facing similar trust challenges, such as the Internet of Things (IoT), edge computing, serverless architectures, and decentralized finance (DeFi) platforms.

7. Game-Theoretic Rule Adaptation Strategies: Researching how game theory could inform the CMM or a higher-level strategic module to adapt rule thresholds and penalty structures in anticipation of rational adversaries attempting to ”game” the system, moving beyond purely reactive rule adjustments.

8. Privacy-Preserving HRTOS Operations: In scenarios involving sensitive data or cross-jurisdictional federations, exploring techniques like homomorphic encryption or secure multi-party computation to enable HRTOS modules to operate on encrypted data or share trust-relevant insights without revealing raw underlying data.

In conclusion, the Hierarchical Rule-Based Trust Orchestration System offers a significant step towards more transparent, efficient, and resilient trust management in cloud federations. While the pursuit of perfect security remains an ongoing journey, HRTOS provides a robust and practical framework for proactively mitigating some of the most challenging trust-related threats in contemporary and future cloud ecosystems.

**Corresponding author**

**Qais Al-Na’amneh**
Al-Na'amnehadd@gmail.com

**Acknowledgements**
NA

**Funding**
NA

**Contributions**
**Conceptualization**, Q.A; M.A; A.S.A; R.H; S.M.S;W.D; **Methodology**, Q.A; M.A; A.S.A; R.H; S.M.S;W.D; **Software**, Q.A; M.A; A.S.A; R.H; S.M.S;W.D; **Validation**, Q.A; M.A; A.S.A; R.H; S.M.S;W.D; **Formal Analysis**, Q.A; M.A; A.S.A; R.H; S.M.S;W.D; **Investigation**, Q.A; M.A; A.S.A; R.H; S.M.S;W.D; **Resources**; **Data Curation**, Q.A; M.A; A.S.A; R.H; S.M.S;W.D; **Writing (Original Draft)**, Q.A; **Writing (Review and Editing)**, Q.A; **Visualization**, Q.A; **Supervision**; Q.A; **Project Administration**, Q.A; **Funding Acquisition**, Q.A. All authors have read and agreed to the published version of the manuscript.

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

**Consent for publication**
Not applicable.

**Competing interests**
All authors declare no competing interests

**References**

[1] Mona Soleymani, Navid Abapour, Elham Taghizadeh, Safieh Siadat, and Rasoul Karkehabadi. Fuzzy rule-based trust management model for the security of cloud computing. *Mathematical problems in engineering*, 2021(1): 6629449, 2021.

[2] Rajanpreet Kaur Chahal and Sarbjeet Singh. Fuzzy rule-based expert system for determining trustworthiness of cloud service providers. *International Journal of Fuzzy Systems*, 19:338–354, 2017.

[3] Poorva Rathi, Himanshu Ahuja, and Kavita Pandey. Rule based trust evaluation using fuzzy logic in cloud computing. In *2017 6th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*, pages 510–514. IEEE, 2017.

[4] Shweta Loonkar, Neeti Taneja, and N Beemkumar. Fuzzy rule-based trust management for cloud security. In *2024 1st International Conference on Sustainable Computing and Integrated Communication in Changing Landscape of AI (ICSCAI)*, pages 1–12. IEEE, 2024.

[5] Venkatarama Reddy Kommidi, Srikanth Padakanti, and Vasudev Pendyala. Securing the cloud: A comprehensive analysis of data protection and regulatory compliance in rule-based eligibility systems. *Technology (IJRCAIT)*, 7(2), 2024.

[6] Abhishek Kesarwani and Pabitra Mohan Khilar. Development of trust based access control models using fuzzy logic in cloud computing. *Journal of King Saud University-Computer and Information Sciences*, 34(5):1958– 1967, 2022.

[7] Md Shihabul Islam, Mustafa Safa Ozdayi, Latifur Khan, and Murat Kantarcioglu. Secure iot data analytics in cloud via intel sgx. In *2020 IEEE 13th international conference on cloud computing (CLOUD)*, pages 43–52. IEEE, 2020.

[8] C Veena, S Ramalakshmi, V Bhoopathy, Minakshi Dattatraya Bhosale, CG Magadum, and Abirami SK. Effective intrusion detection and classification using fuzzy rule based classifier in cloud environment. In *2022 International Conference on Automation, Computing and Renewable Systems (ICACRS)*, pages 497–502. IEEE, 2022.

[9] Yuqing Wang and Xiao Yang. Research on enhancing cloud computing network security using artificial intelligence algorithms. *arXiv preprint arXiv:2502.17801*, 2025.

[10] Sabah M Alturfi, Dena Kadhim Muhsen, Mohammed A Mohammed, Israa T Aziz, and Mustafa Aljshamee. A combination techniques of intrusion prevention and detection for cloud computing. In *Journal of Physics: Conference Series*, volume 1804, page 012121. IOP Publishing, 2021.

[11] Qais Al-Na'amneh, Ammar Almomani, Ahmad Nasayreh, Khalid MO Nahar, Hasan Gharaibeh, Rabia Emhamed Al Mamlook, and Mohammad Alauthman. Next generation image watermarking via combined dwt-svd technique. In *2024 2nd International Conference on Cyber Resilience (ICCR)*, pages 1–10. IEEE, 2024.

[12] Ameera S Jaradat, Ahmad Nasayreh, Qais Al-Na'amneh, Hasan Gharaibeh, and Rabia Emhamed Al Mamlook. Genetic optimization techniques for enhancing web attacks classification in machine learning. In *2023 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, pages 0130–0136. IEEE, 2023.

[13] Dena Abu Laila, Qais Al-Na'amneh, Mohammad Aljaidi, Ahmad Nawaf Nasayreh, Hasan Gharaibeh, Rabia Al Mamlook, and Mohammed Alshammari. Simulation of routing protocols for jamming attacks in mobile ad-hoc network. In *Risk Assessment and Countermeasures for Cybersecurity*, pages 235–252. IGI Global, 2024.

[14] Saydul Akbar Murad, Abu Jafar Md Muzahid, Zafril Rizal M Azmi, Md Imdadul Hoque, and Md Kowsher. A review on job scheduling technique in cloud computing and priority rule based intelligent framework. *Journal of King Saud University-Computer and Information Sciences*, 34(6):2309–2331, 2022.

[15] Hamza Nasir, Azeem Ayaz, Shahzmaan Nizamani, Saima Siraj, Shahid Iqbal, and M Kamran Abid. Cloud computing security via intelligent intrusion detection mechanisms. *International Journal of Information Systems and Computer Technologies*, 3(1):84–92, 2024.

[16] Sunil Kumar Parisa and Somnath Banerjee. Ai-enabled cloud security solutions: A comparative review of traditional vs. next-generation approaches. *International Journal of Statistical Computation and Simulation*, 16(1), 2024.

[17] Uma Rani, Surjeet Dalal, and Jugnesh Kumar. Optimizing performance of fuzzy decision support system with multiple parameter dependency for cloud provider evaluation. *Int. J. Eng. Technol*, 7(1.2):61–65, 2018.

[18] Jomina John and K John Singh. Trust value evaluation of cloud service providers using fuzzy inference based analytical process. *Scientific Reports*, 14(1):18028, 2024.

[19] Vijay Ramamoorthi. Anomaly detection and automated mitigation for microservices security with ai. *Applied Research in Artificial Intelligence and Cloud Computing*, 7(6):211–222, 2024.

[20] Monika Mehata. Dynamic zero trust access control: Fortifying security in mobile-cloud environment. Master's thesis, Youngstown State University, 2025.

[21] Ivan Parkhomenko, Larysa Myrutenko, Roman Ohiievych, and Mykhailo Savonik. Using zero trust principles for detecting authorization attacks in cloud environments. 2024.

[22] Himadri Shekhar Mondal, Md Tariq Hasan, Md Bellal Hossain, Md Ekhlasur Rahaman, and Rabita Hasan. Enhancing secure cloud computing environment by detecting ddos attack using fuzzy logic. In *2017 3rd international conference on electrical information and communication technology (EICT)*, pages 1–4. IEEE, 2017.

[23] Diwakar Chaudhary, SK Verma, Vijay Mohan Shrimal, Ravikiran Madala, Rashi Baliyan, et al. Ai-based methods to detect and counter cyber threats in cloud environments to strengthen cloud security. In *2024 International Conference on Electrical Electronics and Computing Technologies (ICEECT)*, volume 1, pages 1–6. IEEE, 2024.

[24] Sudakshina Mandal, Danish Ali Khan, and Sarika Jain. Cloud-based zero trust access control policy: an approach to support work-from-home driven by covid-19 pandemic. *new generation computing*, 39(3):599–622, 2021.

[25] Somnath Banerjee, Pawan Whig, and Sunil Kumar Parisa. Cybersecurity in multi-cloud environments for retail: An ai-based threat detection and response framework. *Transaction on Recent Developments in Industrial IoT*, 16 (16), 2024.

[26] Quan Shen and Yanming Shen. Endpoint security reinforcement via integrated zero-trust systems: A collaborative approach. *Computers & Security*, 136:103537, 2024.

[27] Chirag N Modi and Kamatchi Acha. Virtualization layer security challenges and intrusion detection/prevention systems in cloud computing: a comprehensive review. *the Journal of Supercomputing*, 73(3):1192–1234, 2017.

[28] Lewis Golightly, Paolo Modesti, Remi Garcia, and Victor Chang. Securing distributed systems: A survey on´ access control techniques for cloud, blockchain, iot and sdn. *Cyber Security and Applications*, 1:100015, 2023.

[29] Mohammad Amin Hatef, Vahid Shaker, Mohammad Reza Jabbarpour, Jason Jung, and Houman Zarrabi. Hidcc: A hybrid intrusion detection approach in cloud computing. *Concurrency and Computation: Practice and Experience*, 30(3):e4171, 2018.

[30] Lokesh B Bhajantri and Tabassum Mujawar. A survey of cloud computing security challenges, issues and their countermeasures. In *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, pages 376–380. IEEE, 2019.

[31] Naresh Kumar Sehgal, Pramod Chandra P Bhatt, and John M Acken. Future trends in cloud computing. In *Cloud computing with security and scalability. concepts and practices*, pages 289–317. Springer, 2022.

[32] Omer Aslan, Merve Ozkan-Okay, and Deepti Gupta. Intelligent behavior-based malware detection system on¨ cloud computing environment. *IEEE Access*, 9:83252–83271, 2021.

[33] Jibu K Samuel, Mahima Thankam Jacob, Melvin Roy, Sayoojya PM, and Anu Rose Joy. Intelligent malware detection system based on behavior analysis in cloud computing environment. In *2023 International Conference on Circuit Power and Computing Technologies (ICCPCT)*, pages 109–113. IEEE, 2023.

[34] Muzammil Ahmad Khan, Shariq Mahmood Khan, and Siva Kumar Subramaniam. Secured dynamic request scheduling and optimal csp selection for analyzing cloud service performance using intelligent approaches. *IEEE Access*, 11:140914–140933, 2023.

[35] Mufti Mahmud, M Shamim Kaiser, M Mostafizur Rahman, M Arifur Rahman, Antesar Shabut, Shamim AlMamun, and Amir Hussain. A brain-inspired trust management model to assure security in a cloud based iot framework for neuroscience applications. *Cognitive Computation*, 10:864–873, 2018.

[36] Matin Chiregi and Nima Jafari Navimipour. A comprehensive study of the trust evaluation mechanisms in the cloud computing. *Journal of Service Science Research*, 9:1–30, 2017.

[37] M Arunkumar and K Ashok Kumar. Malicious attack detection approach in cloud computing using machine learning techniques. *Soft Computing*, 26(23):13097–13107, 2022.

[38] Faheem Raza and Nasir Hussain. Ai-infused dspm for cloud security: Machine learning-based anomaly detection solutions. 2023.

[39] Sunil Kumar Parisa, Somnath Banerjee, and Pawan Whig. Ai-driven zero trust security models for retail cloud infrastructure: A next-generation approach. *International Journal of Sustainable Devlopment in field of IT*, 15 (15), 2023.

[40] Behnam Mohammad Hasani Zade, Najme Mansouri, and Mohammad Masoud Javidi. Saea: A security-aware and energy-aware task scheduling strategy by parallel squirrel search algorithm in cloud environment. *Expert Systems with Applications*, 176:114915, 2021.

[41] Femi Emmanuel Ayo, Sakinat Oluwabukonla Folorunso, Adebayo A Abayomi-Alli, Adebola Olayinka Adekunle, and Joseph Bamidele Awotunde. Network intrusion detection based on deep learning model optimized with rule-based hybrid feature selection. *Information Security Journal: A Global Perspective*, 29(6): 267–283, 2020.

[42] Shekha Chenthara, Khandakar Ahmed, Hua Wang, and Frank Whittaker. Security and privacy-preserving challenges of e-health solutions in cloud computing. *IEEE access*, 7:74361–74382, 2019.

[43] Himanshu Kale, Pravin Nerkar, and Rupesh Hushangabade. Design of model for data security in cloud computing environment.

[44] Charlotte Muller and Niklas Wagner. Machine learning in cloud security: Enhancing anomaly detection and¨ response. *Eastern-European Journal of Engineering and Technology*, 3(1):42–50, 2024.

[45] S Priya and RS Ponmagal. Trust based reputation framework for data center security in cloud computing environment. In *2023 7th International Conference on Computing Methodologies and Communication (ICCMC)*, pages 1041–1047. IEEE, 2023.

[46] Roberto F Mercado. Identifying the advantages of zero-trust architecture in the cloud environment. Master's thesis, Utica University, 2022.

[47] Qais Al-Na'amneh, Mahmoud Aljawarneh, Rahaf Hazaymih, Laith Alzboon, Dena Abu Laila, and Sahel Albawaneh. Trust evaluation enhancing security in the cloud market based on trust framework using metric parameter selection. 2025.

[48] Qais Al-Na'amneh, Mohammad Aljaidi, Ahmad Nasayreh, Hasan Gharaibeh, Al Mamlook, Rabia Emhamed, Ameera S Jaradat, Ayoub Alsarhan, and Ghassan Samara. Journal of intelligent systems: Enhancing iot device security: Cnn-svm hybrid approach for real-time detection of dos and ddos attacks. 2024.

[49] Mohammad Aljaidi, Ayoub Alsarhan, Dimah Al-Fraihat, Ahmed Al-Arjan, Bashar Igried, Subhieh M El-Salhi, Muhammad Khalid, and Qais Al-Na'amneh. Cybersecurity threats in the era of ai: Detection of phishing domains through classification rules. In *2023 2nd International Engineering Conference on Electrical, Energy, and Artificial Intelligence (EICEEAI)*, pages 1–6. IEEE, 2023.

[50] Qais Al-Na'amneh, Mahmoud Aljawarneh, Rahaf Hazaymih, and Rabia Emhamed Al Mamlook. Ethical issues in cyber-security for autonomous vehicles (av) and automated driving: A comprehensive review. *Utilizing AI in Network and Mobile Security for Threat Detection and Prevention*, pages 173–196, 2025.

[51] Qais Al-Na'amneh, Mohammad Aljaidi, Hasan Gharaibeh, Ahmad Nasayreh, Rabia Emhamed Al Mamlook, Sattam Almatarneh, Dalia Alzu'bi, and Abla Suliman Husien. Feature selection for robust spoofing detection: A chi-square-based machine learning approach. In *2023 2nd International Engineering Conference on Electrical, Energy, and Artificial Intelligence (EICEEAI)*, pages 1–7. IEEE, 2023.