electronics

MDPI

Article

Trust-Based Selective Collaboration for Multi-Agent Intrusion Detection in IIoT

Gustavo Funchal [1,*], Tiago Pedrosa [1], Fernando De la Prieta [2] and Paulo Leitão [1]

[1] Research Centre in Digitalization and Intelligent Robotics (CeDRI), Associate Laboratory for Sustainability and Technology in Mountains Regions (SusTEC), Bragança Polytechnic University, Campus de Santa Apolónia, 5300-253 Bragança, Portugal; pedrosa@unipb.pt (T.P.); pleitao@unipb.pt (P.L.)

[2] BISITE Research Group, University of Salamanca, Edificio I+D+i, C/ Espejos s/n, 37007 Salamanca, Spain; fer@usal.es

* Correspondence: gustavofunchal@unipb.pt

Abstract

In distributed intrusion detection systems for the Industrial Internet of Things (IIoT), collaboration among heterogeneous agents can compensate for each agent's partial competence. However, when done indiscriminately, it exposes the agents most capable of negative transfer, degrading decisions that would otherwise be correct on their own. This work reframes the problem as an admission problem, shifting the question from whether to collaborate to with whom to collaborate, and proposes a mechanism in which each agent maintains, in a strictly local manner, a trust matrix that directs its requests for help only to peers whose historical contributions have proven beneficial, without altering the system's aggregation rule. The study compares four trust-updating strategies (exponential moving average, Beta-Bernoulli, upper confidence bound, and binary) and two admission policies (threshold and top-k) against baseline strategies of isolated decision-making and indiscriminate collaboration, on a testbed with ten heterogeneous agents running on real hardware (Raspberry Pi) and using two benchmark datasets with different degrees of heterogeneity (UNSW-NB15 and CICIoT2023). The results show that selective collaboration based on trust preserves the gains of weak agents and suppresses the degradation of strong agents. The benefit is moderate under purely algorithmic heterogeneity, but becomes the factor that makes collaboration worthwhile when agents are trained on distinct datasets, where naive collaboration offers almost no benefit. The trust mechanism also reduces the total message volume by 44% to 84% depending on the configuration (exceeding 60% under top-k admission) and cuts collaboration latency by approximately 44% to 77% depending on the configuration, without compromising resources, and with its value growing with the degree of heterogeneity, from marginal when agents share knowledge to the factor that decides whether collaboration is beneficial when they do not share it.

Keywords: intrusion detection systems; multi-agent systems; Industrial Internet of Things; selective collaboration

Check for updates

Academic Editors: Hung-Yu Chien and Yu-an Tan

Received: 14 September 2026
Revised: 29 October 2026
Accepted: 6 November 2026
Published: 9 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

The Industrial Internet of Things (IIoT) exposes a broad and heterogeneous attack surface across the edge–cloud continuum, where concentrating detection at a single point becomes a scalability and latency bottleneck [1]. Distributed intrusion detection systems (IDSs) address this by placing detection close to the devices and organizing it as a Multi-Agent System (MAS), in which autonomous agents monitor their local environment, decide

independently, and cooperate with their peers to detect threats that exceed the visibility of any single node [2,3]. This paradigm reduces latency, preserves data privacy, and avoids the bottlenecks of a centralized architecture. However, each agent operates with limited resources and from a necessarily partial view of the environment, observing only its segment of traffic and having been trained with a subset of the available knowledge. As a result, the competence of each agent is incomplete when considered individually.

It is from this limitation that the interest in collaboration arises. When an agent lacks sufficient knowledge to reliably classify a sample, consulting peers allows the agent to draw on expertise that it does not possess, and a decision made collectively can prove much more effective than one made in isolation. The benefit is particularly significant for weaker agents, whose accuracy increases substantially when supported by more capable neighbors in a given situation [4]. Therefore, the collaboration emerges not as an architectural feature, but as a mechanism with a direct impact on detection quality.

However, this benefit is not unconditional. In a realistic distributed system, agents are heterogeneous, differing in their classification algorithms, the data they were trained on, and, consequently, the skills they have acquired. When collaboration is indiscriminate, i.e., when an agent considers the opinion of any peer without distinguishing its suitability for the case at hand, it opens the door for a neighbor unprepared for that situation to compromise a decision that the agent would have made correctly on its own. Collaboration then begins to introduce error rather than correct it, an effect known as negative transfer [5], which tends to worsen precisely in the most heterogeneous environments, where collaboration would be most desirable.

From this perspective, the problem moves from whether an agent should collaborate to with whom an agent should collaborate. If an agent could identify, among its peers, those who are truly competent for the situation it faces, it could preserve the benefits of cooperation while avoiding its negative consequences. Trust among peers is the natural tool for making this distinction; by accumulating evidence about each neighbor's past performance, an agent builds a sense of reliability and uses it to direct requests for help only to those most likely to respond to them. Thus, collaboration becomes a problem of selection.

However, establishing trust is not a one-time process. There are various ways to update it based on interaction, from exponential moving averages to beta probabilistic models or uncertainty-sensitive approaches [6–8], and various ways to convert it into a selection decision, whether by considering all peers above a reliability threshold or by limiting the choice to a fixed number of the most reliable ones. It is not immediately clear which of these combinations produces morore effective collaborations, neither under what conditions of heterogeneity each one pays off. The literature on trust in MAS offers many models but rarely tests them empirically under the constraints of a real-world detection scenario, with real data and execution on limited hardware [9]. It is this gap, the absence of an empirical characterization of *how* and *when* trust improves collaboration, that motivates this work.

In this work, the trust mechanism is isolated and studied. Three operating regimes are compared, namely agents that decide independently (SOLO), agents that collaborate without discriminating among peers (NAIVE), and agents that collaborate selectively based on trust (TRUST). Within the TRUST regime, multiple strategies for updating trust and selecting peers are explored. The evaluation is conducted on real hardware using two benchmark datasets that represent different degrees of heterogeneity: a controlled scenario, in which agents share training data and differ only in their algorithms, and a genuinely heterogeneous scenario, in which each agent is trained and tested with different data. The scope of the analysis was deliberately limited: trust influences only the selection of

whom to consult, while the aggregation method and the system's other adaptive mechanisms remain fixed, in order to isolate the effect of selection and support causal conclusions regarding the reduction of negative transfer.

The main contributions of this study are as follows:

- Formulation of negative transfer in collaborative detection as a selection problem under heterogeneity, explicitly articulating the tension between the benefits and risks of collaboration;
- Presentation of a comparative study of trust strategies, combining update and selection methods across the SOLO, NAIVE and TRUST triad;
- Validation of the analysis on real hardware and under two heterogeneity regimes, which lends the observations a robustness that simulation alone can not guarantee;
- Translation of the results into practical guidelines on when and how to apply each strategy, offering an informed trade-off between the effectiveness and efficiency of collaboration.

The remainder of the article is organized as follows. Section 2 presents related work; Section 3 formalizes the trust mechanism; Section 4 details the setup, datasets, and experimental design; Section 5 discusses the results; and Section 6 concludes and points to future work.

2. Related Work

The limitations of monolithic, centralized intrusion detection have been driving the development of architectures in which multiple network monitors exchange information to detect attacks that exceed the visibility of any single node. Fundamental studies formalize this as a spectrum ranging from centralized to hierarchical to fully distributed collaboration, and consistently justify collaborative IDSs by the need to combine local observations into a broader view of the network, while avoiding the scalability limits of single-node deployments [10,11]. This concern is most prominent in Internet of Things (IoT) and IIoT environments, where latency, bandwidth, privacy, and device constraints make purely centralized monitoring unattractive. Therefore, recent research emphasizes distributed, edge and multi-access edge computing (MEC)-oriented designs that distribute decision-making, the model, or both across the cloud, edge and device layers [12]. Hardware-aware studies confirm that local inference on constrained platforms is feasible [13], and hybrid edge AI designs combine distributed detection with privacy-preserving training with competitive accuracy, although strict real-time feasibility for heavier models remains a challenge [14].

A second approach adopts MAS as the basis for coordination, since agents naturally provide autonomous monitoring, distributed problem-solving, and explicit collaboration between network segments [15,16]. Beyond simple reporting, several architectures allow agents to exchange intermediate detection evidence to improve global assessment, through peer-to-peer and cross-network coordination that reduces false positives [17] or through local–global hierarchical collaboration [18]. The authors' recent works [2,3,19] are based more directly on this vision of evidence exchange, coordinating at the time of inference rather than through centralized retraining. Federated learning (FL) is an important related paradigm that keeps raw traffic local while sharing model updates and is increasingly positioning itself as a solution for privacy-sensitive, non-IID, and resource-constrained IIoT deployments [20–22], demonstrating competitive performance compared to centralized training. However, it differs from the configuration proposed by the authors, since FL distributes model optimization toward a shared global predictor, whereas in the proposed configuration, the decision itself is distributed among agents that maintain permanently

heterogeneous local models and collaborate only at the time of inference. Thus, the two paradigms are complementary, not competing.

In cases where multiple detector systems contribute to a decision, their outputs must be combined. The literature largely addresses this issue through the selection of aggregation rules, such as majority voting, weighted averaging, and stacking, to merge multiple basic classifiers into a single label [23,24], with weighted variants assigning greater influence to members considered more reliable [25]. These works largely optimize the form of the combiner, while treating the set of contributors as fixed and without issues. This assumption is fragile under conditions of heterogeneity, as set theory holds that gains depend jointly on the accuracy and diversity of the members, rather than on the number of members [26], and empirical studies warn that selection based strictly on diversity may underperform selection that takes accuracy into account and may even harm deep ensembles [27]. Therefore, a contributor is only helpful if its errors are sufficiently complementary and its competence overlaps with the target decision region.

This observation is directly related to negative transfer, in which leveraging knowledge from the source domain reduces, rather than improves, performance on the target task [5]. It results from incompatibility between source and target, inadequate source data, or misalignment between subdomains, requiring explicit source selection or the suppression of poorly compatible sources [28]. In our context, the mapping is straightforward, as the source domain is a queried entity, the target domain is the querying agent, and the transferred knowledge is the prediction of the entity incorporated into the local decision at the time of inference. The fact that this is more than just an analogy is demonstrated by federated and collaborative studies that report that naive global aggregation or indiscriminate collaboration can harm some participants under heterogeneous distributions, whether through the formation of coalitions based on distribution differences or through instance-wise competence estimation [29,30]. However, these approaches operate primarily during training through client clustering or model selection mechanisms, and they do not address the deployment scenario considered in this work, in which there is a set of peers and the system must decide, in real time and based on samples, which peers are worth querying within a latency constraint.

Given that, trust modeling provides the natural mechanism for these decisions. Fundamental studies on computational trust and reputation formalize trust as an essential means of supporting partner evaluation and trust-based interaction in situations of uncertainty regarding the reliability of a counterparty [31–34]. A standard taxonomy distinguishes between direct trust models, which infer trust based on an agent's own past observations; indirect or reputation-based models, which incorporate third-party testimonials, at the cost of having to filter out inaccurate or misleading sources; socio-cognitive models, which exploit social relationships; and organizational models, which assess agent reliability using their organizational credentials and trusted certificates [33]. The mechanism proposed in this work deliberately belongs to the first category, estimating trust exclusively from an agent's own interactions with a partner. Among experience-based models, Bayesian formulations are the most relevant, with the Beta Reputation System representing reputation through beta densities over accumulated positive and negative evidence [35], and TRAVOS defines trust as a subjective probability constructed from past interactions with explicit trust [36]. Complementary formulations of the bandit model frame the choice of whom to consult as an exploration–exploitation problem, in which upper confidence bound strategies balance a high estimated reward with partners that have yet to be fully explored [37]. Both approaches favor incremental updates in finite states, which is particularly attractive in distributed environments with resource constraints, where agents must evaluate many partners based on limited evidence.

However, applying trust to collaborative detection requires a shift in what trust represents. The broader literature on trust deals with open systems in which partners may be acting in their own self-interest or in a malicious manner, so trust protects against harmful interactions [32,33]. In contrast, the scenario of this study assumes a fully collaborative IDS, in which peers provide accurate information, and the relevant failure mode is not dishonesty but heterogeneity, since an honest peer may still be an unsuitable advisor when their expertise is limited to traffic conditions different from those of the agent asking for assistance. Therefore, trust is more restricted and relational, as it is not a global property of a peer, but an estimate of how useful that peer's predictions have historically been for a specific consulting agent under its own observation regime. The reason the witness reputation is not suitable is that aggregating it across the entire system would obscure the agent-specific overlap that the mechanism is designed to preserve, in addition to increasing message overhead and creating a filtering problem. This also changes the point at which trust comes into play, as instead of ranking candidates or weighting external information, trust is used as an admission criterion for the set of respondents, since reducing the weighting still allows for the admission of a poorly suited contribution, whereas the admission control can remove it entirely and reduce communication all at once. In the literature reviewed, this specific use of trust based on direct experience for online peer admission, by sampling, in collaborative intrusion detection does not appear to have been explicitly addressed, which supports the position taken in this paper.

3. Trust Mechanism

This section formalizes the trust mechanism. Section 3.1 presents the definition of the system model and the notation, while Section 3.2 presents the central structure, represented by each agent's internal trust matrix, along with its properties. Section 3.3 presents the selective collaboration protocol, and Section 3.4 describes the feedback signal that guides trust adaptation. Then, the four update rules under comparison (EMA, Beta-Bernoulli, UCB, and Binary) are described in Section 3.5, and the two collaborator admission policies are presented in Section 3.6. Finally, Section 3.7 addresses the cold-start problem, and Section 3.8 addresses communication costs.

3.1. System Model

Consider a set $\mathcal{A} = \{1, \ldots, N\}$ of detection agents positioned on the edge. Each agent i hosts a locally trained classifier, h_i, and observes a private stream of traffic samples. Given a sample x, agent i produces a prediction $\hat{y}_i(x) \in \{0, 1\}$ and an associated confidence $\pi_i(x) \in [0, 1]$.

Agents are heterogeneous, differing in the learning algorithm they employ and in the distribution of the data on which they were trained. As a result, the value of consulting a given peer varies substantially from peer to peer, and indiscriminate collaboration exposes an agent to the phenomenon of negative transfer, i.e., the degradation of the quality of its own decision caused by incorporating unreliable external evidence. This work assumes a collaborative threat model, in which peers are honest and report their genuine predictions, and the relevant failure mode is heterogeneity of competence, not adversarial behavior. In other words, an honest peer may still be an inadequate advisor when their expertise does not match the traffic conditions of the inquiring agent, and it is this benign form of unreliability, being the root cause of negative transfer that the trust mechanism is designed to filter out. Adversarial scenarios, in which peers may be compromised or act deliberately deceptively, and suboptimal feedback, such as delayed or noisy labels, are outside the scope of this study, but their expected effect on the mechanism is discussed in Section 6.

The proposed mechanism mitigates this risk through admission control, in which trust determines whom an agent asks for help/collaboration.

Definition 1 (Trust Vector and Matrix)**.** *Each agent i maintains a trust vector $\mathbf{T}_i = (T_{i1}, \ldots, T_{iN}) \in [0,1]^N$, where T_{ij} quantifies the level to which i expects a query to j to improve its decision, based on the history of its own interactions. The collection $\{\mathbf{T}_i\}_{i \in \mathcal{A}}$ forms a trust matrix $\mathbf{T} \in [0,1]^{N \times N}$.*

It should be emphasized that each agent estimates its confidence strictly locally, based on the interactions it observes. There is no shared global matrix, nor reputation propagation among agents. The symbols used throughout the paper are summarized in Table 1.

Table 1. Notation.

Symbol	Meaning
N	number of agents
T_{ij}	agent i's trust in agent j
$\mathcal{C}_i$	set of collaborators admitted by i
r_{ij}	binary reward from an interaction with j
n_{ij}	number of informative interactions with j
n_i	total number of informative interactions of i
η	trust learning rate (EMA)
ζ	update step (binary rule)
(a_{ij}, b_{ij})	parameters of the Beta distribution
(a_0, b_0)	*priors* of the Beta distribution
c	constant of exploration (UCB)
τ	admission threshold
k	number of collaborators (top-k policy)
θ	threshold of self-confidence needed to ask for help
α	host system aggregation threshold

3.2. Properties of the Trust Matrix

The trust matrix is directed and, in general, asymmetric, reflecting the subjective nature of trust in heterogeneous collaborative environments. Its main properties are as follows.

Property 1 (Asymmetry)**.** *In general, $T_{ij} \neq T_{ji}$ for $i \neq j$. Trust is not reciprocal. An agent with lower discriminative capacity (e.g., Gaussian Naïve Bayes) may place high trust in a stronger agent (e.g., Random Forest), whose suggestions tend to improve its decisions, while the reverse may not hold, since the suggestions of the weaker agent tend to introduce bias into the decisions of the stronger agent. This asymmetry is essential for correctly modeling negative transfer.*

Property 2 (Self-confidence)**.** *For every i, there is $T_{ii} = 1$, a fixed value that is never updated. Each agent has complete confidence in its own predictions; the local prediction always plays a role in the final decision (Section 3.3), which means that collaboration complements rather than replaces individual capability.*

Property 3 (Limitation)**.** *For all i and j, $0 \leq T_{ij} \leq 1$, where $T_{ij} = 0$ indicates a complete lack of confidence and $T_{ij} = 1$ indicates full confidence. In practice, T_{ij} takes on intermediate values that reflect the observed utility.*

3.3. Selective Collaboration Protocol

For each sample, agent i commits to its own prediction when it is confident, $\pi_i(x) > \theta$, and requests assistance only when $\pi_i(x) \leq \theta$. In this second case, the three regimes under

study diverge: in SOLO, the agent always decides alone (non-collaborative baseline); in NAIVE, it consults all peers, without discrimination; in TRUST, it consults only the subset $\mathcal{C}_i$ admitted by trust (Section 3.6).

The responses collected from $\mathcal{C}_i$ are combined by the aggregation function of the host system (the detection architecture on which the trust mechanism is integrated, inherited from previous work [2,19]) and kept unchanged in this study. It is important to note that each agent i produces a prediction with probability $\pi_i(x)$ and only requests assistance when $\pi_i(x) \leq \theta$, where θ is the self-confidence threshold. Once the responses from the admitted peers are received, the final decision results from an average of the predictions weighted by the probability reported by each classifier,

$$\bar{y}_i(x) = \frac{\pi_i(x)\,\hat{y}_i(x) + \sum_{j \in \mathcal{C}_i} \pi_j(x)\,\hat{y}_j(x)}{\pi_i(x) + \sum_{j \in \mathcal{C}_i} \pi_j(x)}, \tag{1}$$

where the final label is obtained by comparing $\bar{y}_i(x)$ with the aggregation threshold α, and it is classified as an attack if $\bar{y}_i(x) > \alpha$ and as benign otherwise. The thresholds θ and α and the weighting rule itself belong to the host system and remain fixed under all experimental conditions, and trust does not play a role in this step. The self-confidence probability $\pi_i(x)$ enters the mechanism only through the host system, namely in the gating condition $\pi_i(x) \leq \theta$ and in the aggregation weights of Equation (1). These probabilities are generated by structurally different families of classifiers, whose calibration may vary, and no explicit probability calibration is applied in this study. This choice is deliberate and does not compromise the analysis, since the selection mechanism and aggregation rule are identical in the SOLO, NAIVE, and TRUST regimes. Thus, any calibration bias they introduce is present in all three and, therefore, cancels itself out in the comparisons between the regimes on which the conclusions of this study are based. Therefore, calibration affects the shared host-system, not the selection mechanism under study.

The purpose of this design is that aggregation treats all responses equally, meaning that a single unreliable peer response influences the decision with the same weight as a reliable peer. Therefore, the only effective defense is to avoid consulting it. Designing the mechanism as an admission control rather than a weighting scheme isolates the contribution of trust from that of the aggregation rules, which, being inherited and fixed, do not affect the causal interpretation of the results. It is worth comparing admission control with reliability-weighted aggregation schemes, which is common in collaborative detection, in which each peer is consulted and its contribution is weighted by a reliability weight. Such schemes operate *a posteriori*: the unreliable contribution is still requested and received, and is attenuated only in the average. This leads to two limitations: first, a peer with a low-but-non-zero weight still contaminates the aggregate, whereas a peer that is not admitted contributes exactly zero. Second, and more importantly for a resource-constrained edge deployment, weighting cannot reduce communication, since assigning a weight to a peer requires consulting it in the first place. The message and latency savings reported in this work are therefore unachievable through weighting alone. The communication advantage of admission control over reliability weighting is therefore established by construction: a peer that is not admitted is neither consulted nor transmitted to. Whether admission also matches or exceeds weighting in detection effectiveness is a separate, empirical question, which is not settled here and is treated as a hypothesis to be examined against a weighting baseline in future work; the present comparison is against the probability-weighted (but not trust-weighted) NAIVE baseline. Admission control is thus not a weaker form of weighting, but a distinct operating mode that trades the fine granularity of continuous weights for the ability to remove a contribution entirely and avoid its communication cost.

3.4. Feedback Signal

The trust is updated based on the outcome of collaborations for which the true label becomes available. Let x be a sample in which i consulted j, and let y be its true label. Only the event of disagreement is treated as informative, $\mathcal{D}_{ij}(x) = \{\hat{y}_i(x) \neq \hat{y}_j(x)\}$: when both agents agree, the consultation does not provide any information relevant to the decision; if both were correct, i would have made the right decision without help; if both were wrong, j offered no means of correction. Incorporating these events would bias the estimate toward the baseline agreement rate, rather than toward the peer's utility. The reward is therefore defined exclusively on the basis of disagreement:

$$r_{ij}(x) = \begin{cases} 1, & \text{if } \hat{y}_j(x) = y \\ 0, & \text{if } \hat{y}_j(x) \neq y \end{cases} \qquad \text{for } x \in \mathcal{D}_{ij}, \tag{2}$$

and otherwise, no update is performed. The counter n_{ij} is incremented only during informative events, so that exploration bonuses and posterior variances are expressed in units of evidence rather than elapsed time. Under this definition, T_{ij} estimates

$$p_{ij} = \Pr\left[\hat{y}_j = y \,\middle|\, \hat{y}_i \neq \hat{y}_j\right], \tag{3}$$

the probability that j is correct when it contradicts i. This quantity is precisely the one relevant to the admission decision; it covers the entire interval $[0, 1]$ and has a natural neutral point at $p_{ij} = 1/2$, at which consulting j is just as likely to correct i as it is to lead it to error. A consequence of defining the reward on verified outcomes (Equation (2)), rather than on the peer's reported probability is that the trust estimate is by construction insensitive to the calibration of the peers. A systematically over-confident peer that nonetheless gives poor advice accumulates $r_{ij} = 0$ on disagreements and sees its trust T_{ij} decay, regardless of how high its reported $\pi_j(x)$ may be. Therefore, trust measures demonstrated usefulness under the ground truth, not self-assessed confidence, which makes the admission decision robust to miscalibration across the heterogeneous classifier families.

3.5. Strategies for Updating Trust

The four trust updating rules are compared below. They all consume the same binary reward as in Equation (2) and are therefore directly comparable; they differ only in how they accumulate evidence.

3.5.1. Exponential Moving Average (EMA)

First-order recursive update with a constant rate $\eta \in (0, 1)$:

$$T_{ij} \leftarrow (1 - \eta)\, T_{ij} + \eta\, r_{ij}. \tag{4}$$

The estimator is $O(1)$ in both time and space and discounts past evidence geometrically, giving it the shortest adaptation time of the four rules under nonstationary peer behavior. Its stationary expectation is p_{ij}, with a residual variance of the order of $\eta/(2 - \eta)$ that does not decrease with more evidence.

3.5.2. Beta–Bernoulli

Considering r_{ij} as a Bernoulli observation with an unknown parameter p_{ij} and adopting a conjugate *a priori* distribution $\mathrm{Beta}(a_0, b_0)$, the posterior distribution following a sequence of interactions is $\mathrm{Beta}(a_{ij}, b_{ij})$ with

$$a_{ij} \leftarrow a_{ij} + r_{ij}, \qquad b_{ij} \leftarrow b_{ij} + (1 - r_{ij}), \tag{5}$$

and the trust is taken as the posterior mean,

$$T_{ij} = \frac{a_{ij}}{a_{ij} + b_{ij}}. \tag{6}$$

In contrast to the EMA, this rule introduces an explicit measure of epistemic uncertainty,

$$\mathrm{Var}[T_{ij}] = \frac{a_{ij}\, b_{ij}}{(a_{ij} + b_{ij})^2\,(a_{ij} + b_{ij} + 1)}, \tag{7}$$

which converges as evidence accumulates and allows for a conservative acceptance decision based on a lower trust threshold. The *a priori* $\mathrm{Beta}(a_0, b_0)$ is set in Section 4.4; its mean is chosen to be equal to the uniform initialization, $a_0/(a_0 + b_0) = 1/(N-1)$, which makes it skeptical about peers that have not yet been tested, requiring evidence of correctness for trust to rise above the initial value. The cost is $O(1)$ per update, but it requires $O(N)$ of additional state per agent, and the uniform weighting of all past evidence makes the estimator slower to react to a peer whose behavior changes.

3.5.3. Upper Confidence Bound (UCB)

The estimation is based on Equation (4), but the admission policy uses an optimistic value that adds an exploration bonus:

$$s_{ij} = \begin{cases} \infty, & n_{ij} = 0 \\ T_{ij} + c\sqrt{\dfrac{\ln n_i}{n_{ij}}}, & n_{ij} > 0, \end{cases} \tag{8}$$

where n_i refers to the total number of informative interactions of i, n_{ij} refers to the number of informative interactions with j, and the constant c governs the exploration–exploitation trade-off. The bonus decreases as evidence accumulates and prevents the premature exclusion of a peer whose initial interactions happened to be unfavorable, i.e., a type of failure to which purely greedy admission is susceptible, since an excluded peer does not generate new evidence and its estimate remains frozen.

3.5.4. Binary

The trust is adjusted by a fixed step size ζ in the direction of the reward, truncated to the unit interval:

$$T_{ij} \leftarrow \min\big(1, \max\big(0,\ T_{ij} + \zeta\,(2r_{ij} - 1)\big)\big) \tag{9}$$

It is the least significant of the four rules and serves as a lower bound. The recursion describes a random walk with reflective boundaries whose *drift* is $\zeta\,(2p_{ij} - 1)$; the trust value is close to 1 when $p_{ij} > 1/2$ and close to 0 otherwise. The rule disregards the magnitude of p_{ij}, which makes it relatively insensitive to the admission threshold τ over most of its domain.

3.6. Admission of Collaborators

When $\pi_i(x) \le \theta$, agent i determines the set $\mathcal{C}_i \subseteq \mathcal{A} \setminus \{i\}$ of peers to be queried. Two policies are considered, namely the threshold policy and the top-k policy.

The threshold policy accepts every peer whose selection value reaches a fixed level,

$$\mathcal{C}_i = \{\, j \neq i : s_{ij} \ge \tau \,\}, \tag{10}$$

where $s_{ij} = T_{ij}$ for all strategies except UCB, for which s_{ij} is given by Equation (8). The dimension of $\mathcal{C}_i$ depends on the data and may be zero. Given the interpretation of Equa-

tion (3), the choice $\tau = 1/2$, adopted in this work, precisely admits the peers with the highest probability of correcting i rather than inducing an error.

The top-k policy instead admits a fixed number of peers, defined by the set $\mathcal{C}_i$ of the up to k highest ranked peers whose selection value is positive, all ranking at least as high as any excluded peer,

$$\mathcal{C}_i \subseteq \{\, j \neq i : s_{ij} > 0 \,\}, \quad |\mathcal{C}_i| = \min\big(k,\ |\{\, j \neq i : s_{ij} > 0 \,\}|\big), \quad \min_{j \in \mathcal{C}_i} s_{ij} \ \geq \max_{\substack{j \notin \mathcal{C}_i,\ j \neq i \\ s_{ij} > 0}} s_{ij}, \tag{11}$$

with ties broken arbitrarily. This bounds the communication cost per decision, at the cost of admitting peers that may fall below the neutral point when there is no better option. When fewer than k peers have a positive score, the admitted set is smaller than k, and when none of them do, it is empty. In this case, the agent falls back to broadcast exactly as under the threshold policy (line 9 of Algorithm 1).

The Algorithm 1 combines the previous elements into an agent's operating cycle. Each agent i starts with uniform trust on the peers and interaction counters set to zero (lines 1–2) and processes the stream of samples one by one. For each sample, it produces its local prediction (line 4) and, if it is sufficiently confident ($\pi_i > \theta$), commits to it without consulting anyone (lines 5–6). Otherwise, it determines the set of collaborators $\mathcal{C}_i$ according to the current admission policy (line 8). When this set is empty, a typical situation during cold start, in which no peer has yet reached the threshold, the agent proceeds to consult all peers (lines 9–10), the operating regime discussed below. Once the responses have been collected, the final decision results from the probability-weighted aggregation with respect to α (lines 12–13), without any consideration of trust at this step.

Algorithm 1 Trust-based selective collaboration for agent i.

1: $T_{ij} \leftarrow 1/(N-1)$ for all $j \neq i$; $T_{ii} \leftarrow 1$
2: $n_{ij} \leftarrow 0$ for all j; $n_i \leftarrow 0$
3: **for all** samples x **do**
4: $(\hat{y}_i, \pi_i) \leftarrow h_i(x)$
5: **if** $\pi_i > \theta$ **then**
6: decide $\hat{y}_i$; **continue** ▷ no query
7: **end if**
8: compute $\mathcal{C}_i$ by Equation (10) or (11)
9: **if** $\mathcal{C}_i = \emptyset$ **then**
10: $\mathcal{C}_i \leftarrow \mathcal{A} \setminus \{i\}$ ▷ exploration regime
11: **end if**
12: send request to $\mathcal{C}_i$; collect responses
13: decide by probability-weighted aggregation against α
14: **for all** $j \in \mathcal{C}_i$ that responded, when y is available **do**
15: **if** $\hat{y}_j = \hat{y}_i$ **then**
16: **continue** ▷ uninformative
17: **end if**
18: $r_{ij} \leftarrow \mathbb{1}[\hat{y}_j = y]$
19: update T_{ij} by Equation (4), (5), or (9)
20: $n_{ij} \leftarrow n_{ij} + 1$; $n_i \leftarrow n_i + 1$
21: **end for**
22: **end for**

The learning occurs in the final phase of the cycle (lines 14–21) when the true label becomes available; the agent updates its trust in each consulted peer, but only in cases of disagreement, which are the only informative ones. The update follows the chosen trust rule and increments the evidence counters. Thus, the algorithm combines the three components of the mechanism, namely, the decision to ask for help (θ), the selection

of whom to ask (the admission policy regarding $\mathcal{C}_i$), and the adaptation of trust (the update rule), into a single local loop, executed independently by each agent without central coordination. The following subsection examines the behavior of this cycle during system startup, when no evidence has yet been accumulated, and the conditions under which it operates. Also, communication costs and computational complexity are analyzed next. The update step presupposes that the true label of each sample eventually becomes available to the agent; operationally, this corresponds to delayed ground truth from analyst confirmation, signature updates, or incident post-mortems. The mechanism does not require this feedback to be immediate: a delayed label merely postpones the corresponding trust update and lengthens the cold-start phase, while each decision is always taken with the trust available at that instant.

3.7. Cold Start and Operating Conditions

In the absence of prior evidence, trust is initialized uniformly,

$$T_{ij}^{(0)} = \frac{1}{N-1}, \qquad j \neq i, \tag{12}$$

distributing the unitary belief mass across the $N-1$ peers, in a manner agnostic to their identity and without favoring any agent from the beginning. This initialization places every peer below the admission threshold whenever $N > 1 + 1/\tau$, which holds in the configurations considered. Under the threshold policy, the admitted set is therefore empty at the start, and the mechanism proceeds to consult all peers (line 9 of Algorithm 1). This is not a degenerate case, but a deliberate operating regime in which an agent that has not yet accumulated evidence cannot discriminate, and diffusion is the only way to generate the disagreement events from which discrimination is learned. The regime ends internally, without any additional parameters, as soon as the first peers cross τ. The same effect arises under UCB, through the infinite value assigned to peers not yet visited in Equation (8). The top-k policy admits only peers with a positive selection value. Therefore, it can also produce an empty admitted set and fall back to broadcast; in practice, this is rare and confined to rules that can drive a peer's trust to zero, such as the binary rule, while in the common case it admits exactly k peers.

3.8. Cost and Complexity of Communication

The transition from broadcast to selective collaboration has direct implications for communication complexity, a critical aspect for scalability in edge–cloud environments with limited resources. In broadcast, when all agents request help in a round, the cost is

$$\mathcal{M}_{\text{broadcast}} = N(N-1) = O(N^2) \text{ messages per round.} \tag{13}$$

In the selection protocol, each agent consults only $\mathcal{C}_i$, and thus

$$\mathcal{M}_{\text{trust}} = \sum_{i=1}^{N} |\mathcal{C}_i| = N\bar{k} = O(N\bar{k}), \tag{14}$$

where $\bar{k} = \frac{1}{N}\sum_i |\mathcal{C}_i|$ is the average size of the set of collaborators. In sufficiently heterogeneous systems, $\bar{k} \ll N$ is expected. The message reduction factor,

$$\rho = 1 - \frac{\bar{k}}{N-1} \in [0,1], \tag{15}$$

is 0 under complete diffusion ($\bar{k} = N-1$) and approaches 1 as the collaboration becomes more selective. It is important to note that Equations (13) and (14) express the worst case

per round, in which every agent is uncertain and requests help simultaneously. In practice only a fraction h of decisions trigger a request, so the effective message volume scales as $h\,N(N-1)$ for broadcast and $h\,N\bar{k}$ for selective collaboration. The reduction factor ρ of Equation (15) is a ratio of the two and is therefore independent of h, so it characterizes the per-request reduction regardless of how often help is requested; the measured end-to-end message counts (Section 5.4) are consistent with it under top-k admission and exceeding it under threshold admission, whose consulted sets are the smallest.

The mechanism does not introduce additional traffic. Each agent estimates trust only from its own interactions, so there is no transmission of trust values or propagation of reputation among agents. Its effect on communication is strictly subtractive. Per informative interaction, the trust update is $O(1)$ for all rules; the admission decision is $O(N)$ under the threshold policy and $O(N \log N)$ under the top-k policy, evaluated only on the samples where assistance is requested. The state maintained per agent is $O(N)$ (with two scalars per peer in the Beta-Bernoulli case).

Once the mechanism has been defined, it remains to implement it in an experimental setting. The following chapter describes the datasets, classifiers, and runtime environment and organizes the three operating modes, namely SOLO, NAIVE, and TRUST, along with the update and admission strategies, into a set of comparable configurations, evaluated on real hardware under two distinct degrees of heterogeneity.

4. Experimental Setup

This chapter implements the mechanism described in Section 3 in an experimental setting. The system overview is described in Section 4.1; the datasets, their partitioning by agent and the heterogeneous classifiers are described in Section 4.2; the runtime environment on real hardware in Section 4.3; and the set of configurations that combines operating modes with trust strategies in Section 4.4.

4.1. System Overview

The trust mechanism described in Section 3 is evaluated on a distributed instance of the detection architecture, the design of which is summarized in Figure 1. The system consists of two types of entities with distinct roles, namely a set of local agents, which perform the actual detection at the edge, and a global agent, which performs an exclusively observational function. This separation of roles is deliberate and allows the analysis to focus on the trust mechanism without interference from central coordination.

Each local agent is an autonomous entity that hosts its own classifier, observes its traffic segment, and makes decisions locally about each sample. When the confidence in its prediction is insufficient, an agent can request assistance from its peers; this is where the trust mechanism comes into action, selecting whom to direct the request to. Collaboration is strictly peer to peer, which means each agent can consult any other, and each agent maintains and updates its own view of the reliability of the others, based exclusively on the interactions it observes. There is, therefore, no shared trust matrix or any propagation of reputation throughout the system, being the selection mechanism entirely local and decentralized.

The global agent is a component of the overall architecture responsible for supervisory functions, specifically the adaptive adjustment of detection thresholds through reinforcement learning. In this study, these functions are deliberately disabled so that the analysis focuses exclusively on the trust-based selection mechanism (Section 4.3). The global agent is then left with a single role, which is the periodic collection of reports from local agents, containing resource telemetry and snapshots of the trust state, which are generated by

the local agent every thousand processed packets. It is important to emphasize that this collection is a tool for experimental analysis and not a system requirement, as the same reports could be logged locally on each device without any change to the detection behavior. The decision does not depend on the global agent; even with its supervisory functions active, detection and collaboration remain local and peer-to-peer, so the global agent does not constitute a coordination point or a point of failure in the decision-making process.

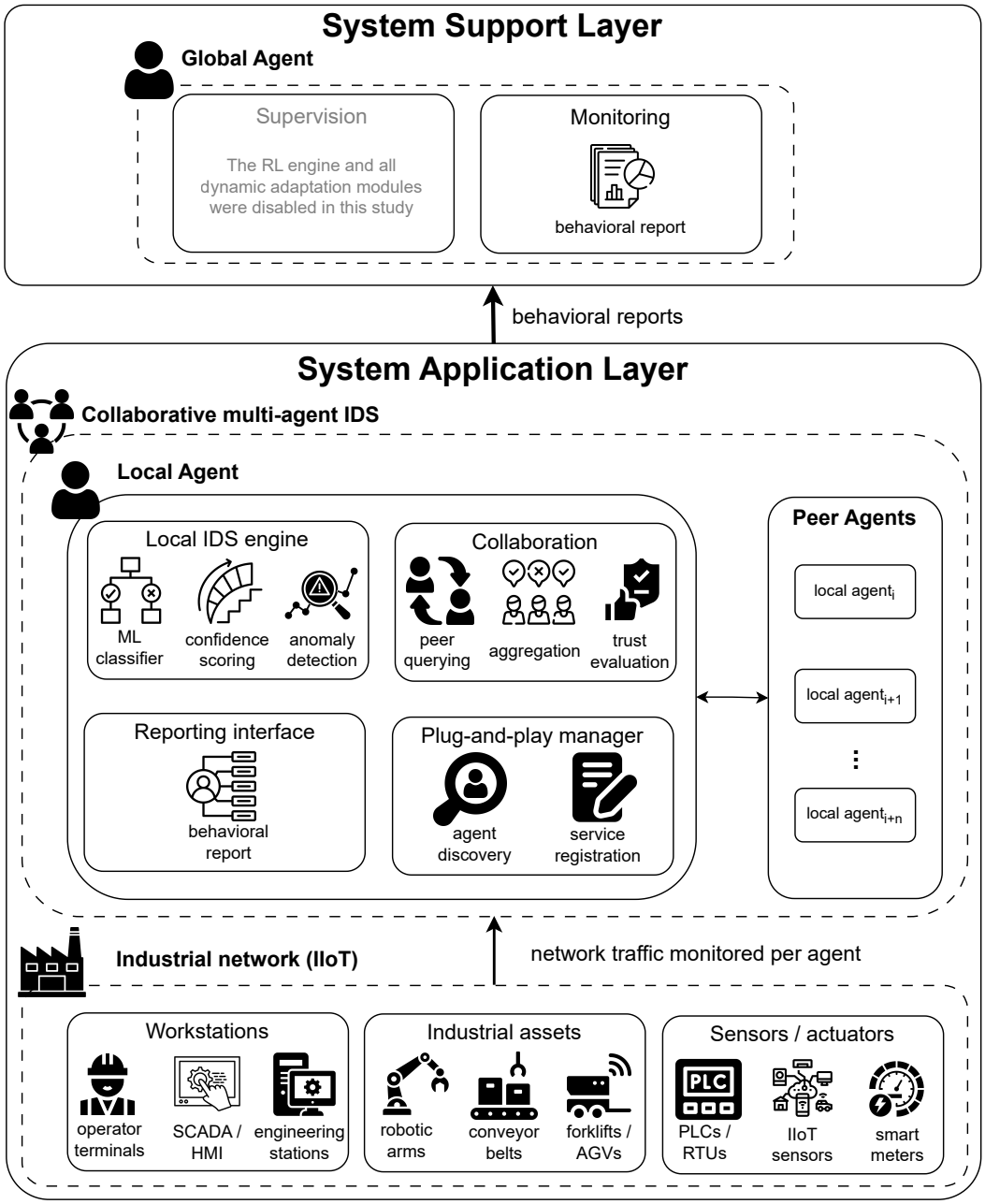


Figure 1. System architecture for studying the trust mechanism among network intrusion detection agents.

This organization replicates the conditions under which a collaborative detection system would operate in a real-world deployment at the edge with autonomous and heterogeneous entities with limited resources, collaborating directly with one another under a trust-based selection policy.

4.2. Datasets and Partitioning

The evaluation relies on two sets of benchmark data for intrusion detection, chosen for representing different degrees of heterogeneity and for being widely used in the literature, which enhances the comparability of the results.

The UNSW-NB15 dataset [38], developed by the Australian Centre for Cyber Security, combines normal traffic with nine attack families, including Fuzzers, Exploits, DoS, Reconnaissance, and Generic, described by a set of flow characteristics. A binary classification is adopted, distinguishing between benign and malicious traffic, and the canonical test partition of 82,332 records is used, against which all agents are evaluated. By bringing together models trained on the same data, UNSW-NB15 provides a controlled analysis scenario, where the only source of heterogeneity is the classification algorithm.

The CICIoT2023 dataset [39], produced by the Canadian Institute for Cybersecurity, is substantially more recent and larger in scale, bringing together tens of millions of records and covering a broad spectrum of attacks on IoT devices, organized into categories such as DDoS, DoS, Recon, Spoofing, and Mirai. Its size and diversity make it well-suited to a heterogeneous scenario in which each agent is trained on a distinct partition of the data, resulting in heterogeneity not only in algorithms but also in knowledge, thereby bringing the test closer to the conditions of a truly distributed system.

The two datasets are processed through a common data preparation pipeline. Categorical features such as protocol, service, and connection status are converted using label encoding, and numerical features are normalized using a scaler tuned on the training data and subsequently applied, without retuning, to the evaluation data. The imbalance between classes is mitigated only on the training data, while the original distribution of the evaluation samples is preserved intact, so that the metrics reflect the system's behavior under representative conditions.

It is in the data partitioning that the two scenarios differ. In UNSW-NB15, all agents share the same training set and are evaluated on exactly the same samples; any divergence among agents stems only from the classifier. In CICIoT2023, the data is divided such that each agent is trained and evaluated on distinct subsets, deliberately inducing partial and non-overlapping competencies. In both cases, the collaborative trials run on a stratified subsample of 20,000 samples per dataset, preserving the class proportions and keeping the experimental cost manageable without compromising representativeness.

The algorithmic heterogeneity is achieved through ten distinct classifiers, being one per agent, namely Logistic Regression (LR), Naïve Bayes (NB), Linear Discriminant Analysis (LDA), K-Nearest Neighbor (KNN), Decision Tree (DT), Multilayer Perceptron (MLP), Support Vector Machine (SVM), AdaBoost (ADA), Random Forest (RF), and Extra Trees (ET). The selection deliberately covers families of different types, such as linear, instance-based, probabilistic, ensemble, and neural network models, ensuring genuinely diverse competence profiles, being a necessary condition for selective collaboration to have an observable effect. Each model is trained offline and deployed on a separate agent, remaining fixed throughout the experiments.

For reproducibility, the classifiers use the scikit-learn implementations with default hyperparameters, except for the following explicitly set values: RF, ADA, and ET use 100 estimators, KNN uses $k = 2$ neighbors, and the iteration limits are set to 1000 for LR, 2000 for the linear SVM, and 300 for the MLP; a fixed random state is used throughout. Numerical features are standardized and categorical features label-encoded, with the transformers fitted on the training split and applied unchanged to the evaluation split. The 20,000 evaluation instances are obtained by proportional stratified sampling with largest-remainder rounding, which preserves the original class proportions exactly and guarantees that no class is dropped (at least one instance per stratum). The two datasets differ in how this

sampling is applied, mirroring their heterogeneity regimes: for UNSW-NB15, stratification uses the attack-category label, and all agents share the same subsample, so that the only source of heterogeneity is the classifier; for CICIoT2023, each agent is assigned a distinct data file, and the stratified sampling is applied within each file independently, with a different seed per file, so that every agent preserves the label distribution of its own segment and the distributions deliberately differ across agents, inducing knowledge heterogeneity. Each configuration is executed with three fixed seeds. Because KNN is configured with $k = 2$ neighbours, its posterior probabilities are quantised in steps of 0.5 and thus provide only a coarse confidence signal; this is why the KNN agent resorts more often to the broadcast fallback, but it affects neither the trust updates, which depend on realised correctness rather than on the reported probability, nor the aggregate results. In particular, because a KNN confidence of 0.5 lies below the self-confidence gate $\theta = 0.8$, this coarse quantisation mainly increases how often the KNN agent requests help rather than how its trust is computed.

4.3. Runtime and Deployment Environment

The test bed implements the architecture described in Section 4.1 in a real-world distributed deployment. Each of the ten local agents runs on a Raspberry Pi 3 B+ (Raspberry Pi Ltd., Cambridge, UK), which represent IoT devices with limited computational resources. The agents are implemented using the JADE framework (version 4.5.0) in Java (including the protocols for requesting help, updating trust, and aggregating received contributions) and expose their classifier through a REST inference service, with this inference layer based on Python (version 3.12.3) using scikit-learn (version 1.4.1.post1). The agents' operational logic is implemented through behaviors defined in JADE, and each agent queries its classifier service via REST, as these are parallel services running internally on the Raspberry Pi.

The collaboration is exclusively peer-to-peer, i.e., when an agent requests assistance, it communicates directly with selected peers via FIPA-ACL messages. The Global Agent, running on a PC, does not participate in this exchange or coordinate the collaboration, it plays a passive role in the system. Its single function during the experiments is to periodically receive reports sent by the local agents, namely the resource telemetry (CPU, temperature, and memory) and snapshots of the trust state and interactions, which serve exclusively for monitoring and for the experimental analysis in the following chapter.

Throughout the entire battery of tests, the reinforcement learning mechanism, which, in the full architecture, would adaptively adjust the detection thresholds, is kept inactive. The self-confidence (θ) and aggregation (α) thresholds therefore remain fixed and uniform across all agents and in all configurations, eliminating any variation unrelated to the selection mechanism under study and supporting a causal interpretation of the reduction in negative transfer.

4.4. Operating Modes and Experimental Tests

The experiment combines operating regimes with trust strategies. The three regimes correspond to those defined in Section 3: in SOLO, each agent decides independently, serving as a reference for non-collaboration; in NAIVE, an uncertain agent consults all peers indiscriminately, isolating the effect of indiscriminate collaboration; in TRUST, an uncertain agent consults only the set admitted by the trust-based selection policy. The NAIVE scheme is the non-selective collaboration baseline: whenever an agent is insufficiently confident in its own prediction, a decision governed by the self-confidence parameter θ, it requests help from all peers and aggregates every response it receives. Both the help-request trigger (θ) and the aggregation rule (the probability-weighted average of Equation (1), which weights each response by the responding classifier's own output probability) are held identical in NAIVE and in every TRUST configuration. The only difference between the two regimes

is therefore the number of peers that take part in each collaboration: NAIVE consults all of them, while TRUST consults only the subset admitted by the selection policy (top-k or threshold). Keeping the trigger and the aggregation fixed across regimes is deliberate, so that the measured differences isolate the effect of selection alone rather than any change in when help is sought or how responses are combined; in particular, trust never enters the aggregation weights, and NAIVE is accordingly not a trust-weighted baseline. Comparing NAIVE with TRUST thus contrasts collaborating with everyone against collaborating with a selected group, and shows that the number of participating agents can be reduced, without losing and often while improving effectiveness, precisely by admitting the most competent peers for each collaboration. An agent is considered uncertain, and therefore eligible to ask for help, when the probability of its prediction does not exceed the self-confidence threshold θ; the final decision results from the probability-weighted average of the classifiers, whose aggregate value is compared with the aggregation threshold α. The contrast between NAIVE and TRUST is what allows us to attribute any observed difference to the selection mechanism, rather than to the aggregation, which remains fixed.

Thus, there are ten different configurations, two of which are reference regimes, SOLO and NAIVE, and the remaining eight are the TRUST regime in all combinations of the four trust update rules (EMA, Beta, UCB, and Binary) with the two admission policies (threshold and top-k). The thresholds θ and α are set uniformly across all agents and remain constant throughout the entire experiment.

In contrast to a static classifier evaluated only once, the system under study is dynamic, in which trust evolves in real time as agents interact, and both the order of interactions and the composition of the sets consulted introduce variability across runs. To characterize this variation, each of the ten configurations is run three times independently, resulting in thirty runs per dataset. The results are reported along with the observed dispersion across runs, which allows us to distinguish consistent differences between strategies from the fluctuations inherent in the system's dynamics.

Table 2 summarizes the parameters common to the entire experimental set. The same values were used in both datasets; only the processed file varied. The EMA rate and the binary rule step share the same implementation constant and therefore appear with the same value; the *a priori* Beta $(1, 8)$ has a mean of $1/9$, which coincides with the uniform initialization $1/(N-1)$ for $N = 10$, starting from the same point as the other strategies but with the initial uncertainty explicitly stated.

The parameters in Table 2 were kept fixed to isolate the effect of the trust mechanism, and their values are based on principles rather than being adjusted. The admission threshold is set to $\tau = 1/2$, which is not an arbitrary choice but the theoretically neutral point of the trust estimate: since T_{ij} estimates $p_{ij} = \Pr[\hat{y}_j = y \mid \hat{y}_i \neq \hat{y}_j]$ (Equation (3)), the value $\tau = 1/2$ admits exactly those peers more likely to correct the agent than to mislead it, and any other value would trade this interpretation for an arbitrary operating point. The self-confidence threshold θ and the aggregation threshold α belong to the host system rather than to the trust mechanism; since they are kept identical in the SOLO, NAIVE, and TRUST regimes, their configurations affect all regimes equally and cancel themselves out in the comparisons between the regimes from which the conclusions were drawn, so the reported effects of selection are not an artefact of these two values. The number of consulted peers k fixes a small, predictable communication budget consistent with the efficiency goal; its effect is not left unexamined, as the per-agent analysis of the weakest classifier (Section 5.2) quantifies the cost of restricting the consulted set, and the threshold-based configurations, whose consulted-set size $\bar{k}$ ranges from 2.36 to 5.08 (according to the results reported later in Section 5), provide empirical evidence of how the number of consulted peers modulates effectiveness and communication cost. A systematic sensitivity

study across θ, α, τ, and k is left for future work. It should be noted that this work is not a sensitivity sweep over these parameters: θ and α are held fixed precisely to remove them as confounders, so that the measured differences are attributable to the admission policy and the update rule alone, and the reported conclusions are stated for the single operating point of Table 2 rather than claimed to hold across the entire parameter space.

Table 2. Parameters of the experimental setup (common to UNSW-NB15 and CICIoT2023).

Parameter	Symbol	Value
Self-confidence threshold	θ	0.8
Aggregation threshold	α	0.6
Admission threshold	τ	0.5
Number of peers (top-k)	k	3
Update rate (EMA)	η	0.1
Update step (Binary)	ζ	0.1
Exploration constant (UCB)	c	0.1
Beta *priors*	(a_0, b_0)	$(1, 8)$
Message interval	-	50 ms
Samples per trial	-	20,000 (stratified)
Number of agents	N	10
Runs per configuration	-	3
Total runs per dataset	-	30

5. Results and Discussion

This chapter presents and discusses the results of the experiment described in Section 4.4, organized around three questions: whether indiscriminate collaboration is beneficial or harmful compared to an individual decision; whether selective trust-based collaboration preserves the benefit while eliminating the negative effect (both addressed in Section 5.2); and which strategies work best, and under what degree of heterogeneity (Section 5.3). Finally, there is a discussion of the cost in terms of actual hardware (Section 5.4), the dynamics of trust (Section 5.5), and a summary of the recommendations (Section 5.6).

5.1. Analysis Protocol

Each configuration was run three times per dataset; the reported values are the average of the three runs, with the standard deviation indicated when relevant. The dispersion among the three replicates, quantified as the standard deviation of the net effect, ranges from about 9 to 324 events on UNSW-NB15 and from 3 to 60 events on CICIoT2023, which corresponds to approximately 0.1% to 3.9% of the mean net effect on UNSW-NB15 and 0.9% to 26% on CICIoT2023, with the highest relative values occurring only in cases where the net effect is small (a few dozen events), since this represents a difference between two larger counts. This variability is consistently smaller than the differences between strategies analysed below, so it does not compromise their comparison: the intrinsic stochasticity of the dynamic system affects the magnitude of the measurements but not the ordering of the strategies. The effect of collaboration breaks down into two complementary quantities: *corrections*, samples in which the isolated prediction was wrong and the collaborative decision corrected it, and *corruptions*, samples in which the isolated prediction was correct and collaboration degraded it. The difference between the two measures the net effect, and the corruptions are the direct signature of negative transfer. The rate of help requests, the average size of the consulted set $\bar{k}$, the message reduction factor ρ (Equation (15)), and the collaboration latency per decision are also reported.

To assess the reliability of the comparisons, the three replicates of each configuration are analysed with pairwise, per-seed differences and 95% confidence intervals (Student's t, two degrees of freedom). Given that only three replicates are available, the formal statistical power of these tests is limited. Therefore, a difference is considered reliable only when its trend is consistent across all three replicates and the corresponding confidence intervals do not overlap, and this per-seed sign-consistency criterion, together with the 95% confidence intervals derived from the three seeds, rather than p-values alone, is used to qualify the classifications below.

5.2. Comparison of Operating Modes

Firstly, the NAIVE regime is compared with the SOLO regime. The *F1-score* per agent and the net effect of collaboration, with agents ranked by individual performance on the UNSW-NB15 dataset, are presented in Table 3 (columns SOLO and NAIVE).

The pattern proved to be clear and asymmetric. In UNSW-NB15, the weaker agents benefit greatly from naive collaboration, in which LDA and SVM gain +2166 and +2307 net classifications, respectively, raising the *F1* score from 0.858 to 0.941 and from 0.880 to 0.979. However, for the stronger agents, the sign reverses, where the random forest suffers a net effect of −225 (290 corruptions versus 65 corrections) and the decision tree of −56. In other words, indiscriminate collaboration redistributes competence from strong agents to weak ones, degrading the strong, which is precisely the manifestation of the negative transfer that the trust mechanism aims to correct.

Table 3. F1-score and net effect per agent on UNSW-NB15, ordered by individual competence (mean of 3 runs).

Classifier	*F1-Score*			Net Effect	
	SOLO	NAIVE	TRUST	NAIVE	TRUST
Naïve Bayes	0.804	0.841	0.839	827	770
LDA	0.858	0.941	0.949	2166	2343
SVM	0.880	0.979	0.983	2307	2392
Logistic Regression	0.884	0.958	0.967	1749	1951
AdaBoost	0.940	0.988	0.989	1084	1124
MLP	0.960	0.978	0.988	424	636
KNN	0.962	0.964	0.982	31	427
Decision Tree	0.987	0.984	0.986	−56	−9
Extra Trees	0.991	0.991	0.995	0	93
Random Forest	0.992	0.982	0.992	−225	0

Illustrative TRUST configuration: EMA rule with top-k admission (not the best-performing configuration; see Table 4).

Having established that indiscriminate (naive) collaboration represents a trade-off, it is important to examine the TRUST column in Table 3, which shows how selective collaboration resolves this issue. In addition, Figure 2 presents a comparison of the net effect per agent in the NAIVE and TRUST strategies, for both datasets, using the EMA rule with top-k admission as an illustrative configuration, which is not the one that performs best, as shown in the full comparison in Table 4, but serves to demonstrate the qualitative effect of selection. The data in the figure makes the correction visible, showing that the negative bars for strong agents in the NAIVE strategy are neutralised or reversed in the TRUST strategy. Supplementing the analysis with the data from the table, it can be seen that the gains of the weak agents remain the same or increase (LDA rises to +2343, and logistic regression from +1749 to +1951), while the strong agents no longer experience degradation. The random forest recovers from −225 to 0, with corruption scores falling from 290 to 93, and the decision tree from −56 to −9. The trust mechanism improves collaboration

not by adding information, but by selectively removing it, by not consulting peers whose historical contributions are harmful, and strong agents are no longer contaminated, while weak agents do not lose the support from which they benefit. It is therefore possible to observe the Properties 1 and 2 manifesting themselves in the data, with self-confidence ensuring that collaboration is additive rather than substitutive, and asymmetry explaining why strong agents are no longer affected by weak peers.

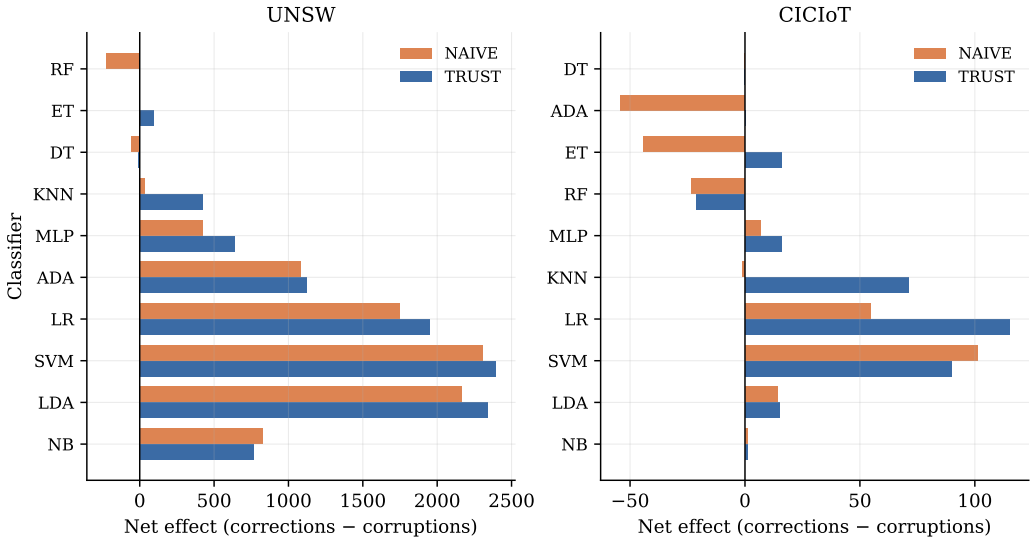


Figure 2. Net effect of collaboration per agent, ranked by individual competence, under NAIVE and TRUST on UNSW-NB15 and CICIoT2023 (illustrative EMA/top-k configuration, not the best-performing setting).

The only exception is the agent using the NB classifier, being the weakest agent, which shows a slight decrease in net effect under the TRUST regime (827 to 770). The reason is explained by the fact that, for this agent, collaboration represents almost exclusively benefit and negligible risk, since its corruption levels remain very low under both regimes (approximately 30) so that the value comes entirely from the corrections. By restricting the set of nine peers under consideration to $k = 3$, admitting the k best peers eliminates some of these beneficial corrections without any compensatory reduction in corruption. The weakest agent is, therefore, the one for whom selectivity is least useful, precisely because there is almost no negative transfer to avoid. In other words, this is the same principle that makes selection valuable for strong agents, viewed from the opposite perspective.

5.3. Strategy Comparison

Having established that trust-based selection resolves the trade-off of naive collaboration, it is also important to compare the trust configurations with one another to determine which strategies work best and under what conditions. Table 4 presents, for each of the ten configurations and for both datasets, the overall *F1-score*, the corrections and corruptions, the net effect with its variance across replicates, and the efficiency metrics, such as the average set of agents consulted $\bar{k}$, the message reduction ρ, and the collaboration latency.

The contrast between the two datasets reveals the decisive role of heterogeneity. In UNSW-NB15, where agents share training data and only the algorithm differs, naive collaboration already produces a high net effect (+8308), and trust adds a moderate improvement in effectiveness, up to +9915 with the binary rule and top-k admission, a 19% gain, while reducing global corruptions from 1288 to values between 420 and 726. In CICIoT2023, the picture is qualitatively different; since each agent is trained on different data, naive collaboration nearly fails, with a net effect of only +54 (373 corrections versus 319

corruptions, which practically cancel each other out). In this case, trust is what makes collaboration viable, multiplying the net effect by a factor of nearly seven, up to +367, while cutting corruptions to about one-third. It is important to note that, in CICIoT2023, the aggregate global *F1* barely changes (from 0.965 to 0.966), since the agents are individually very competent and episodes of collaboration are few compared to the 20,000 samples, meaning that the effect of trust manifests itself in corrections, corruptions, and communication costs, rather than in the aggregate metric. In practical terms, the net effect is summed over the ten agents, i.e., about 200,000 decisions in total (10 × 20,000), so the best net effect on CICIoT2023 (+367) corresponds to only ≈0.18% of all decisions, and the aggregate *F1* stays at 0.965–0.966 across all configurations; the effect of the strategy on this dataset is therefore real but of modest practical magnitude, and matters mainly for communication cost and for the weakest agents.

Table 4. Effectiveness and efficiency of the ten configurations on both datasets (mean over 3 runs). Net effect is reported with its 95% confidence interval (Student's t, $df = 2$).

Dataset	Configuration	*F1*	Corrections	Corruptions	Net Effect [95% CI]	$\bar{k}$	ρ	Collab. (ms)
UNSW-NB15	SOLO	0.926	0	0	0	–	1.00	–
	NAIVE	0.961	9596	1288	8308 [8107, 8509]	9.00	0.00	1590
	EMA/threshold	0.963	9156	427	8729 [8028, 9430]	2.94	0.67	400
	Beta/threshold	0.961	8775	420	8355 [7550, 9161]	2.36	0.74	374
	UCB/threshold	0.967	10,240	610	9631 [9063, 10,198]	3.57	0.60	524
	Binary/threshold	0.964	9658	606	9053 [9030, 9076]	3.20	0.64	400
	EMA/top-*k*	0.967	10,359	631	9728 [9578, 9879]	3.00	0.67	737
	Beta/top-*k*	0.965	10,048	719	9329 [9066, 9592]	3.00	0.67	816
	UCB/top-*k*	0.967	10,392	726	9666 [9521, 9811]	3.00	0.67	743
	Binary/top-*k*	0.968	10,509	593	9915 [9524, 10,307]	3.15	0.65	664
CICIoT2023	SOLO	0.965	0	0	0	–	1.00	–
	NAIVE	0.965	373	319	54 [32, 77]	8.10	0.10	1614
	EMA/threshold	0.965	257	91	166 [138, 193]	4.77	0.47	785
	Beta/threshold	0.965	274	143	131 [84, 178]	5.08	0.44	899
	UCB/threshold	0.966	380	82	298 [291, 305]	4.23	0.53	757
	Binary/threshold	0.965	339	110	229 [80, 378]	4.17	0.54	727
	EMA/top-*k*	0.966	480	177	303 [288, 319]	2.70	0.70	598
	Beta/top-*k*	0.965	470	216	255 [246, 263]	2.70	0.70	605
	UCB/top-*k*	0.966	505	157	348 [339, 357]	2.70	0.70	700
	Binary/top-*k*	0.966	503	137	367 [357, 377]	2.57	0.71	614

The Table 4 enables the eight combinations to be ranked. The admission policy is the most influential factor. Top-*k* admission is consistently more effective, producing the largest net gains on both datasets, because it queries a fixed number of peers and collects more corrections. The magnitude of this advantage, however, must be read against the dispersion across replicates. On UNSW-NB15, the strongest configurations (Binary, EMA, and UCB with top-*k*) form a cluster whose confidence intervals overlap and cannot be strictly ordered: Binary/top-*k* attains the highest mean net effect (+9915), but it is not statistically separable from EMA/top-*k* (+9728) or UCB/top-*k* (+9666). On CICIoT2023, by contrast, the ordering is clean, and Binary/top-*k* (+367) separates from the UCB/top-*k* (+348) with intervals that meet only at their boundary (both touching at 357) and a consistent sign across all three replicates. At the policy level, averaged over the four update rules, top-*k* admission attains the higher mean net effect on both datasets; this advantage is not uniform at the configuration level, however, since on CICIoT2023 the UCB/threshold configuration (+298) significantly outperforms Beta/top-*k* (+255) and on UNSW-NB15 UCB/threshold (+9631) is not separable from the top-*k* cluster. The robust statement is therefore that top-*k* tends to maximise effectiveness while the threshold maximises

communication savings, not that every top-k configuration dominates every threshold one, and the ranking among individual top-k rules is itself only well-resolved under data heterogeneity. Taken across both datasets, the net advantage of the best trust-based configuration over the probability-weighted baseline is sign-consistent across the three seeds in seven of the eight configurations on UNSW-NB15 (the exception being Beta-Bernoulli under threshold admission) and in all eight configurations on CICIoT2023; on UNSW-NB15 in particular, the EMA/threshold and Beta/threshold configurations are not statistically separable from the baseline and are therefore reported as inconclusive rather than favourable. Threshold admission is more resource-efficient, as it queries fewer peers ($\bar{k}$ as low as 2.36 on UNSW-NB15) and thus achieves the greatest communication savings (ρ up to 0.74), at the cost of a slightly lower net effect. The trade-off is therefore clear: top-k when prioritizing effectiveness, threshold when the communication budget is limited. The Figure 3 illustrates this trade-off by plotting each configuration on the effectiveness-communication plane and showing the distinct regions occupied by the two admission policies.

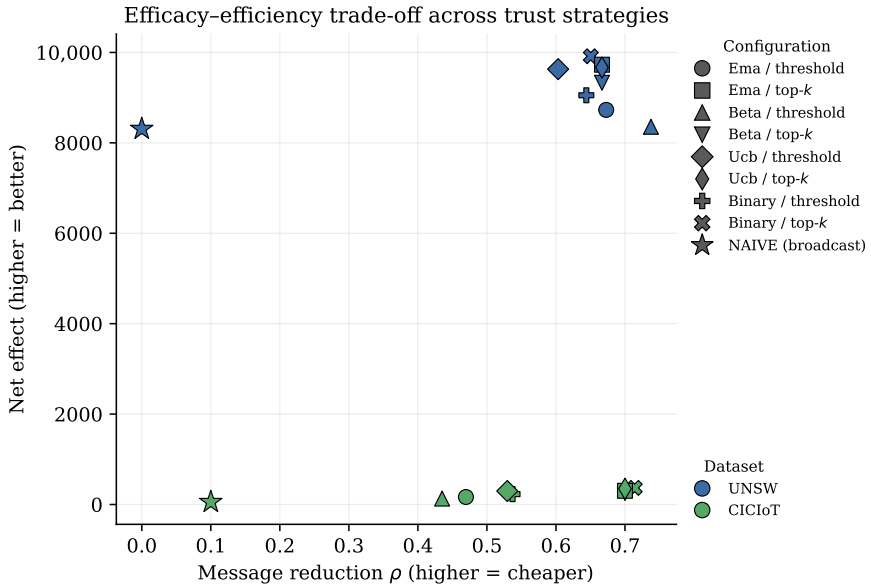


Figure 3. Efficiency-effectiveness trade-off between trust strategies. The NAIVE regime broadcasts to all peers and therefore appears near the origin, with a message reduction of $\rho \approx 0$ (exactly 0 on UNSW-NB15, and 0.10 on CICIoT2023 as not all peers always respond).

The two design axes act differently: the admission policy fixes the direction of the effect in a stable way across update rules, whereas the choice of update rule, although it does not change this direction, modulates its magnitude by a non-negligible amount. Within the threshold admission, the spread of the net effect across the four update rules reaches 1275 events on UNSW-NB15 and 167 on CICIoT2023, exceeding the gap between the two admission policies (718 and 112, respectively), while within top-k admission, the spread is smaller (586 and 112), comparable to the policy gap. Within this spread, the Beta-Bernoulli rule yields the lowest net effect, a consequence of its sceptical prior $(a_0, b_0) = (1, 8)$ that delays the accumulation of trust. However, certain trends can be distinguished; for example, the UCB rule with threshold admission produces the lowest corruption in CICIoT2023 (82), behaving as the most conservative, whereas the binary rule, despite its simplicity, achieves the highest net effects when combined with top-k. The relevance of this choice is itself governed by the degree of heterogeneity: in the controlled regime, the strategy alters the result only by a small margin, whereas in the genuinely heterogeneous regime, it

determines whether collaboration is advantageous at all. The difference between the nearly zero effect of NAIVE and the +367 of the best TRUST configuration represents the entire difference between collaborating ineffectively and collaborating effectively.

The update rule also determines the degree of severity with which the weakest agents are isolated. The Beta rule, with its skeptical a priori distribution $(a_0, b_0) = (1, 8)$, is the most conservative. For the weakest agent (using the NB classifier), this causes the mean of the consulted set to fall below $\bar{k} \approx 1.1$ under threshold admission, effectively excluding it from collaboration and reducing its net effect from +827 (NAIVE) to +770. Since this agent disagrees with almost all peers, its trust values are rarely above the admission threshold under a skeptical prior distribution. Thus, the same mechanism that protects strong agents from corruption may deprive the weakest agent of the extensive collaboration from which it benefits. More permissive rules (EMA and Binary under threshold admission) keep this agent adequately connected and preserve its gains, illustrating that the choice of update rule is most important at the extremes of the competence range.

5.4. Computational and Communication Costs

In addition to effectiveness, practical deployment on edge devices requires that the trust mechanism operate within their limited resources. Table 5 reports, for the SOLO, NAIVE, and TRUST regimes on both datasets, the communication load (help rate, average set of agents consulted $\bar{k}$, and message reduction ρ), the collaboration latency, and the device-level resource usage (CPU, temperature, and memory).

Table 5. Communication cost, latency, and resource usage (TRUST regime with EMA rule and top-k admission; mean of 3 runs).

Dataset	Regime	Help Rate	$\bar{k}$	ρ	Collab. ms	CPU %	Temp. °C	Mem. %
UNSW-NB15	SOLO	-	-	1.00	-	17.6	65.9	38.7
	NAIVE	0.251	9.00	0.00	1590	13.5	66.7	37.8
	TRUST	0.254	3.00	0.67	737	12.7	65.9	38.9
CICIoT2023	SOLO	-	-	1.00	-	21.3	67.1	37.8
	NAIVE	0.136	8.10	0.10	1614	15.9	67.6	36.1
	TRUST	0.137	2.70	0.70	598	15.9	66.5	35.5

The selective collaboration simultaneously reduces both the message volume and latency without compromising the device's resources. Compared to the NAIVE scheme, which queries all nine peers in each request, the TRUST scheme reduces the number of queried peers to less than three on average, corresponding to a reduction in the number of messages ρ between 0.60 and 0.74 on UNSW-NB15 and between 0.44 and 0.71 on CICIoT2023. Collaboration latency per decision decreases by the same proportion, from approximately 1590–1614 ms under NAIVE to 598–737 ms under TRUST in the representative configuration, since the agent waits for fewer responses. In absolute terms, summed over all agents and the full evaluation, the probability-weighted baseline exchanges about 452,000 collaboration messages on UNSW-NB15 and 245,000 on CICIoT2023, while the trust-based configurations exchange between 72,000 and 155,000 on UNSW-NB15 and between 51,000 and 138,000 on CICIoT2023. These counts are consistent with the per-request factor ρ of Table 4: the help-request rate is governed by θ and is essentially the same in NAIVE and in every TRUST configuration (Table 5), so the realized message volume scales with the consulted-set size. For the top-k configurations, the realized reduction (about 66%) coincides with $\rho = 0.67$; for the threshold configurations, it is larger (up to about 84%),

since these admit the fewest peers ($\bar{k}$ as low as 2.36). The per-seed message counts for every configuration are reported in Table A1 in the Appendix A.

This gain comes at no cost in terms of resource consumption. Throughout the entire battery test, CPU utilization remained between 12 and 17% on the UNSW-NB15 and between 16 and 21% on the CICIoT2023, with temperatures ranging from 64.5 and 68 °C, and memory utilization near 36–39%, with no increase attributable to the trust mechanism. On the contrary, the SOLO regime, without collaboration, exhibits the highest CPU load. The inference latency per sample remained equally stable (about 90 ms on UNSW-NB15 and 145 ms on CICIoT2023), regardless of the regime. These values empirically confirm the analysis in Section 3.8; that is, by estimating trust purely locally, in constant time per interaction, and without reputation propagation, the mechanism has a strictly subtractive effect on communication and operates within the resource constraints of edge devices. One detail of the efficiency measurements deserves comment: under top-k admission the average consulted set size $\bar{k}$ occasionally exceeds the nominal $k = 3$ (for instance $\bar{k} = 3.15$ for Binary/top-k on UNSW-NB15). This does not violate the budget but follows from two mechanisms. The top-k rule admits only peers whose selection score is strictly positive, so when fewer than k peers qualify, the consulted set is smaller than k, and the reported average is computed only over requests actually issued, excluding non-responses such as timeouts or empty peer sets. Because a trust update requires an observed outcome, a non-responding peer yields no evidence and its trust value is left unchanged. Conversely, when no peer qualifies, the agent falls back to a broadcast, as specified in Algorithm 1 (lines 9–10); these occasional broadcasts, concentrated in the cold-start phase of rules such as KNN and Binary, pull the average slightly above k. The effect is small and transient and does not alter the reported communication savings.

5.5. Dynamics of Trust

In addition to the aggregate results, it is important to verify that the trust signal behaves as designed. Figure 4 illustrates the evolution of a representative agent's trust values over time, and Figure 5 presents the evolution of the trust matrix as a heat map, both corresponding to an illustrative TRUST configuration (Beta rule with threshold admission, on UNSW-NB15).

The convergence confirms the expected behavior: starting from the uniform initialization $1/(N-1) \approx 0.11$, the trust diverges progressively with the evidence. For the Random Forest, trust in the Extra Trees grows until it exceeds the admission threshold $\tau = 0.5$, while trust in weak peers remains below 0.15. The initial interval during which no peer exceeds τ corresponds to the cold-start exploration phase discussed in Section 3.7, which ends internally as the first reliable peers appear.

The heat map in Figure 5 traces the evolution of the trust matrix at four moments: the uniform initialization and the states after approximately 1000, 5000, and all processed samples. At initialization, every off-diagonal entry sits at $1/(N-1) \approx 0.11$, with no structure yet visible. As evidence accumulates, the matrix differentiates, the columns of the strong agents progressively standing out as they become widely trusted, while the rows remain sparse. The final matrix consolidates a notably asymmetric structure, empirically confirming Property 1. The trust that an agent receives is positively correlated with its individual competence ($r = +0.71$ with *F1* alone), while the trust it grants is negatively correlated ($r = -0.66$); that is, strong agents are widely searched for but perform few searches on their own. In the extreme case, the Random Forest is trusted by peers at an average value of 0.87 , but grants trust only at 0.18, while the Naïve Bayes model is trusted at 0.20 and grants trust at 0.46. It is this asymmetry that reinforces the reduction in messages observed in the previous section, i.e., strong agents, by trusting almost no one, rarely collaborate, while weak

agents direct their requests precisely to the most capable peers. Table 6 summarizes the final trust matrices of all TRUST configurations through three indicators, the matrix sparsity, mean trust, and the correlation of received and granted trust with individual F1. Sparsity is the fraction of off-diagonal entries below the admission threshold $\tau = 0.5$ (higher means more selective); Mean trust is the average off-diagonal trust; $r(\text{F1}, \text{in})$ and $r(\text{F1}, \text{out})$ are the Pearson correlations between individual F1 and the trust each agent receives and grants, respectively. Under algorithmic heterogeneity (UNSW-NB15), the asymmetry is consistent across all configurations: received trust correlates strongly and positively with competence (r from $+0.65$ to $+0.80$) and granted trust negatively (r from -0.50 to -0.83), confirming that competent agents are widely trusted but grant trust moderately. Under knowledge heterogeneity (CICIoT2023), where competence is local rather than global, this outgoing asymmetry weakens and reverses in sign ($r(\text{F1}, \text{out})$ becomes mildly positive), consistent with agents relying on peers that cover regions of the input space they do not, rather than on a global competence ranking. Two cautions apply to these correlations. First, with only ten agents, they are estimated on small samples: on UNSW-NB15, the positive association of received trust with competence ($r = +0.71$) and the negative association of granted trust ($r = -0.66$) are statistically significant ($p = 0.02$ and $p = 0.04$, respectively), while on CICIoT2023, the corresponding coefficients do not reach significance ($p \geq 0.25$ for $n = 10$, none significant at the 0.05 level), so the asymmetry is securely established only under algorithmic heterogeneity. Second, because the two datasets induce different notions of competence, the individual *F1* values are not comparable across datasets, and these correlations should be read within each regime rather than between them. The complete matrices for all configurations are provided in the Appendix A (Figures A1 and A2).

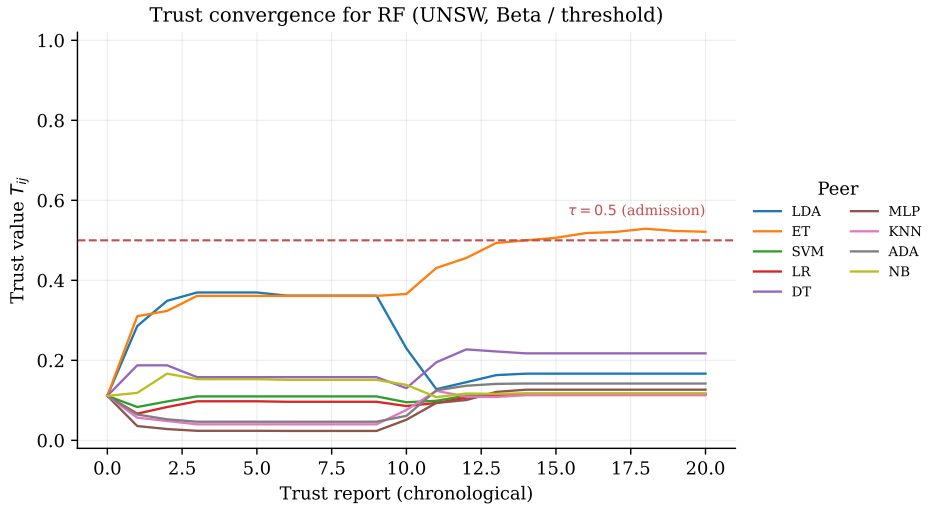


Figure 4. Convergence of the trust values of a representative agent (with Random Forest classifier) over time.

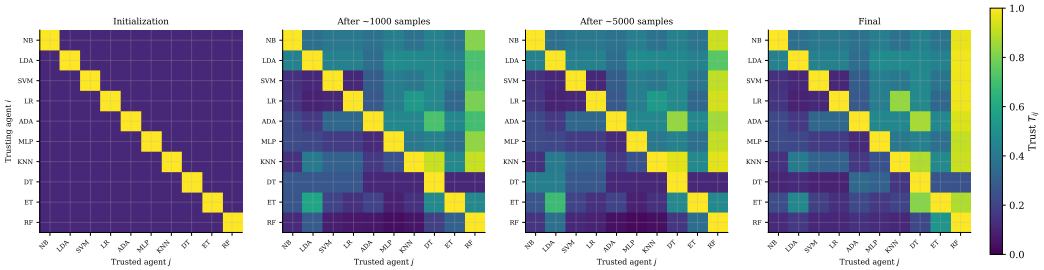


Figure 5. Evolution of trust matrix illustrated using a heat map for the Beta/threshold configuration for the UNSW-NB15 dataset.

Table 6. Quantitative summary of the final trust matrices per TRUST configuration.

Dataset	Configuration	Sparsity	Mean Trust	$r(\text{F1}, \text{in})$	$r(\text{F1}, \text{out})$
UNSW-NB15	EMA/threshold	0.88	0.34	+0.76	−0.68
	EMA/top-k	0.73	0.30	+0.75	−0.50
	Beta/threshold	0.86	0.37	+0.71	−0.66
	Beta/top-k	0.76	0.29	+0.65	−0.60
	UCB/threshold	0.81	0.34	+0.80	−0.77
	UCB/top-k	0.73	0.32	+0.70	−0.65
	Binary/threshold	0.84	0.29	+0.76	−0.83
	Binary/top-k	0.74	0.26	+0.73	−0.75
CICIoT2023	EMA/threshold	0.83	0.35	+0.36	+0.34
	EMA/top-k	0.82	0.25	+0.28	+0.37
	Beta/threshold	0.84	0.33	+0.36	+0.35
	Beta/top-k	0.84	0.23	+0.30	+0.40
	UCB/threshold	0.76	0.38	+0.38	+0.32
	UCB/top-k	0.79	0.29	+0.27	+0.40
	Binary/threshold	0.82	0.32	+0.27	+0.20
	Binary/top-k	0.79	0.25	+0.30	+0.24

5.6. Summary

The results support three conclusions. First, indiscriminate collaboration is a trade-off rather than a guaranteed gain, as it benefits weak agents at the expense of strong ones through negative transfer, and its net effect is nearly zero when heterogeneity is high. Second, the trust mechanism resolves this trade-off by reframing collaboration as an admission problem, preserving the corrections and eliminating the corruptions without altering the aggregation rule, with gains ranging from moderate (controlled regime) to the deciding factor (heterogeneous regime). Third, the choice of strategy follows a clear trade-off, in which top-k admission maximizes effectiveness and threshold admission maximizes communication savings; while the admission policy sets the direction of the effect, the updating rule still modulates its magnitude by a non-negligible amount. In practical terms, top-k admission is recommended when the goal is net effect, threshold admission when the communication budget is limited, and, across the board, the use of trust whenever agents are trained on heterogeneous data, i.e., where blind collaboration ceases to be worthwhile and selection by trust becomes essential. The behaviour of the weakest agent suggests a further practical refinement, since a small fixed k can deprive a weak, high-disagreement agent of beneficial collaboration; the number of admitted peers could be made adaptive and larger for agents whose individual competence is low so that selectivity is applied where there is negative transfer to avoid and relaxed where collaboration has almost pure benefit.

6. Conclusions

This work addressed an inherent tension in collaborative intrusion detection in heterogeneous environments, in which collaboration among distributed agents can compensate for each agent's partial competence, but when done indiscriminately, it exposes the most capable agents to negative transfer, degrading decisions that would otherwise be correct on their own. The problem was reformulated as a question of admission, not whether to collaborate, but with whom, and a mechanism was proposed in which each agent maintains, in a strictly local manner, a trust matrix that directs its requests for help only to peers whose historical contributions have proven beneficial.

The evaluation was conducted on real hardware with ten heterogeneous agents and on two benchmark datasets, and supports three main conclusions. First, it confirmed that naive collaboration is a trade-off rather than a guaranteed benefit, as it benefits weak agents at the expense of strong ones, and under conditions of high heterogeneity, its net effect

approaches zero, with the improvements being almost entirely offset by corruption. Second, selective collaboration based on trust resolves this trade-off without altering the aggregation rule, preserving the corrections from weak agents and suppressing the corruptions from strong agents. The gain is moderate when heterogeneity is purely algorithmic, but becomes the factor that makes collaboration worthwhile when agents are trained on distinct datasets, where it multiplies the net effect by a factor of nearly seven. Third, under algorithmic heterogeneity (UNSW-NB15), the trust structure that emerges is markedly asymmetric, with competent agents being widely requested but requesting few others, while under knowledge heterogeneity (CICIoT2023), this outgoing asymmetry does not hold; in both regimes, selectivity reduces the message volume by up to 84% (more than 60% under top-k admission) and collaboration latency by approximately 44% to 77%, without any measurable penalty in terms of CPU, temperature, or memory on edge devices.

A systematic comparison of the strategies provides practical guidance, with the admission policy setting the direction of the effect and the trust update rule modulating its magnitude, as admitting the k best agents maximizes effectiveness, while admission by threshold maximizes communication efficiency, offering the designer an explicit trade-off between the two selection strategies. In general, the use of trust proves to be unnecessary when agents share knowledge and most valuable when they do not share knowledge, as the degree and type of heterogeneity determine the value of selection. A practical refinement suggested by the per-agent analysis is to make the number of admitted peers k adaptive to each agent's competence, where it is larger for the weakest agents, whereas a small fixed k can otherwise deprive of beneficial collaboration; this is left for future work.

The study has limitations that restrict the scope of these conclusions. The analysis deliberately focused on the selection mechanism, keeping the aggregation rule and the other adaptive mechanisms of the architecture fixed. Therefore, the interaction between trust and aggregation, or between trust and threshold adjustment via reinforcement learning, still needs to be characterized, which has the potential to yield even better results. The evaluation covered ten agents and two datasets in a binary setting; thus, scalability to dozens of agents and to multi-class detection, although suggested by the complexity analysis, was not verified experimentally. Finally, the trust parameters were set uniformly, without optimization per agent or per context. Accordingly, all conclusions are drawn for the single operating point of Table 2 and are not claimed to hold across the full parameter space.

A further limitation concerns the threat model, which also delimits the scope of the guarantees. Because trust is learned from verified outcomes rather than from a peer's self-reported confidence, the mechanism degrades gracefully against a peer whose advice becomes consistently poor: such a peer accumulates negative evidence and its trust decays; because trust is re-estimated after every interaction, the peer is progressively down-ranked under both admission policies. Under threshold admission, this enforces an absolute floor; once its trust falls below τ, it leaves the admitted set entirely. Under top-k admission, the criterion is relative, as it drops out of the admitted set as soon as k better-scoring peers exist (with nine candidate peers and $k = 3$, the typical situation), and, because the top-k rule admits only peers with a strictly positive selection score, a peer whose score becomes non-positive is excluded outright regardless of k. In both regimes, a compromised or faulty peer that returns confident but incorrect predictions is progressively demoted to an incompetent advisor; the only residual case is an adversarial majority, where fewer than k trustworthy peers remain and top-k, unlike threshold, would still fill its k slots. The residual vulnerability is the strategically adaptive peer that behaves well until it has earned trust and only then defects, which an evidence-averaging estimator would detect with a delay proportional to its adaptation rate. Non-ideal feedback is expected to affect the speed rather than the direction of adaptation: delayed labels would lengthen the cold-start

exploration phase, while label noise would bias the estimate of p_{ij} in proportion to the noise rate, an effect that the evidence-accumulating rules (Beta, EMA) should partially smooth. These statements are expectations grounded in the structure of the update rules rather than results measured in the present experiments. A rigorous treatment of adversarial peers and imperfect supervision, including robustness experiments under corrupted feedback, is an important direction for future work.

These limitations point to future work, where integrating trust with the adaptive adjustment of the thresholds θ and α through reinforcement learning (a feature already present in the architecture but left inactive in this study) would allow the selectivity and sensitivity of detection to co-evolve. The trust structure, being a weighted directed graph, also suggests the formation of collaborative clusters that could improve scalability in larger deployments. Finally, extending the approach to non-stationary scenarios, in which agent competence varies over time, would exploit the adaptability that rules based on moving averages and optimistic exploration were designed to provide. A final direction is to compare admission control against trust-weighted aggregation, in which the trust value T_{ij} is used as a weight inside Equation (1) rather than as an admission gate; the present design deliberately keeps the two separate to isolate the causal effect of selection, but a systematic comparison of the two uses of trust would clarify their relative merits under different communication budgets.

Author Contributions: Conceptualization, G.F.; methodology, G.F.; software, G.F.; validation, G.F.; formal analysis, G.F.; investigation, G.F.; writing—original draft preparation, G.F.; writing—review and editing, G.F., T.P., F.D.l.P. and P.L.; supervision, T.P., F.D.l.P. and P.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by FCT—Fundação para a Ciência e Tecnologia, I.P. by projects: CeDRI, UID/05757/2025 (DOI: 10.54499/UID/05757/2025) and UID/PRR/05757/2025 (DOI: 10.54499/UID/PRR/05757/2025); SusTEC, LA/P/0007/2020 (DOI: 10.54499/LA/P/0007/2020). This work has also been supported and received funding from "CyberPRAISE - Cybersecurity research for PRivAte, Intelligent and truStablE solutions"—NORTE2030-FEDER-01820300. The author Gustavo Funchal thanks FCT Portugal for the PhD Grant 2022.13712.BD.

Data Availability Statement: The implementation and configuration files used in this study will be made available as part of the corresponding doctoral thesis; they are not released with this article because the thesis is currently under submission. The datasets used in this study are publicly available at the following URL (https://www.unb.ca/cic/datasets/iotdataset-2023.html, accessed on 1 February 2026 and https://research.unsw.edu.au/projects/unsw-nb15-dataset, accessed on 1 February 2026).

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Table A1. Per-seed net effect and absolute collaboration-message counts for every configuration on both datasets. The "Mean" column repeats the net effect reported in Table 4 with its standard deviation over the three seeds; messages are the total number of help-request messages exchanged, summed over all agents and the full evaluation.

Dataset	Configuration	Seed 1	Seed 2	Seed 3	Mean ± S.D.	Messages ($\times 10^3$)
UNSW-NB15	SOLO	0	0	0	0 ± 0	0
	NAIVE	8216	8368	8340	8308 ± 81	452.4
	EMA/threshold	8431	8992	8765	8729 ± 282	74.2
	Beta/threshold	8419	8004	8643	8355 ± 324	72.4
	UCB/threshold	9894	9485	9513	9631 ± 228	133.3
	Binary/threshold	9059	9042	9057	9053 ± 9	105.0
	EMA/top-k	9660	9776	9749	9728 ± 61	152.2
	Beta/top-k	9227	9438	9322	9329 ± 106	151.0
	UCB/top-k	9599	9701	9699	9666 ± 58	152.6
	Binary/top-k	9793	9860	10,093	9915 ± 157	155.4

Table A1. *Cont.*

Dataset	Configuration	Seed 1	Seed 2	Seed 3	Mean ± S.D.	Messages ($\times 10^3$)
CICIoT2023	SOLO	0	0	0	0 ± 0	0
	NAIVE	45	55	63	54 ± 9	245.4
	EMA/threshold	153	174	170	166 ± 11	83.9
	Beta/threshold	142	142	109	131 ± 19	137.5
	UCB/threshold	297	301	296	298 ± 3	61.5
	Binary/threshold	233	167	287	229 ± 60	109.3
	EMA/top-k	296	307	307	303 ± 6	82.1
	Beta/top-k	251	255	258	255 ± 4	82.2
	UCB/top-k	344	350	350	348 ± 3	82.0
	Binary/top-k	369	362	369	367 ± 4	50.7

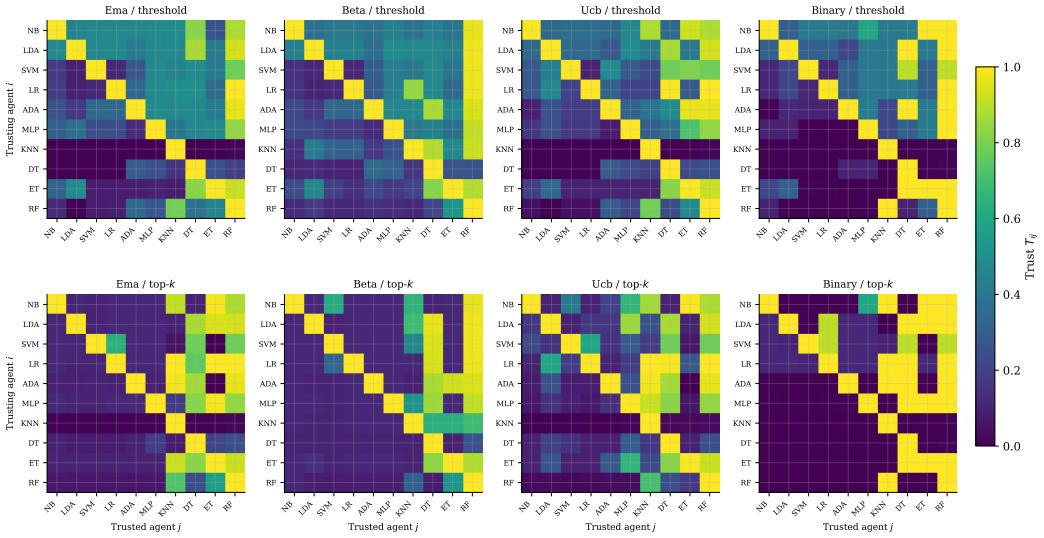


Figure A1. Final trust matrices for all TRUST configurations in UNSW-NB15.

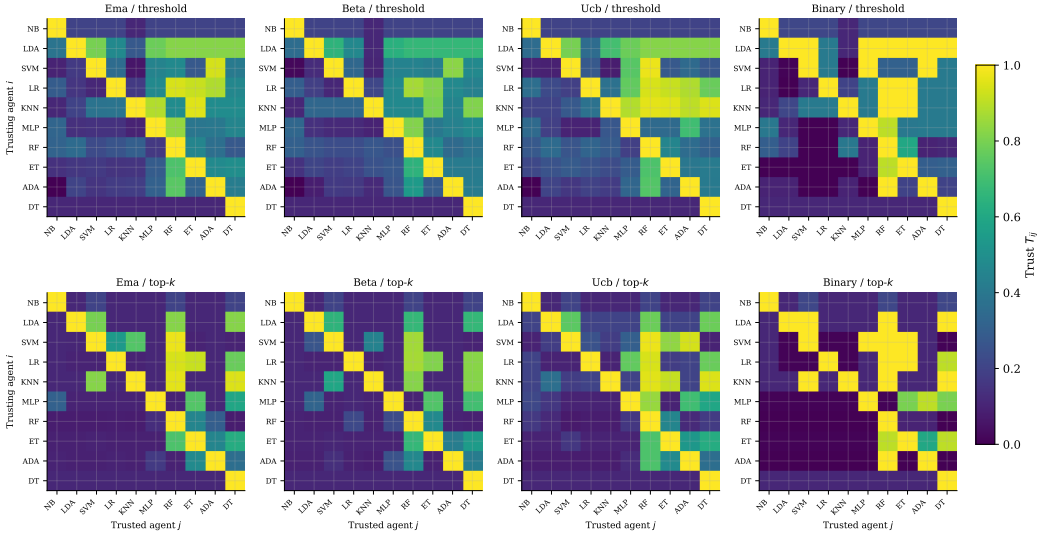


Figure A2. Final trust matrices for all TRUST configurations in CICIoT2023.

References

1. Sanjalawe, Y.; Fraihat, S.; Al-E'mari, S.; Makhadmeh, S.N. A review of artificial intelligence-based intrusion detection in industrial internet of things. *Discov. Internet Things* **2026**, *6*, 44. [CrossRef]
2. Funchal, G.; Pedrosa, T.; Prieta, F.D.L.; Leitao, P. Edge Multi-agent Intrusion Detection System Architecture for IoT Devices with Cloud Continuum. In Proceedings of the 2024 IEEE 7th International Conference on Industrial Cyber-Physical Systems (ICPS), St. Louis, MO, USA, 12–15 May 2024. [CrossRef]
3. Funchal, G.; Pedrosa, T.; De la Prieta, F.; Leitão, P. A Cloud-Driven Support Layer for Enhancing Distributed IDS in IoT Networks. *SN Comput. Sci.* **2026**, *7*, 186. [CrossRef]

4. Jin, D.; Kannengießer, N.; Rank, S.; Sunyaev, A. Collaborative Distributed Machine Learning. *ACM Comput. Surv.* **2024**, *57*, 1–36. [CrossRef]
5. Zhang, W.; Deng, L.; Zhang, L.; Wu, D. A Survey on Negative Transfer. *IEEE/CAA J. Autom. Sin.* **2023**, *10*, 305–329. [CrossRef]
6. Qi, M.; Zhu, T.; Zhang, L.; Li, N.; Tan, Y.a.; Zhou, W. Toward Transparent and Incentive-Compatible Collaboration in Decentralized LLM Multi-Agent Systems: A Blockchain-Driven Approach. *IEEE Trans. Netw. Sci. Eng.* **2026**, *13*, 6706–6720. [CrossRef]
7. Liu, S.; Chalkiadakis, G.; Polyzos, G.C. MAMA-V: A Verifiable Distributed Multi-Agent Model AI Framework for Secure Orchestration. In Proceedings of the IEEE INFOCOM 2026—IEEE Conference on Computer Communications, Tokyo, Japan, 18–21 May 2026; pp. 1–6. [CrossRef]
8. Hao, J.; Bai, H.; Dong, X.; Dong, Z.; Ji, W.; Xie, D.; Guo, S. Multi-agent Dynamic Trust Computational Deduction and Adaptive Collaborative Decision-Making for Grid Edge Intelligence. In *Big Data and Security*; Springer: Singapore, 2026; pp. 220–231. [CrossRef]
9. Granatyr, J.; Botelho, V.; Lessing, O.R.; Scalabrin, E.E.; Barthès, J.P.; Enembreck, F. Trust and Reputation Models for Multiagent Systems. *ACM Comput. Surv.* **2015**, *48*, 1–42. [CrossRef] [PubMed]
10. Vasilomanolakis, E.; Karuppayah, S.; Mühlhäuser, M.; Fischer, M. Taxonomy and Survey of Collaborative Intrusion Detection. *ACM Comput. Surv.* **2015**, *47*, 1–33. [CrossRef]
11. Li, W.; Meng, W.; Kwok, L.F. Surveying Trust-Based Collaborative Intrusion Detection: State-of-the-Art, Challenges and Future Directions. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 280–305. [CrossRef]
12. Gyamfi, E.; Jurcut, A. Intrusion Detection in Internet of Things Systems: A Review on Design Approaches Leveraging Multi-Access Edge Computing, Machine Learning, and Datasets. *Sensors* **2022**, *22*, 3744. [CrossRef] [PubMed]
13. Diab, A.; Chehade, A.; Ragusa, E.; Gastaldo, P.; Zunino, R.; Baghdadi, A.; Rizk, M. Intrusion Detection on Resource-Constrained IoT Devices with Hardware-Aware ML and DL. *arXiv* **2025**, arXiv:2512.02272. [CrossRef]
14. Nassef, L.; Alghamdi, M.I.; CHAABANE, S.B.; Abbas, Q.; Alawad, W.M.; Albalawi, O.H.; Alqaisi, O.I.; Fakieh, B. Lightweight and Energy-Aware Intrusion Detection for Industrial IoT Using TinyML and Edge AI. *Sci. Rep.* **2026**, *16*, 18524. [CrossRef] [PubMed]
15. Bougueroua, N.; Mazouzi, S.; Belaoued, M.; Seddari, N.; Derhab, A.; Bouras, A. A Survey on Multi-Agent Based Collaborative Intrusion Detection Systems. *J. Artif. Intell. Soft Comput. Res.* **2021**, *11*, 111–142. [CrossRef]
16. Shamshirband, S. Cooperative Multi Agents for Intelligent Intrusion Detection and Prevention Systems. Ph.D. Thesis, University of Malaya, Kuala Lumpur, Malaysia, 2014.
17. A. M., R.; Ahmed, M.S.I.; Khan, R.L.R. An adaptive distributed Intrusion detection system architecture using multi agents. *Int. J. Electr. Comput. Eng. (IJECE)* **2019**, *9*, 4951. [CrossRef]
18. Hardegen, C.; Petersen, M.; Ezelu, C.; Geier, T.; Rieger, S.; Buehler, U. A Hierarchical Architecture and Probabilistic Strategy for Collaborative Intrusion Detection. In Proceedings of the 2021 IEEE Conference on Communications and Network Security (CNS), Tempe, AZ, USA, 4–6 October 2021; pp. 128–136. [CrossRef]
19. Funchal, G.; Pedrosa, T.; de la Prieta, F.; Leitão, P. Distributed Machine Learning and Multi-Agent Systems for Enhanced Attack Detection and Resilience in IoT Networks. In Proceedings of the 11th International Conference on Information Systems Security and Privacy, Porto, Portugal, 20–22 February 2025; pp. 192–203.
20. Zhang, J.; Luo, C.; Carpenter, M.; Min, G. Federated Learning for Distributed IIoT Intrusion Detection Using Transfer Approaches. *IEEE Trans. Ind. Inform.* **2023**, *19*, 8159–8169. [CrossRef]
21. Ruzafa-Alcazar, P.; Fernandez-Saura, P.; Marmol-Campos, E.; Gonzalez-Vidal, A.; Hernandez-Ramos, J.L.; Bernal-Bernabe, J.; Skarmeta, A.F. Intrusion Detection Based on Privacy-Preserving Federated Learning for the Industrial IoT. *IEEE Trans. Ind. Inform.* **2023**, *19*, 1145–1154. [CrossRef]
22. Rashid, M.M.; Khan, S.U.; Eusufzai, F.; Redwan, M.A.; Sabuj, S.R.; Elsharief, M. A Federated Learning-Based Approach for Improving Intrusion Detection in Industrial Internet of Things Networks. *Network* **2023**, *3*, 158–179. [CrossRef]
23. Gheni, H.Q.; Al-Yaseen, W.L. Using Ensemble Techniques Based on Machine and Deep Learning for Solving Intrusion Detection Problems: A Survey. *Karbala Int. J. Mod. Sci.* **2023**, *9*, 5. [CrossRef]
24. Upadhyay, D.; Manero, J.; Zaman, M.; Sampalli, S. Intrusion Detection in SCADA Based Power Grids: Recursive Feature Elimination Model With Majority Vote Ensemble Algorithm. *IEEE Trans. Netw. Sci. Eng.* **2021**, *8*, 2559–2574. [CrossRef]
25. Albashish, D.; Aburomman, A. Weighted heterogeneous ensemble for the classification of intrusion detection using ant colony optimization for continuous search spaces. *Soft Comput.* **2022**, *27*, 4779–4793. [CrossRef]
26. Wood, D.; Mu, T.; Webb, A.; Reeve, H.; Luján, M.; Brown, G. A Unified Theory of Diversity in Ensemble Learning. *arXiv* **2023**, arXiv:2301.03962. [CrossRef]
27. Abe, T.; Buchanan, E.K.; Pleiss, G.; Cunningham, J.P. Pathologies of Predictive Diversity in Deep Ensembles. *arXiv* **2023**, arXiv:2302.00704. [CrossRef]
28. Peng, Z.; Zhang, W.; Han, N.; Fang, X.; Kang, P.; Teng, L. Active Transfer Learning. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *30*, 1022–1036. [CrossRef]

29. Mueller, B.; Street, W.N. FedDES: Graph-Based Dynamic Ensemble Selection for Personalized Federated Learning. *arXiv* **2026**, arXiv:2603.28006.
30. Bao, Y.; Gong, W.; Yang, K. A Literature Review of Human–AI Synergy in Decision Making: From the Perspective of Affordance Actualization Theory. *Systems* **2023**, *11*, 442. [CrossRef]
31. Sabater, J.; Sierra, C. Review on Computational Trust and Reputation Models. *Artif. Intell. Rev.* **2005**, *24*, 33–60. [CrossRef]
32. Pinyol, I.; Sabater-Mir, J. Computational trust and reputation models for open multi-agent systems: A review. *Artif. Intell. Rev.* **2011**, *40*, 1–25. [CrossRef]
33. Yu, H.; Shen, Z.; Leung, C.; Miao, C.; Lesser, V.R. A Survey of Multi-Agent Trust Management Systems. *IEEE Access* **2013**, *1*, 35–50. [CrossRef]
34. Braga, D.D.S.; Niemann, M.; Hellingrath, B.; Neto, F.B.D.L. Survey on Computational Trust and Reputation Models. *ACM Comput. Surv.* **2018**, *51*, 1–40. [CrossRef]
35. Ismail, R.; Jøsang, A. The Beta Reputation System. In Proceedings of the 15th Bled Electronic Commerce Conference, Bled, Slovenia, 17–19 June 2002; pp. 41–55.
36. Teacy, W.T.L.; Patel, J.; Jennings, N.R.; Luck, M. TRAVOS: Trust and Reputation in the Context of Inaccurate Information Sources. *Auton. Agents Multi-Agent Syst.* **2006**, *12*, 183–198. [CrossRef]
37. Auer, P.; Cesa-Bianchi, N.; Fischer, P. Finite-time Analysis of the Multiarmed Bandit Problem. *Mach. Learn.* **2002**, *47*, 235–256. [CrossRef]
38. Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 10–12 November 2015; pp. 1–6. [CrossRef]
39. Neto, E.C.P.; Dadkhah, S.; Ferreira, R.; Zohourian, A.; Lu, R.; Ghorbani, A.A. CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment. *Sensors* **2023**, *23*, 5941. [CrossRef] [PubMed]

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