future internet

MDPI

Article

Socially Grounded IoT Protocol for Reliable Computer Vision in Industrial Applications

Gokulnath Chidambaram [1,*], Shreyanka Subbarayappa [1,*] and Sai Baba Magapu [2]

[1] Department of Electronics and Communication Engineering, M. S. Ramaiah University of Applied Sciences, Bengaluru 560058, India
[2] School of Natural Sciences and Engineering, National Institute of Advanced Studies (NIAS), Bengaluru 560012, India; msaibaba@nias.res.in
* Correspondence: c.gokulnath@gmail.com (G.C.); shreyanka.ec.et@msruas.ac.in (S.S.)

Abstract

The Social Internet of Things (SIoT) enables collaborative service provisioning among interconnected devices by leveraging socially inspired trust relationships. This paper proposes a socially driven SIoT protocol for trust-aware service selection, enabling dynamic friendship formation and ranking among distributed service-providing devices based on observed execution behavior. The protocol integrates detection accuracy, round-trip time (RTT), processing time, and device characteristics within a graph-based friendship model and employs PageRank-based scoring to guide service selection. Industrial computer vision workloads are used as a representative testbed to evaluate the proposed SIoT trust-evaluation framework under realistic execution and network constraints. In homogeneous environments with comparable service-provider capabilities, friendship scores consistently favor higher-accuracy detection pipelines, with F1-scores in the range of approximately 0.25–0.28, while latency and processing-time variations remain limited. In heterogeneous environments comprising resource-diverse devices, trust differentiation reflects the combined influence of algorithm accuracy and execution feasibility, resulting in clear service-provider ranking under high-resolution and high-frame-rate workloads. Experimental results further show that reducing available network bandwidth from 100 Mbps to 10 Mbps increases round-trip communication latency by approximately one order of magnitude, while detection accuracy remains largely invariant. The evaluation is conducted on a physical SIoT testbed with three interconnected devices, forming an 11-node, 22-edge logical trust graph, and on synthetic trust graphs with up to 50 service-providing nodes. Across all settings, service-selection decisions remain stable, and PageRank-based friendship scoring is completed in approximately 20 ms, incurring negligible overhead relative to inference and communication latency.

Keywords: Social Internet of Things (SIoT); trust management; trust-aware service selection; industrial computer vision; edge computing; heterogeneous IoT; performance-aware trust; graph-based trust modeling

Academic Editors: Domenico Ursino, Francesco Cauteruccio and Luca Virgili

Received: 30 December 2025
Revised: 21 January 2026
Accepted: 22 January 2026
Published: 27 January 2026

1. Introduction

The rapid growth of the Internet of Things (IoT) has transformed industrial environments by enabling continuous monitoring, automation, and data-driven decision-making across the manufacturing, logistics, energy, and process control domains. Among these technologies, computer vision (CV) has emerged as a critical enabler of intelligent industrial systems. High-resolution cameras, edge processors, and AI-based visual analytics

now support applications such as quality inspection, defect detection, robotic guidance, workplace safety monitoring, and equipment anomaly detection [1–3]. These applications often operate continuously and must deliver reliable, real-time performance under harsh and dynamic factory conditions.

Despite their importance, deploying computer vision solutions in Industrial IoT settings remains challenging. Modern factory floors contain highly heterogeneous devices ranging from resource-constrained embedded platforms to industrial PCs and GPU-enabled workstations with significant variation in computational capacity, energy constraints, model support, and communication capabilities. At the same time, industrial networks are exposed to bandwidth fluctuations, interference, and congestion, which can degrade inference latency, increase frame loss, and destabilize throughput [4–6]. Ensuring that the "right" device processes a given vision task at the "right" time is therefore a non-trivial problem, especially when multiple devices are capable of performing the same task.

A further challenge lies in trust and security. Conventional Industrial IoT deployments typically rely on static configurations, fixed routing, or centralized schedulers that do not explicitly account for device reliability, historical behavior, or quality of service delivered over time. For computer vision workloads, an incorrect, delayed, or malicious result can lead directly to production defects, safety incidents, or regulatory non-compliance. While prior research on IoT and SIoT has proposed various trust and reputation models, including graph-based trust computation and blockchain-assisted trust management, these approaches are rarely tailored to vision-centric metrics such as detection accuracy, frame-level latency, jitter, or throughput, nor are they evaluated in realistic industrial CV scenarios with heterogeneous hardware [7–11].

The Social Internet of Things (SIoT) [12,13] introduces a complementary perspective by treating devices as socially aware entities that can form, manage, and evolve "friendships" based on interaction history and shared roles. SIoT research has shown that social relationships between devices can improve network navigability, decentralize decision-making, and provide a natural basis for trust and collaboration. However, most existing SIoT frameworks focus on generic service discovery, social relationship modeling, or small-scale simulations, with limited consideration of data-intensive workloads and strict real-time constraints [14,15]. As a result, the application of socially grounded relationships and trust scoring to industrial computer vision pipelines, where high-volume video streams and frame-level latency are critical, remains largely unexplored.

Existing Social Internet of Things (SIoT) trust models primarily rely on abstract social relationships, reputation aggregation, or historical interaction patterns to evaluate device reliability [12–15]. While these approaches effectively capture social proximity and long-term behavioral trends, they remain largely domain-agnostic and do not explicitly incorporate execution-level performance characteristics. In particular, they do not account for computer vision-specific factors such as detection accuracy under workload variation, execution feasibility on heterogeneous hardware, or failure modes arising from real-time streaming constraints. In contrast, the proposed framework grounds trust evaluation directly in observed computer vision execution outcomes, enabling socially inspired relationships to emerge from measurable service behavior rather than from static or self-reported attributes.

To operationalize these socially inspired interactions, the proposed SIoT protocol integrates proven distributed systems technologies. Device discovery is achieved using a lightweight Gossip mechanism, enabling decentralized and fault-tolerant propagation of presence and capability information across heterogeneous IIoT networks [16,17]. Secure and efficient inter-device communication is handled through gRPC, which supports bidirectional streaming and structured payloads that are well-suited for transmitting computer vision results and test frames [18]. For friendship evaluation, the protocol employs

a PageRank-inspired scoring model that aggregates multi-attribute performance metrics including latency, accuracy, throughput, and reliability into a single trust value [19,20]. These components collectively enable devices to autonomously identify suitable partners, validate their capabilities, and select the optimal node for executing industrial computer vision workloads under varying network and hardware conditions.

To assess the effectiveness of the proposed protocol, the evaluation includes both homogeneous and heterogeneous test setups. In the homogeneous setting, multiple devices with similar hardware configurations and identical vision models undergo comparison to reveal how the protocol behaves when nominal capabilities are equivalent. In the heterogeneous setting, a Social IoT requester coordinates with diverse devices, including embedded platforms and GPU-enabled systems running different models and operating under varying video resolutions, frame rates (FPSs), and network bandwidths. These scenarios reflect realistic industrial conditions, where camera streams and processing nodes differ in capability and network quality. Across both setups, the evaluation includes measurements of vision accuracy, latency, and throughput, along with an analysis of how SIoT friendship scores evolve and influence device selection.

Unlike conventional edge-service selection approaches that rely on direct multi-metric aggregation or static reputation scores [4,5,12,14], the proposed framework operationalizes socially grounded trust through explicit relationship modeling and propagation. Trust is derived from observed execution outcomes and decomposed into multiple logical trust nodes per service-providing device, capturing distinct performance and execution characteristics. PageRank is employed not as a generic ranking mechanism, but as a means of propagating trust through the resulting relational graph, enabling indirect influence among services based on shared behavioral patterns. The social grounding of the proposed framework is motivated by established principles of relational interaction, where persistent relationships provide structure and interpretability beyond isolated performance measurements [21]. This design allows feasibility constraints, execution failures, and performance trade-offs to be reflected naturally in the friendship scores, distinguishing the proposed approach from metric-only selection or reputation-based SIoT models.

The proposed framework targets intra-site industrial deployments, such as factory floors or localized edge clusters, where service-providing devices operate within a managed network environment. The assumed setting corresponds to local-area or edge-network connectivity with variable but bounded bandwidth and latency, rather than wide-area cloud federation. This deployment model reflects common industrial computer vision scenarios, in which devices exhibit heterogeneous computational capabilities and experience execution failures due to resource constraints rather than persistent network disconnections.

This study makes the following contributions:

1. A socially grounded SIoT protocol that aligns device interactions with relational communication principles, enabling structured collaboration in industrial CV environments.
2. A trust-evaluation framework that integrates accuracy, latency stability, throughput, and model behavior to produce reliable and explainable friendship scores.
3. A mathematical scoring model that applies RPD-based attribute comparison, weighted parameter importance, and PageRank-inspired trust propagation.
4. A unified testing methodology covering homogeneous and heterogeneous hardware conditions, including variations in compute capability, model type, video resolution, frame rate, and network bandwidth.
5. A validated industrial testbed demonstrating stable trust evolution, consistent device selection, and improved operational reliability under realistic industrial CV workloads.

The remainder of this paper is organized as follows. Section 2 presents the background and related work in SIoT, trust management, industrial computer vision, and distributed

edge intelligence. Section 3 outlines the social-theoretic foundations that shape the interaction model. Section 4 describes the SIoT protocol architecture, including discovery, capability exchange, validation, and friendship scoring. Section 5 introduces the mathematical model for attribute weighting and PageRank-based trust propagation. Section 6 describes the test setup and presents the experimental results. Section 7 discusses the observed results, challenges, and implications of the proposed SIoT protocol. Section 8 concludes the study and outlines future research directions.

2. Background and Related Work

Industrial IoT deployments increasingly rely on distributed computer vision (CV) systems for tasks such as defect detection, asset tracking, anomaly monitoring, and safety enforcement. These workloads operate under fluctuating bandwidth conditions, heterogeneous device capabilities, and strict real-time performance requirements. Industrial environments include embedded processors, GPU-accelerated edge nodes, and high-performance industrial PCs that process high-resolution video streams while handling electromagnetic interference, thermal variation, and dynamic network conditions. Existing industrial CV research highlights scheduling optimization, resource-aware offloading, and adaptive inference strategies [22]. However, most frameworks do not integrate explicit trust evaluation, historical device reliability, or socially inspired collaboration mechanisms [12–14].

2.1. Industrial Computer Vision and Edge-Based Processing

Edge computing decreases latency and bandwidth consumption in industrial CV pipelines [1,2,23]. Prior studies explore hybrid cloud–edge CV architectures, GPU-assisted inference servers, and partitioned execution across heterogeneous hardware [1,4,5]. These approaches improve responsiveness and reduce computational bottlenecks, although they rely primarily on load-based or latency-driven heuristics for task assignment.

Industrial CV workloads often operate with varying resolutions, frame rates, and model complexities [24,25]. Existing edge-based solutions seldom incorporate performance-driven trust indicators, anomaly-aware node selection, or relationship-based collaboration, despite the volatility of real industrial networks [1,4,7,8,10].

Recent industrial and UAV-enabled vision pipelines increasingly integrate modern YOLO variants with upstream enhancement or adaptation modules to improve detection robustness under constrained sensing conditions. For example, Succulent-YOLO combines a customized YOLOv10-based detector with UAV-assisted monitoring to address image-quality variation and deployment constraints in agricultural environments [26]. Similarly, Mussel-YOLO integrates super-resolution reconstruction with a YOLOv10-based detector to enable object detection from degraded acoustic video streams in resource-constrained environments [27]. These studies highlight that detection performance and feasibility are strongly influenced by workload characteristics, sensing conditions, and execution environments, reinforcing the need for system-level mechanisms that can select reliable services under heterogeneous operational constraints.

2.2. Trust, Security, and Reliability in IoT and Industrial IoT

Trust management remains a core requirement for reliable IoT and Industrial IoT collaboration [8,9,28–30]. Traditional trust models use direct observation, recommendation aggregation, or behavioral scoring to evaluate node reliability [8,10,31]. Blockchain-enabled trust introduces verifiable interaction logs and decentralized validation, while federated trust schemes distribute trust scoring across multiple devices to minimize single-point failures [11,32–35].

However, most trust research remains domain-agnostic and does not incorporate performance metrics specific to computer vision, such as detection accuracy, inference latency, jitter, and frame-processing throughput. These omissions limit trust evaluation in industrial CV environments where safety-critical outcomes depend on frame-level correctness and timing precision.

Recent surveys have further emphasized the tight coupling between trust management and security mechanisms in Social Internet of Things deployments. A comprehensive SIoT security survey synthesizes recent advances across trust protocols, threat mitigation strategies, technological integrations, tools, and performance metrics, highlighting how trust evaluation is often intertwined with system resilience and security posture in distributed IoT environments [36]. While the present work focuses on trust-aware service selection grounded in observed execution behavior, these security-oriented studies provide important complementary perspectives and motivate future extensions toward adversarial resilience and fault-tolerant SIoT operation.

2.3. Social Internet of Things (SIoT) Foundations

Device selection strategies in edge and fog computing rely on resource availability, current load, latency minimization, or proximity-based heuristics [1,4,5]. More advanced approaches include reinforcement learning-based offloading, cost-optimized scheduling, and cooperative inference, which improve performance under dynamic workloads [5,6]. These strategies, however, typically overlook socially inspired collaboration, long-term device behavior, or trust-based partner evaluation [7,10,12].

Distributed CV workloads introduce additional complexity due to differences in model accuracy, inference speed, thermal constraints, and frame processing performance [24,25]. Existing device selection frameworks do not integrate these factors into trust scoring or social-relationship reasoning [14,15].

2.4. Distributed Inference and Device Selection in Edge Systems

Distributed inference has emerged as a key strategy for supporting real-time computer vision workloads in edge and fog computing environments [37]. Prior work investigates task offloading, workload partitioning, and cooperative execution across heterogeneous devices to reduce latency and balance computational load [1,4,5]. These approaches typically rely on resource availability, queue length, processing latency, or proximity-based heuristics when selecting execution nodes.

More advanced techniques incorporate reinforcement-learning-based scheduling, cost-aware optimization, and adaptive inference pipelines to improve performance under dynamic workloads and network conditions [5,6]. While effective in optimizing throughput and response time, such systems primarily focus on instantaneous performance metrics and short-term system state.

For distributed computer vision workloads, device selection is further complicated by differences in model accuracy, inference speed, thermal constraints, and sustained frame-processing capability across devices [24,25]. Existing distributed inference frameworks do not integrate these factors into long-term trust evaluation or relationship-aware device selection, limiting their applicability in collaborative and socially driven Industrial IoT scenarios [14,15].

Recent work has explored multi-source and cross-modal fusion frameworks for distributed edge intelligence, where heterogeneous data streams such as visual sensing, wireless signals, and contextual metadata are combined to improve perception robustness and situational awareness [38]. These approaches typically focus on collaborative inference and data fusion under the assumption that participating services are reliable and exe-

cutable. In contrast, the proposed SIoT framework addresses the complementary problem of trust-aware service selection by identifying feasible and reliable computer vision services prior to any fusion or aggregation, thereby providing a coordination layer that can support downstream multi-source fusion pipelines. In industrial monitoring and control pipelines, infeasible or delayed computer vision execution can invalidate downstream fusion outputs, leading to missed events, delayed actuation, or false alarms, thereby amplifying operational cost even when individual perception errors are small.

2.5. Limitations of Existing Research

The literature presents several critical gaps:

- IoT and SIoT trust models do not incorporate computer vision-specific performance metrics such as precision, recall, inference latency, jitter, or throughput [8,10,39].
- Existing SIoT frameworks seldom evaluate performance using heterogeneous industrial hardware executing real CV models [14,15].
- Distributed inference approaches focus on latency and resource optimization but do not adopt socially inspired trust evolution [1,4].
- Blockchain-based trust methods introduce latency overhead unsuitable for real-time object detection [11,33,35].
- Existing solutions do not account for dynamic camera resolutions, frame-rate variations, or bandwidth conditions that influence industrial CV reliability [24,25].
- Abed's CSIoT framework [21] advances relationship modeling but does not integrate vision-based validation or industrial CV behavior.
- These limitations highlight the need for a socially grounded SIoT protocol that integrates trust-aware device selection, CV performance evaluation, and relationship-driven collaboration for industrial environments.

The present work substantially extends a preliminary conference paper [40] by providing a complete protocol specification, formal trust modeling, and extensive experimental validation.

3. Theoretical Foundations

The proposed SIoT protocol is grounded in established theories from social science and interpersonal communication, which provide a principled basis for modeling trust, relationship formation, and interaction dynamics among autonomous devices. These theories offer a conceptual bridge between human social behavior and socially inspired device collaboration in distributed IoT environments.

3.1. Knapp's Relational Development Model

Knapp's relational development model describes relationship formation as a staged process through which entities progressively build interaction depth, mutual understanding, and trust over time [41]. The model identifies distinct phases of relationship initiation, intensification, maintenance, and potential dissolution, each characterized by increasing levels of information exchange and commitment.

In the context of the Social Internet of Things, these stages can be mapped to device interactions that evolve from initial discovery and capability advertisement to sustained cooperation and long-term service provision. Early interactions correspond to exploratory exchanges, while repeated successful task execution reinforces trust and strengthens the device relationship.

By adopting Knapp's model, the proposed SIoT protocol formalizes friendship establishment as a gradual and evidence-driven process rather than a binary or static association. This perspective aligns naturally with trust accumulation based on historical performance,

reliability, and behavioral consistency, which are central to industrial computer vision service collaboration.

1. Initiating reflects early-stage contact, aligning with device discovery under uncertain conditions.
2. Experimenting corresponds to capability exchange, where devices explore functional compatibility.
3. Intensifying aligns with trial interactions such as test-frame execution or controlled performance evaluation.
4. Integrating reflects stable collaboration, where devices begin to exchange operational workloads.
5. Bonding represents persistent cooperation supported by consistent trust scores and long-term operational alignment.

This staged progression offers predictable behavioral logic that maps to device interactions in industrial CV workloads. Devices progress toward stable collaboration as they demonstrate accuracy, reliability, and operational consistency.

3.2. Social Exchange Theory

Social Exchange Theory (SET) views relationships as outcomes of cost–benefit interactions, where each participant evaluates the utility returned from the relationship [42]. In SIoT, utility corresponds to performance indicators such as

- Inference accuracy.
- Processing latency.
- Throughput stability.
- Resource consumption.
- Reliability under load.

Devices prefer partners that deliver durable returns like lower latency, higher accuracy, stable throughput, or robust performance under bandwidth variation. Industrial CV environments generate fluctuating demands, making SET principles particularly relevant for evaluating long-term cooperative value.

Relationships become more stable when interaction outcomes consistently meet or exceed expectations. SET therefore informs the trust scoring logic, where favorable results strengthen collaborative ties and inconsistent behavior introduces penalties.

3.3. Social Penetration Theory

Social Penetration Theory (SPT) describes how relationships deepen through progressively richer and more meaningful exchanges [43]. In interpersonal communication, this process involves disclosure of layered information.

In SIoT, the following facts are true:

- Early interactions involve surface-level capability advertisement.
- Deeper interactions involve controlled test-frame processing.
- Mature interactions involve full operational cooperation with real streaming data.

This model supports a graded approach to relationship validation and trust evolution. Devices reveal increasing levels of capacity and reliability as interactions intensify. For industrial CV workloads, this staged approach reduces the risk of selecting unreliable devices before performance validation occurs.

3.4. Fiske's Relational Models

Fiske's Relational Models Theory outlines four interaction patterns: Communal Sharing, Authority Ranking, Equality Matching, and Market Pricing [44]. These categories help classify device–device relationships in industrial environments.

- Communal Sharing corresponds to cooperative processing, where devices share capabilities for mutual benefit (e.g., CV offloading among similar edge nodes).
- Authority Ranking reflects hierarchical collaboration, such as GPU-enabled servers supporting lower-tier embedded devices.
- Equality Matching aligns with symmetric exchanges, such as devices with equivalent models and workloads.
- Market Pricing corresponds to performance-based collaboration, where attribute weighting and trust scoring influence partner selection.

Industrial CV deployments often include diverse hardware tiers, making Fiske's framework useful for structuring role-based collaboration.

3.5. Homophily Theory

Homophily theory states that entities with similar characteristics exhibit a higher likelihood of forming meaningful relationships [7,12]. In SIoT, similarity factors include the following:

- Hardware characteristics (CPU/GPU capability);
- Supported vision models;
- Operating conditions;
- Communication bandwidth;
- Device role (requesting or providing services).

In industrial computer vision systems, homophily influences the clustering of devices with shared operational profiles. Devices processing similar resolutions, FPS levels, or model architectures exhibit more predictable performance, making them suitable collaboration candidates. Homophily therefore supports efficient search-space reduction when identifying suitable peers in socially driven device-selection processes [14,15].

3.6. Proximity, Reciprocity, and Competence

Abed's CSIoT framework [21] expands SIoT trust formation using proximity, similarity, reciprocity, and competence factors. These factors mirror behavioral influences observed in human social networks:

- Proximity reflects physical or network closeness, relevant for bandwidth-dependent CV cooperation.
- Reciprocity captures consistency of responses and timely interaction.
- Competence aligns with device capability, model support, and resource availability.
- Similarity strengthens relationship formation among devices with aligned performance profiles.

These principles support multi-dimensional trust formation and align effectively with industrial CV contexts, where device quality depends on accuracy, latency, and stability.

3.7. Summary of Theoretical Relevance

The theories reviewed above collectively influence the behavioral model of the SIoT protocol:

- Knapp's stages model device interaction phases.
- Social Exchange Theory drives utility-oriented trust evaluation.
- Social Penetration Theory informs graded validation through test frames.

- Fiske's relational types guide role-based collaboration.
- Homophily reduces search complexity and improves compatibility.
- Abed's CSIoT principles strengthen multi-factor trust reasoning.

These theories provide conceptual grounding for the protocol architecture introduced in the next section.

4. SIoT Protocol Architecture

The proposed SIoT protocol adopts a layered architecture that structures device interaction into progressive stages aligned with relational communication theory. Each layer represents a functional phase in the collaboration lifecycle, mapping directly to Knapp's relational stages of initiation, experimentation, intensification, integration, and bonding. This layered organization supports scalable discovery, reliable capability exchange, secure interaction, and trust-oriented device selection in industrial computer vision environments. Figure 1 presents the proposed seven-layer SIoT protocol architecture, illustrating the progressive stages of device interaction from discovery and capability exchange to trust-based friendship scoring and service selection.

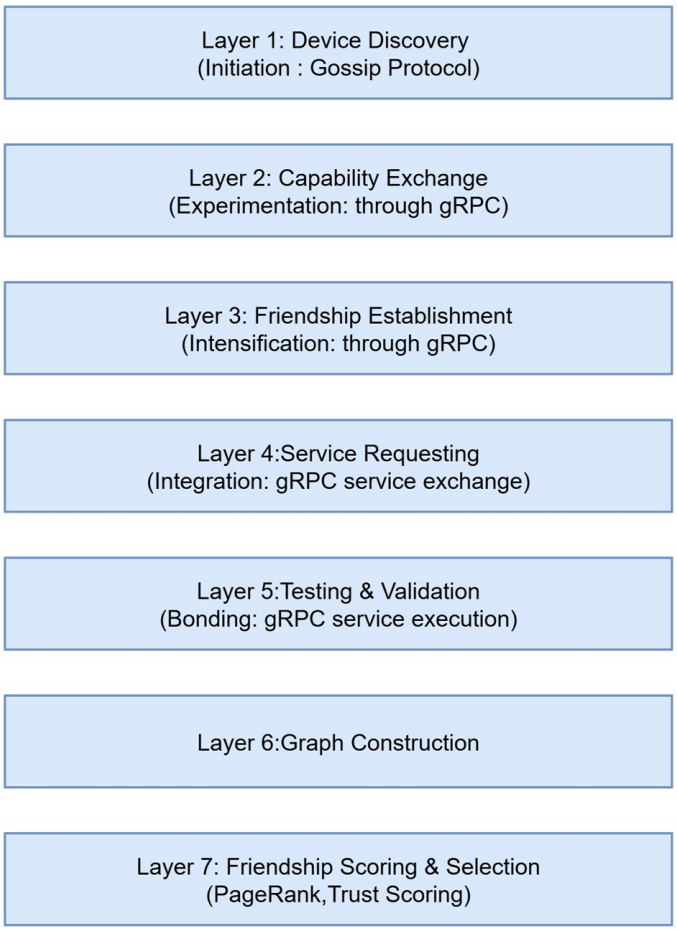


Figure 1. Seven-layer SIoT protocol architecture, aligned with relational development stages and operational phases of industrial device collaboration.

4.1. Trust Model Terminology and Decision Flow

For clarity, the following terms are used consistently throughout the manuscript:

- Trust: a quantitative assessment of a service provider's reliability derived from observed execution outcomes, including detection accuracy, communication latency, and processing behavior.

- Reputation: an aggregated or historical perception of reliability commonly employed in prior SIoT models but not directly used in the proposed framework.
- Friendship score: the final trust-derived ranking value computed through PageRank-based propagation over the SIoT trust graph, reflecting both direct and indirect trust relationships.
- Service selection: the process of choosing one or more service-providing devices based on their friendship scores while respecting execution feasibility constraints.

Scope of Trust Evaluation and Threat Model

Building on the trust definition introduced in the previous subsection, trust in the context of this study is defined as an execution-grounded measure reflecting the reliability and suitability of a service provider for industrial computer vision workloads. The selected metrics—detection accuracy, communication latency, and processing time—directly capture the primary factors influencing the correctness and timeliness of vision-based decisions in operational environments. Other dimensions commonly associated with trust in distributed systems, such as long-term availability, fault history, security posture, and adversarial robustness, are intentionally not modeled in this work. The proposed framework assumes a managed industrial deployment in which devices are authenticated and operate within controlled network boundaries, and the focus is placed on performance-driven trust under realistic execution constraints. These additional trust dimensions are complementary and can be incorporated as additional trust attributes in future extensions without altering the core socially grounded trust computation.

4.2. Layer 1—Device Discovery (Initiation Stage)

Layer 1 is responsible for discovering devices that operate within the SIoT environment. Discovery uses a Gossip protocol to propagate presence information without central coordination. Each device periodically publishes minimal identifiers, operational metadata, and availability status. This approach ensures robust discovery under fluctuating industrial network conditions and maps to the initiation stage of relational development, where entities identify potential partners without deep interaction.

4.3. Layer 2—Capability Exchange (Experimentation Stage)

Layer 2 supports structured exchange of device capabilities, including CV model support, hardware attributes, GPU characteristics, and operational constraints. The interaction occurs through gRPC messaging, allowing devices to share structured and authenticated capability data. This layer maps to Knapp's experimentation stage, where devices explore compatibility based on functional characteristics.

4.4. Layer 3—Friendship Establishment (Intensification Stage)

Layer 3 formalizes initial cooperation by initiating explicit friendship requests. Devices issue gRPC calls to confirm participation, exchange service metadata, and prepare for controlled test execution. This layer reflects the intensification stage, where interaction depth increases and devices begin to form targeted collaborative ties.

4.5. Layer 4—Service Requesting (Integration Stage)

Layer 4 executes service requests using gRPC service operations. Devices in the service-requesting mode transmit CV test frames, object-detection requests, or feature-execution tasks to suitable peers. Devices in the providing mode process these tasks and return structured responses. This stage represents integration, where cooperative processing becomes operational rather than exploratory.

4.6. Layer 5—Testing and Validation (Bonding Stage)

Layer 5 performs controlled validation of peer behavior. Devices transmit test frames to evaluating peers, assess response latency, compare inference outcomes with expected results, and update local metrics. This process reflects the bonding stage, where relationships strengthen based on validated reliability and performance consistency. Vision-specific metrics, such as frame-level precision, latency stability, and throughput, play a central role.

4.7. Layer 6—Graph Construction

Layer 6 constructs a directed friendship graph. Each device represents a node, and edges encode multi-attribute similarity using Relative Percentage Difference (RPD) and attribute weighting. This graph becomes the computational backbone for trust propagation. The graph adapts dynamically as devices update performance characteristics, ensuring alignment with current industrial conditions. Figure 2 provides a conceptual view of the directed SIoT friendship graph constructed at this stage, illustrating how performance metrics, device characteristics, and weighted trust edges are organized to support PageRank-based friendship score propagation.

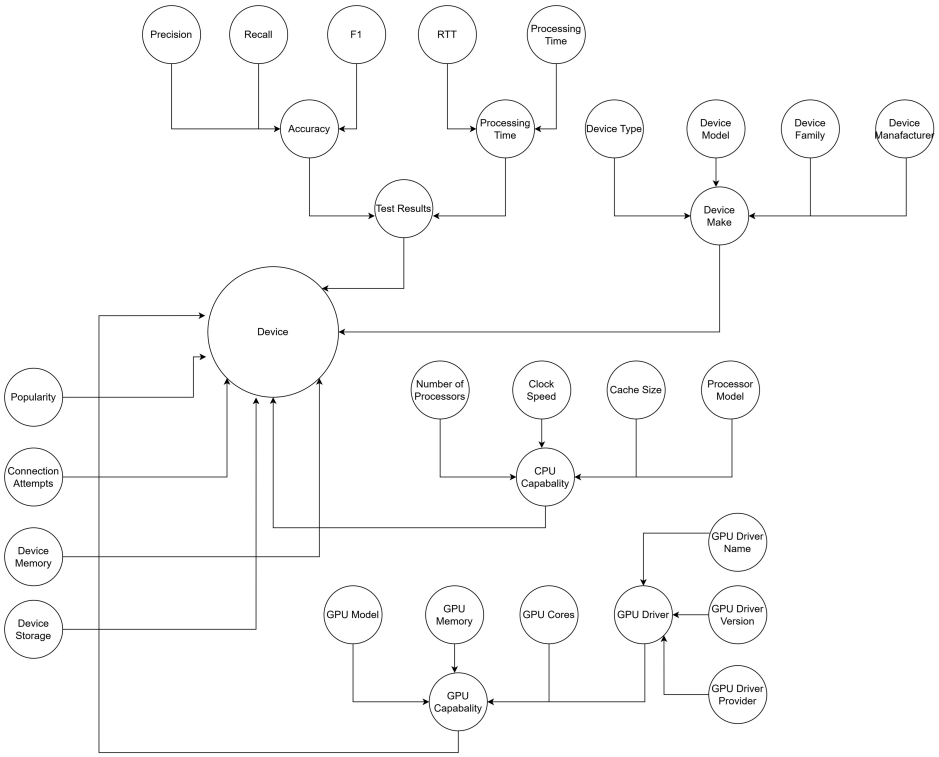


Figure 2. Conceptual representation of the SIoT friendship graph used for trust evaluation and service selection, showing device nodes, attribute-level weighting, directed trust edges, and PageRank-based friendship score propagation.

The SIoT trust graph is constructed dynamically from observed execution data rather than predefined social links or manually assigned relationships. For each service-providing device, normalized performance attributes are computed from measured detection accuracy, communication latency, and processing time. Logical trust nodes represent these attributes, and directed edges are instantiated by comparing relative performance across devices using the Relative Percentage Difference (RPD) formulation. Edge weights therefore reflect empirically observed dominance or degradation relationships rather than assumed trust levels. The relative importance weights assigned to accuracy- and latency-related attributes are not used to define the existence of trust edges, but rather to express application-level

preference during trust aggregation. These weights remain fixed within a given experimental configuration to ensure comparability across runs and reflect industrial computer vision priorities, where both correctness and timeliness are critical. Sensitivity analysis results in Section 6.6.4 show that weight variation affects friendship-score magnitude but does not alter service-selection outcomes.

4.8. Layer 7—Friendship Scoring and Selection (PageRank Trust Evaluation)

Layer 7 applies a PageRank-based scoring process [19] over the constructed SIoT friendship graph to derive trust scores for all potential collaborators. Each device receives a score proportional to its reliability, performance stability, interaction history, and consistency across prior evaluations. Devices with favorable performance attributes accumulate higher scores and become preferred partners for real-time industrial CV workloads.

PageRank-based trust propagation operates on the directed friendship graph constructed in Layer 6, where weighted directed edges are derived from accuracy, latency, and processing attributes produce a global ranking that supports robust service-provider selection under heterogeneous hardware and network conditions.

4.9. Implementation Considerations

The SIoT protocol implementation adopts a lightweight and platform-agnostic design that supports heterogeneous industrial environments. The prototype development uses Python 3.x, selected for its extensive ecosystem, rapid prototyping capability, and compatibility across embedded and high-performance platforms.

gRPC provides the structured communication layer for service requesting, capability exchange, and testing. Its support for bidirectional streaming, HTTP/2 transport, and Protocol Buffer serialization ensures efficient delivery of computer vision payloads and device metadata across diverse hardware. Protocol Buffers define compact message structures for device capabilities, service execution requests, and response exchanges, enabling robust and consistent interaction formats.

Distributed discovery relies on a Gossip mechanism, implemented using the Serf agent, which propagates presence information and device identifiers without central coordination. This approach supports dynamic membership changes, fault tolerance, and efficient propagation under varying industrial network conditions.

Graph construction and trust scoring use the NetworkX version 3.4.2 library, which provides optimized implementations for directed graph structures and PageRank-based evaluation. The library supports dynamic updates to graph nodes and edges, enabling real-time trust computation as device performance, network conditions, or capabilities evolve.

Execution and streaming failures observed during service execution are handled according to their severity in the proposed framework. Hard execution failures, including gRPC stream termination, incomplete frame delivery, or resource exhaustion that prevent completion of inference, are classified as FAIL and result in exclusion of the corresponding configuration from service selection. No friendship edge or validation reward is generated for such infeasible executions. Intermittent execution or streaming degradations that do not halt execution are explicitly treated as negative trust evidence. Such degradations do not receive validation rewards and contribute negatively to the corresponding friendship edge weights in the trust graph. As a result, devices that repeatedly exhibit degraded execution behavior experience a progressive reduction in their PageRank-based friendship scores and are deprioritized in subsequent service-selection rounds. The implications of this failure-handling strategy on performance metrics and service-selection outcomes are reflected in the experimental evaluation presented in Section 6.

The implementation approach ensures portability across industrial hardware, including embedded platforms, GPU-enabled devices, and high-performance workstations. Each device executes the protocol as a self-contained Python microservice, exposing the required SIoT interfaces and maintaining local friendship tables and configuration stores. The modular design supports integration into existing industrial automation platforms and enables interoperability across heterogeneous systems.

5. Mathematical Model

The SIoT trust evaluation framework quantifies the suitability of each device for industrial computer vision workloads through a combination of attribute comparison, weighted importance, and network-aware trust propagation. The mathematical model integrates performance indicators such as detection accuracy, latency stability, throughput, GPU characteristics, and network sensitivity. The scoring process comprises three stages:

1. Relative Percentage Difference (RPD) computation.
2. Weighted attribute aggregation.
3. PageRank-based trust propagation.

5.1. Attribute Definitions and Categories

Each device is evaluated using a set of n attributes, indexed by $j \in \{1, \ldots, n\}$. These attributes represent operational and performance characteristics relevant to industrial CV workloads. Attributes fall into two categories:

- Ascending attributes: Higher values indicate preferable performance. Examples include vision accuracy metrics (precision, recall, F1-score), processing throughput (frames per second), GPU memory size, GPU core count, and supported backbone models.
- Descending attributes: Lower values indicate preferable performance. Examples include network latency, per-frame processing time, and end-to-end response delay.

For each attribute j, the desired value is denoted by X_j^{desired} and the observed value for friend device i is represented as $X_{\text{friend}}^{(i,j)}$.

5.2. Weight Percentage (WP)

Table 1 presents the weight distribution of model parameters used for friendship computation in service requesting mode, expressed using the Weight Percentage (WP). The Weight Percentage $WP_{X,j}$ expresses the relative importance of each attribute j in the trust-evaluation process. Industrial computer vision workloads assign different priorities to performance indicators such as accuracy, latency, throughput, and compute capability. The value $WP_{X,j} \in [0, 1]$ determines the influence of deviations in attribute j on the final friendship score.

The weighting parameters adopted in this work are application-specific and reflect the operational requirements of industrial computer vision workloads. In such environments, detection accuracy is essential due to safety and process-reliability considerations, while timely execution is equally critical because delayed results are often operationally unusable. At the same time, neither accuracy nor processing latency alone is sufficient to characterize service trust, as execution feasibility, reliability, and validation feedback also play key roles. Accordingly, accuracy and processing time are assigned moderate, bounded weights to ensure balanced influence within the overall trust computation, preventing dominance by any single metric. These weights are design-time policy parameters rather than optimized coefficients and may be adjusted for other application domains with different performance priorities.

Table 1. Weight distribution of model parameters for friendship computation in service-requesting mode.

Category	Model Parameters	Weight (%)
Test Results	Accuracy (Precision, Recall, F1-score)—Ascending	15
	Processing Time (RTT, Processing Time)—Descending	15
Device Make	Device Type	0.5
	Device Family	0.5
	Device Model	0.5
	Device Manufacturer	0.5
Device Memory	Installed Memory	4
Device Storage	Storage Capacity	4
CPU Capability	Number of Processors	2.5
	Clock Speed	2.5
	Cache Size	2.5
	Processor Model	2.5
GPU Capability	GPU Model	0
	GPU Memory	15
	GPU Cores	15
	GPU Driver Name	0
	GPU Driver Provider	0
	GPU Driver Version	0
Connection Attempts	Successful Interaction Count—Descending	10
Friend Popularity	Popularity Score	10

Attributes with higher operational significance receive larger $WP_{X,j}$ values. For example, precision, recall, and F1-score retain higher weights in safet- critical inspection tasks, while throughput or GPU-memory capacity receive moderate weights when continuous processing speed becomes essential. Latency-sensitive applications assign higher weights to end-to-end delay and per-frame processing time.

The Weight Percentage does not require prior normalization across attributes, since the PageRank stage stabilizes relative contributions based on the SIoT graph structure. During scoring, $WP_{X,j}$ modifies the impact of the Relative Percentage Difference (RPD) term, amplifying penalties for deviations in strongly weighted attributes and reducing their effect for attributes with lower relevance.

All WP values originate from the device's Friendship Configuration file, allowing each node to express workload-specific preferences when selecting collaborators for industrial CV execution.

5.3. Relative Percentage Difference (RPD)

The Relative Percentage Difference (RPD) expresses the deviation between the expected and observed values while normalizing differences across attributes. For each device i and attribute j, the RPD is

$$\mathrm{RPD}_{i,j} = \frac{\left| X_{\text{friend}}^{(i,j)} - X_j^{\text{desired}} \right|}{\max\left(X_{\text{friend}}^{(i,j)}, X_j^{\text{desired}} \right)}.$$

When attribute j contains multiple subcomponents indexed by k (e.g., precision, recall, and F1-score contributing to accuracy), the RPD for each subcomponent is

$$\mathrm{RPD}_{i,j,k} = \frac{\left| X_{\text{friend}}^{(i,j,k)} - X_{\text{desired}}^{(j,k)} \right|}{\max\left(X_{\text{friend}}^{(i,j,k)}, X_{\text{desired}}^{(j,k)} \right)}.$$

5.4. Ascending Attributes

Ascending attributes represent performance indicators where higher values reflect stronger suitability for industrial CV workloads. These attributes describe accuracy, throughput, compute capability, and overall device strength.

For each ascending attribute j, the weight contribution for device i is

$$W^{(i,j)} = \begin{cases} 0, & X_{\text{friend}}^{(i,j)} = 0, \\ 1, & X_j^{\text{desired}} = 0, \\ \frac{1}{WP_{X,j} \cdot \mathrm{RPD}_{i,j} + b}, & X_{\text{friend}}^{(i,j)} < X_j^{\text{desired}}, \\ WP_{X,j} \cdot \mathrm{RPD}_{i,j} + b, & X_{\text{friend}}^{(i,j)} \geq X_j^{\text{desired}}. \end{cases}$$

Higher accuracy or throughput strengthens the ascending-attribute weight, while lower-than-expected performance reduces the weight through inverse scaling.

5.5. Descending Attributes

Descending attributes represent performance indicators where lower values reflect more desirable behavior. Industrial CV pipelines favor minimal network delay, low processing time, and reduced end-to-end latency.

For each descending attribute j, the weight contribution for device i is

$$W^{(i,j)} = \begin{cases} 0, & X_{\text{friend}}^{(i,j)} = 0, \\ 1, & X_j^{\text{desired}} = 0, \\ \frac{1}{WP_{X,j} \cdot \mathrm{RPD}_{i,j} + b}, & X_{\text{friend}}^{(i,j)} > X_j^{\text{desired}}, \\ WP_{X,j} \cdot \mathrm{RPD}_{i,j} + b, & X_{\text{friend}}^{(i,j)} \leq X_j^{\text{desired}}. \end{cases}$$

Network latency, processing time, and end-to-end delay strengthen the weight when they remain below target thresholds.

5.6. Split Attributes

When attribute j comprises K subcomponents, each subcomponent weight is computed as

$$W^{(i,j,k)} = \text{ascending or descending definition with bias } b_k,$$

and the aggregated attribute weight becomes

$$W^{(i,j)} = \sum_{k=1}^{K} W^{(i,j,k)}.$$

Split attributes are relevant for multi-metric characteristics such as accuracy (precision, recall, F1).

5.7. Composite Friendship Weight

The composite friendship weight for device i is

$$FW_i = \sum_{j=1}^{n} W^{(i,j)}.$$

Higher composite weights reflect stronger alignment with industrial CV requirements.

5.8. SIoT Graph Construction

The SIoT friendship graph is defined as

$$G = (V, E),$$

where each node represents a device and each edge (u, v) encodes directional trust with weight proportional to FW_u. The graph structure adapts dynamically as devices update metrics or network conditions shift.

5.9. PageRank-Based Trust Scoring

Trust propagation uses a PageRank formulation. The trust score for device T_i is

$$PR(T_i) = (1 - d) + d \sum_{T_j \in M(T_i)} \frac{PR(T_j)}{C(T_j)},$$

where d is the damping factor, $M(T_i)$ denotes devices with links directed to T_i and $C(T_j)$ denotes outgoing edges from T_j.

Devices with consistent accuracy, reduced latency, and favorable composite weights achieve higher PageRank scores. PageRank-based friendship scoring is computed using a damping factor of 0.85, consistent with standard practice for influence propagation in directed graphs. All nodes are initialized with equal rank values at the start of computation. Iterative updates are performed until the ℓ_1-norm difference between successive PageRank vectors falls below 10^{-6} or a maximum of 50 iterations is reached, whichever occurs first. In all evaluated scenarios, convergence is achieved well within the iteration limit, ensuring stable and reproducible friendship scores.

5.10. Final Device Selection

The final trust value guiding device selection is

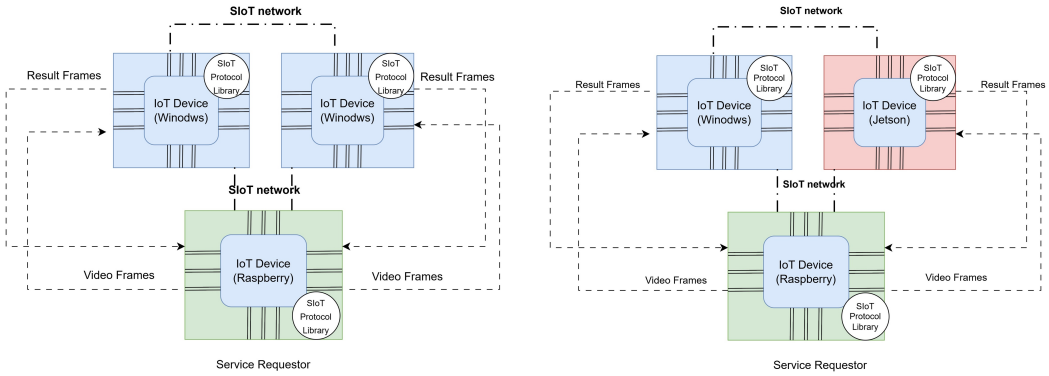
$$S_i = PR(T_i),$$

where S_i ranks the suitability of each device for subsequent industrial CV workloads.

6. Experimental Results and Analysis

6.1. Experimental Setup

This section describes the experimental configuration used to evaluate the proposed SIoT-based protocol for industrial computer vision workloads. The setup is designed to assess protocol behavior under controlled yet realistic operating conditions, reflecting both homogeneous and heterogeneous device environments commonly found in industrial IoT deployments. Figure 3 illustrates the SIoT experimental test setup used in this study, showing both the homogeneous and heterogeneous deployment environments considered for evaluation.



(**a**) Homogeneous environment (**b**) Heterogeneous environment

Figure 3. SIoT test setup: (**a**) homogeneous environment with two Windows-based service providers and a Raspberry Pi requester; (**b**) heterogeneous environment with Windows and Jetson service providers interacting with a Raspberry Pi requester. Dashed lines represent application-level data exchange, including video frames and result frames transmitted between devices, while solid bold lines denote the underlying SIoT network connectivity. Different colors are used to distinguish device types (Windows, Jetson, and Raspberry Pi).

6.1.1. Test Environments

Two experimental environments are considered:

1. Homogeneous environment: Two Windows-based devices (D1 and D2) with similar hardware capabilities act as service providers, while a Raspberry Pi device (D4) functions as the service requester. This configuration evaluates SIoT behavior when service providers exhibit comparable computational resources.
2. Heterogeneous environment: A Windows device (D1) and an embedded GPU-based Jetson device (D3) act as service providers, with the Raspberry Pi device (D4) as the service requester. This configuration reflects practical industrial scenarios involving mixed hardware platforms with varying compute and memory characteristics.

6.1.2. Device Configuration

Each device is assigned a unique identifier for clarity throughout the evaluation:

- D1: Windows Device 1;
- D2: Windows Device 2;
- D3: Jetson Device;
- D4: Raspberry Pi Device.

The detailed hardware and software configurations of all devices, including processor type, memory capacity, storage, operating system, and accelerator support are summarized in Table 2. These configurations are kept constant throughout all experiments to ensure repeatability.

Table 2. Hardware and software configuration of devices used in the SIoT experimental setup.

Component	Raspberry Pi 4B (D4)	Jetson Nano (D3)	Windows Laptop (D1, D2)
Operating System	Debian GNU/Linux 12 (Bookworm)	Ubuntu 20.04.6 LTS (Focal)	Windows 11 Pro (Build 22631)
CPU	ARM Cortex-A72 Quad-core, 1.8 GHz	ARM Cortex-A57 Quad-core, 1.47 GHz	13th Gen Intel® Core™ i7-13700H 14 cores, up to 5.0 GHz

Table 2. *Cont.*

Component	Raspberry Pi 4B (D4)	Jetson Nano (D3)	Windows Laptop (D1, D2)
GPU	Broadcom VideoCore VI	NVIDIA Maxwell GPU 128 CUDA cores	Integrated Intel GPU + NVIDIA GPU (workstation class, configuration dependent)
Memory (RAM)	4 GB LPDDR4 (∼3.8 GB usable)	2 GB LPDDR4 (∼1.9 GB usable)	32 GB DDR5
Storage	32 GB microSD (∼14 GB free)	64 GB microSD (∼95% utilized)	1 TB NVMe SSD
Swap/Virtual Memory	199 MB	5 GB (zram)	Managed by OS (up to 38 GB virtual memory)
Network Interface	Gigabit Ethernet/Wi-Fi	Gigabit Ethernet	Gigabit Ethernet/Wi-Fi
Role in SIoT	Service requester	Service provider (heterogeneous setup)	Service provider (homogeneous and heterogeneous setups)

6.1.3. Object Detection Algorithms

- A1: YOLO-based detector.
- A2: SSD Inception v2 detector.
- A3: EfficientDet-Lite0 detector.

The architectural characteristics and deployment variants of these algorithms are summarized in Table 3. The detection algorithms are instantiated from publicly available pretrained implementations. The YOLO-based detector is deployed using the Ultralytics YOLO framework, while the SSD Inception v2 and EfficientDet-Lite0 models are obtained from the TensorFlow Detection Model Zoo and TensorFlow Hub, respectively.

In the homogeneous environment, algorithms A1 and A2 (as defined in Table 3) are deployed on D1 and D2 and systematically swapped to eliminate device–algorithm bias. In the heterogeneous environment, algorithm A3, as summarized in Table 3, is additionally deployed on the Jetson device (D3) to evaluate embedded GPU behavior under high-resolution streaming conditions.

Table 3. Object-detection algorithms deployed in the SIoT experimental setup.

Algorithm ID	Model Name	Detection Type	Backbone/Variant	Target Deployment	Reference
A1	YOLO	Single-stage	YOLO family	GPU-enabled real-time systems (Windows, Jetson)	[45]
A2	SSD Inception v2	Single-stage	Inception v2 + SSD	CPU/Edge GPU (Windows, Raspberry Pi)	[46]
A3	EfficientDet-Lite0	Single-stage	EfficientNet-Lite0 + BiFPN	Embedded and edge devices (Jetson Nano)	[47]

6.1.4. Workload and Input Configuration

All experiments use a traffic-surveillance workload focused on detecting and counting vehicles (cars and trucks). Video frames are streamed from the service requester (D4) to service providers using a gRPC-based streaming interface.

The workload is evaluated across the following input parameters:

- Resolutions:
  - Ultra HD (3840 × 2160).

  - Full HD (1920 × 1080).
  - HD (1080 × 720).

- Frame rates: 30 FPS and 60 FPS.
- Network bandwidths: 100 Mbps, 50 Mbps, and 10 Mbps.

Each experiment maintains a fixed combination of resolution, frame rate, and bandwidth to isolate the impact of individual parameters on protocol behavior.

6.1.5. Reproducibility Details

All experiments are conducted using publicly available datasets and pretrained detection models to ensure reproducibility. Video workloads are selected to represent common industrial surveillance and monitoring scenarios, with annotated object classes corresponding to vehicles and pedestrians. The datasets are used solely as inference workloads, and no additional model training or fine-tuning is performed as part of this study. The sources of pretrained model implementations are documented in References [48–50].

Network conditions are emulated by explicitly constraining available bandwidth between requestor and service-providing devices. Bandwidth limits are applied at the operating-system level to reflect realistic industrial network conditions. Packet loss and jitter are not artificially injected; instead, the evaluation focuses on bandwidth-induced latency effects, which dominate performance variability in the targeted deployment setting.

6.1.6. Performance Metrics

The evaluation considers both computer vision performance metrics and SIoT protocol-level metrics:

- Precision, Recall, and F1-score: Computed using ground-truth annotations associated with the test video content. These metrics quantify object-detection accuracy and are used as ascending attributes in the trust model.
- Round-Trip Time (RTT): Defined as the elapsed time between frame transmission from the requester and receipt of the processed result, including preprocessing, data transfer, and post-processing. RTT is treated as a descending attribute.
- Processing time: Measures algorithm execution time at the service provider, excluding network delay.
- Friendship score: Represents the final trust value assigned to each service provider after SIoT graph construction and PageRank-based propagation.
- SIoT Graph build time and PageRank computation time: Measure the overhead introduced by SIoT trust evaluation.

In the experimental evaluation, execution failures are treated consistently with the trust-handling logic described in Section 4.9. Device–algorithm configurations that cannot complete inference due to hard execution failures—such as gRPC stream termination, incomplete frame delivery, or resource exhaustion (e.g., Jetson execution at 4K resolution)—are classified as FAIL and excluded from friendship-score computation and service selection. In contrast, configurations that complete execution but exhibit degraded performance or intermittent streaming instability remain feasible and contribute negative trust evidence, resulting in reduced friendship edge weights and lower PageRank-based friendship scores in subsequent selection rounds.

6.1.7. SIoT Protocol Execution

For each experimental run, the SIoT protocol operates as follows:

1. All participating devices are preloaded with the proposed SIoT protocol library, and the requester device (D4) streams video frames to available service providers.

2. Service providers execute object-detection inference using the deployed algorithm.
3. Performance metrics are collected and normalized using the proposed mathematical model.
4. A directed SIoT graph is constructed using composite friendship weights.
5. PageRank-based trust propagation computes final friendship scores.
6. The service provider with the highest friendship score is selected for subsequent execution.

Weight Percentages used in the trust-evaluation process are defined in Table 1.

6.1.8. Experimental Scope

All experiments are conducted using identical datasets, configurations, and evaluation criteria across both environments. Each experimental configuration was executed three times under identical conditions to assess result stability. Reported values correspond to mean measurements across runs. Observed run-to-run variability was small relative to absolute execution time and did not affect relative device ranking or service-selection outcomes. For this reason, standard deviation values are omitted for clarity. This consistency ensures that observed differences in performance and trust behavior arise from device heterogeneity, algorithm characteristics, and network constraints rather than experimental bias.

6.2. Homogeneous Environment Results

Experimental results obtained in the homogeneous environment are summarized in Tables 4–6. Table 4 presents results for 4K resolution , Table 5 reports results for Full HD (1920p) resolution, and Table 6 summarizes results for HD (1080p) resolution. The corresponding trends are illustrated in Figures 4–7. The homogeneous and heterogeneous test results are reported separately in Sections 6.2 and 6.3, respectively.

Table 4. Homogeneous environment results for 4K resolution (3840 × 2160). RTT denotes round-trip communication latency measured in seconds, and processing time denotes the per-frame inference execution time on the service-providing device, measured in seconds.

Experimental Setup					Detection Performance			Runtime Performance		SIoT Metrics			Decision
Content	**Res.**	**FPS**	**BW (Mbps)**	**Device–Algo**	**Precision**	**Recall**	**F1 Score**	**RTT (s)**	**Proc. Time (s)**	**Friendship Score**	**Graph Time (s)**	**PR Time (s)**	**Chosen Device**
Traffic Surveillance	4K	30	100	D1–A1	0.120	0.353	0.175	6.502	0.269	0.182	1.812	0.019	D1
				D2–A2	0.053	0.033	0.036	6.704	0.354	0.135	–	–	–
				D1–A2	0.053	0.033	0.036	6.847	0.337	0.135	1.649	0.019	D2
				D2–A1	0.120	0.353	0.175	6.364	0.246	0.182	–	–	–
			50	D1–A1	0.120	0.353	0.175	15.419	0.243	0.182	1.661	0.019	D1
				D2–A2	0.053	0.033	0.036	15.888	0.336	0.135	–	–	–
				D1–A2	0.053	0.033	0.036	15.632	0.330	0.135	1.661	0.019	D2
				D2–A1	0.120	0.353	0.175	15.136	0.225	0.182	–	–	–
			10	D1–A1	0.120	0.353	0.175	59.973	0.286	0.182	1.670	0.019	D1
				D2–A2	0.053	0.033	0.036	61.413	0.353	0.135	–	–	–
				D1–A2	0.053	0.033	0.036	61.892	0.314	0.137	1.651	0.019	D2
				D2–A1	0.120	0.353	0.175	59.347	0.269	0.180	–	–	–
		60	100	D1–A1	0.174	0.508	0.253	6.512	0.262	0.191	1.661	0.019	D1
				D2–A2	0.061	0.048	0.047	6.673	0.352	0.126	–	–	–
				D1–A2	0.061	0.048	0.047	6.737	0.385	0.124	1.646	0.019	D2
				D2–A1	0.174	0.508	0.253	6.495	0.262	0.192	–	–	–
			50	D1–A1	0.174	0.508	0.253	15.850	0.239	0.191	1.660	0.019	D1
				D2–A2	0.061	0.048	0.047	13.869	0.338	0.126	–	–	–
				D1–A2	0.061	0.048	0.047	13.368	0.367	0.125	1.651	0.019	D2
				D2–A1	0.174	0.508	0.253	15.246	0.244	0.192	–	–	–
			10	D1–A1	0.174	0.508	0.253	59.356	0.261	0.191	1.671	0.019	D1
				D2–A2	0.061	0.048	0.047	61.323	0.349	0.126	–	–	–
				D1–A2	0.061	0.048	0.047	61.125	0.337	0.127	1.639	0.019	D2
				D2–A1	0.174	0.508	0.253	59.036	0.287	0.190	–	–	–

Table 5. Homogeneous environment results for Full HD resolution (1920p). RTT denotes round-trip communication latency measured in seconds, and processing time denotes the per-frame inference execution time on the service-providing device, measured in seconds.

Experimental Setup					Detection Performance			Runtime Performance		SIoT Metrics			Decision
Content	**Res.**	**FPS**	**BW (Mbps)**	**Device–Algo**	**Precision**	**Recall**	**F1 Score**	**RTT (s)**	**Proc. Time (s)**	**Friendship Score**	**Graph Time (s)**	**PR Time (s)**	**Chosen Device**
Traffic Surveillance	1920p	30	100	D1–A1	0.195	0.441	0.261	2.522	0.248	0.177	1.776	0.019	D1
				D2–A2	0.060	0.046	0.043	1.983	0.140	0.139	–	–	–
				D1–A2	0.060	0.046	0.043	2.676	0.133	0.131	1.649	0.019	D2
				D2–A1	0.195	0.442	0.261	1.849	0.284	0.185	–	–	–
			50	D1–A1	0.195	0.441	0.261	11.904	0.233	0.183	1.654	0.019	D1
				D2–A2	0.060	0.046	0.043	11.216	0.140	0.134	–	–	–
				D1–A2	0.060	0.046	0.043	11.569	0.133	0.133	–	–	–
				D2–A1	0.195	0.441	0.261	12.245	0.215	0.182	–	–	–
			10	D1–A1	0.195	0.441	0.261	56.404	0.213	0.184	1.656	0.019	D1
				D2–A2	0.060	0.046	0.043	54.707	0.165	0.133	–	–	–
				D1–A2	0.060	0.046	0.043	54.353	0.163	0.133	1.636	0.019	D2
				D2–A1	0.195	0.441	0.261	56.985	0.211	0.184	–	–	–
		60	100	D1–A1	0.186	0.601	0.276	2.830	0.239	0.182	1.657	0.019	D1
				D2–A2	0.101	0.082	0.080	2.242	0.153	0.135	–	–	–
				D1–A2	0.101	0.082	0.080	2.515	0.150	0.133	1.654	0.019	D2
				D2–A1	0.186	0.601	0.276	2.891	0.215	0.183	–	–	–
			50	D1–A1	0.186	0.601	0.276	11.305	0.233	0.184	1.656	0.019	D1
				D2–A2	0.101	0.082	0.080	10.961	0.140	0.133	–	–	–
				D1–A2	0.101	0.082	0.080	10.329	0.149	0.133	1.640	0.019	D2
				D2–A1	0.186	0.601	0.276	11.563	0.212	0.184	–	–	–
			10	D1–A1	0.186	0.601	0.276	56.386	0.235	0.184	1.656	0.019	D1
				D2–A2	0.101	0.082	0.080	53.799	0.151	0.133	–	–	–
				D1–A2	0.101	0.082	0.080	53.355	0.153	0.133	1.636	0.019	D2
				D2–A1	0.186	0.601	0.276	56.135	0.240	0.184	–	–	–

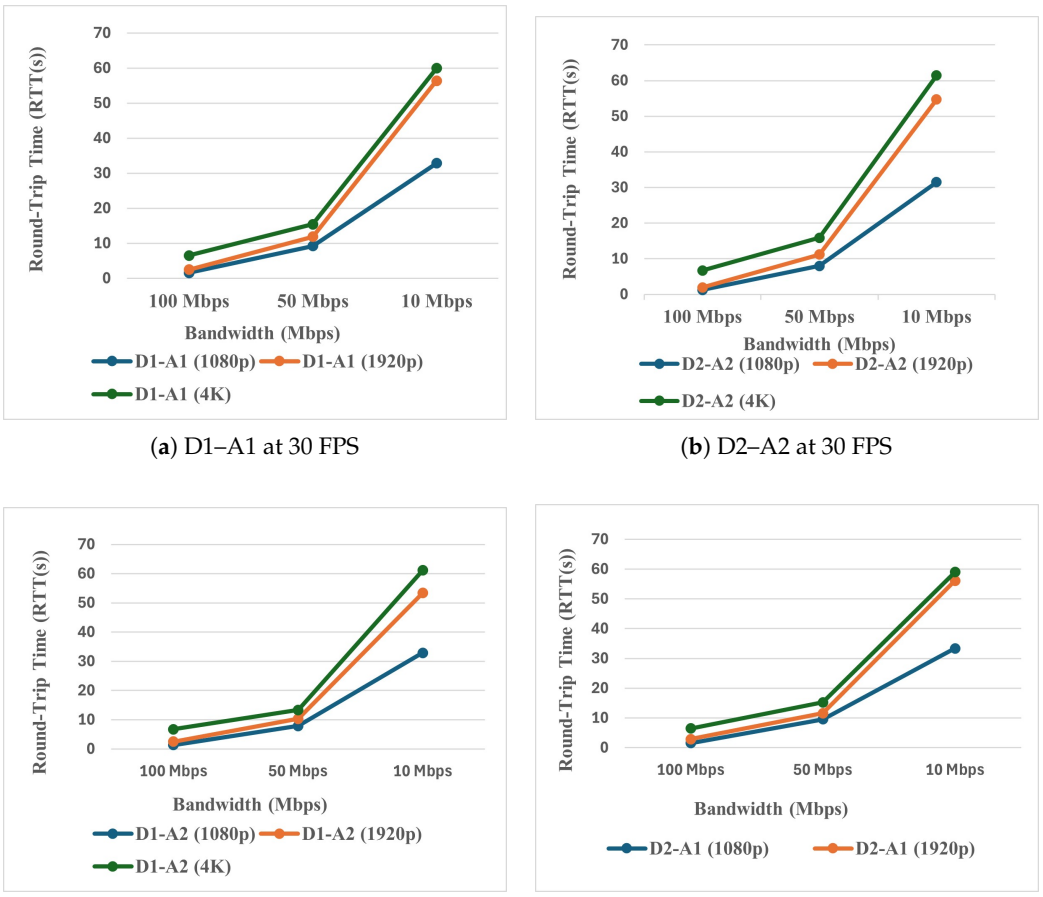
Table 6. Homogeneous environment results for (1080p) resolution . RTT denotes round-trip communication latency measured in seconds, and processing time denotes the per-frame inference execution time on the service-providing device, measured in seconds.

Experimental Setup					Detection Performance			Runtime Performance		SIoT Metrics			Decision
Content	**Res.**	**FPS**	**BW (Mbps)**	**Device–Algo**	**Precision**	**Recall**	**F1 Score**	**RTT (s)**	**Proc. Time (s)**	**Friendship Score**	**Graph Time (s)**	**PR Time (s)**	**Chosen Device**
Traffic Surveillance	1080p	30	100	D1–A1	0.178	0.507	0.258	1.597	0.181	0.167	1.736	0.019	D1
				D2–A2	0.102	0.070	0.076	1.266	0.091	0.150	–	–	–
				D1–A2	0.102	0.070	0.076	1.401	0.097	0.148	1.651	0.019	D2
				D2–A1	0.178	0.507	0.258	1.684	0.188	0.169	–	–	–
			50	D1–A1	0.178	0.507	0.258	9.236	0.190	0.181	1.674	0.019	D1
				D2–A2	0.102	0.070	0.076	7.983	0.091	0.136	–	–	–
				D1–A2	0.102	0.070	0.076	8.093	0.093	0.136	1.640	0.019	D2
				D2–A1	0.178	0.507	0.258	9.544	0.187	0.181	–	–	–
			10	D1–A1	0.178	0.507	0.258	32.853	0.189	0.181	1.670	0.019	D1
				D2–A2	0.102	0.070	0.076	31.502	0.090	0.136	–	–	–
				D1–A2	0.102	0.070	0.076	31.045	0.093	0.136	1.645	0.019	D2
				D2–A1	0.178	0.507	0.258	32.235	0.185	0.181	–	–	–
		60	100	D1–A1	0.188	0.580	0.275	1.626	0.181	0.169	1.658	0.019	D1
				D2–A2	0.118	0.103	0.103	1.325	0.091	0.148	–	–	–
				D1–A2	0.118	0.103	0.103	1.365	0.096	0.147	1.668	0.019	D2
				D2–A1	0.187	0.579	0.274	1.578	0.183	0.170	–	–	–
			50	D1–A1	0.188	0.580	0.275	9.860	0.187	0.182	1.667	0.019	D1
				D2–A2	0.118	0.103	0.103	8.095	0.090	0.135	–	–	–
				D1–A2	0.118	0.103	0.103	7.873	0.093	0.135	1.635	0.018	D2
				D2–A1	0.188	0.580	0.275	9.563	0.183	0.182	–	–	–
			10	D1–A1	0.188	0.580	0.275	33.867	0.187	0.182	1.660	0.019	D1
				D2–A2	0.118	0.103	0.103	32.533	0.090	0.135	–	–	–
				D1–A2	0.118	0.103	0.103	32.893	0.095	0.135	1.660	0.019	D2
				D2–A1	0.188	0.580	0.275	33.346	0.183	0.182	–	–	–

6.2.1. Round-Trip Time (RTT) Analysis

This subsection analyzes the round-trip time (RTT) behavior observed in the homogeneous experimental environment, where two identical Windows devices (D1 and D2) act as service providers and a Raspberry Pi (D4) functions as the service requester. Since D1 and D2 share comparable hardware configurations and operating systems, RTT vari-

ations primarily reflect the effects of network bandwidth, video resolution, frame rate, and protocol-level processing rather than hardware heterogeneity. The corresponding numerical results are summarized in Tables 4–6 and visualized in Figure 4.



(**a**) D1–A1 at 30 FPS

(**b**) D2–A2 at 30 FPS

(**c**) D1–A2 at 60 FPS

(**d**) D2–A1 at 60 FPS

Figure 4. Round-trip time (RTT) comparison for homogeneous device configurations under different device–algorithm assignments and frame rates.

In this study, round-trip time (RTT) is defined as the elapsed time between the transmission of a video frame from the service-requesting device and the reception of the corresponding detection result. RTT includes frame preprocessing, gRPC serialization, uplink transmission, response reception, and post-processing overhead but explicitly excludes the model inference time executed on the service provider, which is reported separately as processing time.

Figure 4 illustrates the RTT measured for different device–algorithm pairs under varying bandwidth conditions (100 Mbps, 50 Mbps, and 10 Mbps) at both 30 FPS and 60 FPS. As reported in Tables 4–6, RTT increases sharply as available bandwidth decreases across all resolutions, demonstrating the dominant influence of network throughput on end-to-end communication delay. At 100 Mbps, RTT remains relatively low and stable, while at 10 Mbps, RTT increases by an order of magnitude, particularly for high-resolution video streams. This behavior is consistent with the increased transmission time required for larger frame payloads under constrained network conditions.

The effect of frame rate is also evident. For a fixed resolution and bandwidth, RTT values at 60 FPS are consistently higher than those observed at 30 FPS. Higher frame rates increase the number of frames transmitted per second, thereby amplifying network congestion and buffering effects within the gRPC streaming pipeline. As a result, RTT reflects cumulative queuing delays rather than instantaneous inference latency alone.

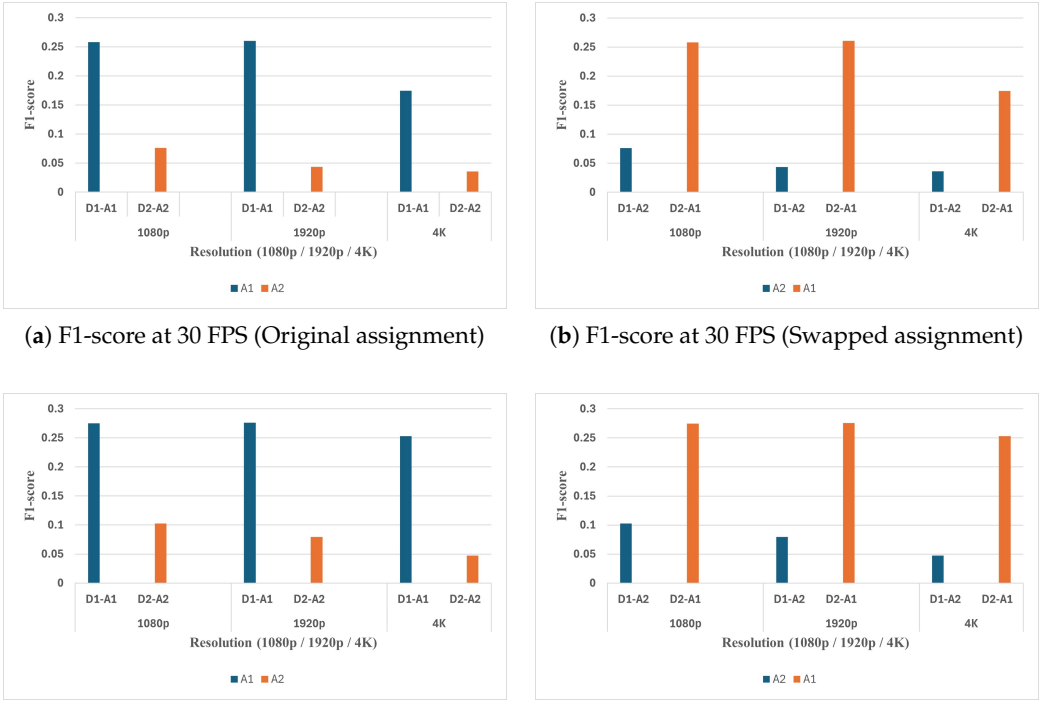
Although D1 and D2 are homogeneous devices, small but consistent RTT differences are observed between them across several test cases, as shown in Tables 4–6. These variations remain within a narrow range and do not exhibit systematic bias toward either device. Such differences arise from operating-system scheduling, thread synchronization, memory allocation behavior, and network stack timing rather than differences in algorithm execution time.

Resolution has a pronounced impact on RTT scaling. 4K resolution streams exhibit significantly higher RTT values compared to 1920p and 1080p resolutions, especially under reduced bandwidth conditions. The larger frame sizes associated with 4K content increase serialization and transmission costs, making RTT more sensitive to bandwidth constraints. In contrast, lower resolutions demonstrate more gradual RTT scaling, indicating improved robustness to network variability.

Overall, the homogeneous RTT results confirm that network bandwidth and frame rate dominate end-to-end delay, while device-level variations remain minimal when hardware configurations are similar. These observations establish a baseline for interpreting the effects of heterogeneity and failure scenarios analyzed in subsequent sections.

6.2.2. Detection Accuracy and F1-Score Analysis

This subsection evaluates detection performance in the homogeneous environment using precision, recall, and F1-score as ascending attributes. The analysis focuses on two object-detection algorithms—A1 and A2—executed on homogeneous service-provider devices (D1 and D2) under varying resolutions, frame rates, and bandwidth conditions. The corresponding quantitative results are reported in Tables 4–6 and visualized in Figure 5.



(**a**) F1-score at 30 FPS (Original assignment)

(**b**) F1-score at 30 FPS (Swapped assignment)

(**c**) F1-score at 60 FPS (Original assignment)

(**d**) F1-score at 60 FPS (Swapped assignment)

Figure 5. F1-score comparison of object-detection algorithms A1 and A2 under homogeneous configurations for original and swapped device–algorithm assignments at different frame rates.

Across all test configurations, Algorithm A1 consistently achieves higher precision, recall, and F1-score than Algorithm A2, irrespective of device assignment. This trend holds for both the original deployment (D1–A1, D2–A2) and the swapped configuration (D1–A2, D2–A1). Since D1 and D2 share comparable hardware characteristics, these results indicate

that detection accuracy in the homogeneous setting is primarily governed by algorithmic capability rather than device-level factors.

Resolution exhibits a noticeable but non-monotonic influence on detection accuracy. While moderate increases in spatial resolution improve object representation, the experimental results reported in Tables 4–6 show that 4K does not consistently yield higher F1-scores than 1920p. In several cases, 1920p achieves superior precision and F1-score compared to 4K. This behavior arises from internal input resizing within detection models, scale-mismatch effects between object size and anchor or grid structures, and increased pipeline pressure associated with higher-resolution streams. Consequently, higher spatial resolution does not automatically translate into improved detection accuracy under real-time streaming conditions.

Bandwidth variations have minimal direct impact on precision, recall, and F1-score. For a fixed resolution and frame rate, detection accuracy remains largely invariant across 100 Mbps, 50 Mbps, and 10 Mbps conditions, as observed in Tables 4–6. This observation confirms that bandwidth constraints primarily affect communication delay rather than inference correctness, provided that frames are successfully delivered and processed.

The swapped deployment experiments further validate the robustness of the accuracy results. When Algorithm A1 executes on either D1 or D2, detection performance remains stable, while Algorithm A2 exhibits similarly consistent but lower accuracy regardless of the hosting device. This outcome reinforces the separation between algorithm-level accuracy and device-level performance in homogeneous environments.

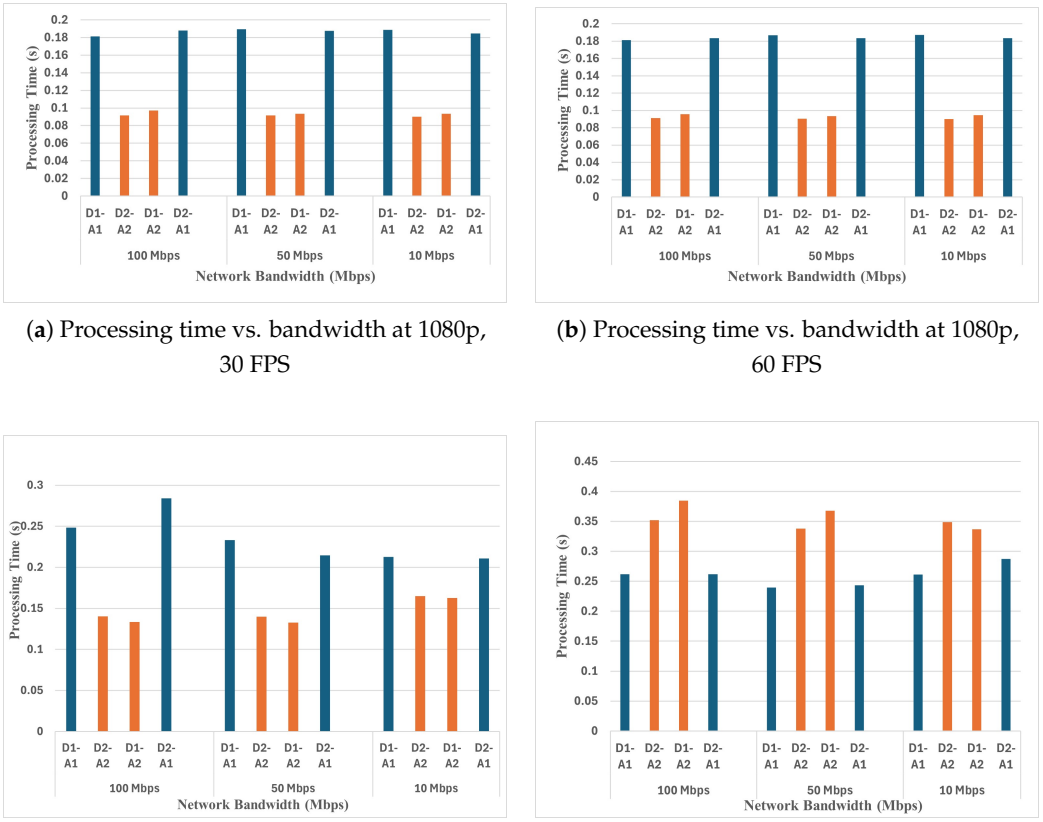
Overall, the homogeneous accuracy results demonstrate that algorithm selection dominates detection quality, while network bandwidth and device assignment exert negligible influence on precision, recall, and F1-score. These findings justify the emphasis placed on accuracy-related metrics in the trust-evaluation model and support their higher weighting in the friendship score computation.

6.2.3. Processing Time vs. Bandwidth

This subsection analyzes the processing time behavior observed in the homogeneous environment under varying bandwidth, resolution, and frame-rate configurations. Processing time refers exclusively to the model inference execution on the service-provider device and excludes communication-related delays, which are captured separately as RTT. The corresponding results are reported in Tables 4–6 and illustrated in Figure 6.

Across all resolutions and frame rates, processing time remains largely insensitive to network bandwidth. For a fixed resolution and FPS, the measured processing time exhibits minimal variation across 100 Mbps, 50 Mbps, and 10 Mbps conditions. This behavior confirms that processing time is dominated by algorithm execution characteristics and device computing capability rather than network constraints.

Algorithm-level differences strongly influence processing time. Algorithm A1 consistently exhibits higher processing time than Algorithm A2 at 1080p and 1920p resolutions, reflecting the increased computational complexity associated with A1. In contrast, at 4K resolution, this trend reverses in several test cases, with Algorithm A2 exhibiting comparable or higher processing time than Algorithm A1. This behavior arises from resolution-dependent internal preprocessing and scaling operations, where Ultra HD inputs introduce additional resizing and memory-handling overheads that affect lighter models disproportionately.



(**a**) Processing time vs. bandwidth at 1080p, 30 FPS

(**b**) Processing time vs. bandwidth at 1080p, 60 FPS

(**c**) Processing time vs. bandwidth at 1920p, 30 FPS

(**d**) Processing time vs. bandwidth at 4K, 60 FPS

Figure 6. Processing time variation with network bandwidth under homogeneous configurations for different resolutions and frame-rate settings.

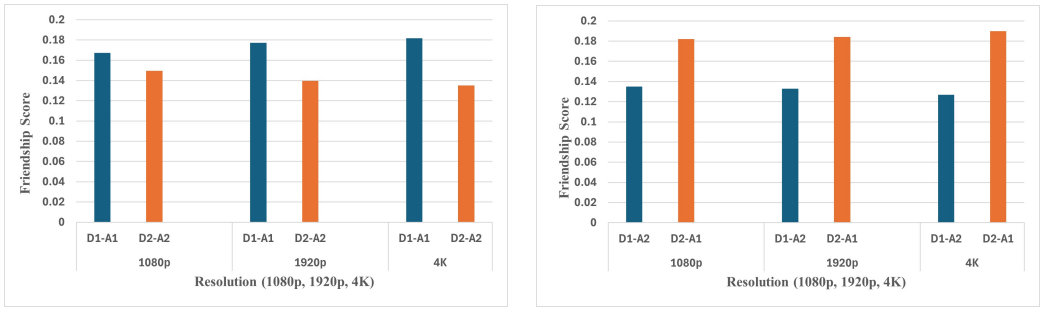
Resolution exerts a pronounced effect on processing time. Processing time increases substantially when moving from 1080p to 1920p and Ultra HD, particularly at higher frame rates. Ultra HD streams impose significantly higher memory bandwidth and tensor-handling demands, amplifying execution overhead even when inference input dimensions are internally normalized. As a result, processing time scales non-linearly with resolution, especially under real-time constraints.

Frame rate further amplifies processing demands. At 60 FPS, processing time increases relative to 30 FPS for all resolutions, reflecting reduced per-frame execution slack and increased scheduling pressure on the service-provider device. However, the relative ranking between algorithms remains consistent within each resolution regime, indicating that frame rate magnifies—but does not fundamentally alter—algorithmic processing characteristics.

Overall, the processing-time analysis demonstrates that algorithm complexity and input resolution dominate inference execution cost, while network bandwidth exerts negligible influence. These findings justify treating processing time as a descending attribute in the trust-evaluation model and motivate its explicit inclusion alongside RTT and accuracy in friendship-score computation.

6.2.4. Friendship Score Analysis

This subsection analyzes the friendship scores derived from the homogeneous experimental setup by integrating detection accuracy, processing time, and round-trip time within the proposed SIoT trust-evaluation framework. Friendship scores are computed using the weighted attribute model followed by PageRank-based trust propagation, as summarized in Tables 4–6 and illustrated in Figure 7.



(**a**) Friendship score at 30 FPS and 100 Mbps (**b**) Friendship score at 60 FPS and 10 Mbps

Figure 7. Friendship score comparison under homogeneous configurations for representative bandwidth and frame-rate conditions.

Across all homogeneous test configurations, service providers executingAlgorithm A1 consistently achieve higher friendship scores than those executing Algorithm A2. This outcome remains stable across different resolutions, frame rates, and bandwidth conditions, indicating that friendship scores are primarily driven by algorithm-level performance rather than transient network variations. The dominance of Algorithm A1 in friendship ranking directly reflects its superior precision, recall, and F1-score, as discussed in Section 6.2.2.

Low RTT alone does not guarantee a high friendship score. Several configurations exhibit comparable RTT values across service providers while yielding significantly different friendship scores. This behavior arises from the weighted trust model, in which accuracy-related attributes receive higher importance than latency-related attributes. As a result, service providers delivering consistently higher detection accuracy retain stronger trust scores even when their RTT or processing time is marginally higher.

Processing time contributes as a descending attribute in the friendship computation but does not override accuracy-dominated behavior in the homogeneous environment. Although Algorithm A1 often incurs higher processing time than Algorithm A2 at moderate resolutions, its superior detection accuracy compensates for this penalty in the weighted aggregation stage. This trade-off ensures that the trust model favors reliable detection performance over marginal gains in execution speed.

Bandwidth variations exert minimal influence on friendship scores in the homogeneous setup. Since accuracy remains stable across bandwidth conditions and processing time is largely network-independent, friendship scores exhibit only minor fluctuations between 100 Mbps, 50 Mbps, and 10 Mbps scenarios. This stability confirms that the proposed trust model effectively isolates intrinsic service quality from transient network effects when devices are homogeneous.

Overall, the homogeneous friendship score results validate the design of the SIoT trust-evaluation framework. By jointly considering ascending attributes such as detection accuracy and descending attributes such as RTT and processing time, the model consistently selects service providers that offer reliable and accurate computer vision performance. These results establish a baseline against which heterogeneous environment behavior is analyzed in the subsequent section.

6.3. Heterogeneous Results

This section presents the experimental results obtained in the heterogeneous environment, where service-provider devices exhibit distinct hardware and computational characteristics. In this setup, a Windows device (D1) and an embedded GPU-based device (D3—Jetson) act as service providers, while a Raspberry Pi (D4) functions as the service requester. Unlike the homogeneous configuration, this environment exposes the proposed SIoT protocol to variations in compute capability, memory architecture, and system-level constraints.

The heterogeneous experiments are designed to evaluate the robustness of the SIoT trust-evaluation framework under realistic industrial conditions, where participating devices differ significantly in performance and reliability. The analysis focuses on round-trip time, detection accuracy, processing time, and friendship score behavior, with particular attention to failure scenarios observed during high-resolution video streaming.

Experimental results obtained in the heterogeneous environment are summarized in Tables 7–9. Table 7 presents results for 4K (3840 × 2160) resolution, Table 8 reports results for Full HD (1920p) resolution, and Table 9 summarizes results for HD (1080p) resolution. The corresponding trends are illustrated in Figures 8–11.

Table 7. Heterogeneous environment results for 4K resolution (3840 × 2160). RTT denotes round-trip communication latency measured in seconds, and processing time denotes the per-frame inference execution time on the service-providing device, measured in seconds. FAIL indicates that the corresponding device–algorithm configuration could not complete inference due to resource exhaustion, runtime termination, or violation of predefined execution-time thresholds and is excluded from service selection.

Experimental Setup					Detection Performance			Runtime Performance		SIoT Metrics			Decision
Content	**Res.**	**FPS**	**BW (Mbps)**	**Device–Algo**	**Precision**	**Recall**	**F1 Score**	**RTT (s)**	**Proc. Time (s)**	**Friendship Score**	**Graph Time (s)**	**PR Time (s)**	**Chosen Device**
Traffic Surveillance	4K	30	100	D1–A1	0.120	0.353	0.174	6.403	0.273	0.316	1.360	0.023	D1
				D3–A3	FAIL	FAIL	FAIL	FAIL	FAIL	NA	–	–	–
			50	D1–A1	0.120	0.353	0.174	15.109	0.255	0.316	1.345	0.023	D1
				D3–A3	FAIL	FAIL	FAIL	FAIL	FAIL	NA	–	–	–
			10	D1–A1	0.120	0.353	0.174	60.523	0.283	0.316	1.349	0.023	D1
				D3–A3	FAIL	FAIL	FAIL	FAIL	FAIL	NA	–	–	–
		60	100	D1–A1	0.174	0.508	0.253	6.211	0.252	0.316	1.347	0.023	D1
				D3–A3	FAIL	FAIL	FAIL	FAIL	FAIL	NA	–	–	–
			50	D1–A1	0.174	0.508	0.253	15.399	0.226	0.316	1.346	0.023	D1
				D3–A3	FAIL	FAIL	FAIL	FAIL	FAIL	NA	–	–	–
			10	D1–A1	0.174	0.508	0.253	59.110	0.246	0.316	1.347	0.023	D1
				D3–A3	FAIL	FAIL	FAIL	FAIL	FAIL	NA	–	–	–

Table 8. Heterogeneous environment results for Full HD resolution (1920p). RTT denotes round-trip communication latency measured in seconds, and processing time denotes the per-frame inference execution time on the service-providing device, measured in seconds. FAIL indicates that the corresponding device–algorithm configuration could not complete inference due to resource exhaustion, runtime termination, or violation of predefined execution-time thresholds and is excluded from service selection.

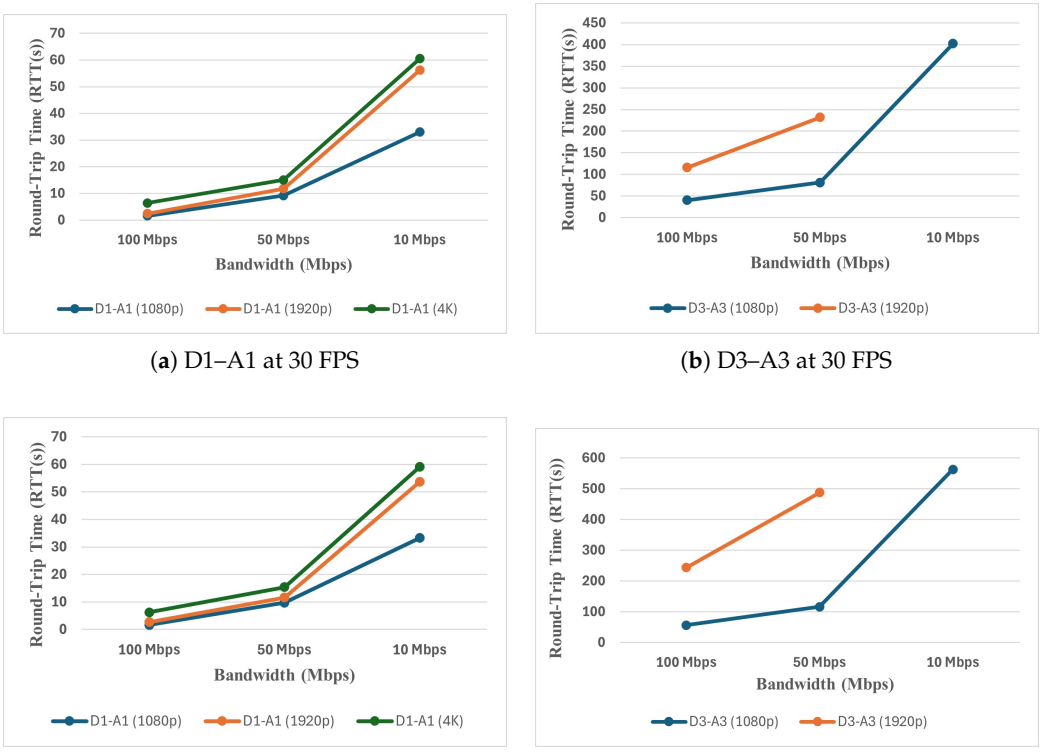
Experimental Setup					Detection Performance			Runtime Performance		SIoT Metrics			Decision
Content	**Res.**	**FPS**	**BW (Mbps)**	**Device–Algo**	**Precision**	**Recall**	**F1 Score**	**RTT (s)**	**Proc. Time (s)**	**Friendship Score**	**Graph Time (s)**	**PR Time (s)**	**Chosen Device**
Traffic Surveillance	1920p	30	100	D1–A1	0.195	0.441	0.261	2.467	0.238	0.199	1.581	0.023	D1
				D3–A3	0.148	0.160	0.149	115.860	0.229	0.118	–	–	–
			50	D1–A1	0.195	0.441	0.261	11.782	0.241	0.199	1.590	0.023	D1
				D3–A3	0.148	0.160	0.149	231.725	0.222	0.118	–	–	–
			10	D1–A1	0.195	0.441	0.261	56.199	0.240	0.316	1.346	0.023	D1
				D3–A3	FAIL	FAIL	FAIL	FAIL	FAIL	NA	–	–	–
		60	100	D1–A1	0.186	0.601	0.276	2.679	0.229	0.199	1.584	0.023	D1
				D3–A3	0.127	0.202	0.150	243.756	0.228	0.118	–	–	–
			50	D1–A1	0.186	0.601	0.276	11.594	0.221	0.199	1.591	0.023	D1
				D3–A3	0.127	0.202	0.150	487.511	0.222	0.118	–	–	–
			10	D1–A1	0.186	0.601	0.276	53.634	0.246	0.316	1.348	0.023	D1
				D3–A3	FAIL	FAIL	FAIL	FAIL	FAIL	NA	–	–	–

Table 9. Heterogeneous environment results for (1080p) resolution. RTT denotes round-trip communication latency measured in seconds, and processing time denotes the per-frame inference execution time on the service-providing device, measured in seconds. FAIL indicates that the corresponding device–algorithm configuration could not complete inference due to resource exhaustion, runtime termination, or violation of predefined execution-time thresholds and is excluded from service selection.

Experimental Setup					Detection Performance			Runtime Performance		SIoT Metrics			Decision
Content	Res.	FPS	BW (Mbps)	Device–Algo	Precision	Recall	F1 Score	RTT (s)	Proc. Time (s)	Friendship Score	Graph Time (s)	PR Time (s)	Chosen Device
Traffic Surveillance	1080p	30	100	D1–A1	0.178	0.507	0.259	1.612	0.187	0.199	1.607	0.023	D1
				D3–A3	0.170	0.233	0.187	40.204	0.222	0.117	–	–	–
			50	D1–A1	0.178	0.507	0.259	9.236	0.181	0.192	1.580	0.023	D1
				D3–A3	0.170	0.233	0.187	81.390	0.226	0.125	–	–	–
			10	D1–A1	0.178	0.507	0.259	32.983	0.189	0.192	1.578	0.023	D1
				D3–A3	0.170	0.233	0.187	402.044	0.240	0.125	–	–	–
		60	100	D1–A1	0.188	0.580	0.275	1.648	0.181	0.196	1.604	0.023	D1
				D3–A3	0.183	0.264	0.207	56.241	0.222	0.121	–	–	–
			50	D1–A1	0.188	0.580	0.275	9.763	0.189	0.189	1.592	0.023	D1
				D3–A3	0.183	0.264	0.207	115.873	0.219	0.128	–	–	–
			10	D1–A1	0.188	0.580	0.275	33.246	0.180	0.189	1.598	0.023	D1
				D3–A3	0.183	0.264	0.207	562.420	0.222	0.128	–	–	–

6.3.1. Round-Trip Time (RTT) Analysis

This subsection analyzes round-trip time (RTT) behavior in the heterogeneous environment, where a Windows device (D1) and an embedded GPU-based device (D3—Jetson) act as service providers and a Raspberry Pi (D4) functions as the service requester. RTT is used here as an end-to-end communication delay metric, as defined earlier. The results are summarized in Tables 7–9 and illustrated in Figure 8.



(**a**) D1–A1 at 30 FPS

(**b**) D3–A3 at 30 FPS

(**c**) D1–A1 at 60 FPS

(**d**) D3–A3 at 60 FPS

Figure 8. Round-trip time (RTT) comparison for heterogeneous device configurations under different device–algorithm assignments and frame rates.

Compared to the homogeneous setup, RTT values in the heterogeneous environment exhibit higher variance and stronger sensitivity to resolution and frame rate. For moderate resolutions (1080p and 1920p) at sufficient bandwidth, RTT remains within operational limits for both service providers. However, for 4K streams, stable RTT measurements are not observed for the Jetson-based service provider across multiple bandwidth settings. Instead, repeated gRPC streaming failures occur, preventing sustained frame transmission.

The observed failures for 4K streaming on the Jetson device arise from the combined effects of high-resolution frame payloads, limited memory headroom, and sustained gRPC streaming pressure. 4K frames significantly increase serialization overhead and buffer occupancy, leading to backpressure within the gRPC pipeline. Under these conditions, frame queues grow rapidly, triggering transport-level timeouts or resource exhaustion before inference results can be returned. As a result, RTT becomes unbounded, and the service effectively fails rather than degrading gracefully.

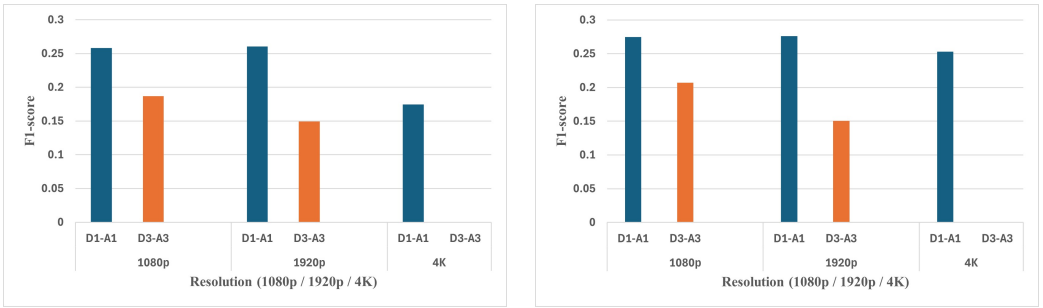
In contrast, the Windows-based service provider maintains stable RTT behavior under identical 4K streaming conditions. This difference reflects the availability of greater system memory, more aggressive buffering capacity, and more robust operating-system scheduling on workstation-class hardware. These characteristics allow the Windows device to absorb transient surges in data rate and maintain bidirectional gRPC communication even at high resolutions.

For resolutions where both service providers operate successfully, RTT follows expected scaling behavior. RTT increases as bandwidth decreases and as frame rate increases, consistent with the accumulation of transmission and buffering delays. However, even under these conditions, the Jetson-based service provider exhibits higher RTT values than the Windows device, indicating reduced tolerance to sustained streaming load.

Overall, the heterogeneous RTT results highlight the importance of device capability awareness in SIoT-based service selection. While homogeneous environments mask such limitations, heterogeneous deployments expose failure thresholds that cannot be captured by average RTT values alone. These findings motivate the integration of reliability and failure-awareness into the trust-evaluation framework, as discussed in subsequent subsections.

6.3.2. Detection Accuracy and F1-Score Analysis

This subsection evaluates detection accuracy in the heterogeneous environment using precision, recall, and F1-score as ascending attributes. The analysis considers object-detection algorithms A1, A2, and A3 deployed on heterogeneous service providers with distinct computational capabilities, namely a Windows-based device (D1) and an embedded GPU-based Jetson device (D3). The corresponding results are summarized in Tables 7–9 and illustrated in Figure 9.



(**a**) F1-score comparison for A1 and A3 at 30 FPS (**b**) F1-score comparison for A1 and A3 at 60 FPS

Figure 9. F1-score comparison of object-detection algorithms A1 and A2 under heterogeneous device configurations at different frame rates.

For configurations where stable streaming is achieved, detection accuracy primarily reflects algorithmic capability rather than device class. Algorithm A1 consistently delivers higher precision, recall, and F1-score than Algorithm A3 when executed on the Windows-based service provider, mirroring the behavior observed in the homogeneous environment. These results indicate that, in the absence of failures, algorithm selection remains the dominant factor governing detection accuracy even under heterogeneous hardware conditions.

The Jetson-based service provider exhibits acceptable detection accuracy for moderate resolutions such as 1080p and 1920p when bandwidth is sufficient. In these cases, Algorithm A3 achieves reasonable precision and recall, although its F1-score remains lower than that of Algorithm A1 executed on the Windows device. This difference reflects both algorithmic design and the constrained computational and memory resources available on embedded platforms.

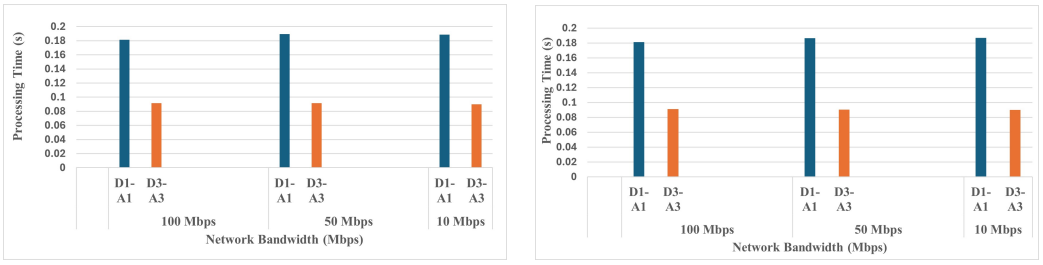
For 4K streaming, detection accuracy metrics are not reported for the Jetson device due to repeated gRPC streaming failures. In these scenarios, frames are not processed reliably, resulting in incomplete or missing inference outputs. Consequently, precision, recall, and F1-score values are marked as failed rather than degraded. This distinction is important, as it emphasizes that the absence of accuracy measurements arises from system-level instability rather than poor inference quality.

Frame rate exerts a secondary influence on detection accuracy in the heterogeneous environment. At 60 FPS, detection accuracy remains comparable to 30 FPS for resolutions where streaming is stable, indicating that temporal sampling does not significantly degrade inference correctness prior to system saturation. However, higher frame rates reduce the operational margin for embedded devices, increasing susceptibility to streaming failure under high-resolution workloads.

Overall, the heterogeneous accuracy results demonstrate that algorithmic strength determines detection quality when execution is feasible, while device capability governs whether such execution remains stable. These findings reinforce the need for trust models that account for both accuracy and reliability, particularly in heterogeneous SIoT deployments where failure behavior cannot be inferred from accuracy metrics alone.

6.3.3. Processing Time vs. Bandwidth

This subsection analyzes processing time behavior in the heterogeneous environment under varying bandwidth, resolution, and frame-rate conditions. Processing time represents the model inference execution time on the service-provider device and excludes communication-related delays captured by RTT. The corresponding results are reported in Tables 7–9 and illustrated in Figure 10.



(**a**) Processing time vs. bandwidth at 1080p, 30 FPS

(**b**) Processing time vs. bandwidth at 1080p, 60 FPS

Figure 10. Processing time variation with network bandwidth under heterogeneous device configurations at 1080p resolution for different frame-rate settings.

For configurations where stable streaming is achieved, processing time remains largely independent of network bandwidth for both service providers. Across 100 Mbps, 50 Mbps, and 10 Mbps conditions, processing time exhibits only minor variation for a fixed resolution and frame rate. This observation confirms that inference execution cost is dominated by algorithm complexity and device compute capability rather than network throughput.

Clear differences emerge between workstation-class and embedded platforms. The Windows-based service provider consistently achieves lower and more stable processing times compared to the Jetson device for moderate resolutions such as 1080p and 1920p. This behavior reflects higher available CPU and memory bandwidth, more aggressive parallelism, and reduced contention for system resources on the Windows platform.

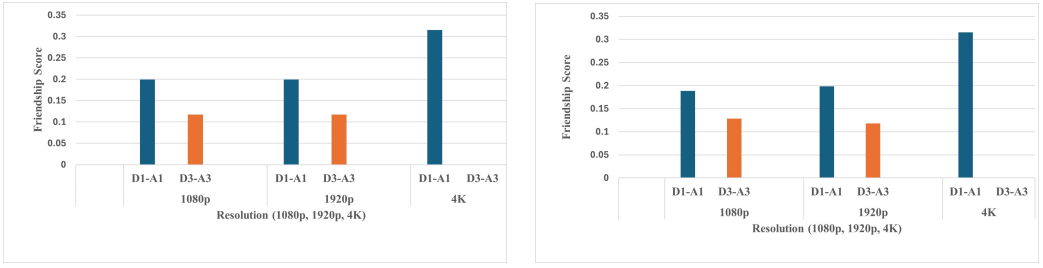
The Jetson-based service provider exhibits significantly higher processing time variability as resolution and frame rate increase. At 1920p resolution, processing time increases sharply at higher frame rates, indicating reduced scheduling slack and increased memory pressure on the embedded device. These effects are amplified at lower bandwidths, not because of network influence on inference execution, but due to upstream buffering and backpressure that reduce effective processing throughput.

For 4K configurations, processing time measurements are not reported for the Jetson device due to repeated streaming failures. In these cases, inference execution does not reach a steady state, preventing reliable processing-time measurement. This behavior highlights a fundamental limitation of embedded platforms when subjected to sustained high-resolution, high-frame-rate streaming workloads.

Overall, the heterogeneous processing-time results demonstrate that algorithm execution cost and device capability jointly determine inference feasibility. While bandwidth does not directly affect processing time, its interaction with buffering and streaming stability indirectly influences whether inference can proceed reliably on resource-constrained devices. These findings further motivate the inclusion of processing time as a descending attribute in the SIoT trust-evaluation framework.

6.3.4. Friendship Score Analysis

This subsection analyzes friendship score behavior in the heterogeneous environment by integrating detection accuracy, processing time, RTT, and reliability within the proposed SIoT trust-evaluation framework. Friendship scores are computed using weighted attribute aggregation followed by PageRank-based trust propagation. The corresponding results are summarized in Tables 7–9 and illustrated in Figure 11.



(**a**) Friendship score at 30 FPS and 100 Mbps (**b**) Friendship score at 60 FPS and 10 Mbps

Figure 11. Friendship score comparison under heterogeneous device configurations for representative bandwidth and frame-rate conditions.

In contrast to the homogeneous environment, friendship scores in the heterogeneous setup exhibit stronger differentiation across service providers. The Windows-based service provider consistently achieves higher friendship scores than the Jetson-based device across all stable configurations. This outcome reflects the combined effect of superior detection

accuracy, lower processing time, and stable RTT behavior under varying network and workload conditions.

Failure behavior plays a decisive role in friendship-score computation. For 4K streaming scenarios, the Jetson-based service provider repeatedly experiences gRPC streaming failures, resulting in incomplete inference outputs. In these cases, friendship scores associated with the Jetson device are either not computed or significantly penalized, effectively excluding the device from selection. This behavior demonstrates that the trust model captures reliability implicitly, favoring service providers that maintain consistent execution rather than those that intermittently fail.

Low RTT alone does not guarantee a high friendship score in the heterogeneous environment. Several configurations exhibit comparable RTT values across service providers while yielding substantially different friendship scores. This outcome arises from the weighting strategy employed in the trust model, where accuracy and execution reliability exert stronger influence than latency under industrial computer vision workloads. Consequently, devices delivering stable and accurate inference are consistently prioritized even when latency differences are marginal.

Processing time contributes as a descending attribute but remains secondary to accuracy and reliability in determining friendship scores. Although the Jetson device may achieve acceptable processing times under moderate workloads, its susceptibility to failure under high-resolution streaming reduces its overall trustworthiness. In contrast, the Windows-based service provider maintains stable performance across all tested configurations, resulting in consistently higher trust scores.

Overall, the heterogeneous friendship score results validate the ability of the proposed SIoT framework to distinguish between capable and unreliable service providers under realistic industrial conditions. By jointly considering accuracy, latency, processing cost, and failure behavior, the trust-evaluation mechanism selects service providers that offer dependable and sustained computer vision performance. These results confirm that socially grounded trust modeling is essential for robust service selection in heterogeneous SIoT deployments.

6.4. SIoT Graph Construction and Friendship Scoring Overhead

This subsection evaluates the computational overhead introduced by SIoT graph construction and PageRank-based friendship scoring across both homogeneous and heterogeneous experimental environments. Unlike inference execution and communication delay, these operations depend primarily on the size and topology of the SIoT graph rather than video resolution, frame rate, bandwidth, or device capability. The corresponding measurements are reported in Tables 4–9.

Across all experimental configurations, the SIoT graph structure remains fixed at 11 nodes and 22 directed edges. Consequently, graph construction time remains stable in both homogeneous and heterogeneous setups, typically within the range of approximately 1.3 s to 1.8 s. Minor variations are observed across experiments; however, these variations do not correlate with changes in resolution, frame rate, bandwidth, or algorithm selection. Instead, they arise from runtime factors such as memory allocation, interpreter scheduling, and graph initialization overhead.

The PageRank-based friendship scoring time exhibits minimal variation across all experiments, typically ranging between 0.019 s and 0.023 s. Since the graph topology and edge weights remain structurally consistent, this variation does not reflect changes in trust computation complexity. Rather, it is attributed to runtime effects, including thread scheduling, cache behavior, memory allocation, and floating-point convergence

characteristics during PageRank iteration. Importantly, the observed variation is negligible when compared to end-to-end communication delay and inference execution time.

Overall, these results demonstrate that SIoT graph construction and friendship scoring do not constitute performance bottlenecks in the proposed protocol. Compared to the latency introduced by video streaming and object-detection inference, the overhead associated with trust evaluation is minimal. This efficiency is particularly relevant for industrial SIoT deployments, where trust relationships may require frequent recomputation without compromising real-time responsiveness or system stability.

6.5. Scalability Analysis of the SIoT Trust Graph

To evaluate the scalability of the proposed SIoT trust computation beyond the size of the physical testbed, a synthetic trust-graph analysis was conducted. As illustrated in Figure 2, the SIoT framework operates at the trust-evaluation layer, where each service-providing device contributes multiple logical trust nodes representing performance metrics, execution characteristics, and device capabilities. As a result, the size of the SIoT graph increases with the number of participating devices and their associated evaluation attributes, rather than with the physical device count alone.

In this context, stability refers to the consistency of the relative PageRank ordering and the selected service provider as the SIoT trust graph scales in size, rather than to statistical variance or sensitivity metrics. In the scalability study, synthetic SIoT trust graphs were constructed to emulate deployments with increasing numbers of service providers. For each service provider, multiple logical trust nodes were instantiated to capture detection accuracy, communication latency, processing time, and device-level characteristics. Weighted directed edges were computed using the Relative Percentage Difference (RPD) formulation combined with workload-specific attribute weights. PageRank-based friendship scoring was then applied to the resulting graphs to assess convergence behavior and computational overhead under increasing graph size.

The evaluation considered configurations with up to 50 service providers, corresponding to approximately 500 logical trust nodes and nearly 30,000 weighted edges, as summarized in Table 10. Across all evaluated network sizes, PageRank convergence was stable under the fixed damping factor and convergence threshold described in Section 5, and the relative ordering of top-ranked trust nodes remained consistent across repeated executions. While absolute PageRank friendship scores varied with graph size, the relative ordering of top-ranked devices and the resulting trust-based service-selection decisions remained stable and were not sensitive to network scale.

The computational overhead of PageRank-based trust evaluation remained low as the graph size increased. Even for the largest evaluated configuration, the trust-scoring computation completed within a few tens of milliseconds, confirming that PageRank-based friendship evaluation can be executed frequently without compromising system responsiveness. For very small graphs, observed execution-time variability was dominated by Python runtime and library initialization overhead rather than graph size and therefore does not reflect scalability trends. It is noted that the reported PageRank execution times correspond exclusively to the in-memory trust-scoring computation and exclude protocol-level communication and inference overheads, which dominate end-to-end latency in the physical experimental testbed.

These results demonstrate that the proposed socially grounded SIoT trust model scales efficiently with increasing trust-graph dimensionality and introduces negligible overhead relative to inference and communication costs. This confirms the suitability of the framework for frequent trust re-evaluation in dynamic industrial IoT environments.

Table 10. Scalability and convergence behavior of PageRank-based SIoT trust evaluation with increasing network size.

Devices	Trust Nodes	Edges	PR Time (ms)	Ranking Stability
2	20	200	2.1	Stable
10	100	1800	1.6	Stable
20	200	5600	3.1	Stable
50	500	29,000	21.4	Stable

6.6. Baseline Selector Comparison

To isolate the contribution of socially grounded trust computation from conventional metric-based decision strategies, this section compares the proposed SIoT framework against three non-social baseline selectors. The comparison focuses on decision outcomes, rather than re-measuring performance, and operates exclusively on the summary metrics already reported in Sections 6.2 and 6.3.

6.6.1. Baseline Selector Definitions

Three baseline selectors are considered:

- Accuracy-First Selector: Selects the service provider with the highest detection accuracy, measured using the F1-score.
- Latency-First Selector: Selects the device with the lowest round-trip time (RTT), prioritizing communication responsiveness.
- Weighted-Sum Selector (Non-Social): Applies direct normalization and fixed weighting to accuracy, RTT, and processing time using the same relative importance as the proposed trust model, but without SIoT graph construction or PageRank-based trust propagation.

All baseline selections are derived deterministically from the reported summary metrics, and no additional experimental executions are performed.

6.6.2. Baseline Comparison in the Homogeneous Environment

Table 11 compares service-selection outcomes under representative homogeneous configurations spanning high-load, moderate-load, and constrained operating conditions. Representative configurations are selected to span low-load, moderate-load, and high-stress operating regimes; all underlying metrics for other configurations are reported in Tables 4–6. Since service providers exhibit comparable hardware and network characteristics in this environment, performance differences primarily arise from algorithmic behavior rather than device heterogeneity. Across homogeneous scenarios, baseline selectors and the proposed SIoT method frequently converge on the same device. This convergence reflects clear dominance in detection accuracy and minimal variation in runtime behavior between service providers. The proposed SIoT framework preserves these dominant performance signals and does not introduce artificial bias when metric differences are unambiguous.

Table 11. Comparison of service selection strategies in the homogeneous environment under representative conditions.

Experimental Configuration			Baseline Selectors			Proposed Method
Resolution	FPS	BW (Mbps)	Accuracy First	Latency First	Weighted Sum	SIoT (PageRank)
4K	30	100	D1–A1	D1–A1	D1–A1	D1–A1
4K	60	10	D1–A1	D1–A1	D1–A1	D1–A1
1920p	60	50	D1–A1	D2–A2	D1–A1	D1–A1
1080p	30	10	D1–A1	D2–A2	D1–A1	D1–A1

Baseline selections are derived deterministically from the summary metrics reported in Tables 4–6.

6.6.3. Baseline Comparison in the Heterogeneous Environment

Table 12 presents baseline selection outcomes for representative heterogeneous configurations involving resource-diverse devices. These scenarios introduce competing objectives between accuracy and runtime feasibility, reflecting realistic industrial deployment conditions. In several heterogeneous cases, baseline selectors and the proposed SIoT method coincide due to strong and consistent performance dominance by a single device. This outcome indicates that PageRank-based trust propagation respects clear evidence rather than overriding it. Divergence between selectors arises only under conditions where metric trade-offs or performance variability become significant, at which point the proposed framework emphasizes stability and consistency across repeated interactions.

Table 12. Comparison of service selection strategies in the heterogeneous environment under representative conditions.

Experimental Configuration			Baseline Selectors			Proposed Method
Resolution	**FPS**	**BW (Mbps)**	**Accuracy First**	**Latency First**	**Weighted Sum**	**SIoT (PageRank)**
4K	30	100	D1-A3	D1-A3	D1-A3	D1-A3
4K	60	10	D1-A3	D1-A3	D1-A3	D1-A3
1920p	60	50	D1-A3	D1-A3	D1-A3	D1-A3

Baseline selections are derived deterministically from the summary metrics reported in Tables 7–9.

6.6.4. Weight Sensitivity Analysis

To evaluate the robustness of the proposed SIoT trust model with respect to weight assignment, a sensitivity analysis is conducted by varying the top-level category weights assigned to accuracy and processing time while preserving internal metric ratios. Accuracy and processing-time weights are varied across a broad but reasonable range (5–25%), reflecting different application priorities.

Table 13 summarizes the resulting service-selection outcomes for representative homogeneous and heterogeneous configurations. Across all tested weight combinations, the selected service provider remains unchanged. Absolute PageRank friendship scores exhibit only minor, smooth variations under weight perturbation, while the relative ordering of service providers and final selection remain unchanged, indicating that decision outcomes are driven by dominant performance relationships rather than finely tuned weight values. In the heterogeneous configuration, devices that fail to execute under given resource constraints are excluded from the candidate set, and weight variation does not override feasibility constraints. These results demonstrate that the chosen weights are not tuned post hoc and that the proposed SIoT-based selection strategy is robust to reasonable weight perturbations.

Table 13. Sensitivity analysis of SIoT-based service selection under varying accuracy and processing-time weights.

Environment	Resolution	FPS	Accuracy: Processing Time (%)	Selected Device	Top Friendship Score
Homogeneous	4K	30	15:15	D1–A1	0.182
Homogeneous	4K	30	20:10	D1–A1	0.186
Homogeneous	4K	30	10:20	D1–A1	0.178
Homogeneous	4K	30	5:25	D1–A1	0.173
Homogeneous	4K	30	25:5	D1–A1	0.189
Heterogeneous	1920p	60	15:15	D1-A3	0.199
Heterogeneous	1920p	60	20:10	D1-A3	0.204
Heterogeneous	1920p	60	10:20	D1-A3	0.192

Table 13. *Cont.*

Environment	Resolution	FPS	Accuracy: Processing Time (%)	Selected Device	Top Friendship Score
Heterogeneous	1920p	60	5:25	D1-A3	0.185
Heterogeneous	1920p	60	25:5	D1-A3	0.210

Accuracy and processing-time weights are varied at the category level while preserving internal metric ratios. Friendship scores correspond to PageRank values of the selected service provider.

7. Discussion

While the proposed framework does not aim to benchmark or improve object-detection models, the validity of trust-aware service selection necessarily depends on the baseline reliability of the deployed detection pipelines. In this study, detection algorithms and datasets are intentionally chosen to reflect commonly used industrial computer vision workloads rather than to achieve state-of-the-art accuracy. Consequently, absolute F1-score values should be interpreted as representative of realistic operating conditions rather than as indicators of model optimality. Trust-aware selection is therefore meaningful within feasible operating regimes where detection quality is acceptable, and the framework is designed to identify the most reliable service provider under these practical constraints.

The discussion of homogeneous and heterogeneous results reveals that trust-aware service selection in the Social Internet of Things (SIoT) is shaped by both algorithmic performance and system-level feasibility constraints. This study evaluates whether socially grounded trust computation can reliably guide service selection when devices differ in performance, execution constraints, and network conditions, rather than assessing detector performance in isolation.

7.1. Key Observations Across Experimental Environments

Results from the homogeneous environment establish a controlled baseline in which service-provider devices exhibit similar hardware configurations and operating systems. Under these conditions, detection accuracy emerges as the primary differentiating factor, while round-trip time and processing latency remain largely comparable across devices. Consequently, the friendship score consistently favors the service provider executing the higher-accuracy algorithm, confirming that the proposed weighting strategy behaves as expected when device-level variability is minimal.

In contrast, the heterogeneous environment introduces realistic asymmetry in compute capability, memory architecture, and execution efficiency. Here, performance differentiation arises not only from algorithmic accuracy but also from device suitability and runtime stability. The observed divergence in processing time and RTT highlights the importance of incorporating multiple performance dimensions into trust evaluation, particularly under high-resolution and high-frame-rate workloads.

7.2. Homogeneous vs. Heterogeneous Trust Dynamics

A key insight from this work is the fundamentally different role played by trust metrics across homogeneous and heterogeneous environments. In homogeneous configurations, trust computation is dominated by algorithm-level performance, as device capabilities are effectively interchangeable. In heterogeneous configurations, however, trust emerges from the interaction between algorithm capability and device execution feasibility.

Importantly, the heterogeneous experiments intentionally preserve realistic deployment constraints. Algorithm–device pairings are restricted to configurations that are stable and deployable under real-time streaming conditions, reflecting practical SIoT deployments rather

than synthetic benchmarking scenarios. This design choice ensures that trust evaluation reflects service reliability rather than forced execution of unsuitable algorithm–device combinations.

The resulting friendship scores demonstrate that high algorithmic accuracy alone does not guarantee superior trust when deployed on resource-constrained platforms. Instead, SIoT service selection favors configurations that achieve a balanced trade-off between detection performance, communication latency, and processing efficiency. This behavior confirms that the proposed framework captures performance asymmetries introduced by heterogeneity in a principled and context-aware manner.

7.3. Robustness Under Feasibility Constraints

Observed detection performance, including absolute F1-scores, is also influenced by the intrinsic characteristics of the deployed detection algorithms and their pretrained configurations. The evaluated models are selected to represent commonly used industrial detection pipelines rather than to maximize benchmark accuracy on a specific dataset. Consequently, variations in F1-score reflect both algorithmic capability and system-level execution constraints. The proposed SIoT framework does not seek to optimize detection models themselves but instead provides a trust-aware mechanism to select the most reliable service provider given the available algorithms and operating conditions.

7.4. Protocol Overhead and Practical Feasibility

The computational overhead associated with SIoT trust evaluation is separated into two distinct components: trust-graph construction and PageRank-based friendship scoring. The reported graph construction time corresponds to initializing or updating the trust graph based on aggregated execution outcomes rather than performing per-frame recomputation. In the proposed framework, trust-graph construction is triggered at coarse temporal granularity, such as after completion of a workload batch or upon significant changes in device availability or execution behavior. As a result, the one-time construction overhead is distributed across multiple inference executions and does not dominate end-to-end latency.

From a practical standpoint, the experimental setup reflects realistic industrial scenarios, where a limited number of heterogeneous edge devices collaborate to process high-rate visual data streams. The observed behaviors mirror conditions commonly encountered in smart surveillance, automated inspection, and industrial monitoring systems, where device capability, network variability, and workload intensity interact in complex ways.

Trust-graph construction is therefore treated as an event-driven operation, triggered during initial system setup or when significant changes occur, such as the addition or removal of service-providing devices, administrative reconfiguration, or sustained changes in execution behavior. In contrast, PageRank-based friendship scoring operates on an already constructed in-memory graph and completes within a few tens of milliseconds, enabling frequent trust re-evaluation without compromising system responsiveness.

7.5. Practical Relevance and Experimental Scope

Recent work on distributed industrial edge systems has shown that lightweight consensus mechanisms enable consistent sensing data sharing under severe resource constraints by reducing communication and computation overhead through techniques such as node sampling and dynamically adjusted trust or reputation. These approaches focus on maintaining data integrity, provenance, and system-wide consistency while remaining scalable and fault tolerant in large, heterogeneous IoT deployments [51]. The proposed SIoT trust-evaluation framework complements such mechanisms by operating at a higher service-selection layer. While lightweight consensus protocols coordinate shared system state and data synchronization across edge servers, the SIoT framework leverages locally observed execution outcomes to guide trust-aware selection of feasible and reliable computer vision services.

In a combined deployment, consensus mechanisms provide an efficient coordination and consistency substrate, whereas the SIoT framework exploits execution-level feedback to adapt service selection under heterogeneous device capabilities and variable network conditions. This separation of concerns allows both approaches to work together without tight coupling, enhancing scalability and practical deployability in distributed industrial edge environments.

A common concern in SIoT evaluation is whether experiments involving a limited number of devices adequately reflect real-world deployments. In this work, the experimental setup is intentionally scoped to a small but representative set of service providers to enable controlled analysis of trust dynamics. SIoT trust computation is inherently relative: service providers are evaluated based on comparative performance rather than absolute scale.

Even with a small number of devices, the underlying friendship graph construction and PageRank-based scoring mechanism exhibit stable and interpretable behavior. These results indicate that meaningful trust differentiation can emerge without requiring large-scale deployments, particularly during early-stage system validation or in localized industrial settings where device populations are naturally constrained.

Moreover, industrial IoT deployments often evolve incrementally, beginning with a limited number of heterogeneous devices before scaling. The presented evaluation reflects this practical progression and demonstrates that the proposed SIoT framework remains effective under such conditions.

7.6. Limitations and Future Directions

While the proposed SIoT trust-evaluation framework demonstrates consistent and stable service selection under realistic execution constraints, several limitations should be acknowledged. First, the physical experimental evaluation involves a limited number of service-providing devices, resulting in a trust graph comprising 11 logical nodes and 22 edges. Although synthetic scalability analysis indicates stable PageRank convergence for larger graphs, the physical results primarily reflect small- to medium-scale industrial deployments and may not capture all dynamics present in large-scale systems.

Second, the experimental study relies on a selected set of pretrained object detection models and publicly available traffic surveillance datasets. Detection accuracy, failure characteristics, and execution behavior are therefore influenced by the intrinsic properties of these models and datasets. While the proposed SIoT framework is model- and dataset-agnostic, absolute performance values and failure rates may vary under alternative architectures, training regimes, or application domains.

Finally, failure handling in this work focuses on execution feasibility and performance degradation rather than on safety-critical fault tolerance or adversarial resilience.

Configurations that cannot complete inference are excluded from service selection, and repeated execution failures reduce trust scores over time. This approach supports robust service selection under resource constraints but does not replace dedicated safety mechanisms required in mission-critical industrial systems.

Future work will explore larger and more diverse device populations, adaptive trust-weight tuning, and dynamic model migration to further evaluate scalability and autonomy. Incorporating additional trust dimensions, such as availability history and security posture, and exploring cross-vendor acceleration backends represent promising directions for extending the framework toward more resilient industrial deployments.

8. Conclusions

This work presented a socially driven protocol for the Social Internet of Things (SIoT) that enables trust-aware service selection for distributed industrial computer vision workloads. The proposed framework integrates multi-dimensional performance metrics, including detection accuracy, communication latency, processing time, and device-level execution characteristics—within a graph-based trust model, allowing service providers to be selected based on observed behavior rather than static device descriptions.

Experimental evaluation across both homogeneous and heterogeneous environments shows that the proposed SIoT framework supports consistent and stable trust-based service selection within feasible operating regimes.

In homogeneous settings, where service providers exhibit comparable hardware capabilities, algorithmic accuracy dominates trust outcomes, and higher-performing detection pipelines achieve higher friendship scores. This behavior validates the adopted weighting strategy in environments where execution feasibility is uniform and resource constraints are minimal.

In heterogeneous environments comprising resource-diverse devices, trust computation reflects the combined influence of algorithm capability and device execution feasibility. Under high-resolution and high-frame-rate workloads, service providers with stronger processing capacity and stable execution profiles are preferentially selected, resulting in clear differentiation as hardware resources, network conditions, and workload intensity vary. Device–algorithm configurations that fail to execute under given constraints are excluded from service selection, allowing the framework to operate reliably within feasible operating regions.

The experimental results further indicate that network bandwidth primarily affects communication delay without significantly impacting detection correctness. Specifically, reducing available bandwidth from 100 Mbps to 10 Mbps increases round trip communication latency by approximately one order of magnitude, while detection accuracy remains largely invariant. In contrast, processing capability and execution stability emerge as decisive factors under demanding workloads. From a systems perspective, SIoT trust evaluation introduces limited computational overhead: trust-graph construction incurs a one-time or event-driven cost on the order of seconds, while PageRank-based friendship scoring is completed within tens of milliseconds and can be executed frequently without impacting end-to-end service latency.

The practical applicability of the proposed SIoT framework is conditioned on operating regimes where candidate device–algorithm configurations are feasible and detection performance remains within acceptable bounds for the target application. Experimental results show that service selection is fundamentally constrained by device capability and algorithm feasibility under given workload conditions. Within these bounded feasibility regions, the framework consistently enables stable and trust-aware service selection, but it does not eliminate limitations imposed by hardware resources or algorithm suitability.

Overall, the proposed SIoT protocol demonstrates the effectiveness of socially grounded trust computation for coordinating heterogeneous edge devices in industrial IoT environments. By jointly considering accuracy, efficiency, and runtime behavior, the framework supports principled and explainable service selection under dynamic conditions. Future work will explore scaling the framework to larger SIoT deployments, adaptive trust-weight tuning under evolving workloads, and the incorporation of additional trust attributes such as availability history and fault resilience to further enhance applicability in safety-critical industrial systems.

https://doi.org/10.3390/fi18020069

Author Contributions: Conceptualization, G.C.; Methodology, S.S. and G.C.; Software, G.C.; Validation, G.C. and S.S.; Formal Analysis, G.C. and S.S.; Investigation, G.C.; Writing—Original Draft Preparation, G.C.; Writing—Review and Editing, G.C., S.S. and S.B.M.; Supervision, S.S. and S.B.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The object detection models used in this study are standard pre-trained implementations provided by their respective publicly available frameworks, with canonical algorithm references and corresponding implementation links provided in References [48–50]. The video datasets employed for evaluation are publicly available traffic surveillance datasets commonly used in computer vision research. Configuration files, evaluation scripts, and detailed execution logs supporting the reported results can be made available from the corresponding author upon reasonable request.

Acknowledgments: The author G.C. respectfully acknowledges the spiritual guidance and inspiration of Sai Baba Magapu, which played a significant role in sustaining this research effort.

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

Abbreviations

The following abbreviations are used in this manuscript:

CV	Computer Vision
FPS	Frames Per Second
GPU	Graphics Processing Unit
IIoT	Industrial Internet of Things
IoT	Internet of Things
PR	PageRank
RTT	Round-Trip Time
SIoT	Social Internet of Things

References

1. Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge Computing: Vision and Challenges. *IEEE Internet Things J.* **2016**, *3*, 637–646. [CrossRef]
2. Satyanarayanan, M. The Emergence of Edge Computing. *Computer* **2017**, *50*, 30–39. [CrossRef]
3. Mao, Y.; You, C.; Zhang, J.; Huang, K.; Letaief, K.B. A Survey on Mobile Edge Computing: The Communication Perspective. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2322–2358. [CrossRef]
4. Wang, X.; Han, Y.; Leung, V.C.M.; Niyato, D.; Yan, X.; Chen, X. Convergence of Edge Computing and Deep Learning: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 869–904. [CrossRef]
5. Ananthanarayanan, G.; Bahl, P.; Bodík, P.; Chintalapudi, K.; Philipose, M.; Ravindranath, L.; Sinha, S. Real-Time Video Analytics: The Killer App for Edge Computing. *Computer* **2017**, *50*, 58–67. [CrossRef]
6. Khan, W.Z.; Ahmed, E.; Hakak, S.; Yaqoob, I.; Ahmed, A. Edge Computing: A Survey. *Future Gener. Comput. Syst.* **2019**, *97*, 219–235. [CrossRef]
7. Nitti, M.; Girau, R.; Atzori, L. Trustworthiness Management in the Social Internet of Things. *IEEE Trans. Knowl. Data Eng.* **2014**, *26*, 1253–1266. [CrossRef]
8. Yan, Z.; Zhang, P.; Vasilakos, A.V. A Survey on Trust Management for Internet of Things. *J. Netw. Comput. Appl.* **2014**, *42*, 120–134. [CrossRef]
9. Sicari, S.; Rizzardi, A.; Grieco, L.A.; Coen-Porisini, A. Security, Privacy and Trust in Internet of Things: The Road Ahead. *Comput. Netw.* **2015**, *76*, 146–164. [CrossRef]
10. Alam, S.; Zardari, S.; Noor, S.; Ahmed, S.; Mouratidis, H. Trust Management in Social Internet of Things (SIoT): A Survey. *IEEE Access* **2022**, *10*, 108924–108954. [CrossRef]
11. Lu, Y. Blockchain and the Related Issues: A Review of Current Research Topics. *J. Manag. Anal.* **2018**, *5*, 231–255. [CrossRef]
12. Atzori, L.; Iera, A.; Morabito, G.; Nitti, M. The Social Internet of Things (SIoT)—When Social Networks Meet the Internet of Things: Concept, Architecture and Network Characterization. *Comput. Netw.* **2012**, *56*, 3594–3608. [CrossRef]

13. Atzori, L.; Iera, A.; Morabito, G. From "Smart Objects" to "Social Objects": The Next Evolutionary Step of the Internet of Things. *IEEE Commun. Mag.* **2014**, *52*, 97–105. [CrossRef]
14. Hosseinzadeh, M.; Mohammadi, V.; Lansky, J.; Nulicek, V. Advancing the Social Internet of Things (SIoT): Challenges, Innovations, and Future Perspectives. *Mathematics* **2024**, *12*, 715. [CrossRef]
15. Roopa, M.S.; Pattar, S.; Buyya, R.; Venugopal, K.R.; Iyengar, S.S.; Patnaik, L.M. Social Internet of Things (SIoT): Foundations, Thrust Areas, Systematic Review and Future Directions. *Comput. Commun.* **2019**, *139*, 32–57. [CrossRef]
16. Demers, A.; Greene, D.; Hauser, C.; Irish, W.; Larson, J.; Shenker, S.; Sturgis, H.; Swinehart, D.; Terry, D. Epidemic Algorithms for Replicated Database Maintenance. In Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing (PODC 1987), Vancouver, BC, Canada, 10–12 August 1987; pp. 1–12. [CrossRef]
17. Jelasity, M.; Voulgaris, S.; Guerraoui, R.; Kermarrec, A.-M.; van Steen, M. Gossip-Based Peer Sampling. *ACM Trans. Comput. Syst.* **2007**, *25*, 1–36. [CrossRef]
18. gRPC Authors. gRPC: A High Performance, Open Source Universal RPC Framework. Available online: https://grpc.io (accessed on 10 December 2025).
19. Brin, S.; Page, L. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Comput. Netw. ISDN Syst.* **1998**, *30*, 107–117. [CrossRef]
20. Lodigiani, C.; Melchiori, M. A PageRank-Based Reputation Model for VGI Data. *Procedia Comput. Sci.* **2016**, *98*, 566–571. [CrossRef]
21. Abed, J. *Digital Strategies and Organizational Transformation*; World Scientific Publishing Company: Baltimore, MD, USA, 2023; Chapter 9, pp. 151–178. [CrossRef]
22. Shi, W.; Dustdar, S. The Promise of Edge Computing. *Computer* **2016**, *49*, 78–81. [CrossRef]
23. Satyanarayanan, M. Edge Computing for Situational Awareness. In Proceedings of the IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN 2017), Osaka, Japan, 12–14 June 2017; pp. 1–6. [CrossRef]
24. Subbarayappa, S.; Rao, K.R. Overview and Extensions of the High Efficiency Video Coding (HEVC) and Beyond (Versatile Video Coding). *Int. J. Emerg. Technol. Adv. Eng.* **2019**, *9*, 70–97.
25. Subbarayappa, S.; Rao, K.R. Video Quality Evaluation and Testing Verification of H.264, HEVC, VVC and EVC Video Compression Standards. *IOP Conf. Ser. Mater. Sci. Eng.* **2021**, *1045*, 012028. [CrossRef]
26. Li, H.; Zhao, F.; Xue, F.; Wang, J.; Liu, Y.; Chen, Y.; Wu, Q.; Tao, J.; Zhang, G.; Xi, D.; et al. Succulent-YOLO: Smart UAV-Assisted Succulent Farmland Monitoring with CLIP-Based YOLOv10 and Mamba Computer Vision. *Remote Sens.* **2025**, *17*, 2219. [CrossRef]
27. Zhao, F.; Xu, D.; Ren, Z.; Shao, X.; Wu, Q.; Liu, Y.; Wang, J.; Song, J.; Chen, Y.; Zhang, G.; et al. Mamba-based Super-Resolution and Semi-Supervised YOLOv10 for Freshwater Mussel Detection Using Acoustic Video Camera: A Case Study at Lake Izunuma, Japan. *Ecol. Inform.* **2025**, *90*, 103324. [CrossRef]
28. Kouicem, D.E.; Bouabdallah, A.; Lakhlef, H. Internet of Things Security: A Top-Down Survey. *Comput. Netw.* **2018**, *141*, 199–221. [CrossRef]
29. Roman, R.; Zhou, J.; Lopez, J. On the Features and Challenges of Security and Privacy in Distributed Internet of Things. *Comput. Netw.* **2013**, *57*, 2266–2279. [CrossRef]
30. Conti, M.; Dehghantanha, A.; Franke, K.; Watson, S. Internet of Things Security and Forensics: Challenges and Opportunities. *Future Gener. Comput. Syst.* **2018**, *78*, 544–546. [CrossRef]
31. Khan, A.R.; Awan, K.A.; Alruwaili, F.F.; Ara, A.; Song, H.; Saba, T. Trust-Enhanced Lightweight Security Framework for Resource-Constrained Intelligent IoT Systems. *IEEE Internet Things J.* **2025**, *12*, 10175–10182. [CrossRef]
32. Kang, J.; Yu, R.; Huang, X.; Wu, M.; Maharjan, S.; Xie, S.; Zhang, Y. Blockchain for Secure and Efficient Data Sharing in Vehicular Edge Computing and Networks. *IEEE Internet Things J.* **2019**, *6*, 4660–4670. [CrossRef]
33. Zhao, T.; Foo, E.; Tian, H. A Lightweight Blockchain-Based Trust Management Framework for Access Control in IoT. In *Secure and Trusted Cyber Physical Systems*; Pal, S., Jadidi, Z., Foo, E., Eds.; Smart Sensors, Measurement and Instrumentation, Vol. 43; Springer: Cham, Switzerland, 2022. [CrossRef]
34. Wang, L.; Li, Y.; Zuo, L. Trust Management for IoT Devices Based on Federated Learning and Blockchain. *J. Supercomput.* **2025**, *81*, 1. [CrossRef]
35. Huynh, T.T.; Nguyen, T.D.; Tan, H. A Survey on Security and Privacy Issues of Blockchain Technology. In Proceedings of the International Conference on System Science and Engineering (ICSSE 2019), Dong Hoi, Vietnam, 19–21 July 2019; pp. 362–367. [CrossRef]
36. Patil, D.A.; Shyamala, G. A Comprehensive Survey on Securing the Social Internet of Things: Protocols, Threat Mitigation, Technological Integrations, Tools, and Performance Metrics. *Sci. Rep.* **2025**, *15*, 40190. [CrossRef]
37. Li, L.; Ota, K.; Dong, M. Deep Learning for Smart Industry: Efficient Manufacture Inspection System with Fog Computing. *IEEE Trans. Ind. Inform.* **2018**, *14*, 4665–4673. [CrossRef]
38. Li, M.; Wang, W. Hybrid Zone: Bridging Acoustic and Wi-Fi for Enhanced Gesture Recognition. In Proceedings of the IEEE Conference on Computer Communications (INFOCOM 2024), Vancouver, BC, Canada, 20–23 May 2024; pp. 981–990. [CrossRef]

39. Um, T.-W.; Kim, J.; Lim, S.; Lee, G.M. Trust Management for Artificial Intelligence: A Standardization Perspective. *Appl. Sci.* **2022**, *12*, 6022. [CrossRef]
40. Chidambaram, G.; Subbarayappa, S.; Magapu, S.B. Designing a Socially Driven Protocol for the Social Internet of Things (SIoT) and Bridging Technical Design with Public Perception. In Proceedings of the IEEE 7th PhD Colloquium on Emerging Domain Innovation and Technology for Society (PhD EDITS 2025), Bangalore, India, 10–12 November 2025; pp. 1–2. Available online: https://ieeexplore.ieee.org/document/11288990 (accessed on 16 January 2026).
41. Knapp, M.L.; Vangelisti, A.L. *Interpersonal Communication and Human Relationships*, 6th ed.; Pearson: Boston, MA, USA, 2010.
42. Homans, G.C. Social Behavior as Exchange. *Am. J. Sociol.* **1958**, *63*, 597–606. [CrossRef]
43. Altman, I.; Taylor, D.A. *Social Penetration: The Development of Interpersonal Relationships*; Holt, Rinehart and Winston: New York, NY, USA, 1973.
44. Fiske, A.P. The Four Elementary Forms of Sociality. *Psychol. Rev.* **1992**, *99*, 689–723. [CrossRef] [PubMed]
45. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
46. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *Proceedings of the European Conference on Computer Vision (ECCV 2016)*, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37. [CrossRef]
47. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2020), Seattle, WA, USA, 14–19 June 2020; pp. 10778–10787. [CrossRef]
48. Ultralytics. YOLO Object Detection Framework. Available online: https://github.com/ultralytics/ultralytics (accessed on 15 January 2026).
49. TensorFlow. SSD Object Detection Models. Available online: https://github.com/tensorflow/models/tree/master/research/object_detection (accessed on 15 January 2026).
50. TensorFlow. EfficientDet-Lite0 Object Detection Model. Available online: https://tfhub.dev/tensorflow/efficientdet/lite0/detection/1 (accessed on 15 January 2026).
51. Liu, X.; Zheng, Z.; Xu, H.; Liang, Z.; Shi, G.; Zhang, C.; Li, K. Enabling Consistent Sensing Data Sharing Among IoT Edge Servers via Lightweight Consensus. *IEEE Trans. Comput.* **2025**, *74*, 2045–2057. [CrossRef]