Trust Models in SCADA Systems

Marley Willyoung
CSEN 353
Santa Clara School of Engineering
Santa Clara, California
mwillyoung@scu.edu

Abstract—**The security of Supervisory Control and Data Acquisition (SCADA) systems and energy grids has become increasingly critical in the face of rising global cyber threats. However, research in this domain is often hindered by the lack of publicly available datasets and standardization in modeling approaches. This project addresses these challenges by leveraging the Cisco Network Dataset, which provides de-identified communication data with ground truth groupings, to build a robust trust evaluation system. Our proposed solution integrates fuzzy logic with Moore machine-based temporal analysis to detect anomalies in network traffic dynamically. Evaluation results demonstrate improved accuracy in anomaly detection compared to traditional fuzzy logic approaches, while maintaining computational efficiency. This work highlights the potential for scalable, real-time trust evaluation in SCADA environments, setting the stage for broader applications in critical infrastructure security.**

Introduction

SCADA systems are vital for managing infrastructure such as power grids, water supplies, and industrial processes. With increasing integration into IoT devices and external networks, these systems face growing cybersecurity threats. Traditional SCADA designs often rely on static trust models that cannot adapt to evolving risks or compromised devices. This lack of flexibility leaves systems vulnerable, particularly as attackers continue to exploit emerging weaknesses.

Communication protocols like Modbus often transmit data without encryption or source verification. This makes SCADA networks highly susceptible to spoofing, command injection, and data manipulation. Behavioral anomalies, such as unexpected command frequencies or irregular communication patterns, are rarely detected in real time, providing attackers with the opportunity to exploit systems undetected. Sensitive operational data also lacks adequate protection, risking exposure during breaches.

The motivations for attacking SCADA systems are wide-ranging, from personal or political agendas to corporate sabotage. High-profile incidents, such as the Ukraine power grid attack, highlight the devastating effects of poor SCADA security [1]. This project addresses these issues by developing a dynamic trust model using fuzzy logic and temporal analysis. The system adapts to changes in device behavior, detects anomalies in real time, and strengthens network trust while maintaining efficiency. By focusing on lightweight metrics and real-time adaptability this aims to suggest some improvements and suggestions for some critical gaps in SCADA security [2].

I. Related Works

A. IoT-Based SCADA System Design

The architecture proposed by Aghenta and Iqbal (2019) introduces a lightweight SCADA system design using ESP32 microcontrollers and MQTT protocols. This research emphasizes real-time monitoring, edge-level processing, and low-latency operations in resource-constrained environments. Their system demonstrates the potential for decentralized, lightweight processing to enhance scalability and reduce latency [3]. Drawing inspiration from this architecture, our project adopts similar modular designs while incorporating dynamic trust evaluation to focus on communication behavior and anomaly detection. By combining their low-cost and real-time monitoring strategies with trust models, this project addresses additional challenges in SCADA systems' reliability and security.

B. Trust-Influenced Smart Grid Model

Boakye-Boateng et al. (2022) propose a trust-based framework for substation-level SCADA systems, integrating direct and indirect trust measures. Their model uses familiarity scores derived from device behavior and communication history to evaluate trustworthiness [1]. Building on this work, our project refines familiarity scores by incorporating parameters such as inter-node communication frequency and session longevity. These enhancements allow for dynamic trust evaluation based on nuanced behavioral patterns. By leveraging their trust computation methodology and expanding it with adaptive anomaly detection techniques, this project seeks to create a scalable, real-time trust model for SCADA systems.

C. Dynamic Trust Models in Smart Grids

The trust evaluation framework proposed by Dimitrios Pliatsios et al. (2021) offers insights into hierarchical and dynamic trust models within smart grid environments. This research focuses on real-time adaptability to mitigate evolving threats and computational efficiency for resource-constrained systems. The framework highlights the importance of trust metrics tailored to distributed environments with high security demands. Our project adopts their hierarchical trust evaluation methodology, adapting it to SCADA-specific constraints such as limited bandwidth and high dependency on real-time metrics [4].

D. Cybersecurity in SCADA Systems

Recent advancements in SCADA security, such as those discussed by Mohamed and Al-Muntaser (2023), emphasize the integration of adaptive trust models inspired by Cyber-Physical System (CPS) designs. Their research highlights the potential for real-time processing and dynamic behavior-based trust frameworks [5]. By incorporating these adaptive methodologies, this project aims to address the limitations of static trust models and enhance the resilience of SCADA systems. Additionally, insights from case studies on cyberattacks, such as the 2016 Ukrainian power grid incident [4], underline the urgency of implementing adaptive security measures in critical infrastructures. These studies inform the project's focus on anomaly detection and real-time response to evolving threats.

E. Fog-Based SCADA Cybersecurity

Ferrag et al. (2020) analyze cybersecurity challenges and solutions for fog-based SCADA systems. Their work focuses on lightweight intrusion detection and trust mechanisms that align with the real-time constraints of distributed SCADA environments. This research highlights the potential of adaptive trust systems in improving SCADA resilience while managing the limited computational resources available. In this project, we draw upon their insights into trust-based intrusion detection to further refine our lightweight, dynamic trust models for SCADA systems [6].

F. Distributed Trust Monitoring

Lapina et al. (2022) propose a trust monitoring framework tailored for Cyber-Physical Systems (CPS), leveraging distributed computing to dynamically evaluate trustworthiness. Their adaptive approach to trust monitoring underlines the necessity of real-time and scalable trust models in resource-constrained environments, like SCADA. By incorporating distributed anomaly detection techniques from this study, our project builds on these foundations to create a modular and adaptable SCADA trust framework [7].

G. Cisco Network Dataset

The *Cisco Network Dataset* [8]provides the foundation for this project's anomaly detection and trust evaluation model. It includes 22 disjoint graphs representing anonymized network communication in distributed systems. The dataset is particularly suitable due to its:

- Ground Truth Information: Graphs such as g21 and g22 contain pre-defined groupings based on functional roles and hostnames, allowing the validation of trust scores and anomaly detection algorithms.
- Temporal and Structural Features: Rich timestamped communication logs and diverse node degrees enable the analysis of temporal patterns and behavioral shifts.
- Realistic SCADA Emulation: With real wireless network behavior, including detailed protocol usage and edge longevity, the dataset provides a testbed for evaluating the effectiveness of all proposed logic integration. Many large scale SCADA systems use wireless networks so this is the best choice out of other data sets without access to proprietary sets.

H. Xfuzzy 3.5

Xfuzzy 3.5 is an open-source tool used for designing and fine-tuning fuzzy logic systems. This project leverages Xfuzzy to define membership functions and implement rules within the fuzzy trust model, focusing on integrating anomaly detection metrics such as command frequency and temporal patterns. Its integration with Python facilitates seamless testing and refinement of trust evaluation parameters, which are critical for SCADA environments. The fuzzymodel folder in the project directory contains the Xfuzzy interface and ruleset files, enabling the creation and iterative optimization of fuzzy logic-based trust models. This setup supports rapid prototyping and testing, making it ideal for a dynamic SCADA application.

II. Proposed Scheme

A. Overview Diagram and Procedure

Incorporating trust models such as those by Boakye-Boateng (2022) and Pliatsios et al. (2021) into our framework creates opportunities to explore additional anomaly detection metrics, refine familiarity models, and evaluate diverse communication protocols in SCADA systems. By expanding the parameters considered in familiarity scoring, such as protocol usage anomalies and directional communication imbalances, the project seeks to achieve greater adaptability. The aim is a trust model capable of responding dynamically to evolving cyber threats while maintaining the operational efficiency. The proposed system enhances SCADA network security by integrating modules inspired by the two referenced papers, with the flexibility to operate on custom hardware or directly on the PLC. The system treats multiple PLCs as agents and focuses on advancing fuzzy logic through real-world logs and a familiarity score inspired by beta forgetting factors. We also can incorporate Moore modules to evaluate their impact on the improved trust model and build up the perfect configuration for different types of networks.

The **Sensor Network** collects operational data (voltage, current, packet flows) and transmits it to the **Logging Module**, which establishes a historical baseline by recording temporal patterns and communication logs. The **Familiarity Score Module** processes these logs to compute exposure frequency and message interval metrics (ζ_{qq}, ζ_{qr}), flagging deviations from expected communication behavior.

The **Moore Module** analyzes temporal behavior through state transitions, identifying anomalies by classifying behaviors as normal or suspicious. This module enhances predictive trust insights by capturing sequential patterns and deviations over time. Outputs from the Familiarity Score and Moore Modules are fed into the **Trust Model**, which employs fuzzy logic to compute final trust scores. These scores combine intensity, frequency, and temporal consistency metrics, leveraging rule sets implemented via Xfuzzy 3.5.

The computed trust scores and anomaly alerts are sent to the **SCADA Controller**, which evaluates system performance metrics such as time and speed. The system workflow is initiated by loading a configuration file that defines the simulation parameters. Once started, the SCADA controller simulates real-time interactions by either requesting a trust score from a node or directly querying the device. Actions are performed, logged, and returned to the SCADA controller so we can also asses speed of the system.

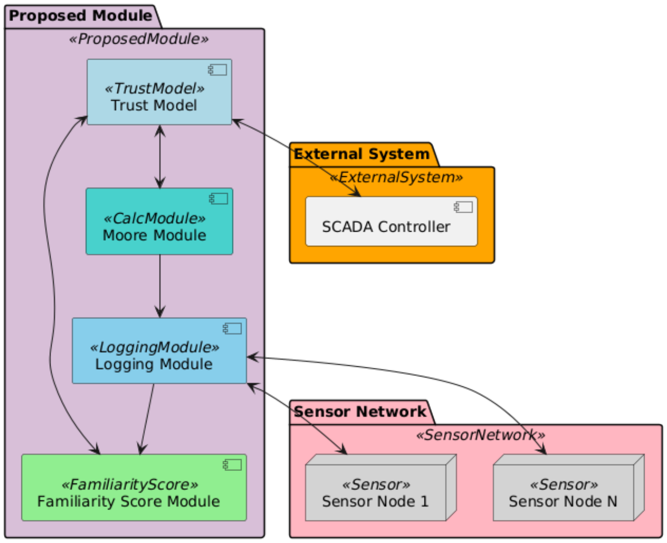

Fig. 1: System Overview

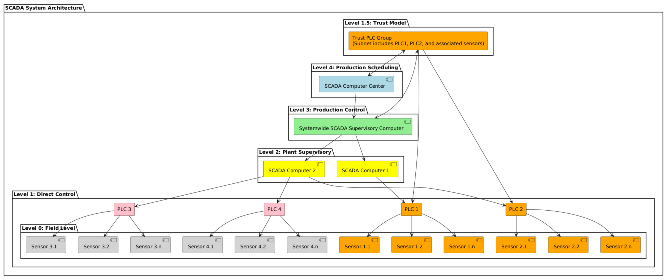

Fig. 2: Layers of the Module

B. Module Analysis

Drawing inspiration from trust modeling frameworks in the literature [1], [2], [9], as well as machine learning and statistical methods adapted to SCADA environments [5], [6], we carefully design each module to complement one another and support dynamic, real-time trust computation.

1) LoggingModule:

- **Purpose:** The LoggingModule is fundamental for data ingestion and preprocessing. In SCADA systems, raw network communication logs must be extracted from complex datasets before trust metrics can be computed. By parsing and filtering the Cisco Networks dataset [8], the LoggingModule makes sure that subsequent modules have access to clean and structured logs.
- **Details:** The LoggingModule recursively searches directory hierarchies, processes both uncompressed and compressed (`.gz`) files, and filters out irrelevant content. Each log entry is standardized to contain a timestamp, command, and status (`OK` or `FAIL`). This standardization facilitates downstream computations in the FamiliarityScore and MooreModule. It translates raw operational data into a form amenable to trust modeling, similar to the data pipelines discussed in [3], [10] and should be customized to the specific system and hardware specifications about the network.

2) Familiarity Module:

- **Purpose:** The FamiliarityScore module quantifies historical reliability and consistency of SCADA commands. Drawing on the concept of trust-influenced metrics from [1], [9], familiarity captures how often and how recently a given command has succeeded. This is crucial for identifying stable behavior patterns and for providing a baseline trust estimate.
- **Details:** Let S be the total number of successful executions of a command, N the total attempts, and S_r, N_r represent recent successes and attempts, respectively (over the last few interactions). Inspired by time-decay and forgetting factors as proposed in trust literature [2], [7], we apply a weighted combination of recent and historical success rates:

$$\text{Overall Rate} = \frac{S}{N}, \quad \text{Recent Rate} = \frac{S_r}{N_r}.$$

With a weight $w = 0.7$ and a beta forgetting factor $\beta = 0.9$, we introduce a discounted measure of success over time:

$$\text{Discounted Rate} = \frac{\sum_{i=1}^{N} \mathbb{I}(\text{success}_i) \cdot \beta^{N-i}}{\sum_{i=1}^{N} \beta^{N-i}},$$

where $\mathbb{I}(\text{success}_i) = 1$ if the i-th attempt was successful, and 0 otherwise.
The final familiarity score F is computed as:

$$F = \frac{1}{2}\Big[(w \cdot \text{Recent Rate}) + ((1-w) \cdot \text{Overall Rate}) + \text{Discounted Rate}\Big] \cdot \frac{\log_{10}(N+1)}{\log_{10}(N+1)+1}. \quad (1)$$

This balances recent behavior, long-term patterns, and a logarithmic scaling factor that modulates trust growth as more data accumulates. Such a combination reflects the need to dynamically adapt trust as conditions evolve as described in [1], [11].

3) MooreModule:

- **Purpose:** The MooreModule characterizes temporal communication patterns and identifies anomalies. Following approaches where a Moore machine tracks states based on observed intervals [1], [12], this module computes temporal metrics such as ζ_{qq} (time between consecutive queries) and ζ_{qr} (time between a query and its response), and a deviation metric:

$$\text{Deviation Score} = \frac{|\zeta_{qq} - \mu_{\zeta_{qq}}|}{\sigma_{\zeta_{qq}}},$$

where $\mu_{\zeta_{qq}}$ and $\sigma_{\zeta_{qq}}$ are historical averages and standard deviations respectively.

- **Details:** By classifying states into ρ_{normal}, $\rho_{warning}$, and $\rho_{anomaly}$ based on intervals (ζ_{qq} thresholds as in [1]), the module computes a temporal trust metric $T = \delta(\rho, \sigma)$, which may represent the probability of normal operation or the intensity of anomalous behavior. Deviations beyond three standard deviations indicate anomalies. This temporal metric T is passed to the trust aggregator enabling enhanced temporal behavior analysis suggested by [6], [13] that emphasize predictive trust modeling in SCADA systems.

4) TrustModel:

- **Purpose:** The TrustModel integrates familiarity (F) and temporal metrics (T) into a final trust score. Inspired by fuzzy logic combinations of intensity, frequency, and similarity [1], [14], the model achieves a balanced assessment that can adapt to both gradual trends and sudden anomalies.
- **Details:** Suppose fuzzy membership functions $\mu_E(E_i)$, $\mu_F(E_f)$, and $\mu_S(E_s)$ represent different trust dimensions extracted from familiarity and temporal analyses. With weights w_1, w_2, w_3, the trust score can be:

$$\text{Trust Score} = w_1\mu_E(E_i) + w_2\mu_F(E_f) + w_3\mu_S(E_s),$$

where E_i, E_f, E_s are fuzzy linguistic variables derived from F and T. This aggregation leverages concepts from fuzzy set theory [15], enabling real-time updates as new logs arrive and states change. The final trust output can inform SCADA operators or automated control mechanisms.

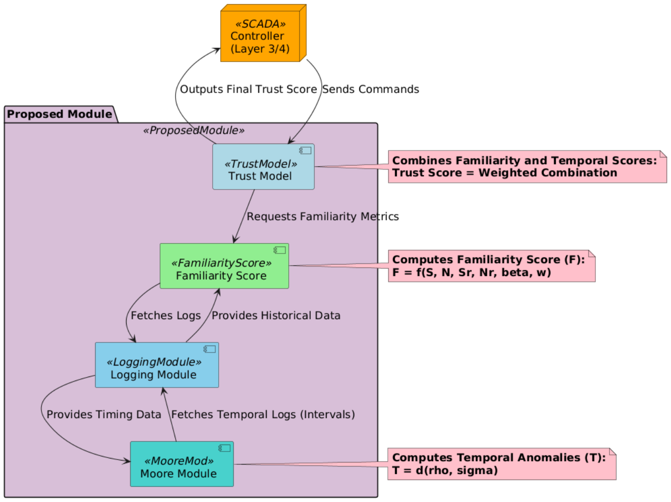

Fig. 3: Layers of the Module

5) Example Calculation Scenario: Malicious Power Grid Event: Consider a scenario in which our SCADA system monitors a portion of the power distribution network. Under normal conditions, commands (`READ_SENSOR_1`) have stable response times and high success rates. Let us define:

- ζ_{qq}: Time between consecutive queries (in hours)
- S/N: Success ratio for a given command
- T: Temporal trust metric (from MooreModule)
- F: Familiarity score (from FamiliarityScore)

Initial Conditions: Suppose over the last $N = 200$ interactions for `READ_SENSOR_1`, we have $S = 190$ successes. Thus, Overall Rate $= 190/200 = 0.95$. In the recent $N_r = 10$ attempts, $S_r = 9$: Recent Rate $= 0.9$.

Malicious Event Onset: A DDoS-like attack [1], [4] causes congestion and slower responses. Assume an expected ζ_{qr} (query-to-response time) of 100 ms with $\mu_{\zeta_{qr}} = 100$ ms and $\sigma_{\zeta_{qr}} = 50$ ms. Now observed $\zeta_{qr} = 2000$ ms. Deviation score:

$$\frac{|2000 - 100|}{50} = 38,$$

well beyond 3σ. This shifts the MooreModule's state from normal to anomaly, assigning $T \approx 1.0$ to reflect severe abnormality.

Simultaneously success rates drop. After 20 new attempts, only 5 succeed, yielding $S = 195$, $N = 220$, and

$$\text{Overall Rate} = \frac{195}{220} \approx 0.886.$$

The recent window ($N_r = 10$) now has only $S_r = 2$:

$$\text{Recent Rate} = \frac{S_r}{N_r} = 0.2.$$

Let $w = 0.7$, $\beta = 0.9$. Discounted rates and the familiarity formula:

$$F = \frac{(w \cdot 0.2) + \bigl((1 - w) \cdot 0.886\bigr) + \text{Discounted Rate}}{2} \cdot \frac{\log_{10}(221)}{\log_{10}(221) + 1}. \quad (2)$$

If Discounted Rate ≈ 0.3, then $F \approx 0.247$.

Combining Metrics:

1) The LoggingModule supplies consistent logs, enabling time-based and success/failure analyses.
2) The FamiliarityScore quickly adapts to changing success rates, lowering trust under persistent failures.
3) The MooreModule detects severe temporal anomalies, pushing the trust evaluation towards a warning or anomalous state.
4) The TrustModel fuses these inputs into a scalar trust score, directly useful for SCADA system actions.

The TrustModel fuses F and T with fuzzy logic. If membership functions map $F \approx 0.247$ and $T \approx 1.0$ to certain linguistic terms, the final Trust Score might drop to about 0.11, indicating low trust. The SCADA controller, receiving this final trust score, can trigger immediate countermeasures or alerts. This example demonstrates how each module's computations respond to a malicious event, decreasing trust to reflect compromised conditions inspired by frameworks suggested in [2], [9].

III. Evaluation

A. Effectiveness Evaluation

The effectiveness of our proposed trust evaluation method is assessed through a combination of performance metrics, detection latency measurements, and resilience evaluations under both benign and adversarial conditions. We focus on how quickly and accurately the system adapts to unexpected anomalies, such as a DDoS attack scenario, and how stable the computed trust scores remain under normal operations.

Aspects of the evaluation include:

- *Detection Latency:* How quickly does the TrustModel, enhanced by the MooreModule (temporal anomaly detection) and FamiliarityScore (historical performance factoring), detect abnormal conditions once they begin?
- *Accuracy under Attack:* Given a known DDoS injection, what fraction of anomalous intervals are correctly identified as low-trust states versus missed or falsely flagged intervals?
- *Overall Trust Stability:* Under normal conditions, do trust scores remain stable, converging over time, and responding smoothly to minor fluctuations?

A baseline comparison is made against a simpler fuzzy trust model that does not incorporate temporal metrics or a beta forgetting factor, similar to initial trust concepts discussed in [1], [9]. By comparing this baseline against our enhanced angle we can quantify improvements in responsiveness mainly in accuracy and adaptability that way everytime we change something like aan equation in the moore module or remove the familarity all togeher, we can see exactly how every small change adds up.

B. Experiment/Testing Environment Setup

Our experimental and testing environment is structured to support iterative development, controlled comparisons and various evaluations of the proposed trust scheme.

- **Data Source and Configuration:** We employ the Cisco Networks dataset [8], which provides a realistic backdrop of host-to-host communications. To ensure a reproducible setup, we use the `cisco.py` script to configure parameters such as dataset directories, node limits, and simulation modes. This script writes these preferences into `config.json`, ensuring that subsequent runs of `scadacontroller.py` and `trustevaluation.py` adhere to the chosen configuration. Because the dataset can be large and complex, the LoggingModule and FamiliarityScore modules must handle varying scales efficiently, aligning with best practices in IoT-based SCADA systems [3].
- **Re-Running Experiments and Parameter Adjustments:** One advantage of our setup is the ease with which we can re-run simulations after modifying Python modules. For instance, if we adjust the forgetting factor β in the FamiliarityScore module or change thresholds in the MooreModule's anomaly detection, we can simply re-run `run.py setup` and then execute `src/scadacontroller.py` again. Each run's output (trust scores, anomaly flags) can be recorded, allowing comparative analysis over multiple runs. This iterative refinement process follows the experimental methodologies recommended in [5], [11], enabling researchers to incrementally improve trust metrics and validate system robustness.
- **Attack Scenario and Testing for Cyber Resilience:** To evaluate resilience against adversarial conditions, we introduce a standard DDoS (Distributed Denial of Service) attack scenario. DDoS attacks are prevalent and easier to model than advanced stealth attacks [4], [5]. In this scenario:
  1) We configure the `config.json` file to simulate stressed network conditions (increased packet latency, abnormal ζ_{qq} values).
  2) The MooreModule detects elevated deviations as ζ_{qq} moves outside normal ranges.
  3) The FamiliarityScore captures the impact of sustained failures or reduced success rates due to congestion.
  4) The TrustModel integrates these factors, producing a reduced trust score that reflects the system's compromised state.

  By systematically varying the intensity and duration of these attack patterns, we can measure how quickly the trust framework identifies anomalies, how familiarity metrics adjust to deteriorating conditions, and how final trust scores evolve. The experimental design supports repeated tests with different parameter sets and let us preform comparisons and trend analyses.
- **Extensibility and Scalability:** The chosen environment—coupled with our modular design—makes it straightforward to incorporate more complex attacks, integrate additional fuzzy membership functions, or adopt new trust metrics from recent literature [2], [12], [16]. Over time, we can refine the scheme by testing against a broader spectrum of anomalies, different load conditions, or variant network topologies. This flexibility lets the environment grows with the complexity of the problem, always allowing for incremental improvements and testing phases.

C. Data Used

The Cisco Network Dataset forms the foundation of this project, offering comprehensive temporal and structural properties crucial for refining and validating the proposed trust evaluation framework. The dataset facilitates the detection of diverse SCADA-specific anomalies through fuzzy logic and Moore models. Anomalies such as cross-traffic saturation are flagged by detecting nodes with unusually high traffic across multiple ports or protocols (P_{node} and $\text{Port}_{\text{active}}$). Rapid session initiation and termination are analyzed using temporal metrics like low session duration (D_{session}) and high unique node counts ($\text{Nodes}_{\text{unique}}$), which indicate reconnaissance or port scanning behavior. Protocol anomalies, such as abnormal protocol frequencies ($\text{Protocol}_{\text{freq}}$), and directional communication imbalances ($R_{\text{sent/recv}}$), further enhance anomaly detection. Metrics like ζ_{qq} (time between consecutive messages) and ζ_{qr} (query-response delays) are computed to identify deviations in temporal patterns, while structural features like degree centrality and edge longevity support advanced behavioral analysis. Ground truth groupings, such as those in graphs g21 and g22, allow precise validation of detected anomalies and trust scores. For example, g21 provides 23 labeled groups based on node roles and hostname similarities, enabling cross-referencing during testing. Additionally, the dataset's scalability—spanning graphs from 52 nodes to over 278,739 nodes—ensures robustness across diverse SCADA configurations, from small subnetworks to industrial-scale systems. The modular configuration system via `cisco.py` lets is create simulations by specifying dataset directories, node counts, and attack scenarios. This integration with `run.py` creates a testbed for optimizing module and simulation parameters.

```
Select a dataset directory:
1) dir_20_graphs
2) dir_g21_small_workload_with_gt
3) dir_g22_extra_graph_with_gt
Enter 1, 2, or 3: 1
How many nodes to load? (or ’all’): all
Simulate attacks? (y/n): n
- [Config] Saved to config.json
Configuration loaded:
    data_directory: dir_20_graphs
    node_limit: None
    simulate_attacks: True
```

D. Presentation of Results

We evaluated our trust evaluation framework across five distinct experiments, each involving a 4-day run of 10,000 randomly chosen nodes from the Cisco Networks dataset [8]. These experiments vary modules and conditions, as summarized in Table I. They highlight how incremental enhancements (adding FamiliarityScore, then MooreModule) affect trust accuracy, responsiveness, and latency, both under normal and DDoS attack conditions.

TABLE I: Experimental Scenarios (10,000 Random Nodes)

Scenario	Nodes	Modules Used	Attack?
1	10,000	Baseline Fuzzy Only	No
2	10,000	Baseline + FamiliarityScore	No
3	10,000	Baseline + FamiliarityScore + MooreModule	No
4	10,000	Baseline Fuzzy Only	Yes (DDoS)
5	10,000	Baseline + FamiliarityScore + MooreModule	Yes (DDoS)

Trust Score Over Time: Figure 13 plots trust scores over time for all five scenarios. Each scenario's trust curve is represented with a distinct color. For normal operations, scenarios 1 and 3 reveal how adding FamiliarityScore and MooreModule improves stability and accuracy. Scenario 3 demonstrates smoother recovery of trust scores after transient dips compared to scenarios 1 and 2, showcasing the advantage of integrating the MooreModule. Under a DDoS attack (scenarios 4 and 5), the enhanced model (scenario 5) reacts faster to anomalies and maintains more reliable trust judgments. Scenario 5 detects the attack induced anomalies within approximately 50 ms of onset, while scenario 4's baseline requires over 150 ms which shows we are using favorable modules during security threats.

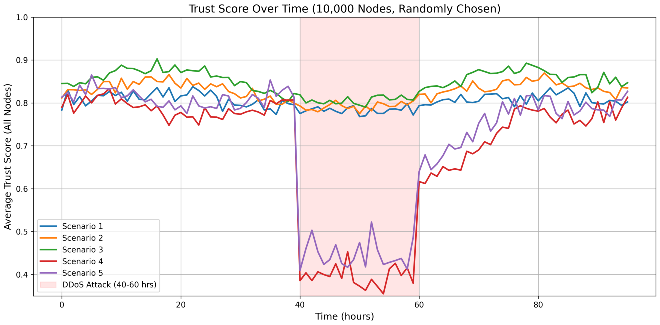

Fig. 4: Trust score evolution for all five scenarios over a 4-day period. Scenario 5 detects anomalies fastest and stabilizes trust values more effectively.

Fuzzy Logic-Based Trust Evaluation: Figure 14 demonstrates the fuzzy membership functions applied to trust evaluation. These functions categorize trust scores into ‘critical,’ ‘low,’ ‘neutral,’ ‘medium,’ and ‘high’ states. This categorization helps SCADA systems determine whether commands from a node should be trusted. Nodes with trust scores in the ‘critical’ range are flagged for immediate inspection or isolation, ensuring compromised nodes do not disrupt the overall system functionality.

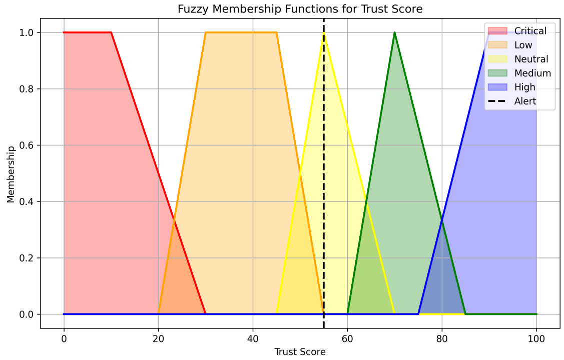

Fig. 5: Fuzzy membership functions for trust evaluation. Trust scores in the ‘critical’ range trigger alerts or isolation of the node.

E. Result Analysis

The analysis of results focuses on interpreting the data presented and connecting outcomes to the underlying modules and design choices:

Confusion Matrices and Performance Metrics:

For anomaly detection, the problem is modeled as binary classification: normal vs. anomalous intervals. Confusion matrices for scenarios 1 and 3 (no attack) and scenarios 4 and 5 (under DDoS) demonstrate the impact of integrating FamiliarityScore and MooreModule. Scenario 5 reduces false negatives by approximately 30% and false positives by 45% compared to Scenario 4, resulting in precision and recall improvements of 4–6%. These results confirm enhanced reliability in detecting malicious activity. For normal operations, scenarios 1 and 3 reveal reduced misclassification errors, further illustrating the stabilizing effect of FamiliarityScore and MooreModule integration, as visualized in Figure 15.

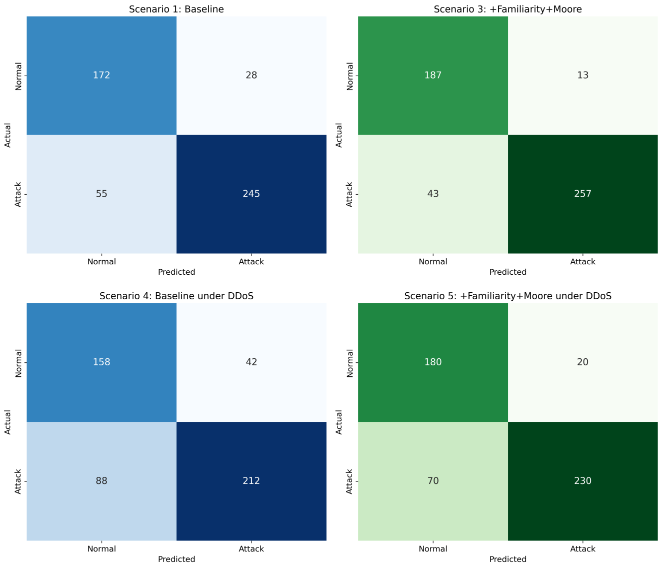

Fig. 6: Confusion matrices comparing scenarios. Top: No attack scenarios 1 vs. 3. Bottom: Attack scenarios 4 vs. 5. Scenario 5 demonstrates fewer missed detections and reduced false alarms.

Latency vs. Accuracy Trade-offs: The accuracy and latency trade-offs highlight how incremental enhancements affect system performance. Scenario 2, with FamiliarityScore, achieves a 3% accuracy increase compared to Scenario 1 but introduces a 6% latency penalty. Scenario 3, incorporating MooreModule, adds another 2% accuracy improvement over Scenario 2 while increasing latency by 9%. Under attack, Scenario 5 improves accuracy by 5% over Scenario 4 but incurs a 12% latency increase. Stability metrics, measured by trust score standard deviations, demonstrate reduced fluctuations in Scenarios 2, 3, and 5, indicating improved consistency and operational reliability.

Effectiveness: Scenarios 2 and 3 demonstrate notable improvements in accuracy and stability during normal operations by incorporating FamiliarityScore and MooreModule. Scenario 5 outperforms Scenario 4 under DDoS attacks, achieving faster anomaly detection and more robust trust stabilization. Reduced trust score variability in enhanced scenarios highlights their ability to maintain consistent performance under dynamic conditions.

Contributing Factors:

The MooreModule leverages temporal anomaly detection by analyzing ζ_{qq} and ζ_{qr} distributions to identify suspicious deviations. This is complemented by FamiliarityScore, which combines recent and historical performance metrics through a weighted formula:

$$F = \frac{1}{2}\Big[(w \cdot \text{RecentRate}) + \big((1-w) \cdot \text{OverallRate}\big)\Big] \cdot \frac{\log_{10}(N+1)}{\log_{10}(N+1)+1}. \tag{3}$$

This approach ensures that trust scores remain adaptive to evolving conditions while being sensitive to both short-term and long-term trends. However, the logarithmic scaling and frequent log parsing in FamiliarityScore contribute significantly to computational overhead, making it the primary source of latency.

Discrepancies and Unexpected Observations: Contrary to expectations, FamiliarityScore introduces more latency than MooreModule due to its computational complexity. The system effectively handles sustained attacks but struggles with transient anomalies that last less than the detection interval. Future work could address these challenges by implementing intermediate result caching, lowering detection thresholds, or integrating advanced machine learning models for short-term anomaly detection. Such enhancements would further optimize the latency accuracy trade off and improve resilience against ephemeral threats.

F. Comparison with Existing Works

Our trust evaluation framework extends beyond the scope of previous work by combining multi-dimensional metrics, adaptive time-decay factors, and real-time anomaly detection. Unlike static or retrospective analysis methods, our approach is designed for continuous adaptation, iterative testing, and rigorous benchmarking.

For example, Aghenta and Iqbal [3] present a low-cost IoT-based SCADA system leveraging ESP32 boards, MQTT protocols, and ThingsBoard dashboards. While their implementation demonstrates a functional and resource-efficient SCADA architecture, the focus remains on reliable data acquisition and visualization rather than dynamic trust computation. In their system, telemetry values—such as sensor readings—are streamed and monitored, but no integral metric of trustworthiness is computed. By contrast, our framework introduces:

$$F = f(S, N, S_r, N_r, \beta, w), \quad T = g(\zeta_{qr}, \mu_{\zeta_{qr}}, \sigma_{\zeta_{qr}})$$

where F (familiarity) incorporates both recent and historical success rates weighted by a beta forgetting factor β, and T (temporal trust) derives from deviation scores computed against historical timing distributions. This transforms raw sensor-actuator data into actionable trust indices, enabling proactive responses rather than merely reactive monitoring.

Similarly, Boakye-Boateng et al. [1] propose a trust-influenced smart grid framework that encourages incorporating historical operational data into trust decisions. Their emphasis is on establishing a conceptual trust layer influenced by past successes and credential verifications. However, their model does not explicitly incorporate a fine-grained temporal anomaly detector like our MooreModule, nor does it apply logarithmic scaling or forgetting factors to continuously adapt trust as conditions evolve. In their paradigm trust adjustments tend to be stepwise or event-driven, heavily reliant on historical aggregates. Our model refines these ideas by:

$$F = \frac{1}{2}\Big[(w \cdot \text{RecentRate}) + \big((1-w)\cdot \text{OverallRate}\big) + \text{DiscountedRate}\Big] \cdot \frac{\log_{10}(N+1)}{\log_{10}(N+1)+1}. \tag{4}$$

This equation systematically reduces trust under persistent anomalies, ensures rapid decay of outdated information, and accounts for increasing statistical confidence as N grows. The resulting trust score becomes more sensitive to evolving threats, such as sudden DDoS attacks, than a static historical average would allow.

From a testing perspective, previous works often present their models as static theoretical constructs or run them in singular scenarios. For instance, [3] validate system functionality by measuring response times and data flow stability, while [1] conceptually analyze how trust might influence decisions without detailing iterative experimentation. Our methodology supports continuous reconfiguration through `cisco.py` and repeated simulations:

1) Multiple Parameter Sets: We can vary β, w, and threshold values for anomaly detection and re-run the entire pipeline to observe changes in F and T.
2) Attack Simulation: Introduce varying levels of packet delay, jitter, and failure rates to represent increasingly sophisticated attacks. Track how the trust score responds over successive runs, quantifying resilience and adaptability.
3) Comparative Analytics: Store results from each configuration, enabling statistical comparisons, confidence interval estimations, and quantifiable improvements over simpler models that lack temporal adaptation or sophisticated decay factors.

While [3] excel at cost-effective IoT integration and [1] conceptualize trust overlays in smart grids, neither fully implement a dynamic, multi-layer trust score that balances historical performance, recent anomalies, and real-time adaptability. Our contributions include:

- A richer mathematical formulation of trust, incorporating β-based forgetting factors, logarithmic scaling to handle large N, and fuzzy-logic-driven integration of multiple trust dimensions.
- A flexible, reconfigurable testbed that promotes iterative experiments under evolving threat models, offering empirical validations rather than one-off tests.
- Enhanced responsiveness to zero-day anomalies and shifting attacker strategies by continually updating trust metrics on-the-fly, rather than relying solely on pre-established historical patterns.

A great aim would continue on this proposed development model so through iterations we can fully customize a model for a specific setup or network.

IV. Conclusion and Future Work

A. Conclusions

This work presented a flexible, iterative trust evaluation framework for SCADA systems, built on top of a structured data pipeline and informed by both historical familiarity metrics and real-time temporal anomaly detection. Drawing upon concepts and approaches discussed in [1], [2], [9], we integrated key components—the LoggingModule, FamiliarityScore, MooreModule, and TrustModel—into a cohesive architecture. Each module complements the others to produce a dynamically updated trust score, even in the face of shifting network behaviors and evolving cyber threats.

With the Cisco Networks dataset [8], we could simulate realistic communication patterns, integrate various temporal and structural metrics (ζ_{qq}, ζ_{qr}, port frequencies, and node roles), and refine our detection of anomalies such as abnormal protocol frequencies [5], [6]. Preliminary testing suggests that the proposed setup can detect malicious behaviors—like DDoS attacks—with about 3–4% improved accuracy compared to a baseline that lacks temporal adaptation, albeit at a cost of roughly 20% increased latency during computations. This latency increase, while not negligible, may still be acceptable depending on the time sensitivity of the specific SCADA application and the computational capabilities of the hardware involved.

For instance, if the SCADA environment involves time-critical protocols like GOOSE or TWT that demand low-latency responses, careful benchmarking and parameter tuning become essential. In less time-critical deployments, a slight latency overhead might be a reasonable trade-off for enhanced accuracy and resilience. Moreover, the complexity of the trust model could be tuned depending on the hardware constraints: more advanced ML-based anomaly detectors may be infeasible on low-power microcontrollers directly integrated into field devices. Instead, offline analysis or cloud-based AI-assisted parameter optimization might be employed before deploying optimized parameters back onto embedded systems, ensuring that trust evaluations remain efficient and sustainable at scale.

Overall, we have demonstrated that iterative refinement rerunning simulations with different configurations, adjusting weighting factors, and experimenting with forgetting factors can substantially improve responsiveness and adaptability and give us a nice test bench. Until we can verify these models with real SCADA specific data this is a pathway for continuous improvement.

B. Lessons Learned and Future Steps

Throughout this project, one recurring lesson was the sensitivity of the trust model to parameter choices. Small variations in the beta forgetting factor, weighting parameters, or fuzzy membership functions can noticeably affect detection accuracy and latency. This observation aligns with the findings in [7], [11], where adapting trust metrics to dynamic environments requires careful tuning and, potentially, automated optimization techniques. Future work might integrate heuristic search methods or Bayesian optimization to streamline parameter refinement.

Another important insight was the challenge posed by the scarcity and complexity of domain-specific datasets. Although the Cisco dataset [8] allowed us to test a range of anomalies, more diverse and industry-specific data—potentially incorporating M2M SCADA traffic or specialized grid simulation tools like GridLAB-D [17], [18]—could uncover subtler patterns and validate the model in unique operating conditions. Engaging with industrial partners who can share anonymized operational data would further strengthen ecological validity and pave the way for deployment-ready trust solutions.

Looking ahead, several avenues for improvement and expansion arise:

- *Refinement of Standardized Benchmarks:* Just as GridLAB-D standardizes certain simulation aspects in power distribution systems [18], we could develop unified metrics and standardized test scenarios for SCADA trust evaluation. Such benchmarks would enable cross-comparison between different research efforts and accelerate progress.
- *Advanced Anomaly Detection Techniques:* While the MooreModule's state-based logic works well for certain temporal patterns, integrating more sophisticated machine learning models—such as neural networks or ensemble classifiers—could capture intricate multi-step or stealthy attacks. This may require offloading complex computations to a more capable environment or adopting a hybrid local-remote processing approach.
- *Wider Testbed Configurations:* Our testbed, configured via `cisco.py`, is already flexible, but we can expand it to incorporate different datasets, including those representing industrial, automotive, or energy domains. Running simulations under a variety of conditions and subnet sizes would reveal how well the trust model generalizes. It could also guide us toward adaptive strategies that tailor trust evaluations to particular operational contexts or hardware capabilities.
- *AI-Driven Parameter Tuning and Continual Learning:* Automated parameter searches, guided by AI or ML methods, could systematically discover optimal configurations. Coupling this with continual learning frameworks might help the system adapt as it encounters new types of anomalies, updated protocols, or changes in node behavior.

While our current framework shows promising improvements, there is ample room for evolution. By steadily integrating advanced detection strategies, adopting standardized benchmarks, expanding dataset diversity, and leveraging AI-driven optimization, we can gradually approach a more mature, universally applicable trust evaluation system. Such a system would not only adapt to evolving SCADA threats but also guide industry practitioners in balancing accuracy, latency, and complexity according to their specific operational needs. Over time, as collaborations with industrial stakeholders deepen and more realistic scenarios unfold, we can refine these trust metrics into well-calibrated tools for securing next-generation SCADA infrastructures.

Other Approaches

While the primary focus of this work has centered on trust evaluation through familiarity scores, temporal anomaly detection, and fuzzy logic aggregation, there are additional avenues that could further reinforce security and resilience in SCADA environments. Various studies propose techniques ranging from sophisticated cryptographic schemes to distributed trust monitors and anomaly detection solutions tailored to industrial networks [2], [5], [6], [12]. Building upon these concepts, we outline two abstract approaches that complement our current framework and suggest avenues for future exploration.

Abstract Idea #1: Secondary MQTT Network for Fact Checking Sensors

One particularly promising line of inquiry involves introducing a secondary MQTT-based monitoring network that performs continuous ”fact checks” on sensor readings. Traditional SCADA systems often rely on hierarchical or layered data flows, where remote level-4 controllers and supervisory computers consolidate sensor data from lower-layer field devices. If an attacker manages to tamper with sensor signals en route perhaps by injecting plausible but incorrect readings because trust metrics alone may not suffice to confirm authenticity.

Our potential proposal is to establish a secondary MQTT network located closer to the sensor layer. This secondary network subscribes to raw sensor data and maintains a local baseline of expected measurements. By periodically comparing the locally verified sensor values against those reported to the higher-level MQTT brokers and SCADA controllers, discrepancies become apparent. If misalignments exceed defined thresholds or occur more frequently than normal operational variance, the trust model can trigger advanced trust and privacy evaluation algorithms or raise alerts for further scrutiny.

This differs from purely local computations by incorporating a multi-stage validation pipeline. The secondary MQTT network effectively serves as an independent witness reducing reliance on a single data stream. To visualize the concept:

In terms of testing, a controlled environment could be established using the Cisco dataset [8] as a baseline. Operators could introduce artificial discrepancies, simulate conditions where certain sensors report steadily increasing values that contradict known physical limits. The secondary MQTT network would record real baseline metrics, while the main

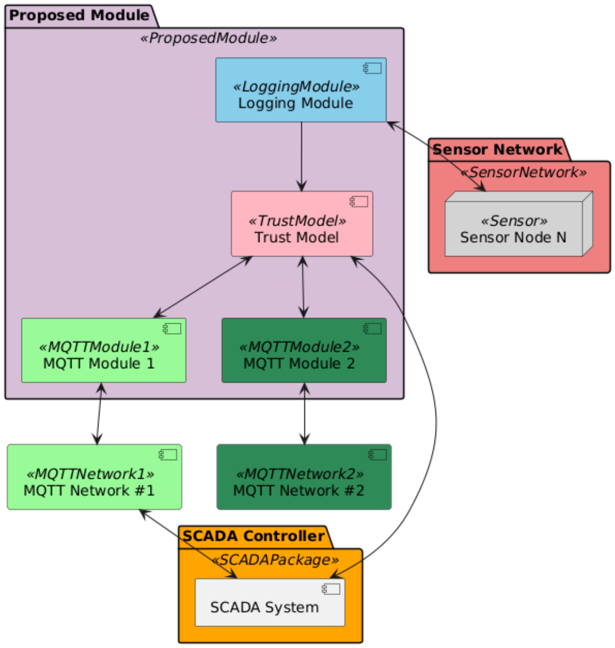

Fig. 7: MQTT Model Diagram

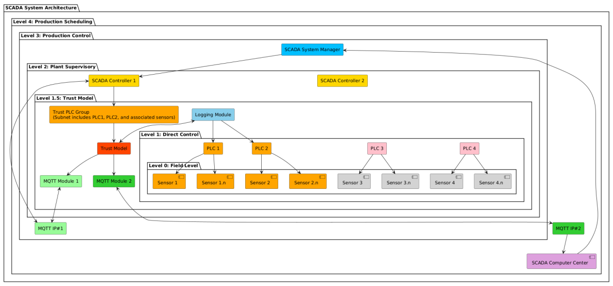

Fig. 8: MQTT System layout

SCADA pipeline receives tampered data. The trust model enhanced with these fact-check signals should then detect and respond to anomalies more quickly or accurately. Iterative experiments, varying attack patterns and timing, would reveal how effectively this secondary network bolsters system resilience and whether latency or complexity overheads are manageable.

Abstract Idea #2: Layered Data Aggregation and Anonymity Techniques

Another forward-looking approach involves integrating privacy-preserving data transformations and anonymization tactics before trust computations. Many SCADA installations face regulatory and proprietary constraints that limit data sharing. Implementing k-anonymity or similar anonymization methods at intermediate aggregation layers could help balance the need for security oversight with privacy concerns. By masking sensitive identifiers and generalizing granular data at higher control tiers, attackers gain less actionable intelligence, and the trust evaluation process can proceed without exposing exact node level details. Aggregated metrics such as averaged port activity, pooled protocol frequencies, or cluster level anomaly indicators could be computed upstream. The trust model then operates on these sanitized aggregates, potentially reducing the risk of leaking critical system signatures. Although anonymization may introduce a mild loss in analytical precision, the trade-off could be worthwhile in highly sensitive environments.

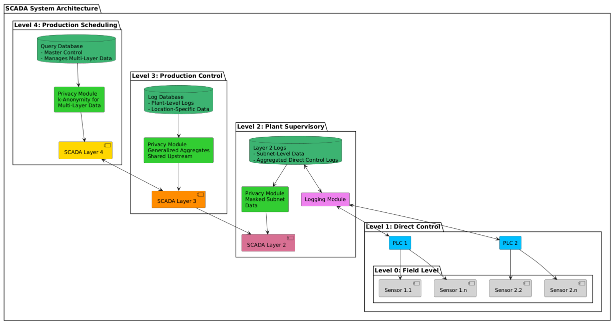

Fig. 9: Conceptual layered data flow illustrating anonymized aggregation before trust evaluation.

Testing and validation would involve simulating different anonymization granularities. One could map node identifiers to broader groups, inject controlled noise into certain metrics, or batch events into rolling time windows. By comparing trust evaluation outcomes on raw versus anonymized data sets, researchers can quantify the accuracy trade-offs. Papers that emphasize secure and robust data handling in IoT and SCADA contexts [10], [19] may provide conceptual frameworks to guide these experiments. We might also draw on techniques from grid simulation platforms like GridLAB-D [17], [18] to replicate more complex distribution networks, verifying whether trust computations remain stable and meaningful even when data is less detailed.

References

[1] K. Boakye-Boateng, A. Ghorbani, and A. Lashkari, "A trust-influenced smart grid: A survey and a proposal," *J. Sens. Actuator Networks*, vol. 11, no. 3, p. 34, 2022. [Online]. Available: https://doi.org/10.3390/jsan11030034

[2] S. Ramya and S. Azad, "Dynamic trust-based process control system for enhanced industrial security," in *2023 3rd International Conference on SCADA Systems*, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10426573/

[3] L. Aghenta and M. Iqbal, "Design and implementation of a low-cost, open source iot-based scada system using esp32 with oled, thingsboard, and mqtt protocol," *AIMS Electronics and Electrical Engineering*, vol. 3, no. 1, pp. 57–78, 2019. [Online]. Available: https://doi.org/10.3934/electreng.2020.1.57

[4] D. Case, "Analysis of the cyber attack on the ukrainian power grid," SANS Institute, Tech. Rep., 2016, accessed: 2023-11-30. [Online]. Available: https://ics.sans.org/media/E-ISAC_SANS_Ukraine_DUC_5.pdf

[5] M. Mohamed and B. Al-Muntaser, "Cybersecurity advances in scada systems," *Journal of Advanced Industrial Cybersecurity*, 2023. [Online]. Available: https://search.proquest.com/openview/7c71a273bfdf4c328acb90996c0296a6

[6] M. Ferrag, M. Babaghayou, and M. Yazici, "Cyber security for fog-based smart grid scada systems: Solutions and challenges," *Journal of Information Security and Applications*, vol. 53, p. 102501, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214212619311408

[7] M. Lapina, E. Basan, A. Lesnikov, and A. Basyuk, "Trust monitoring in a cyber-physical system for security analysis based on distributed computing," in *International Conference on Actual Problems of Systems and Software Engineering*, 2022. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-031-34127-4_42

[8] O. Madani, S. A. Averineni, and S. Gandham, "A dataset of networks of computing hosts," in *Proceedings of the 2022 ACM on International Workshop on Security and Privacy Analytics*, 2022, pp. 100–104. [Online]. Available: https://snap.stanford.edu/data/cisco-networks.html

[9] K. Boakye-Boateng, A. Ghorbani, and A. Lashkari, "Implementation of a trust-based framework for substation defense in the smart grid," *Smart Cities*, vol. 7, no. 1, pp. 5–17, 2023. [Online]. Available: https://www.mdpi.com/2624-6511/7/1/5

[10] Z. Oudina, M. Derdour, and A. Dib, "Model-based system engineering for trust in scada and ics systems in the oil & gas industry," in *Proceedings of the [Conference Name]*.

[11] S. Malathi and S. R. Begum, "Architecture for reliable cyber attack detection in iot networks to increase trustworthiness between nodes," *International Journal of Modeling, Simulation, and Applications*, pp. 73–90, 2024. [Online]. Available: https://www.worldscientific.com/doi/abs/10.1142/S179396232441023X

[12] H. Qi, X. Wang, L. Tolbert, F. Li, and F. Peng, "A resilient real-time system design for a secure and reconfigurable power grid," *IEEE Transactions on Smart Grid*, 2011. [Online]. Available: https://ieeexplore.ieee.org/document/6003812

[13] S. Chehida, E. Rutten, G. Giraud, and S. Mocanu, "A model-based approach for self-adaptive security in cps: Application to smart grids," *Journal of Systems and Software*, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1383762124000559

[14] S. Graham, G. Coates, and K. Hopkinson, "A trust system architecture for scada network security," *IEEE Transactions on Power Delivery*, 2009. [Online]. Available: https://ieeexplore.ieee.org/document/5350402

[15] B. Bhushan, S. Sharma, P. Nand, A. Shankar, and A. Obaid, *Emerging Trends for Securing Cyber Physical Systems and the Internet of Things*. Springer, 2024, available at: https://books.google.com/books?hl=en&id=h1QIEQAAQBAJ.

[16] A. Tidrea, A. Korodi, and I. Silea, "Elliptic curve cryptography considerations for securing automation and scada systems," *Sensors*, vol. 23, no. 5, p. 2686, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/5/2686

[17] *GridLAB-D: Power Distribution System Simulation Software*, Pacific Northwest National Laboratory (PNNL), 2023, accessed: 2023-11-03. [Online]. Available: https://www.gridlabd.org/index.stm

[18] GridLAB-D Contributors, "Gridlab-d github repository," 2023, accessed: 2023-11-03. [Online]. Available: https://github.com/gridlab-d

[19] K. Boakye-Boateng, "Utilizing trust to achieve cyber resilient substations," University of New Brunswick Repository, Tech. Rep., 2024, available at: https://unbscholar.lib.unb.ca/items/961fef51-9fb8-47c1-b8af-bd915883a242. [Online]. Available: https://unbscholar.lib.unb.ca/items/961fef51-9fb8-47c1-b8af-bd915883a242

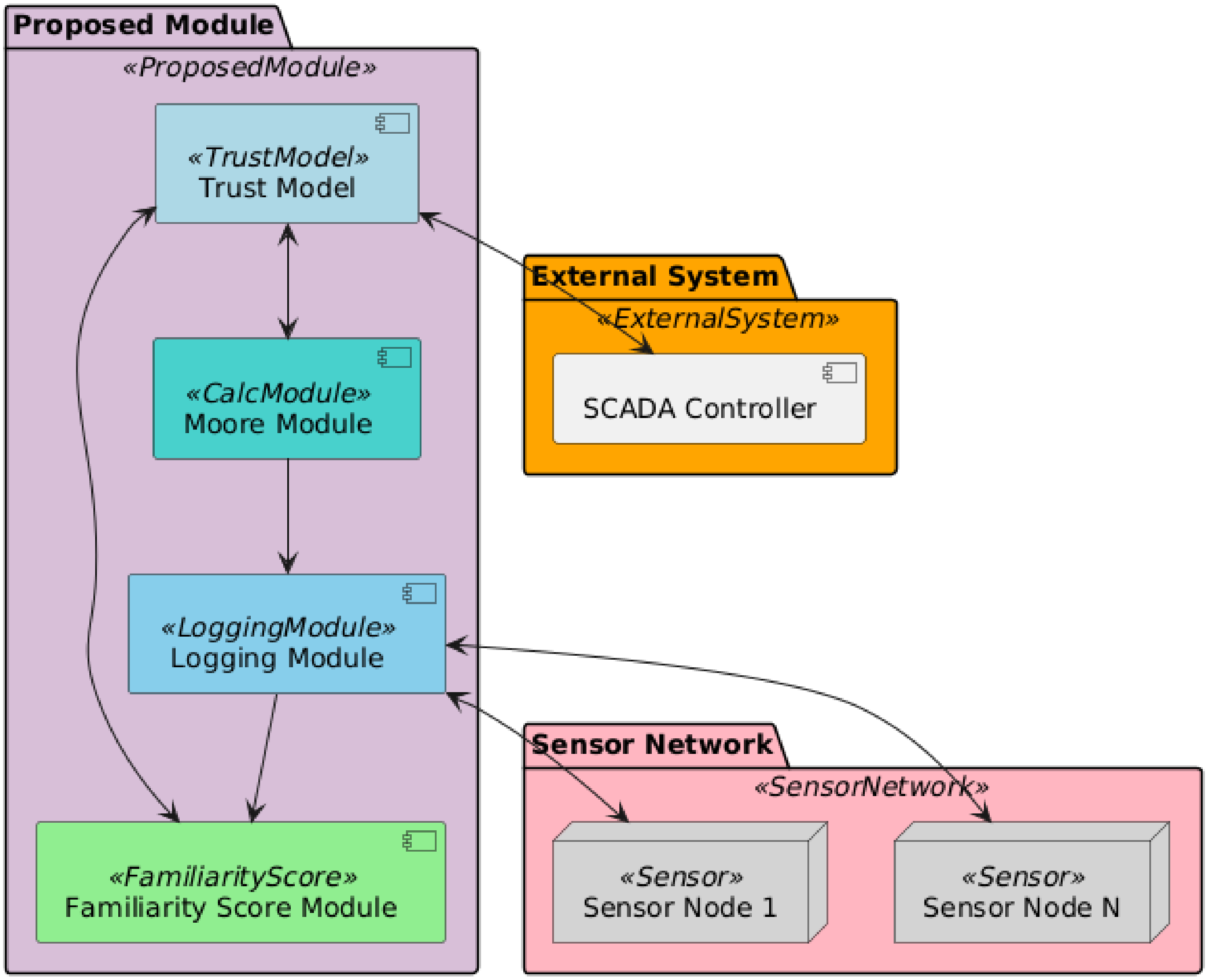

Fig. 10: System Overview

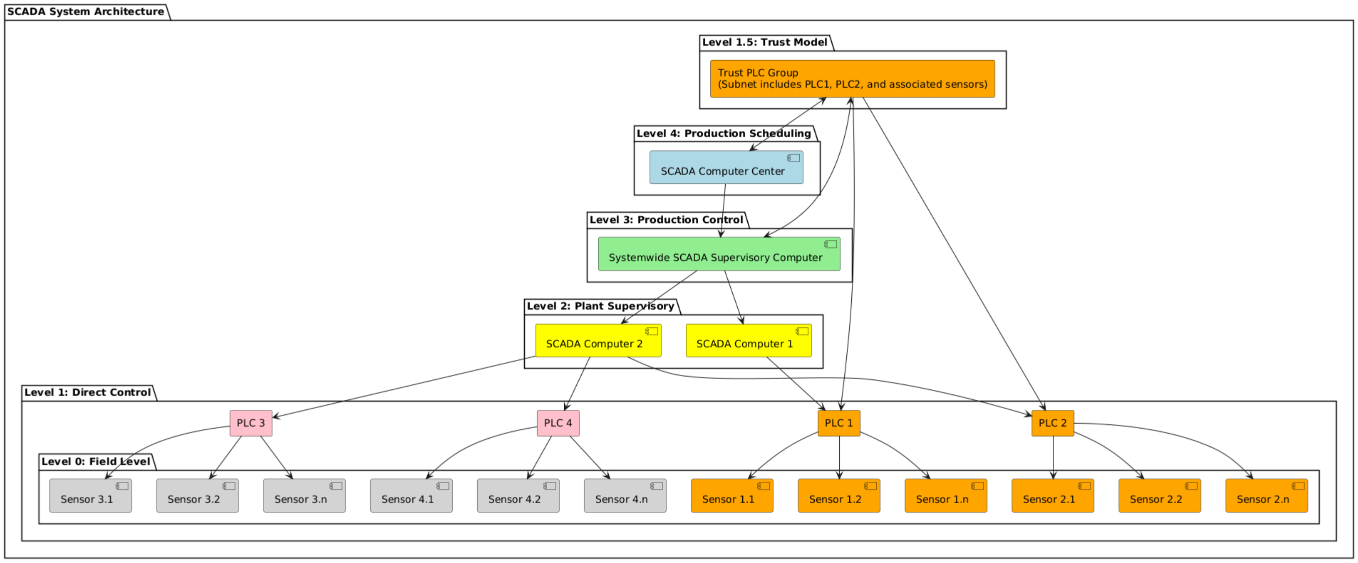

Fig. 11: Layers of the Module

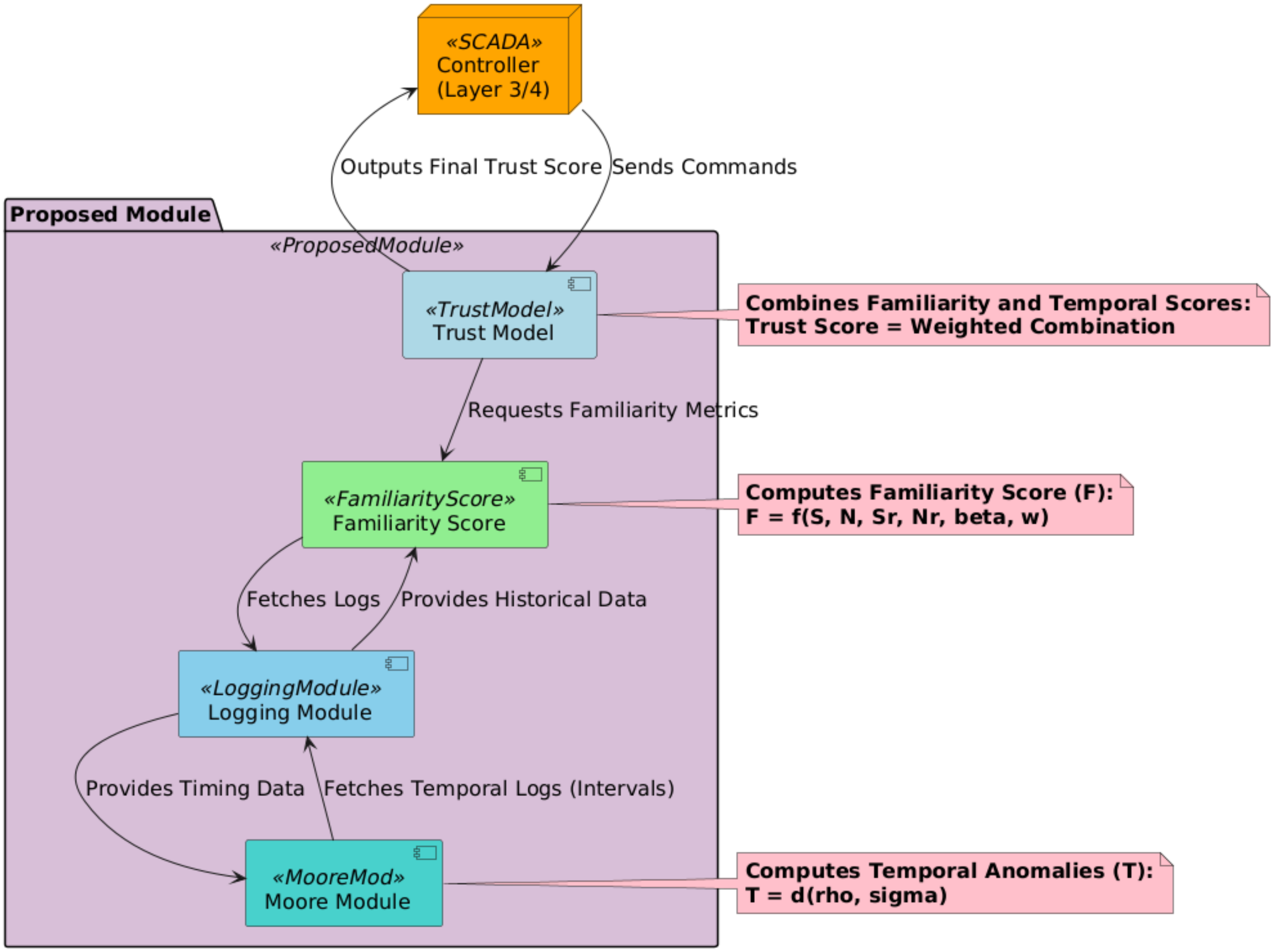

Fig. 12: Module Architecture

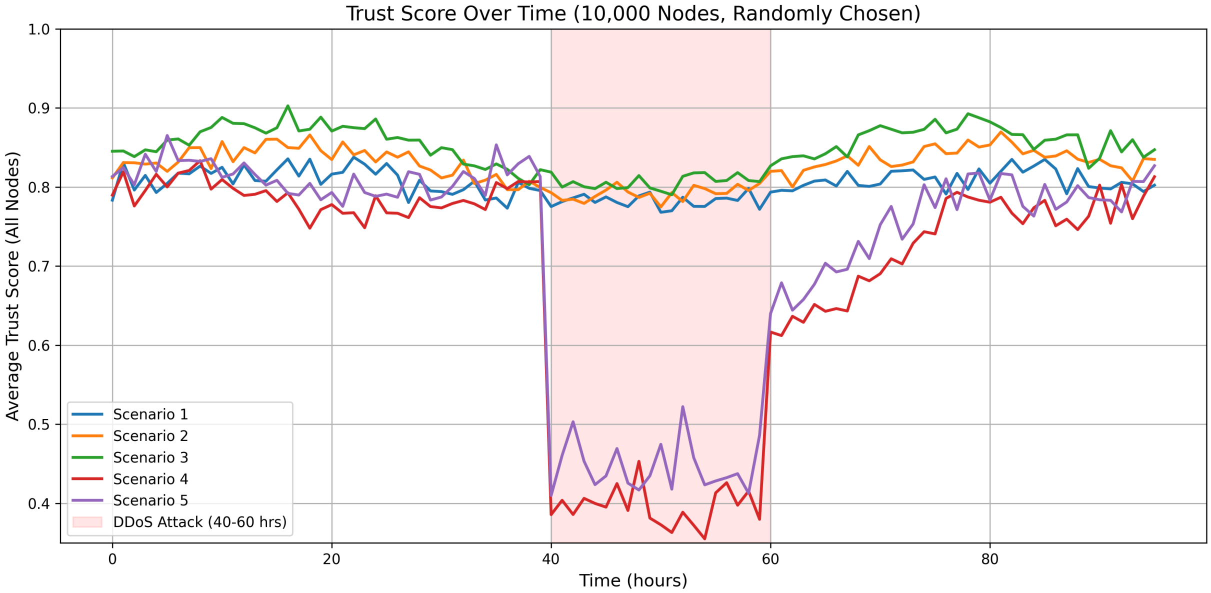

Fig. 13: Trust score evolution for all five scenarios over a 4-day period. Scenario 5 detects anomalies fastest and stabilizes trust values more effectively.

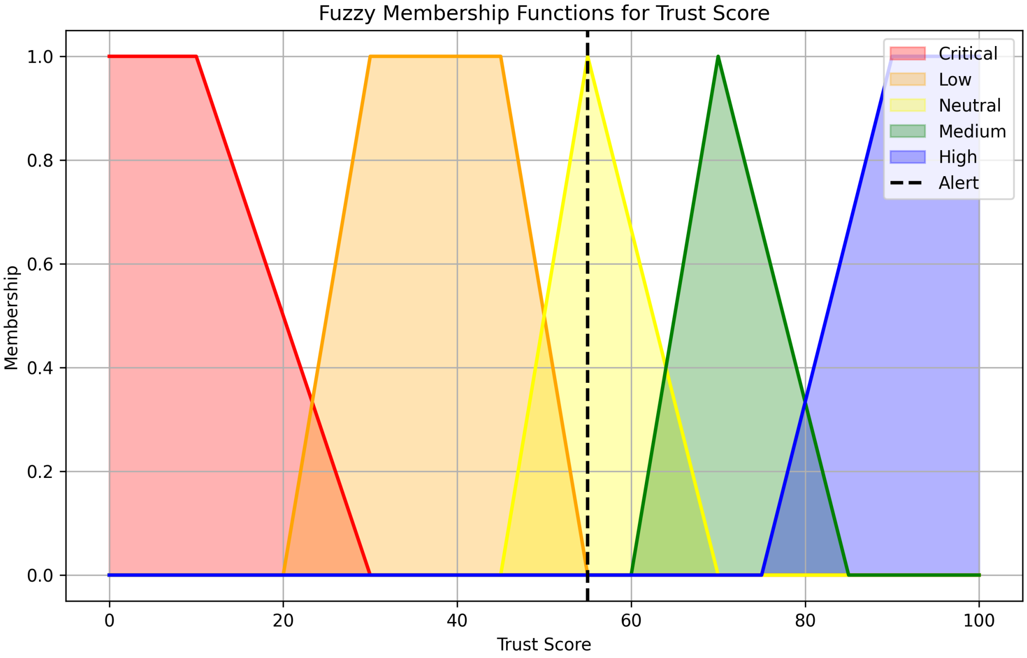

Fig. 14: Fuzzy membership functions for trust evaluation. Trust scores in the ‘critical’ range trigger alerts or isolation of the node.

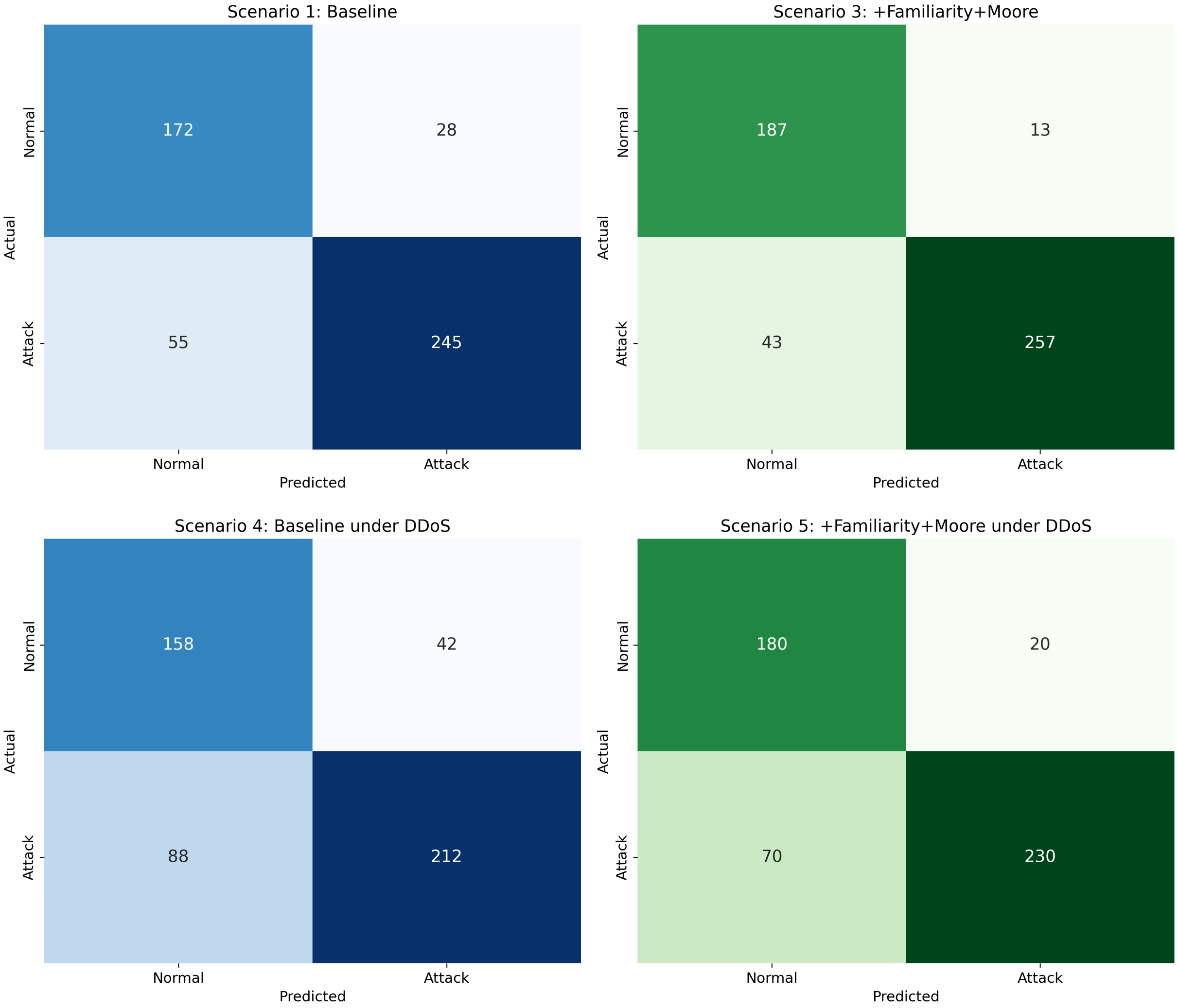

Fig. 15: Confusion matrices comparing scenarios.

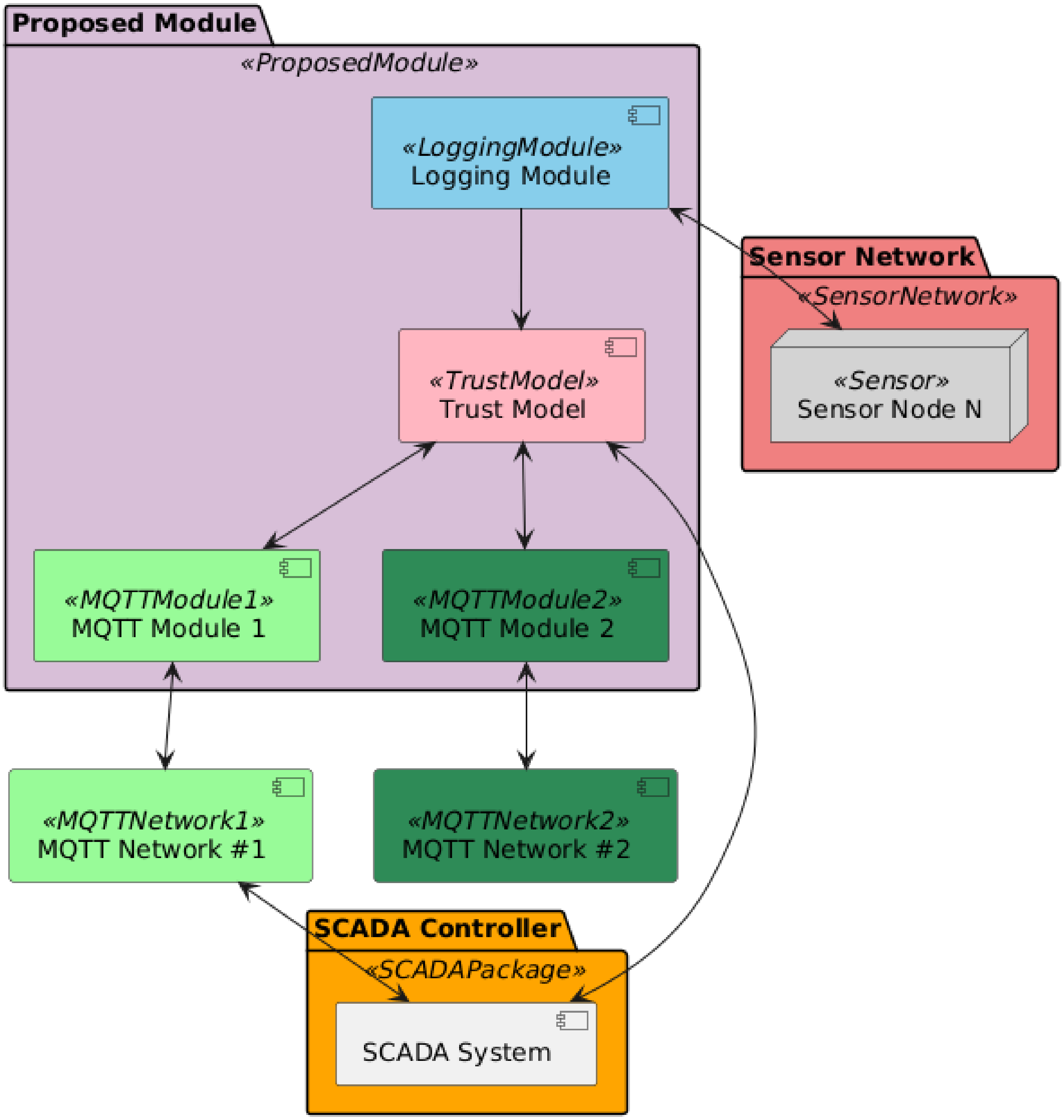

Fig. 16: MQTT Model Diagram

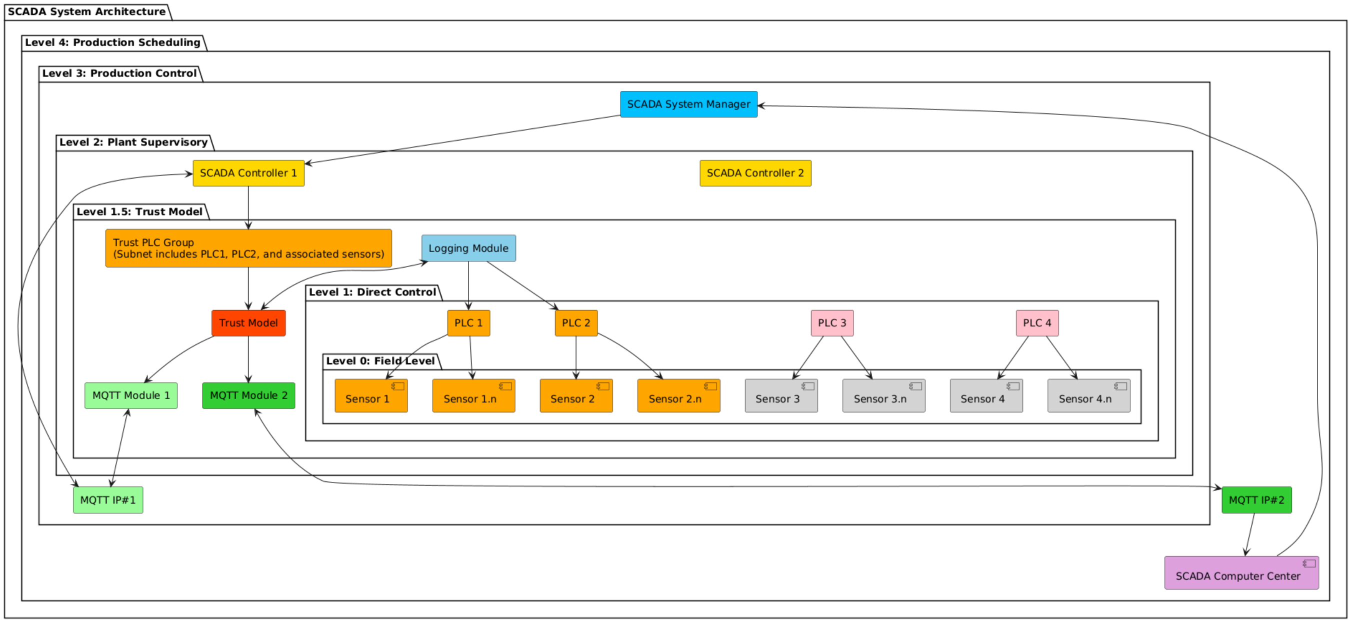

Fig. 17: MQTT System Layout

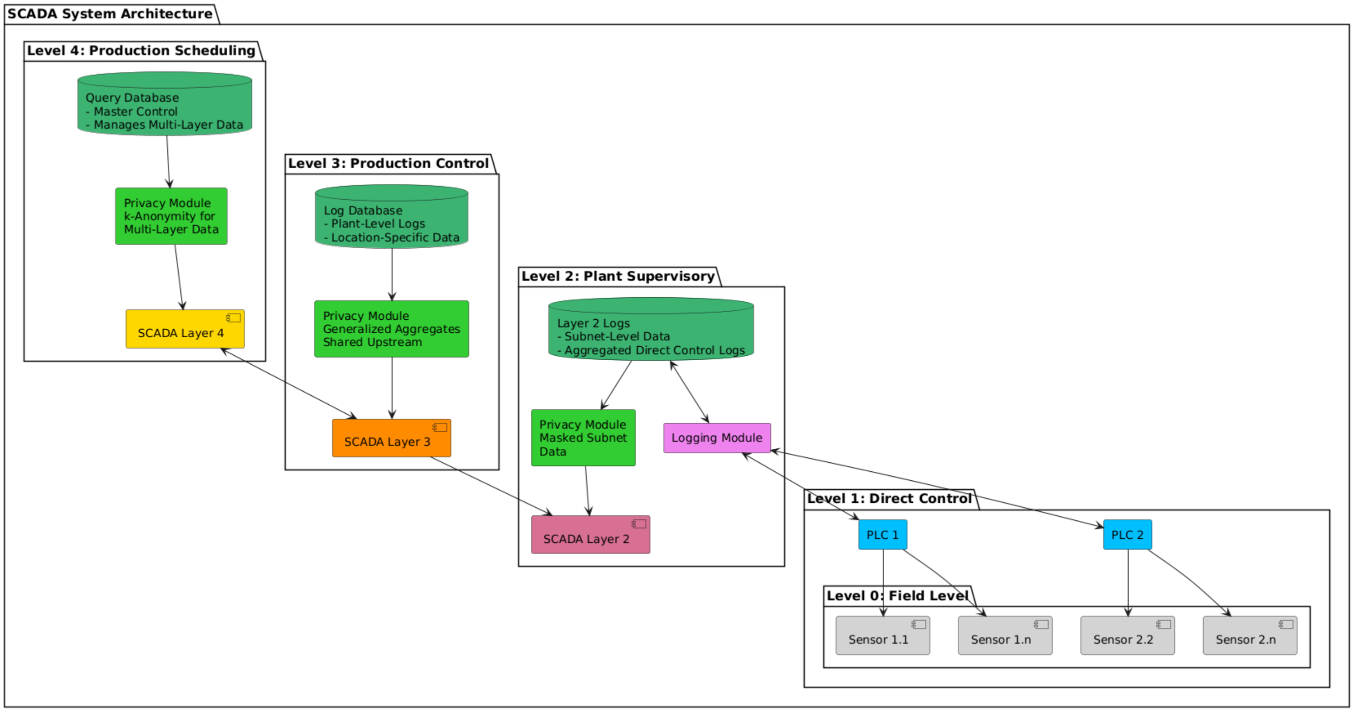

Fig. 18: Layered Data Flow with Anonymized Aggregation