Adaptive Trust-Based Fault Tolerance for Multi-Drone Systems: Theory and Application in Agriculture

Atef GHARBI[1], Faheed A.F. Alrslani[2]
Department of Information Systems-Faculty of Computing and Information Technology,
Northern Border University, Rafha 91911, Saudi Arabia[1]
Department of Information Technology-Faculty of Computing and Information Technology,
Northern Border University, Rafha 91911, Saudi Arabia[2]

***Abstract*—This paper presents RobotTrust, an adaptive trust framework for fault-tolerant coordination in multi-drone systems for precision agriculture. The study aims to improve mission reliability under sensor/actuator faults and uncertain interactions by combining a structured fault taxonomy (behavioral, actuator, sensor) with team-based recovery and an adaptive trust model that integrates direct experience with filtered indirect recommendations. We formalize trust computation (direct, recommended, and global trust) and introduce safeguards such as a minimum-trust threshold and weighted fusion to curb misinformation propagation. The framework is evaluated in simulation using the AgriFleet drone team and is compared against the TReconf baseline across three metrics: (i) time-step efficiency for task completion, (ii) RMSD between predicted and true trustworthiness, and (iii) interaction quality (preference for reliable peers). Results show 20–40% faster task completion, lower RMSD (more accurate trust estimation), and selective interaction patterns that prioritize dependable agents while limiting exposure to unreliable ones. These findings indicate that RobotTrust enhances responsiveness and robustness in decentralized, fault-prone environments typical of agricultural deployments. The work contributes a practical, generalizable approach to trust-aware coordination in multi-robot systems and outlines directions for context-aware weighting, explainable trust signals, heterogeneous teams, adversarial robustness, and large-scale field trials.**

Keywords*—*Adaptive trust model; trust-aware robotics; multi-drone coordination; fault-tolerant systems; precision agriculture applications

I. Introduction

Multi-Robot Systems (MRS) are new approaches in automation and enable collaborative tasks in complex and dynamic environments, such as precision agriculture [1-3], search and rescue missions [4-6], and industrial logistics [7-8]. These systems provide significant scalability, redundancy and adaptation advantages using distributed intelligence, qualities that are essential to real-world applications. However, its decentralized nature poses crucial challenges in fault management and trust-based coordination, where unrecognized failures or poor communication can lead to systemic inefficiencies or total failures of missions [9-11]. The latest progress in MRS focuses on improving fault tolerance and cooperative decision-making. Traditional fault detection methods, such as rule-based diagnostics [12] and anomaly detection based on machine learning [13], are widely accepted. These approaches are effective in controlled environments, but these solutions often lack the sensitivity required for dynamic environments where robot interactions are intrinsically uncertain. At the same time, the trust model in MRS has evolved significantly, and frameworks such as TReconf [14] provide mechanisms for reliability assessment. However, these models demonstrated limitations in dealing with real-time adaptation and reducing errors caused by misinformation, especially in systems affected by sensor noise or partial observation [15]. The recent work of Li et al. [16] has made progress in the integration of trust and fault recovery, but there is still a lack of a comprehensive framework for combining dynamic fault classification, team-based recovery and adaptive trust calculation.

While existing models like TReconf offer basic trust-based coordination, they fall short in handling real-time fault adaptation and mitigating misinformation spread, especially under sensor noise and partial observability. This gap underscores the need for a more robust framework that integrates dynamic fault classification with adaptive trust evaluation—motivating the development of RobotTrust. This paper presents an integrated framework for fault-resistance MRS, addressing these challenges. Our approach begins with a systematic fault classification system that classifies faults into behavioral, actuator and sensor errors, enabling targeted diagnosis and recovery. Based on this, we have developed a decentralized fault handling platform, which uses dynamic task assignment, shared knowledge and self-diagnosis mechanisms to maintain the continuity of the mission. The core innovation of the framework is the RobotTrust model, an adaptive trust evaluation system that quantifies robot reliability through direct interactions, transitional recommendations and global trust aggregation, significantly improving resilience to sensor failures and coordination failures. We validate our framework by extensive testing with AgriFleet, a multi-drone platform designed for precision agricultural applications. Our comparative analysis compares RobotTrust with the established TReconf model [14] in scenarios that simulate the probability of defects and produces three key results. First, our approach has shown that there are approximately 20% improvements in the efficiency of the time-step for the

completion of tasks. Secondly, it reaches greater accuracy of trust, as shown by lower values of root average square deviation than TReconf. Thirdly, the system effectively reduces unnecessary interactions with unreliable agents and optimizes the overall performance of the system.

Our work has made four important contributions in this field. We introduced a structured fault taxonomy for MRS, supported by real-world case studies of the AgriFleet platform. We have developed a new team-based fault recovery mechanism to ensure robust operation in dynamic environments. The RobotTrust model represents a significant advance over existing trust computation frameworks. Finally, our extensive empirical validation shows that the system is more resistant to errors under fault-prone operation conditions.

This paper is organized as follows. Section II provides a detailed examination of fault classification and handling strategies. Section III introduces the computational framework underlying RobotTrust. Section IV presents our comparative simulation results and performance analysis. By integrating fault resilience with trust-aware coordination, this work advances the state of MRS robustness, offering practical solutions that address real-world deployment challenges.

II. Collaborative Fault Management Framework for Multi-Robot Systems

Exploration of multi-robot systems (MRSs) requires proactive fault management, especially by early identification and classification of possible system failures. Establishing clear fault categories provides the basis for developing a structured response strategy. In this context, we present a collaborative fault management framework that integrates team-based coordination to detect, isolate and mitigate faults between multiple robots. This strategy emphasizes the system's collective capability and promotes resilience and maintenance of functionality even in the case of failure of individual components.

A. Fault Classification

To enable systematic fault detection and processing, we classify faults that affect intelligent robot performance in MRS environments into three main areas: behavioral, actuator and sensor faults.

1) Behavioral deficiencies: Behavioral faults refer to the decision-making and planning processes of robots:

a) Action deficiencies: occur when a robot does not perform a commanded action correctly or completely.

b) Plan fault: a failure of a planned action plan caused by a robot not achieving the intended goal.

c) Unexpected conditions: When robots encounter unpredicted environmental conditions and tasks outside their operational design, triggers are activated.

2) Actuator errors: These errors affect the robot's ability to interact physically with the environment:

a) Blocked actuator errors: actuators cannot initiate the necessary movements or operations despite receiving a command.

b) Blocking fault: Even in the absence of a control signal, the actuator remains inactive or active.

3) Sensor errors: These involve errors in the acquisition of sensory data, affecting situational consciousness:

a) Sensor bias errors: sensors provide constant dispersion or misreading, which affects perception and decision-making.

b) Sensor freezing fault: The sensor becomes unresponsive, ceasing to update or transmit data.

B. Fault Handling Using a Team Approach

In collaborative MRS, effective fault management and dynamic task coordination are essential, especially when individual agents encounter operational failures during collective missions. Consider the scenario in which robot R1 suffers a failure (F1) while performing a collective task. The fault management process involves several key participants: a defective robot (R1), other participating robots (such as R2, R3), and a central coordinate entity called a fault manager. Two main system components support this architecture:

- Shared knowledge: A repository containing the team's current understanding of tasks objectives and allocations.
- The Fault Registry is a structured record of known faults, related robots, and current resolution status.

Communication between system components is promoted by structured messages, including report Fault, query Capabilities, updateTaskPlan, performTask, diagnoseFault, reportStatus, and updateFaultRegistry.

III. Multi-Robot Systems Trust Modeling

This section presents a new trust assessment framework, RobotTrust, and compares it to the existing TReconf trust model described in study [14]. The robotic trust methodology is described in detail, focusing on its application to fault detection and trust-based decision making in multi-robot systems (MRSs). The RobotTrust model is an effective mechanism for assessing the trustworthiness of robots in a multi-agent environment. It facilitates the monitoring and detection of faulty robots by analyzing historical interaction data. The model computes multiple trust indicators, including direct trust, recommended trust, and global trust values, and provides a multilevel understanding of the reliability between robots. Trust (i) represents the global trust value assigned to robot i, and Trust (i, j) represents the local trust value assigned to robot j by robot i in direct or indirect interactions. The trust model distinguishes between robots that have previously interacted and those that have not, structuring trust computation accordingly.

A. Types of Trust Metrics

1) Direct trust value: The direct trust value quantifies the relationship between two robots engaged in observable interactions. Using the difference between the number of satisfactory and unsatisfactory interactions as shown in Eq. (1).

$$S_{a,b} = \text{sat}(a,b) - \text{unsat}(a,b) \quad (1)$$

This defines the direct trust between robots (a) and (b), as defined in Eq. (2).

$$trust(a,b) = f(S_{a,b}) \quad (2)$$

and f represents a transformation function that normalizes the satisfaction score.

2) Recommended trust value: If there is no direct interaction between two robots (e.g. (a) and (d)), the recommended trust value is derived using transition relationships defined in Eq. (3).

$$Trust(a,d) = Trust(a,k) \times Trust(k,d) \quad (3)$$

$$\text{where } k \in \text{DirectPeers}(a) \cap \text{DirectPeers}(d)$$

This method uses mutual neighbors to estimate the trust between agents indirectly connected.

3) Global trust value: Global trust value represents the trustworthiness of a robot in the network, aggregating direct and recommended trust metrics of all other robots, as shown in Eq. (4).

$$T_i^{k+1} = \sum_{j=1}^{n} Trust(j,i) \cdot T_j^{k+1} \quad (4)$$

where:

- T_i^{k+1}: global trust of a robot i at the k+1 iteration,
- Trust(j,i): local trust of a robot from j to i,
- n: total number of robots in the network.

Firstly, the global trust values of all robots are distributed uniformly, i.e. for N robot networks. Algorithm 1 calculates the adaptive trust between robots in two main phases: the direct trust calculation of immediate peers and the indirect trust propagation of non-direct peers. Firstly, all trust values are initialized as neutral values (0.5), which represent uncertainty regarding unknown peers. For each direct peer (the robot j with which robot i has directly interacted), the algorithm calculates a trust score based on the historical interaction. It counts a satisfactory (pos) and an unsatisfactory (neg) interaction, and then combines these interactions with a parameter α (set to 0.7). Trust value mixes the satisfaction ratio (pos/total interaction) and the penalty factor (1/(1+neg)), which is degraded with negative experiences. The result is limited between 0 and 1 to ensure a valid trust value. If robot R_1 (trust = 0.9) shares accurate data, its trust score increases; if robot R_2 (trust = 0.6) reports false data, its score decays.

In the case of the indirect peers (robot j with which robot i has not directly interacted), the algorithm calculates trust through intermediary robots connecting i and j. It only considers intermediaries that meet the minimum trust threshold (min_trust = 0.1) to prevent unreliable spread. The indirect trust is calculated as a weighted average of the path of trust, and the contribution of each path depends on the relationship between trust (i,k) and trust (k,j). Parameter β (0.3) controls the amount of indirect trust that updates the existing trust value. If there is no valid path, trust remains at initial neutral value. The algorithm generates a trust value of 0 (complete distrust) to 1 (complete trust) and combines direct experience with carefully filtered indirect information to form a complete assessment of each peer's trust.

Algorithm 1: ComputeAdaptiveTrust

```
Output:
Trust(i,j): trust value assigned by robot i to each robot j
Parameters:
 α ← 0.7   // weight for direct experience
 β ← 0.3   // weight for indirect trust propagation
min_trust ← 0.1   // minimum trust threshold for propagation
neutral_trust ← 0.5   // default neutral trust value

Initialize Trust(i,j) ← neutral_trust for all robot j ∈ AllRobots
// First calculate direct trust
for each robot j ∈ DirectPeers(i) do
    pos ← countSatisfactory(i, j)
    neg ← countUnsatisfactory(i, j)
    total ← pos + neg
     if total > 0 then
         satisfaction_ratio ← pos / total
         penalty_factor ← 1 / (1 + neg)
         Trust(i,j) ← α * satisfaction_ratio + (1 - α) *
penalty_factor
         Trust(i,j) ← max(0, min(1, Trust(i,j)))
    end if
end for

// Calculate indirect trust for non-direct peers
for each robot j ∈ IndirectPeers(i) \ DirectPeers(i) do
    candidates ← findBridgeRobots(i, j)
    weighted_trust_sum ← 0
    total_weights ← 0
    for each robot k ∈ candidates do
        // Only with sufficient trust
        if Trust(i,k) ⩾ min_trust and Trust(k,j) ⩾ min_trust
            weight ← Trust(i,k) * Trust(k,j)
           weighted_trust_sum ← weighted_trust_sum +
weight
            total_weights ← total_weights + weight
        end if
    end for
    if total_weights > 0 then
        indirect_trust ← weighted_trust_sum / total_weights
        // Combine with existing trust if any
        Trust(i,j) ← β * indirect_trust + (1 - β) * Trust(i,j)
    end if
  end for
```

IV. Simulation Scenario and Comparative Trust Evaluation

A. AgriFleet: Trust-Based Fault-Tolerant Drone Coordination

The AgriFleet system consists of a team of specially equipped autonomous air drones (D1, D2, D3, and D4), each equipped with specific capabilities required for the precise

surveillance of agriculture and field data collection. D1 performs high-resolution aerial imaging, D2 achieves soil moisture detection, D3 makes multispectral crop health analysis, and D4 performs environmental mapping and terrain classification. Each drone operates independently, but its operational interdependence is essential for the success of complex agricultural tasks. During the execution of missions, drones constantly communicate, and delegate tasks based on environmental data in real time. For example, D1 can detect crop stress signs by image and transmit this information to D3, which then performs detailed spectral analysis of the affected area. Similarly, D3 can identify areas of suspected soil dilution and prompt D2 to carry out targeted moisture assessment. This type of cooperation enables better coverage, avoids redundancy and increases the overall efficiency of the air monitoring process. To ensure consistency of coordination and reliable task execution, the system incorporates dynamic trust models, based on the accuracy and utility of shared data, where the trust value of drones evolves. The level of trust is quantified on a continuous scale of [0.0, 1.0] and 0.5 represents a neutral baseline. Accurate assessments and successful completion of tasks lead to higher confidence, whereas false reports or failures lead to a decrease in trustworthiness. Despite structured cooperation, drones may suffer from sensor failures, each with unique probability of occurrence. These deficiencies are unknown to drones themselves but affect how their output is trusted by peers. The probability of sensor failure and the corresponding initial trust score are presented in Table I.

TABLE I. DRONE SENSOR RELIABILITY AND TRUSTWORTHINESS

Drone	Function	Sensor malfunction	Prob Trust
D1	High-resolution Imaging	0.1	0.9
D2	Soil Moisture Detection	0.4	0.6
D3	Spectral Crop Health Assessment	0	1
D4	Terrain Mapping and Profiling	0.3	0.7

We evaluated the efficiency of RobotTrust models compared to previously developed TReconf model [14]. The objective is to determine how each trust framework adapts to an environment characterized by interdependent behaviour and probabilistic misinformation due to sensor errors. In 85 interaction rounds, drones act as both trustors and trustees, engaging in communication and task delegation, while considering the probability of sensor-induced false information. The formation of trust is influenced by the interaction between trust scores, feedback on success and interaction results in this time window.

B. Comparative Evaluation of RobotTrust and TReconf Models

The performance of the RobotTrust and TReconf models is evaluated based on three basic parameters:

- Time-step analysis: measures the system's response and task completion speed under different trust conditions.
- Root Mean Square Deviation (RMSD): Evaluate the coherence and convergence of trust assessments throughout the simulation timeline.
- Interaction analysis: investigating the quality and frequency of interactions between robots, especially under uncertainties and partial trust conditions.

This comparative analysis aims to determine which models better capture trust dynamics and improve system reliability in collaborative, fault-prone and multi-robot environments.

1) Time step analysis for trust-based aerial coordination: Fig. 1 shows a clear performance advantage over the TReconfig baseline in various simulation iterations. As shown by time-step analysis, RobotTrust consistently requires fewer time-steps to complete coordination tasks, indicating better system reaction and efficient task execution under the proposed trust-based framework. The data reveal several important trends. First, the performance gap between RobotTrust and TReconfig is most prominent during mid-range iterations (40-60 simulation cycles), and the trust model seems to maximize direct experience evaluation and indirect trust propagation. This suggests that adaptive weighting mechanisms ($\alpha = 0.7$ for direct trust, $\beta = 0.3$ for indirect trust) filter unreliable information effectively under typical operational conditions. Secondly, although the performance of the two systems has increased considerably in higher iteration (approximately 80), RobotTrust maintains an important advantage in the testing range. The relative stability of the RobotTrust time step requirement shows that its fault-tolerant design successfully mitigates coordination failures that increasingly impact baseline systems as the complexity of operations increases. The reduced time step of RobotTrust can bring tangible operational benefits to agricultural applications. More short-term work is required, which means faster response to field conditions, more efficient allocation of resources, and ultimately higher completion rates of critical operations such as crop monitoring or precision spraying. These results validate two key characteristics of RobotTrust design: (1) Minimum Trust Limit (min_trust = 0.1) effectively prevents the spread of errors by unreliable intermediaries; and (2) a mixed trust measurement combining satisfaction rates and penalty factors maintains accuracy despite changes in environmental conditions.

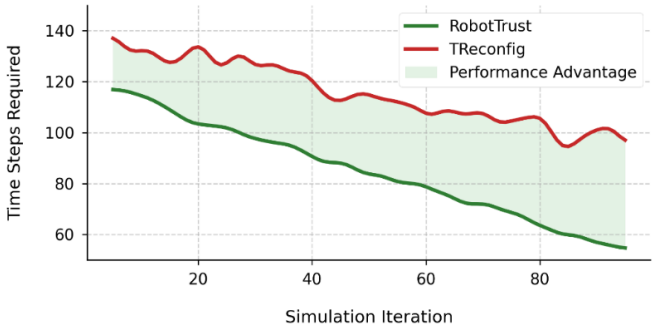

Fig. 1. Comparative analysis of time step efficiency: robottrust vs. treconf in the presence of probabilistic sensor malfunctions.

Performance advantages are particularly present in the most common operational scenario, while modest convergence in

extreme iterations suggests areas for potential improvements in the handling of edge cases. Experimental evidence supports the adoption of trust-based coordination in the agricultural drone fleet, where unpredictable field conditions and the reliability of equipment require a robust and responsive decision-making process.

2) RMSD Evaluation for trustworthiness estimation: The root mean square deviation (RMSD) measurement is used in this study to quantitatively measure the accuracy with which a trust model estimates the true trustworthiness of individual drones in the AgriFleet system. RMSD provides statistical assessments of differences between predictability values generated by trust models and real trust levels derived from established drone attributes such as sensor reliability, mission coherence, and functional performance (as described below). Lower RMSD values indicate a higher degree of alignment between predicted and actual trustworthiness, indicating more accurate and reliable trust models. RMSD is particularly useful in identifying cases of overestimation or underestimation, which may have adverse effects on the coordination of multi-drones, the allocation of resources and the robustness of operations of agricultural missions. For each drone in AgriFleet, the actual value of trust is set as the ground truth reference based on predefined operational parameters. The predicted trust score is derived from the results of the RobotTrust and TReconf models after simulations. The RMSD is computed as shown in Eq. (5).

$$RMSD = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(T_{\text{pred},i} - T_{\text{actual},i})^2} \quad (5)$$

Where:

$T_{pred,i}$: Predicted trust value of drone i at the end of simulation,

$T_{actual,i}$: Actual trust value of drone i based on initial drone specifications,

n: Total number of drones (in this case, 4).

Fig. 2 shows a comparison analysis of the coordination performance of the RobotTrust system with TReconf baseline, measured with the root mean square deviation (RMSD) value between different combinations of drones (T1,2 to T3,4). RMSD measurement serves as a quantitative indicator of coordination accuracy, and the lowest value represents more precise and stable collaboration behaviors between drone pairs. The results show that RobotTrust consistently achieves better coordination performance than TReconf in all tested drone pairings. The RMSD values show particularly significant improvements in certain pair combinations (especially T1,3 and T2,4), suggesting that a trust-based approach can effectively address challenging coordination scenarios in which conventional methods fail. This increased performance is based on RobotTrust's adaptive trust mechanism, which dynamically adjusts interaction weights based on real-time reliability assessments. The variation in RMSD reductions in different drone pairs highlights the effectiveness of trust models based on context. Some pairings have seen dramatic improvements, while others have seen more modest but still significant gains. This pattern suggests that, although the trust framework provides universal benefits, its influence is more significant in situations that require complex interdependent procedures or those that are vulnerable to communication failures. These findings confirm the robustness of the Decentralized Coordination Approach of RobotTrust, especially for agricultural applications, where environmental factors and equipment reliability may vary significantly. The continuous reduction in RMSD in all tested pairs confirms that the system maintains its coordination accuracy advantage, regardless of which specific drones are involved, and proves the reliability of the trust framework. The results highlight the practical value of incorporating adaptive trust indicators into multi-drone systems to improve operation.

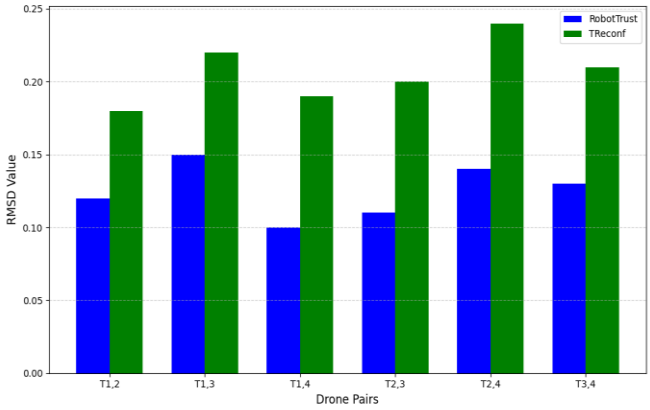

Fig. 2. Root Mean Square Deviation (RMSD) of trust estimation.

3) Interaction analysis: The effectiveness of trust development in multi-drone agricultural systems such as AgriFleet depends primarily on the frequency and quality of interaction between the drones. In this context, interaction analysis evaluates the effectiveness of each trust model (RobotTrust and TReconf) in identifying and prioritizing reliable agents to unreliable agents in field operations.

An optimal trust framework must enable the system to:

- Maximize the value of interactions with reliable drones that are continuously reliable;
- Minimize the dependence on drones that are vulnerable to defects or misinformation;
- Ensure mission progress through reliable communication links for accurate and timely decision-making.

Reliable drones are identified through consistent, accurate and mission-enhancing contributions in various interactions. The trust model that gives these drones a clear priority helps reduce flight redundancy, avoid conflicting data collection, and simplify the overall performance of missions. In the AgriFleet system, the D1 and D3 drones showed higher interaction frequencies than D2 and D4. This distribution is closely linked to the lower probability of sensor malfunction, making it a more reliable data source for collaborative air missions.

Fig. 3 shows the comparison of the thermal maps of the frequencies of interaction between drones under two models: TReconf and RobotTrust. Each subplot shows a 4x4 matrix that represents the number of times each drone interacts with each other, and the rows and columns represent drones D1 to D4. Because drones do not interact with each other, the diagonal entry is zero.

In the TReconf model, interaction frequencies appear to be widely distributed and have relatively high and consistent values for all drone pairs. This indicates a more uniform or dispersed communication model, indicating that TReconf does not differentiate between reliable and unreliable agents in its trust decisions. For example, D1 interacts frequently with other users, including D4, and is unsuitable if D4 is untrustworthy. On the other hand, the RobotTrust model shows a more selective pattern of interaction. The strongest interactions are concentrated in specific pairs (D1 and D3) and show much higher values (e.g. 18 interactions). Interactions with D2 and D4 are less frequent across the board, indicating that RobotTrust is actively avoiding collaborations with drones that might exhibit unreliable behavior. This behavior reflects RobotTrust's intention to isolate unreliable agents and prioritize reliable communication channels. Overall, Fig. 3 effectively shows how trust-based decision-making strategies can shape multi-agent robotic communication patterns.

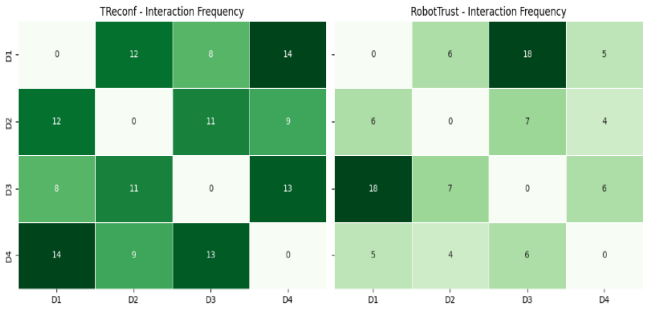

Fig. 3. Heatmap of interaction frequencies among drones under sensor faults, comparing TReconf and RobotTrust models.

C. Discussion

This study shows that RobotTrust improves multi-drone coordination by reducing time-steps to task completion, aligning estimated trust more closely with ground truth (lower RMSD), and prioritizing interactions with reliable peers. These gains appear to stem from two design choices: (i) weighting direct experience more than indirect recommendations and (ii) filtering low-quality recommendations via a minimum-trust threshold. Practically, this combination limits the propagation of misinformation under sensor noise and helps maintain mission progress in fault-prone conditions typical of precision agriculture.

There are, however, several alternative paths we could have taken. First, rather than a hand-crafted trust update, a learned model (e.g., graph neural networks or Bayesian filters) could infer trust dynamics from data, potentially capturing richer inter-robot dependencies at the cost of interpretability and data requirements. Second, instead of rule-based task reassignment, a planning layer based on reinforcement learning or POMDPs could map trust states to actions; this may improve adaptability but raises concerns about sample efficiency and safety guarantees. Third, our probabilistic fault model could be complemented by domain-randomized simulators or hardware-in-the-loop experiments to narrow the sim-to-real gap. Finally, comparisons against consensus-based reputation or Byzantine-resilient aggregation would help position RobotTrust among broader trust frameworks.

We also acknowledge limitations and threats to validity. Communication latency, wind disturbances, GPS/IMU drift, and battery degradation were simplified, and fixed α/β weights may underperform under rapidly changing conditions. Our small-team evaluations (four drones) limit claims about scalability and heterogeneity (air/ground/IoT), and we considered stochastic faults rather than coordinated adversaries.

V. Conclusion

The paper introduced an adaptive trust model to address the fault tolerance coordination of agricultural multidrone systems. The proposed approach enables responsive decision-making even when the individual agent fails or provides unreliable data by dynamically balancing direct experience and indirect trust recommendations through a weighted mechanism. By implementing minimum trust threshold, the system eliminates non-trustworthy intermediaries and significantly reduces the spread of errors across the network. Across simulation studies with the AgriFleet team (D1–D4), RobotTrust reduced coordination time-steps by ~20–40% relative to TReconf, lowered RMSD in trust estimation (indicating closer alignment to true reliability), and prioritized interactions with reliable agents while limiting exposure to unreliable peers. Design elements such as a minimum trust threshold and weighted fusion (α for direct, β for indirect) were pivotal in filtering misinformation and sustaining mission progress under sensor faults. Collectively, these results demonstrate that adaptive, trust-aware coordination can improve responsiveness, reliability, and operational efficiency in fault-prone field conditions typical of agricultural deployments. By accelerating task completion, improving trust accuracy, and curbing counterproductive interactions, RobotTrust supports safer, more efficient flight operations (e.g., crop monitoring, targeted spraying), and offers a generalizable mechanism for decentralized fault management in other mission-critical, multi-robot settings.

Future work will focus on making RobotTrust more adaptive, explainable, and field-ready: first, we will replace fixed trust weights with context-aware weighting that adapts to mission urgency, environmental uncertainty, and sensor health; second, we will add online fault-severity estimation and dynamic task reallocation to shorten recovery latency during actuator/sensor anomalies; third, we will introduce explainable trust signals so operators can audit why trust increased or decreased for a given agent; fourth, we will validate on heterogeneous teams (air/ground/IoT) and new domains beyond agriculture (e.g., disaster response, autonomous logistics) to assess transferability; fifth, we will harden the framework against adversarial misinformation and Byzantine behavior using reputation damping and consensus safeguards; and finally, we will run scalable field trials with larger fleets to

measure compute/communication overheads, energy usage, and mission-level KPIs under realistic fault conditions.

ACKNOWLEDGMENT

The authors extend their appreciation to the Deanship of Scientific Research at Northern Border University, Arar, KSA for funding this research work through the project number "NBU-FFR-2025-2441-02".

REFERENCES

[1] Del Cerro, J., Cruz Ulloa, C., Barrientos, A., & de León Rivas, J. (2021). Unmanned aerial vehicles in agriculture: A survey. Agronomy, 11(2), 203.

[2] Velusamy, P., Rajendran, S., Mahendran, R. K., Naseer, S., Shafiq, M., & Choi, J. G. (2021). Unmanned Aerial Vehicles (UAV) in precision agriculture: Applications and challenges. Energies, 15(1), 217.

[3] Alotaibi, F., Al-Dhaqm, A., & Al-Otaibi, Y. D. (2023). A conceptual digital forensic investigation model applicable to the drone forensics field. Engineering, Technology & Applied Science Research, 13(5), 11608-11615.

[4] Murphy, R. R. (2004). Human-robot interaction in rescue robotics. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 34(2), 138-153.

[5] Chan, T. H., Halim, J. K. D., Tan, K. W., Tang, E., Ang, W. J., Tan, J. Y., ... & Foong, S. (2023, June). A Robotic System of Systems for Human-Robot Collaboration in Search and Rescue Operations. In 2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM) (pp. 878-885). IEEE.

[6] Pan, D., Zhao, D., Pu, Y., Wang, L., & Zhang, Y. (2024). Use of cross‐training in human‐robot collaborative rescue. Human Factors and Ergonomics in Manufacturing & Service Industries, 34(3), 261-276.

[7] Dzedzickis, A., Subačiūtė-Žemaitienė, J., Šutinys, E., Samukaitė-Bubnienė, U., & Bučinskas, V. (2021). Advanced applications of industrial robotics: New trends and possibilities. Applied Sciences, 12(1), 135.

[8] Ebel, H. (2021). Distributed control and organization of communicating mobile robots: design, simulation, and experimentation.

[9] Nandanwar, A., Tripathi, V. K., & Behera, L. (2021, July). Fault-tolerant control for multi-robotics system using variable gain super twisting sliding mode control in cyber-physical framework. In 2021 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM) (pp. 1147-1152). IEEE.

[10] Govoni, L., & Cristofaro, A. (2023, July). A fault-tolerant task allocation framework for overactuated multi-robot systems. In 2023 9th International Conference on Control, Decision and Information Technologies (CoDIT) (pp. 287-292). IEEE.

[11] Ding, S. X. (2021). Advanced methods for fault diagnosis and fault-tolerant control (Vol. 184). Berlin/Heidelberg, Germany: Springer.

[12] Ai, X., Liu, Y., Shan, L., Xie, C., & Zhou, H. (2024). A concurrent fault diagnosis method for electric isolation valves in nuclear power plants based on rule-based reasoning and data-driven methods. Progress in Nuclear Energy, 171, 105190.

[13] Li, G., & Jung, J. J. (2023). Deep learning for anomaly detection in multivariate time series: Approaches, applications, and challenges. Information Fusion, 91, 93-102.

[14] Gharbi, A., & Altowaijri, S. M. (2023). A multi-robot-based architecture and a trust model for intelligent fault management and control systems. Electronics, 12(17), 3679.

[15] Billaud, O., Porcher, E., & Maclouf, E. (2025). Experience and trust to handle uncertainty of biodiversity responses to innovative agricultural practices: Lessons from citizen science on farms. People and Nature.

[16] Guo, Y., Pang, Y., Lyons, J., Lewis, M., Sycara, K., & Liu, R. (2024). Trust-Aware Reflective Control for Fault-Resilient Dynamic Task Response in Human–Swarm Cooperation. AI, 5(1), 446-464.