Impostors Among Us: An Agentic Approach to Identifying and Resolving Conflicts in Collaborative Network Environments

Sai Sree Laya Chukkapalli*, David Ledbetter†, Anupam Joshi†, Tim Finin†, Jade Freeman‡
* IBM Research, NY, USA
saisreelaya.chukkapalli@ibm.com
†University of Maryland, Baltimore County, MD, USA
{davidl2, joshi, finin}@umbc.edu
‡DEVCOM Army Research Lab, MD, USA
jade.l.freeman2.civ@army.mil

***Abstract*—Today's networked cyber-physical environments contain a wide range of agents, including fixed sensors, unmanned aerial vehicles (UAVs), and unmanned ground vehicles (UGVs), which help accomplish the objective(s) of a given mission. These collaborating agents support informed decision making for humans to accomplish mission objectives such as surveillance or search and rescue. However, these agents and their sensors are subject to various failures, including power, communication, hardware, and environmental factors. In contested environments, these failures also result from the kinetic or cyber actions of the adversary. This can result in conflicting information and the loss of a shared notion of the truth, leading to impaired situational awareness and poor decision-making. To overcome this challenge, we present CONFLICTRESOLVER, a policy-driven knowledge-graph framework that can identify agents that share conflicting information and resolve conflicts using agent negotiation and reasoning. In doing so, it also infers/updates the trustworthiness measures of the sensing agents. This framework analyzes the information available within the collaborative agents to identify conflicts, acquire mission-critical objectives from the operator, determine triggering actions, and self-organize and reconfigure the agents' capabilities in accordance with mission-criticality to remain resilient in contested environments. As part of our test bed, we deploy UGVs equipped with multimodal sensors to demonstrate how agents handle conflicts and establish trust.**

***Index Terms*—Collaborative Autonomous Systems, Information Conflict, Resilience, Autonomous Agents, Machine Learning**

I. INTRODUCTION

A distributed system of collaborating, autonomous agents can provide independent observations from on-board sensors to enhance situational awareness and improve decision-making for operators [3] in a variety of contexts, including the battlefield. The data generated by these autonomous agents are heterogeneous in nature, as they can come from various types of sensors (e.g. cameras, LIDAR, RADAR, GPS), from operational metrics (e.g. speed, direction, fuel or battery levels, maintenance status), from measures of environmental conditions (e.g. weather, traffic patterns, object detection), and communication logs (e.g., logs of interactions between the autonomous agent and control centers, other autonomous systems, and infrastructure). The real-time data generated by the agents aid in effective decision making by providing the most current and accurate information.

However, the information reported by the autonomous agents is not always accurate, making it difficult to achieve a shared and coherent notion of the truth. These inaccuracies can arise from sensor malfunctions, environmental interference, and adversarial cyber attacks, especially in contested environments. They can cause challenges [16] such as incorrect decision-making, mission failures, increased risks to personnel and equipment, and resource misallocation. Therefore, there is a need to demonstrate resilience in collaborative autonomous agents to have them automatically adapt and recover from situations when there are inconsistencies in sensed information.

Recent efforts have focused on improving reliability by proposing solutions to identify inaccuracies in the data generated by the agents [19], [27]. These solutions may not always be effective and can significantly hinder the decision-making process during mission-critical events. Our own previous work has described the ontologies that these agents can use to share data and potential actions that might resolve conflicts [7]. But it is also essential to consider how and when to trust the information the agents may share.

We present CONFLICTRESOLVER, which allows for the automatic resolution of inconsistencies across distributed agents, helping to maintain a shared and coherent notion of truth. We identify conflicts by integrating the information shared by agents to analyze and correlate data from similar sensors across different autonomous systems. Metadata about the information generated by the autonomous agents is stored in a knowledge graph. We use this information to calculate a composite score that combines the standard deviation, range, and entropy to provide a single measure of the overall disparity in the information. This disparity can lead to the resolution process being invoked. Information about the mission objectives and their criticality is also stored in the knowledge graph [7]. The key contribution of this work is a policy-driven, agent

negotiation based engine that aims to resolve inconsistencies and conflicts by querying [25] and reasoning over information in the knowledge graph to dynamically choose a resilience strategy. This strategy can involve multiple steps in an agent negotiation process.

To validate and evaluate our approach, we set up an indoor testbed with autonomous agents (UGVs) that mimics realistic deployment scenarios for a surveillance mission. In these scenarios, the UGVs report conflicting data. With our framework, we show how we can identify conflicts by monitoring the information stream from the agents and storing them as facts in the knowledge graph, resolve disputes in soft real-time by reasoning over the facts while querying for mission criticality in the policy engine using RDFox [18], and update the agent trust scores post-resolution.

II. Related Work

The problem of truth maintenance [21] and Byzantine agreement [12] in distributed systems has a long history. However, these models often assume benign faults rather than adversarial agents that deliberately inject misleading information, as considered in our threat model. Several fault diagnosis schemes have been developed in the past to identify false information shared by sensors [1], [14], [26] without the benefit of cross-checks from other sensors. Identifying conflicts in collaborative autonomous systems from multiple sources results in higher data accuracy and system reliability as it allows for a system-wide perspective where the behavior of one sensor can be assessed in the context of others. There has been an increase in interest in collaborative inference at the edge across many different domains, such as autonomous driving [22], swarm robotics [8], and smart homes [15]. However, our scope primarily focuses on identifying and resolving conflicts in the information about the shared environment.

Moreover, the tools developed in the past for identifying conflicts require large volumes of data to train machine learning models and may only work in known environments [4], [9], [11], [13]. This makes them less suitable for dynamic and resource-constrained deployments like battlefield or disaster-response scenarios. Our framework can scale with additional new devices by grouping the information shared based on proximity and modality. As a result, we do not require large volumes of data to identify conflicts. Knowledge graphs and ontologies can help integrate information from multiple sources. For example, Mittal et al. [17] issued real-time alerts to security analysts by reasoning about contextual information gathered from security-related tweets. We monitor autonomous agents to identify and reason over conflicts to resolve them.

Autonomous systems must be capable of dynamically adjusting their operations to mitigate risks and maintain continuity of service. However, current techniques isolate or ignore devices that share inaccurate information [2], restart, or perform a fail-safe rollback to handle faults [19]. These methods may not work in contested or constrained environments, especially when there is a need for immediate situational awareness. Other methods isolate supposedly malfunctioning components without considering mission-critical context or long-term trust dynamics. Through our framework, we can also provide an immediate response where indicated by our reasoning process by activating additional modalities or bringing in trusted devices to share details about the region of interest based on the criticality of the situation.

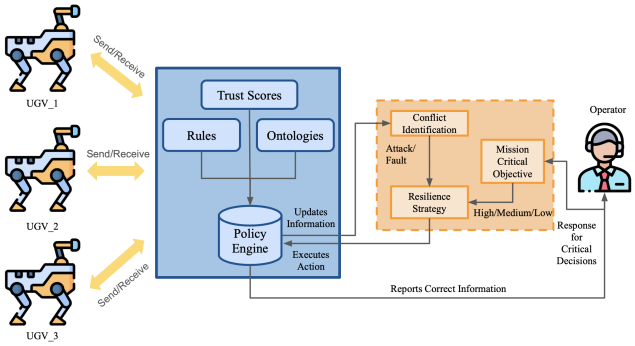

Fig. 1: Architecture of our ConflictResolver framework.

III. Threat Model

As discussed earlier, variations in sensed data among agents can happen for a variety of causes. For the variations caused by adversarial attacks, we assume that the adversary is able to subvert one of more of the sensors by gaining unauthorized access to the autonomous agents. False data can then be injected into a subset of sensor systems to create inconsistencies. We make no assumptions about how many sensors are attacked, so do not assume that we always have $3n+1$ total sensors for n attacked ones. The adversary can also use kinetic action to disable agents and thus prevent them from sharing information.

IV. ConflictResolver Framework

The information shared by autonomous agents can be inaccurate due to constrained environments susceptible to adversarial attacks, harsh environmental conditions, or sensor failures. This can result in inconsistencies between the sensors. ConflictResolver framework shown in Figure 1 aims to resolve inconsistencies by monitoring the data continuously to check for inconsistent information in the same modality. We resolve the conflicts based on mission criticality and additional sensor modalities. In this section, we describe the components of this framework.

A. Context Gathering from Autonomous agents

This module gathers the data generated by the autonomous agents and sensors and stores it as facts. The data types collected are diverse and crucial for a robust understanding of the operational environment. The sensor data includes output from cameras and radars, providing real-time insights into activities on the ground. Environmental data includes information about weather conditions, terrain, and different geographical features that can significantly affect the operations.

We store the above information in the Resource Description Framework (RDF) data model in the policy engine described in Section IV-G. The RDF triples are composed of three components: Subject, which denotes the entity being described; Predicate, which indicates the attribute or relationship of the subject; and Object, which represents the value or another entity related to the subject through the predicate. RDF triples provide a standardized data-sharing format, allowing the operators to efficiently understand and use the data from multiple autonomous agents and their components. In addition, the data captures the context and relationships between data points, offering a richer semantic understanding of the sensor data.

B. Knowledge Graphs Population

Integrating ontology with sensor-generated data, as described in Section IV-A, into knowledge graphs enables a representation of, and reasoning over, situational contexts. These graphs represent entities and their intricate relationships and allow for dynamic querying and updating of data.

In this paper, we use two ontologies from our previous work, IoBT [6] to support information integration from multiple autonomous agents and sensors and the Unified Cybersecurity Ontology (UCO) [23] to capture information about cybersecurity-related events. These ontologies are linked with the data generated by the agents to populate knowledge graphs. We use SPARQL queries [25] to efficiently retrieve specific data and insights from the knowledge graphs.

C. Conflict Identification among Autonomous agents

In this module, we identify inconsistencies in the data generated by the autonomous agents. We start by querying data from various agents, including sensors, devices, or other data sources, each possibly using different modalities (e.g., visual, thermal, acoustic) required for the mission objectives. The ontology from our prior work [7] aids in grouping sensors of similar modalities in a specific region of interest, as the data generated by multiple agents are in different modalities. For example, all thermal sensors in a region would form one group. We calculate the conflict score with the data grouped by modality and region. This score quantifies the disparity or conflict within the data of each group.

There are many reasons for the sensors to report an anomalous value, such as low, high, or zero. During a network failure, the sensors might report no value or a missing value. Hardware failure or an attack can cause the sensor to report a static or stuck value.

In these cases, we use the following well known metrics:

- Mean Imputation: Handle missing values by imputing them with the mean of the observed values
- Standard Deviation: Measure overall spread of the data
- Range: Measure the difference between the maximum and minimum values
- Entropy: Measure data's randomness or unpredictability

Finally, these metrics are combined into a composite score that reflects the overall disparity. We establish pre-defined ranges for the composite score that indicate a conflict or no conflict, considering the components that make up the score and their typical values in scenarios considered normal versus abnormal.

D. Mission Criticality

The mission's overall criticality plays a pivotal role in determining appropriate resilience strategies. Mission criticality refers to the importance and urgency assigned to a mission based on potential consequences, risks, and strategic value. This criticality determines the need to allocate resources, the response strategies that need to be triggered, and the attention required. In addition, the importance of a particular sensed value to the mission is also factored in. In our current implementation, We categorize the mission's criticality into low, medium, and high levels. Our framework also permits a finer or coarser categorization. In the mode where the operator is "in the loop", the operator is requested to respond once to ascribe importance to the mission, and each time an inconsistency is detected to identify how important that information is to the overall mission. High criticality tasks are directly linked to immediate response and safety, medium ones support strategic objectives, and lower ones address broader, often non-immediate issues. By properly understanding and classifying mission criticality and the importance of particular sensed information, operators can effectively achieve their objectives and use resources more efficiently. Where the operator is "on the loop", these measures can be provided to the system ahead of time and the operator merely informed as decisions are made.

E. Resilience Strategies

We present a flexible and extensible framework that enables the integration of diverse resilience strategies aligned with varying levels of mission criticality. These strategies are formalized as Datalog rules within the RDFox reasoning engine [18] and are activated based on contextual cues and agent negotiation. For example, if the mission is not very critical or the particular sensed data is not that important, one available strategy involves majority voting to resolve conflicts by prioritizing consistent sensor inputs. However, this is just one of many strategies that agents can adopt based on predefined rules and mission requirements. At medium criticality, for instance, the system can negotiate with the agent to verify its sensed data by triggering supplementary sensing modalities. For example, if a camera sensor is not detecting tanks that other nearby sensors are showing, it can be asked to send data from its microphones, which can be used to estimate if tanks are nearby using engine noise. If the mission is critical, or if the sensed value is key to the decision making, the central agent can for instance retask a trusted agent. The negotiation process allows agents to dynamically evaluate and agree on the most appropriate strategy, ensuring that conflict resolution remains context-aware, scalable, and resilient across mission scenarios.

F. Trust Scores for Autonomous agents

In this module, we describe how to calculate the trust score for each agent once the conflict is resolved. We create a Bayesian Belief Network representing the dependencies between sensor readings and their initial trust scores. The network will have nodes for each sensor and trust score, with edges indicating the influence of sensor accuracy on trust.

Consider a surveillance setting where $S = \{S_1, S_2, \ldots, S_n\}$ represents sensors across autonomous agents, each with a trust score T_i. Observations $O_i \in \{\text{detects}, \text{does\_not\_detect}\}$ are reported per sensor. Each trust score is initialized as $T_i^{(0)} = 5$. When conflicts arise, we update trust scores using Bayes' theorem:

$$P(T_i|O_i) = \frac{P(O_i|T_i) \cdot P(T_i)}{\sum_{T_i} P(O_i|T_i) \cdot P(T_i)} \quad (1)$$

The updated trust scores are stored in the knowledge base for future decisions. This allows operators to assess the reliability of sensor reports during conflict evaluation.

G. Policy Engine

The policy engine is integral to our CONFLICTRESOLVER framework. It enables the integration of data from multiple agents in the form of RDF triples and stores it as facts as described in Section IV-A and evaluates policies, which are rules written in RDFox's Datalog-like format [18] by querying the facts using SPARQL queries [25]. Here, the facts represent data generated by autonomous agents, environmental conditions, and mission-criticality levels. The policy engine queries the knowledge graph described in Section IV-B to reason over contextual information aligned with the structured semantic understanding of the operational environment to identify inconsistent information as described in Section IV-C.

Contextual information encapsulates details about the interaction of autonomous agents in conflict, the mission criticality, the importance of particular sensed values, etc. The policy engine reasons about the context and available conflict resolution strategies to identify the appropriate resilience strategy, such as the examples mentioned in Section IV-E. The policy engine also supports updating the trust scores of autonomous agents that share incorrect information as described in Subsection IV-F.

V. EVALUATION

This section presents the surveillance of a region of interest as a use-case scenario. We set up the testbed, which consists of multiple autonomous agents continuously monitored in real-time to identify conflicts. We simulate conflicts caused by an adversarial attack, hardware failures, and hostile environmental conditions to show the resilience capabilities of our framework.

A. Experimental Setup

In this section, we outline the resources used for the indoor experiment, which included four Boston Dynamics Spot robots equipped with multiple sensors and processing units.

- Navigation and Object Avoidance: The Velodyne LiDAR VLP-16 captures LiDAR data processed on a Jetson Xavier NX. This unit features a 6-core NVIDIA Carmel ARM V8.2 64-bit CPU and a 384-core NVIDIA Volta GPU with 48 Tensor cores, enabling efficient navigation and crash prevention through object avoidance.
- Object Detection: The Jetson Orin Nano runs the YOLOv8 object detection model [24] to identify and track objects in the environment using the video feeds captured by an Intel RealSense Depth Camera D435i.
- Moving Target Detection: We detect moving targets using a combination of digital signal processing (DSP) methods [5] by employing the AWR 1642 BOOST millimeter-wave radar from Texas Instruments.

These autonomous systems communicate through a client-server architecture, sending and receiving data over the network to coordinate their actions and responses with the RDFox reasoning engine [18] setup on a Jetson Nano. The reasoning engine acts as a centralized hub that captures interactions in the region of interest to identify conflicts and determine resilience strategies to resolve them as described in Section IV.

B. Implementation

CONFLICTRESOLVER demonstrates resilience under simulated conditions that create conflicts and inconsistencies among the autonomous agents in the region of interest. We describe two of them below:

1) Surveillance Inconsistencies with Video Feed Cameras: The surveillance system uses cameras mounted on three UGVs positioned at different angles to maximize coverage and reliability in detecting people. In one scenario, UGV-1 detects 10 people, UGV-3 detects 50, while UGV-2 reports none, simulating an adversarial takeover of UGV-2 as in prior work [10]. The framework identifies this conflict via a composite score and queries the mission's criticality level.

We evaluate the framework across low, medium, and high mission-criticality levels, illustrating possible resolution strategies. For low criticality, the system applies majority voting. For medium, the inconsistent agent activates a secondary sensor (e.g. radar) for validation. For high, a trusted reserve UGV is retasked for independent verification. Trust scores are continuously updated using a Bayesian Belief Network based on data accuracy.

2) Inconsistencies in Surveillance with Millimeter Wave Radar: In this scenario, a millimeter wave radar on a UGV detects motion in the area of interest. To verify its reliability, the framework queries the radar's trust score from a Bayesian Belief Network, which accounts for past false positives. The policy engine for this instance checks the mission's criticality: for low criticality, it accepts the radar's inference; for medium, it consults weather station data to rule out wind-induced errors; and for high, a nearby UGV is retasked to provide visual confirmation. This tiered approach resolves inconsistencies while minimizing resource use.

In addition, we conduct ten experimental runs for each criticality level (low, medium, high) resulting in total 30 runs with 50 simulated conflict instances. Each run involved 4–10 UGVs in a controlled indoor setting, simulating adversarial control (e.g., spoofed data from UGV-2), hardware failures (e.g., constant sensor value), and environmental noise (e.g., radar misreadings due to wind). All these scenarios (e.g., inconsistencies with video feeds and millimeter wave radar) were controlled using a fixed random seed (seed=42) to ensure repeatability.

C. Results

We evaluate the CONFLICTRESOLVER framework under varying mission criticality levels to understand its performance in detecting and resolving conflicting sensor data. Our evaluation focuses on three metrics: *conflict detection F1-score*, *resolution latency*, and *trust convergence time*.

1) Conflict Detection Accuracy: We evaluate conflict detection accuracy using F1-score, based on composite scores computed from standard deviation, range, and entropy. Ground-truth conflict labels are assigned to simulated scenarios and predictions are compared using precision, recall, and F1-score. As shown in Table I, detection accuracy improves with mission criticality, as it should. CONFLICTRESOLVER achieves an F1-score of **0.94** in high-criticality settings, compared to **0.85** under low-criticality, due to the use of additional modalities and trusted agents that reduce false positives. The more critical the mission or the sensed value, the more resources we use to ensure that it is correct.

TABLE I: Evaluation Across Mission Criticality Levels

Criticality	Detection F1-Score	Resolution Latency (s)	Trust Convergence (s)
Low	0.85	0.7	3.5
Medium	0.91	1.2	4.1
High	0.94	2.5	6.8

2) Resolution Latency: Resolution latency is defined as the time elapsed between the detection of the conflict and the implementation of a resilience strategy. For high-criticality missions, latency increases due to the invocation of alternate modalities or the retasking of additional agents. Nonetheless, response times remain under **2.5s**, demonstrating the system's capability to function in near real-time.

3) Trust Score Convergence: We analyze the time taken by the Bayesian Belief Network to converge to update trust values for sensors involved in conflicting reports. Higher mission criticality produces longer convergence due to increased contextual evidence, ranging from **3.5s** to **6.8s**.

To illustrate the ability of the framework to identify and suppress malicious agents over time, we plot the evolution of trust scores for ten UGV agents during a high-criticality surveillance scenario. Figure 2a presents a heatmap where each cell reflects the trust score of an agent at a specific time step, with color encoding to highlight changes in trust.

In this scenario, UGV-3 and UGV-7 are configured to share adversarial or conflicting data. As the mission progresses, CONFLICTRESOLVER detects these inconsistencies and reduces their trust scores accordingly through the Bayesian trust update process. By contrast, UGV-1, UGV-2, and UGV-5 consistently report aligned observations and are rewarded with increased trust scores. Agents with limited or neutral participation maintain relatively stable trust levels. This dynamic adjustment highlights the system's resilience to incorrect or adversarial data and reinforces its ability to prioritize information from reliable agents when resolving conflicts.

4) Baseline Comparison: To evaluate the effectiveness of CONFLICTRESOLVER, we compare it with three baseline techniques lacking the comprehensive integration of context, mission criticality, and dynamic trust modeling.

- **Simple Majority Voting** is a naive approach that assumes the most frequently observed value is correct.
- **Rule-Based Filtering** Employs static heuristics (e.g., range and threshold rules) for conflict detection. This method does not incorporate a reasoning engine or knowledge graph for adaptive conflict resolution, and does not differentiate between high, medium, or low criticality levels, as discussed in Section IV-E.
- **Random Forest Classifier [20]** uses a machine learning model trained on historical sensor data. While more sophisticated than the previous two, it requires large volumes of labeled training data and lacks semantic context integration (Section IV-B). Furthermore, it is not robust to novel or adversarial disruptions, which are dynamically handled in our approach (Section IV-C).

Table II summarizes the performance of each method in terms of detection accuracy and resolution latency. The results demonstrate that CONFLICTRESOLVER offers a substantial improvement over the baselines, achieving higher detection accuracy and policy-aware adaptability. Its resilience-oriented architecture ensures suitability for real-world deployment in adversarial and resource-constrained environments.

TABLE II: Comparison for Conflict Detection and Resolution

Method	Detection F1-Score	Resolution Latency (s)
Simple Majority Voting	0.78	0.5
Rule-based Filtering	0.82	0.9
Random Forest Classifier	0.87	1.6
CONFLICTRESOLVER	**0.94**	2.5

D. System Overhead and Scalability

To assess the scalability of the CONFLICTRESOLVER framework, we evaluate its runtime and memory performance across increasing numbers of autonomous agents. We focused on three metrics: average SPARQL query latency, RDFox memory footprint, and overall decision latency as the system scales. Here, simulated environments consist of 4 to 10 UGVs, each contributing real-time sensor data in multiple modalities. Figure 2b shows the increase in RDFox knowledge base (KB) size and average SPARQL query response time. We observed that while KB size grows linearly with the number of agents, SPARQL query latency remains below 120ms even at peak load, supporting near real-time performance.

(a) Trust score evolution (b) System overhead

Fig. 2: (a) Trust dynamics under adversarial conditions. (b) Latency and KB size trends as agents scale from 4 to 10.

VI. Conclusion & Future Work

The ConflictResolver framework aims to resolve conflicts and maintain the shared notion of truth among multiple autonomous agents in collaborative networks. It monitors multiple autonomous agents by grouping information that belongs to a similar modality, identifies conflicting information while gathering mission criticality, and achieves resilience by reasoning over the available and additional information. It can make informed and reliable decisions for mission objectives by considering automatically updated trust scores assigned to each autonomous agent. Using a realistic UGV testbed, we describe multiple scenarios demonstrating the possibility of information disagreement due to the battlefield's constrained environments and how our framework helps resolve possible inconsistencies among collaborative autonomous agents in real time by selecting appropriate resilience strategies based on operational urgency, and update agent trust scores to support reliable decision-making.

While ConflictResolver demonstrates promising results using interpretable components. In the future, we plan to explore online learning techniques to refine trust estimation over time, and investigate data-driven policy optimization to reduce reliance on predefined resolution strategies.

Acknowledgment

This research was supported by U.S. Army Grant W911NF2120076. We thank the UMBC CARDS Center for supporting the experimental setup.

References

[1] Ismail R. Alkhouri, Akram S. Awad, Qun Zhou Sun, and George K. Atia. Imperceptible attacks on fault detection and diagnosis systems in smart buildings. *IEEE Trans. Ind. Informatics*, 20(2):2167–2176, 2024.

[2] Masoud Ardekani, Rayman Singh, Nitin Agrawal, Douglas Terry, and Riza Suminto. Rivulet: a fault-tolerant platform for smart-home applications. In *ACM/IFIP/USENIX Middleware Conf.*, pages 41–54, 2017.

[3] Booz, Allen, Hamilton. The future battlefield is digital. https://www.boozallen.com/markets/defense/digital-battlespace.html, 2025.

[4] Jiwon Choi, Hayoung Jeoung, Jihun Kim, Youngjoo Ko, Wonup Jung, Hanjun Kim, and Jong Kim. Detecting and identifying faulty iot devices in smart home with context extraction. In *48th international conference on dependable systems and networks*, pages 610–621. IEEE, 2018.

[5] Debjyoti Chowdhury, Nikhitha Vikram Melige, Biplab Pal, and Aryya Gangopadhyay. Enhancing outdoor moving target detection: Integrating classical dsp with mmwave fmcw radars in dynamic environments. *Electronics*, 12(24):5030, 2023.

[6] Sai Sree Laya Chukkapalli. Internet of battlefield things ontology. http://bit.ly/ArtIAMAS_iobt, 2022.

[7] Sai Sree Laya Chukkapalli, Anupam Joshi, Tim Finin, and Robert F Erbacher. Capd: a context-aware, policy-driven framework for secure and resilient iobt operations. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications IV*, volume 12113, pages 224–232. SPIE, 2022.

[8] Kasra Eshaghi, Goldie Nejat, and Beno Benhabib. A concurrent mission-planning methodology for robotic swarms using collaborative motion-control strategies. *J. Intell. Robotic Syst.*, 108(2):15, 2023.

[9] Chenglong Fu, Qiang Zeng, and Xiaojiang Du. HAWatcher: Semantics-aware anomaly detection for appified smart homes. In *30th USENIX Security Symposium*, pages 4223–4240, 2021.

[10] Narayana Murari Gowrishetty, Sai Sree Laya Chukkapalli, and Anupam Joshi. Bewitching the battlefield: Repurposing the mousejack attack for crazyflie drones. In *IEEE Conference on Computer Communications Workshops*, pages 1–6, 2023.

[11] David Haldimann, Marco Guerriero, Yannick Maret, Nunzio Bonavita, Gregorio Ciarlo, and Marta Sabbadin. A scalable algorithm for identifying multiple-sensor faults using disentangled rnns. *IEEE Trans. Neural Networks Learn. Syst.*, 33(3):1093–1106, 2022.

[12] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, July 1982.

[13] Dan Li, Yuxun Zhou, Guoqiang Hu, and Costas J. Spanos. Identifying unseen faults for smart buildings by incorporating expert knowledge with data. *IEEE Trans Autom. Sci. Eng.*, 16(3):1412–1425, 2019.

[14] Siliang Lu, Jingfeng Lu, Kang An, Xiaoxian Wang, and Qingbo He. Edge computing on iot for machine signal processing and fault diagnosis: A review. *IEEE Internet of Things Journal*, 10(13):11093–11116, 2023.

[15] Yair Meidan, Dan Avraham, Hanan Libhaber, and Asaf Shabtai. Cadesh: Collaborative anomaly detection for smart homes. *IEEE Internet Things J.*, 10(10):8514–8532, 2023.

[16] Archan Misra, Dulanga Weerakoon, and Kasthuri Jayarajah. The challenge of collaborative iot-based inferencing in adversarial settings. In *INFOCOM Conference on Computer Communications Workshops*, pages 1–6. IEEE, 2019.

[17] Sudip Mittal, Prajit Kumar Das, Varish Mulwad, Anupam Joshi, and Tim Finin. Cybertwitter: Using twitter to generate alerts for cybersecurity threats and vulnerabilities. In *International Conference on Advances in Social Networks Analysis and Mining*, pages 860–867. IEEE, 2016.

[18] Yavor Nenov, Robert Piro, Boris Motik, Ian Horrocks, Zhe Wu, and Jay Banerjee. Rdfox: A highly-scalable rdf store. In *14th International Semantic Web Conference*, pages 3–20. Springer, 2015.

[19] Michael Norris, Z Berkay Celik, Prasanna Venkatesh, Shulin Zhao, Patrick McDaniel, Anand Sivasubramaniam, and Gang Tan. Iotrepair: Flexible fault handling in diverse iot deployments. *ACM Transactions on Internet of Things*, 3(3):1–33, 2022.

[20] Mahesh Pal. Random forest classifier for remote sensing classification. *International journal of remote sensing*, 26(1):217–222, 2005.

[21] M. Stanojevic, S. Vranes, and D. Velasevic. Using truth maintenance systems. a tutorial. *IEEE Expert*, 9(6):46–56, 1994.

[22] Shengchao Su, Xiang Ju, Chaojie Xu, and Yufeng Dai. Collaborative motion planning based on the improved ant colony algorithm for multiple autonomous vehicles. *IEEE Trans. Intell. Transp. Syst.*, 25(3):2792–2802, 2024.

[23] Zareen Syed, Ankur Padia, Tim Finin, Lisa Mathews, and Anupam Joshi. Uco: A unified cybersecurity ontology. In *Workshops at the thirtieth AAAI conference on artificial intelligence*, 2016.

[24] Rejin Varghese and M Sambath. Yolov8: A novel object detection algorithm with enhanced performance and robustness. In *International Conference on Advances in Data Engineering and Intelligent Computing Systems*. IEEE, 2024.

[25] W3C SPARQL Working Group. Sparql 1.1 overview. Technical report, World Wide Web Consortium, 2013.

[26] Yanzhi Wang, Ziyang Yu, Jinhong Wu, Chu Wang, Qi Zhou, and Jiexiang Hu. Adaptive knowledge distillation based lightweight intelligent fault diagnosis framework in iot edge computing. *IEEE Internet of Things Journal*, 2024.

[27] Dulanga Weerakoon, Kasthuri Jayarajah, Randy Tandriansyah, and Archan Misra. Resilient collaborative intelligence for adversarial iot environments. In *22th International Conference on Information Fusion*, pages 1–8. IEEE, 2019.