Intent-driven Data Falsification Attack on Collaborative IoT-Edge Environments

Shima Yousefi
City University of New York
Email: syousefi@gradcenter.cuny.edu

Shameek Bhattacharjee
Western Michigan University
Email: shameek.bhattacharjee@wmich.edu

Saptarshi Debroy
City University of New York
Email:saptarshi.debroy@hunter.cuny.edu

***Abstract*—Collaborative IoT-edge environments, although effective in hosting latency-sensitive applications, are fundamentally vulnerable to data falsification attacks that can potentially impact key system performance objectives. In this paper, we explore and propose an intent-driven energy data falsification attack model for collaborative IoT-edge environments and shed light on the attack's impact on system performance. Our primary contribution lies in developing key intuitions and systemization of threat landscape for attacks with selfish and malicious intents that target one or many key system performance objectives, viz., overall system energy-efficiency and end-to-end latency of hosted applications. The proposed attack model is evaluated, optimized, and validated through 'testbed-in-the-loop' simulations. The results demonstrate that depending on selfish and malicious intents, the proposed attack model can achieve upto 50% increase in energy savings for the compromised IoT devices, accelerate battery drainage of non-compromised devices, and ensure upto 61% success in violating application latency requirements.**

***Index Terms*—Edge computing, data falsification, energy efficiency, IoT, task offloading, collaborative computing.**

I. Introduction

With the widespread adoption of compute-intensive machine learning and artificial intelligence (ML/AI) applications for mission-critical use cases, collaborative IoT-edge environments involving IoT devices (e.g., drones, UAVs, robots) and edge servers that share the burden of the computation workload, are becoming popular [1]. Given the energy and compute resource constraints of IoT devices, fully or partially offloading compute-intensive tasks from the IoT devices to edge servers becomes imperative to enhance the IoT devices' energy efficiency and minimize the strict end-to-end latency requirements of the involved ML/AI workloads.

In most cases, such energy and latency optimization necessitates key system parameters, such as, energy consumption, to be periodically measured and shared with a remote controller that is responsible for overall system resource management. However, sharing such key information over unsecured wireless channels and/or relay nodes to the edge server hosted controller, may leave the overall system vulnerable to data falsification attacks that can potentially impact the system performance objectives.

Falsification of energy consumption information, in collaborative IoT-edge environment poses a significant challenge mission-critical applications [2], [3]. Due to the often multi-objective nature of the system performance optimization, inter-conflict among such optimization objectives (e.g., energy efficiency and latency), and interdependence among system components (e.g., IoT devices and edge servers), data falsification in energy consumption information may lead to degradation in all such system performance objectives, depending on the attack intent. Hence, understanding the attacker's intent and conflicting system performance objectives are critical. This necessitates a comprehensive analysis of the attacker's motives, system vulnerabilities, and the potential impact of such attacks on system performance. Although the existing literature contains extensive studies on data falsification attacks on areas, such as healthcare, power system and Smart Grids , autonomous vehicles, and wireless networks, there exists little to no exploration of intent-driven data falsification attacks on collaborative IoT-edge environments.

In this paper, we explore data falsification attacks, in particular of energy consumption information, within the context of collaborative IoT-edge environments, and shed light on their potential impact on system performance. To this end, we first present a comprehensive system model of a collaborative IoT-edge environment hosting a drone driven video processing application that incorporates three key elements: 1) ML/AI application model (for visual computing) that involves the task and data to be offloaded, 2) widely used IoT device energy consumption model involving sensory, transmission, kinetic, and computational activities [4], and 3) a state-of-the-art task offloading strategy from IoT devices (e.g., drones) to edge servers [5], [6]. Our primary contribution lies in developing a novel data falsification attack model, which is driven by the intentions of the attacker in terms of targeting one or many system performance objectives, while maintaining stealth and remaining undetected. In particular, we explore four specific types of attack intents that are either selfish and malicious, viz., selfishly saving the energy of the compromised devices, maliciously draining the non-compromised devices of battery, maliciously jeopardizing the timely execution of ML/AI tasks associated with non-compromised devices, and inflicting operation impairment or disruption to non-compromised devices.

The effectiveness of the proposed attack model is evaluated, optimized, and validated through 'testbed-in-the-loop' simulations, that encompass a wide range of attack and system scenarios. We show that for different attack scales and intensities, attack that are selfish in their intent can result in upto 50% increase in energy savings for the compromised devices. Furthermore, the results demonstrate that for attacks with malicious intent and under diverse attack scenarios, data falsification can lead to: i) complete battery drainage of non-compromised devices, ii) violation of non-compromised devices' computing task deadlines, and iii) operational harm to non-compromised devices, for upto 61% success rate, , depending on system settings. Overall, the results provide key insights on developing energy data falsification attacks, with diverse intents, that can be effective in achieving their goals, yet difficult to detect.

The rest of the paper is organized as follows. Section II introduces the collaborative IoT-edge system model. Section III proposes the attack model Section IV discusses the experimental evaluation and results. Section V concludes the paper.

II. System Model

This work considers a typical collaborative IoT-edge environment where the IoT devices are predominantly performing kinetic and sensory activities on top of sharing the computation workload with the edge servers. This is achieved by integrating

2024 IEEE/ACM Symposium on Edge Computing (SEC) | 979-8-3503-7828-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/SEC62691.2024.00047

2837-4827/24/$31.00 ©2024 IEEE
DOI 10.1109/SEC62691.2024.00047

Authorized licensed use limited to: CUNY Hunter College Library. Downloaded on February 04,2025 at 21:51:54 UTC from IEEE Xplore. Restrictions apply.

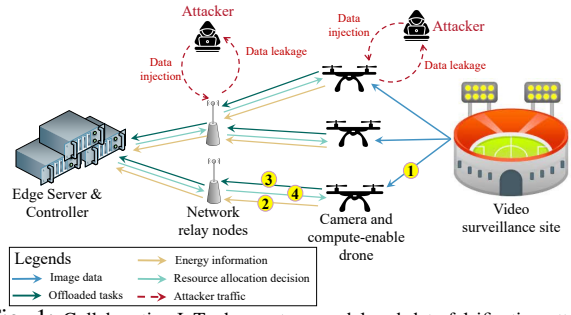


Fig. 1: Collaborative IoT-edge system model and data falsification attack

CPU/GPU computational units, such as NVIDIA Jetson Nano or TX2 units with the devices.

A. Application model

Our exemplary collaborative IoT-edge environment comprises of a swarm of camera enabled drones/UAVs that performs video surveillance of a site of interest in collaboration with a remote edge server which also acts as the overall system controller, as shown in Fig. 1. In particular, the exemplary system performs a set of object detection tasks $\{\gamma_1 \dots \gamma_N\}$ where each $\gamma_i = \{\S_1^i, \S_2^i, ..., \S_M^i\}\ \forall\ i \in N$ consists of a sequence of M stages. We assume that each task has a distinct and strict latency deadline defined by t_d^i executing in drone $\mathcal{D}_i$ where $\{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_N\}\ \forall\ i \in N$ is the set of drones. Given the compute resource constraints of the drones, it is imperative to offload a set of task stages to the edge server to ensure timely execution.

B. Computation model and task offloading

1) Fair-allocation heuristic: Here, initially, the controller leverages the deadlines t_d^i's to determine the maximum number of stages that a given drone can complete within the deadline. This calculation of the maximum stages is derived as $\text{Max}_{\S^i} = \frac{t_d^i - \text{App}_s \times Q^i}{\text{App}_U - \text{App}_s}$, where Q_i is total number of stages, App_s is computation time required for one stage in the server, and App_U is computation time for one stage in the drone. The server allocates part of its computational resources to handle offloaded drone tasks and meet their deadlines. Such relation also provides the minimum server resource requirement to meet such task deadline: $\text{Min}_{\S^i} = Q^i - Max_{\S^i}$

Once the controller collects sufficient data from drones, it employs a fair allocation of the server capacity to the competing drones. Further, as a drone's battery approaches complete depletion, the server redirects the computational capacity initially designated for that drone to support the remaining drones, guaranteeing the continuity of operations with resource efficiency at its core. Thus, for typical fair allocations, the number of computational stages of a ML/AI model assigned to a drone $\text{R}_{\S^i}$ can be expressed as $\text{R}_{\S^i} = R \times \frac{\mathcal{C}^i(t)}{\sum_{i=1}^{N} \mathcal{C}^i(t)}$, where R denotes the remaining number of stages that the server can run and is calculated as $R = C_t - \sum_{i=1}^{N} Min_{\S^i}$ with C_t being the server capacity. Here, $\mathcal{C}^i(t)$ is the task allocation coefficient calculated as $\mathcal{C}^i(t) = \frac{\mathcal{B}_c^i(t)}{\mathcal{B}_n^0}$ with $\mathcal{B}_c^i(t)$ being consumed battery at time t and $\mathcal{B}_n^0$ being the initial drone battery capacity.

C. IoT device/drone energy consumption model

1) Kinetic and data collection activity: Depending on the mission, drones can have different kinetic modes. For simplicity, in this model, we only consider flying and hovering. For each mode, the energy consumption is modeled based on many factors, such as, drone mass, payload, and number of rotor to name a few. Based on such factors, we model the rate of energy, i.e., power consumption for hovering $P^{hov} = f(w, q, g, \rho, \Delta, n)$ based on [4], where w represents the total mass of the drone, q signifies the payload weight, g denotes the gravitational force, ρ is the density of air, Δ refers to the area of the spinning blade, and n is the number of rotors.

Similarly, for flying, the power consumption is modeled as $P^{fly} = f(T, V_a, \alpha, V_i, \eta)$, where T is total thrust, V_a denoted as air speed, α is the pitch angle for steady flight, V_i represents induced speed, and η is power efficiency. The drones are also involved in data collection, specifically video streaming that the ML/AI applications process. Power consumption during such data collection P^{data} is simply modeled according to well accepted model in [7].

2) Data transmission activity: Our system model shown in Fig. 1, assumes direct wireless connection between the drones and the edge server/controller, however, such connectivity can also involve intermediate hops of access points/base stations/relay nodes. The power consumption for transmission P^{trans} for such scenario can be modeled by [5].

III. Attack Model

A. Exploits for Energy Falsification

IoT devices are generally susceptible to cyberattacks due to their limited power, connectivity, processing, data storage capacities, and inherent heterogeneity [8]. Data leakage and falsification are some of the most critical vulnerabilities and potential attacks that exist in different layers of a collaborative IoT-edge environment, viz., application, middleware, and edge layers [9]. For instance, in application layer, IoT device software is often written in unsafe programming languages and poorly maintained due to their limited computational and power resources. Additionally, the hardware used in these devices is not always robust enough to withstand manipulation attacks. This lack of robustness makes it easier for attackers to compromise a device within a network and utilize it as a base to launch attacks against other devices in the network [10], such as exfiltration [11]. The primary type of data falsification that can manifest through IoT middleware layer is the Man-In-The-Middle (MITM) attack. This attack targets the data exchanged between endpoints, maybe remotely or often at intermediate hops/relay nodes through proxy devices compromising both data integrity and confidentiality by falsifying power data associated with the compromised drone[12]. Finally, attacks can also occur through edge layer. However, for this paper, we explore data falsification before the data reaches the edge servers.

B. Categories of Attacker Intents

In this work, we conceptualize four possible intents for data falsification in a collaborative IoT-edge environment, while remaining undetected for each scenario:

Save energy - Here, a compromised IoT device, i.e., a drone in our use case, tries to get an unfairly more allocation of the server's resources. The controller is fooled into allocating additional capacity to the selfish drone, believing it to be on the brink of depletion. This intent is *selfish* in nature.

Drain non-compromised drones - It is a *malicious* intent where the compromised drone seeks to increase the workload of the non-compromised drones by misleading the controller to increase tasks assigned to latter.

Authorized licensed use limited to: CUNY Hunter College Library. Downloaded on February 04,2025 at 21:51:54 UTC from IEEE Xplore. Restrictions apply.

Jeopardize task deadlines - This is another *malicious* intent where compromised drones seek to violate the task deadlines running on other non-compromised drones.
Operational impairment/disruption - Under this *malicious* intent, the adversary seeks to compromise drones by causing operational impairment to them by misleading them believe that they can handle more tasks than what they actually can.

C. Taxonomy of Data Falsification

As part of the system model it is assumed that each of the N drones monitor and record its actual energy consumption for a pre-defined period/time slot/iteration of length t, denoted as $E_i^t(actual)$, consistently and independently. In the event of data falsification attack, the periodic reported energy consumption, denoted as $E_i^t(report)$ is intentionally modified from the $E_i^t(actual)$ for the compromised drones, based on the previously mentioned attack intents. The collected data is sent to the controller which the latter uses for making task offloading decisions. This iteration t is of constant length and independent of δ_t which dictates drone activity. In this work, we specifically explore two types data falsifications.
Additive: for additive falsification, $E_i^t(report) = E_i^t(actual) + \Delta_{avg}$, where the Δ_{avg} represents the false energy value injected to $E_i^t(actual)$ by the attacker.
Deductive: In deductive attack, $E_i^t(report) = E_i^t(actual) - \Delta_{avg}$. Determining the optimal value for Δ_{avg} necessitates a strategic approach to both achieve the intended goal and maintain stealthiness.

For both types, the attacker selects the injected value using a uniformly random distribution within the intervals $\Delta_{min} < \Delta_{avg} < \Delta_{max}$. The uniform random nature results less spikes in $E_i^t(report)$, and thus is less likely to be detected. If Δ_{avg} falls outside the range of Δ_{min} to Δ_{max}, we assume that the system can detect the attack. This assumption leads to further considerations regarding the attacker's intent, which we will discuss later in more detail.

D. Attacker Budget and Objective

We assume that the attacker can compromise $M_{\max}$ out of N drones, based on its budget. The scale of the attack, ζ, is calculated as $\zeta = \frac{M_{\max}}{N}$.

The impact of the attack can vary depending on the attacker's objectives. The attacker may have *long term* or *short term* damage objective. Long-term attacks are characterized by their slow and steady approach, aiming to avoid detection while building towards a significant strategic objective. On the other hand, short-term attacks are designed for maximum quick impact often at the risk of being detected almost immediately[13].

E. Threat Landscape Overview

Next, we explain how the attacker can achieve the intended goals by misreporting energy values additive through additive and deductive attacks. Note that the scale (ζ) and degree of attack (Δ_{avg}) affect the potential attack impact and detection probability. The overall attack intensity, thus can be define as:

$$I(\zeta, \Delta_{avg}) = f(\zeta) \times g(\Delta_{avg}) \tag{1}$$

and the corresponding reported energy for additive:

$$E_i^t(report) = E_i^t(actual) + I(\zeta, \Delta_{avg}) \tag{2}$$

and for deductive:

$$E_i^t(report) = E_i^t(actual) - I(\zeta, \Delta_{avg}) \tag{3}$$

Save Energy - For this intent, the compromised drone reports higher energy consumption than the actual $E_i^t(actual)$ (i.e. $E_i^t(report) > E_i^t(actual)$), i.e., an additive attack type in order to selfishly gain energy/battery advantage. This way, the selfish drone executes fewer tasks, conserves more battery, and has more residual battery level left, by successfully offloading more than its fair share of computation tasks to the edge server. The impact is measured in terms of 'Break-even time' $BET(\zeta, \Delta_{avg})$ (in number of iterations) that measures the difference in iterations between when the attack starts and when the attack effects become noticeable, i.e., the iteration where the residual drone battery level under attack starts to diverge from normal scenario. Thus, it can be measured as (from Fig. 2):

$$\text{BET}_{attacked} = \min\{t \mid B^t_{attacked} < B_{min}\} \tag{4}$$

Drain non-compromised drones - When additive falsification attacks are orchestrated from multiple drones, i.e., with $E_i^t(report) >> E_i^t(actual)$ and/or from many drones (i.e., high ζ), the controller, due to the fair task assignment, assigns fewer tasks to the compromised drones, while non-compromised ones get more than their fair share. This accelerates the battery depletion. The BET can be measured as:

$$\text{BET}_{\text{drained}} = \min_t\{t \mid \mathcal{B}r^i(t) < \tau\} \tag{5}$$

where $\mathcal{B}r^i(t)$ is resual battery and τ is a certain threshold of the initial battery level below which a drone is considered drained.
Jeopardize task deadlines - Here, the adversary compromises even greater number of drones (higher ζ) reporting $E_i^t(report) > E_i^t(actual)$, which results in the server pulling more workload than it can sustain. This cascading effect can lead to the server being unable to perform the minimum required tasks necessary ($\text{Min}_{\S^i}$) for deadline satisfaction of the non-compromised drones, eventually causing the server to violate their task deadline requirements (t_d^i). This limitation arises because of the server's finite capacity. The impact is measured as the number of iterations the attacks takes till the deadline is compromised and can be measured as:

$$\text{Iterations}_{\text{deadline}} = \min\{t \mid \frac{C_t}{N} < \text{Min}_{\S^i}\} \tag{6}$$

Operational Impairment - Under this intent, the adversary launches a deductive attack from compromised drones by under-reporting energy expenditure, i.e., $E_i^t(report) < E_i^t(actual)$, which misleads the server into assigning compromised drones more tasks than they can handle. This leads overloading the drones, eventually leading to drone operation impairment and disruptions, e.g., sudden crash due to rapid energy depletion. The impact of this attack is also measured in BET (in terms of iterations) till the drone is completely drained of energy (and not below the threshold τ). This signifies operational damage or disruption to the drone.

In this paper, in order to develop intuitions about Δ_{avg} which predominantly dictates the aforementioned attacks' impacts and attack detection probability, we choose an experimental approach, rather than analytical as the latter can become intractable due to too many system variables and their unknown characteristics.

IV. Performance Evaluation

A. Experimental environment

The proposed intent-driven attack model is evaluated, optimized, and validated using extensive 'testbed-in-the-loop' simulations. Our lab based hardware testbed mimics a collaborative IoT-edge environment where NVIDIA Jetson TX2 mimics IoT computational units and a Dell PowerEdge Tower PC mimics the edge server. In the testbed, using the simulated drone models, we create a class of heterogeneous collaborative IoT-edge environments for data falsification experiments. Details of such heterogeneous simulation provided in Table I.

Authorized licensed use limited to: CUNY Hunter College Library. Downloaded on February 04,2025 at 21:51:54 UTC from IEEE Xplore. Restrictions apply.

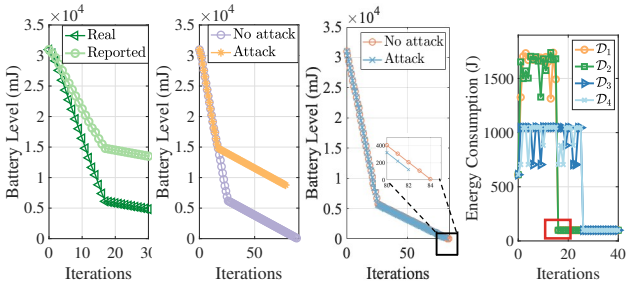


Fig. 2: Drone energy expenditure behavior under *selfish* intent attack: (a) real vs. reported for the compromised, (b) when not under attack vs. under attack for the compromised, (c) when not under attack vs. under attack for the non-compromised, (d) point of deadline jeopardizing for the compromised

B. Quantifying Attack impact

Fig. 2 illustrates additive attacks' impact (for all intents) on compromised and non-compromised drones, for $\zeta = 50\%$ scenario. In this scenario, the compromised drones, viz., $\mathcal{D}_1$ and $\mathcal{D}_2$, misreport their energy consumption by adding Δ_{avg} during each iteration (i.e., reporting period), as depicted in Fig. 2a.

As shown in Fig. 2b, under no attack conditions, i.e., no misreporting by the compromised drones, the controller identifies $\mathcal{D}_1$ to be in critically low-battery condition around iteration 26. However, when the attack is launched, such diagnosis occurs significantly earlier, specifically around iteration 18. In Fig. 2c, we compare the battery drainage effects of such attacks on drones that are not compromised (i.e., on $\mathcal{D}_3$ and $\mathcal{D}_4$),signifying *drain non-compromised drones* intent. We observe that although effects of such *drain non-compromised drones* intent are not as pronounced as *save energy* intent, the effects are felt nonetheless. Finally, Fig. 2d illustrates the attack impact of *jeopardize task deadlines* intent on compromising the task completion deadlines. Due to the limited computational capacity of the server, it can only handle a limited number of drones that are on the verge of running out of power. If the scale of the attack increases, for example, ζ equal to 50%, the server will have to prioritize its resources. This may result in sacrificing deadlines satisfaction due to limited capacity to even support the basic needs of all drones. The iteration in question is shown with a 'red' box in Fig. 2d.

C. Trade-offs for save energy intent

Fig. 3 displays the impact of falsification attacks with *save energy* intent with different Δ_{avg} values to investigate optimum Δ_{avg} and impact trade-offs. Figs. 3a and 3b show BET characteristics under homogeneous settings with identical parameters, deadlines, and YOLOv5 models, demonstrate similar energy/battery conservation among compromised drones. Figs. 3c and 3d, with different YOLOv5 models shows similar BET against different Δ_{avg} values. Such similarity can be attributed to the fact that changing the YOLOv5 models only influences the computational aspect. Figs. 3e and 3f instead show results for drone heterogeneity with same YOLOv5 models scenarios. Here, we observe longer BET characteristics (for both ζ=25% and ζ=50%) against Δ_{avg} when compared to more homogeneous setting of Figs. 3a and 3b. The reason behind the longer BET, i.e., delayed onset of attack effects, can be attributed to drone heterogeneity in terms of their battery capacities and energy consumption rates, as outlined in Table I. Even if a drone over-reports energy consumption, it might not receive less tasks than a drone with limited battery capacity. Figs. 3g and 3h demonstrates BET characteristics for fully heterogeneous systems. Notably, such scenarios show some variations when compared to the previous scenarios with limited heterogeneity. These differences arise due to model heterogeneity, which affects task offloading. The result shows when $\zeta = 25\%$, the server can handle it. But as more drones are compromised, the server must monitor all drained drones, leading to a faster BET.

TABLE I: Simulation details for ζ set at 25% and 50%

ζ=25%, 50%	Model	Battery Capacity (J)	ζ=37.5%
$\mathcal{D}_1$	Yolov5m (medium)	30960 (large)	$\mathcal{D}_1$, $\mathcal{D}_5$
$\mathcal{D}_2$	Yolov5x (extra-large)	40920 (x-large)	$\mathcal{D}_2$, $\mathcal{D}_6$
$\mathcal{D}_3$	Yolov5s (small)	15400 (small)	$\mathcal{D}_3$, $\mathcal{D}_7$
$\mathcal{D}_4$	Yolov5l (large)	21380 (medium)	$\mathcal{D}_4$, $\mathcal{D}_8$

Fig. 4 displays the *save energy* intent's BET vs. Δ_{avg} characteristics for system with ζ = 37.5%. Here, total 8 drones are simulated with 3 being compromised. We observe that for homogeneous scenario (in Figs. 4a), BET characteristics of all compromised drones follow similar trends as in for ζ = 25% and ζ = 50% scenarios. However, with greater heterogeneity in the system, as shown in Figs. 4b-d, there is a discernible impact of the attacks which is notable on $\mathcal{D}_1$ and $\mathcal{D}_2$ as they possess higher battery capacities. As the Δ_{avg} increases, the effect becomes more pronounced, leading to faster BETs.

D. Trade-offs for drain non-compromised drones intent

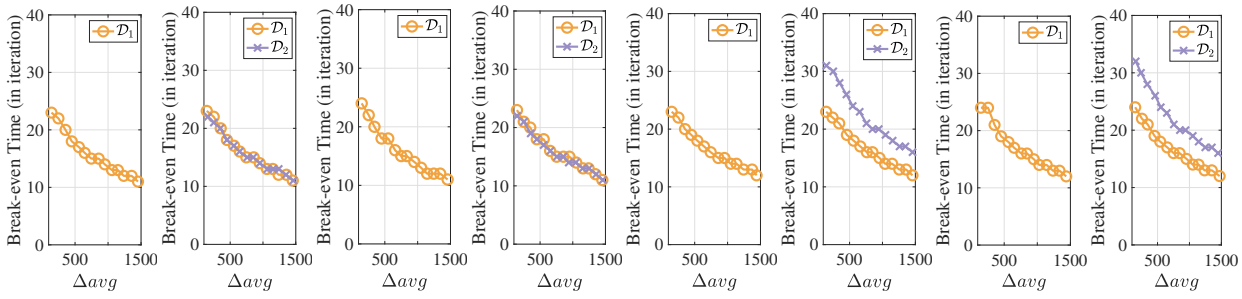
Fig. 5 provides the impact in terms of the number of iterations it takes to drain non-compromised drones' batteries to a critically low level. For fully homogeneous scenario with ζ=25% as shown in Fig. 5a, we see that ζ=25% is too small to cause any malicious effect in terms of battery drainage on non-compromised devices. When the scale of the attack increases to ζ=50% as shown in Fig. 5b, it becomes evident that for a specific range of Δ_{avg} values, the attack accelerates the drainage of non-compromised drones. However, beyond this range, an intriguing shift occurs - the non-compromised drones begin to benefit from the attack. This reversal transpires because the controller excludes the compromised drones from server collaboration in terms of workload sharing, leading to an unexpected advantage for the non-compromised drones.

Figs. 5c and 5d illustrate the model heterogeneity scenario. When Δ_{avg} increases, non-compromised drones deplete their batteries more rapidly due to increased task assignments to them. However, beyond a certain threshold, the time to battery drainage exhibits no further changes. This phenomenon mirrors the observation in Figs. 5a and 5b, where, after reaching a specific Δ_{avg}, the effect of the attack neutralizes. In Figs. 5e-5h, we show *drain non-compromised drones* intent results for diverse drones. We observe drones with higher battery levels take longer to drain, while those with smaller batteries experience a faster depletion.

E. Trade-offs for jeopardize task deadlines intent

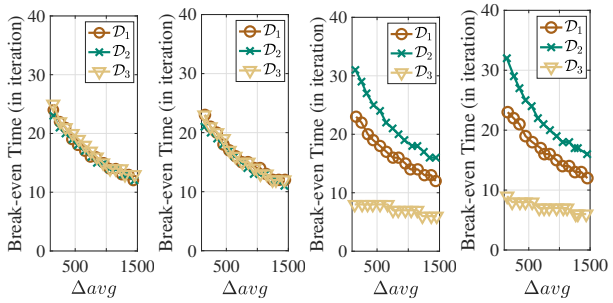
Fig. 6 demonstrates the impact of *malicious* attack on compromised drones in terms of time (i.e., number of iterations) to deadline compromise, for different Δ_{avg} values. In Fig. 6a, when ζ = 25% under homogeneous conditions, the deadline remains unaffected. However, when ζ = 50%, the server's capacity becomes insufficientthus jeopardising the task deadlines. Hence, if the attack scale increases, it affects the exact point where the deadline is jeopardized. It is observed that in Fig. 6b model heterogeneity has little impact on attack success, but drone heterogeneity (as shown in Fig. 6c and Fig. 6d) alter such

Authorized licensed use limited to: CUNY Hunter College Library. Downloaded on February 04,2025 at 21:51:54 UTC from IEEE Xplore. Restrictions apply.



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 3: *Selfish* intent attack performance vs. Δ_{avg} across various scenarios: (a) fully homogeneous system, ζ=25%, (b) fully homogeneous system, ζ=50%, (c) only model heterogeneity, ζ=25%, (d) only model heterogeneity, ζ=50%, (e) only drone heterogeneity, ζ=25%, (f) only drone heterogeneity, ζ=50%, (g) fully heterogeneous system, ζ=25%, (b) fully heterogeneous system, ζ=50%



(a) (b) (c) (d)

Fig. 4: *Save Energy* intent attack performance vs. Δ_{avg} for ζ=37.5% across various scenarios: (a) fully homogeneous system, (b) only model heterogeneity, (c) only drone heterogeneity, (d) fully heterogeneous system

behavior significantly. It accelerates the deadline compromise, i.e., iteration 11 in comparison to iteration 26. This can be attributed to the fact that the compromised drones, i.e., $\mathcal{D}_1$ and $\mathcal{D}_2$ are larger and thus necessitate more cooperation from the server when close to drainage. This causes additional workload on the server which the server cannot sustain, leading to early drainage of the compromised drones. We also observe that for heterogeneous scenarios, when two drones are compromised, the behavior is different from homogeneous scenarios. Because, they are bigger Table I) so the controller prioritizes them. This in turn makes the deadline compromise situation no more worse than it already is, even for higher Δ_{avg}.

F. *Trade-offs for* operational impairment *intent*

Fig. 7a shows the impact of *operational impairment* intent on BET vs. Δ_{avg} for fully heterogeneous system. The results show that the attack affects between intervals 100 and 200. Beyond this point, even significantly increasing Δ_{avg} has no further impact. This is because when Δ_{avg} is large, the controller assigns more tasks to the drone for the next iteration, this can lead to higher energy consumption than the other non-compromised drones, even when they are reporting less. As a result, the action and its corresponding reaction counterbalance each other. Fig. 7b demonstrates the impact of the attack on the battery drainage of non-compromised drones. Initially, D_3 benefits from this manipulation and battery life extends for one more iteration because the compromised drones executing more tasks on their own, while D_4 remains unaffected due to the system's full heterogeneity. Fig. 7c illustrates the impact of *operational impairment* intent on task deadlines. Here, we observe that there is no effect on the deadlines, even when Δ_{avg} increases significantly. The reason for this is that although the compromised drones report lower battery levels, no extra notifications are sent to the controller that their battery levels are critically low.

G. *Computing optimal* Δ_{avg}

As discussed earlier, our objective with these comprehensive experiments is to determine the optimal value for Δ_{avg} in various attack scenarios. It is important to understand that predicting the optimal value for Δ_{avg} is not possible without conducting experiments. This is because there are various factors that influence the controller's decision-making process include the number of compromised drones, their deadline, and physical characteristics. Additionally, there may be instances of falsified data which can further complicate the process of determining the optimal value for Δ_{avg}.

Table II demonstrates the optimal value of Δ_{avg} for all intent in different attack scale. In Fig.3, we can observe that the slope of the lines changes, becoming softer towards the end as having a large value for Δ_{avg} does not aid in reducing the number of iterations required for BET. Instead, it increases the risk of detection. Therefore, the ideal value for Δ_{avg} lies somewhere before the value gets too large with marginal effect. Fig. 5, shows that increasing Δ_{avg} does not continue in the same fashion. This is because reporting huge false data will be neutralized by forcing the non-compromised drones to consume more energy. As the scale of the attack increases, it could have a negative impact on some of the non-compromised ones. As depicted in Fig. 6, $\zeta = 25\%$ is not large enough to jeopardize the deadline. However, when $\zeta = 50\%$, we observe the effect. In a fully homogeneous system and only heterogeneous model, the deadline decreases over time, but the rate at which it decreases varies. In Figs. 6c and 6d, in order to impact the deadline, Δ_{avg} must be set in such as way that the compromised drone's falsified data is higher than that of the other drones.

ACKNOWLEDGEMENTS

This work supported by the National Science Foundation under Award Number: CNS-1943338 and CNS-2401928.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we investigated and proposed an intent-driven data falsification attack model for collaborative IoT-edge environments, covering both selfish and malicious intents. Our theoretical intuitions are evaluated and validated by 'testbed-in-the-loop' simulations that revealed significant effects of data falsification attack, with upto 50% for intents that are selfish in nature, and upto 61% for intents that are malicious in nature, depending on system parameters. In future, we seek to

Authorized licensed use limited to: CUNY Hunter College Library. Downloaded on February 04,2025 at 21:51:54 UTC from IEEE Xplore. Restrictions apply.

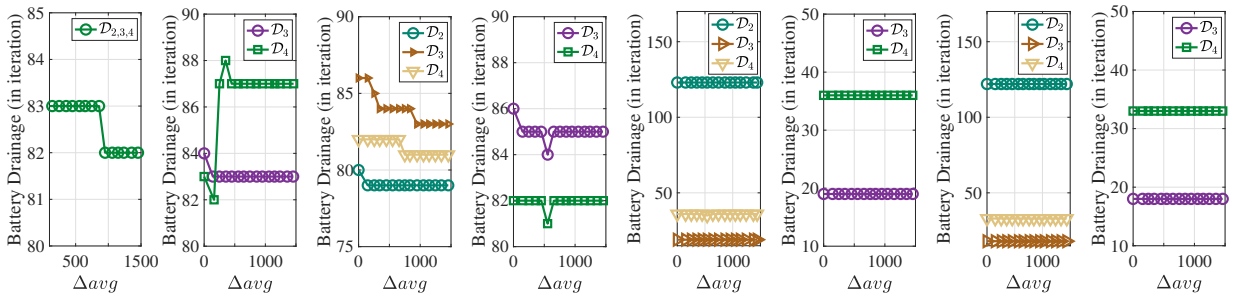


Fig. 5: *Drain non-compromised drones* intent attack performance vs. Δ_{avg} across various scenarios: (a) fully homogeneous system, ζ=25%, (b) fully homogeneous system, ζ=50%, (c) only model heterogeneity, ζ=25%, (d) only model heterogeneity, ζ=50%, (e) only drone heterogeneity, ζ=25%, (f) only drone heterogeneity, ζ=50%, (g) fully heterogeneous system, ζ=25%, (b) fully heterogeneous system, ζ=50%

TABLE II: Optimum value of Δ_{avg} in different scenarios

Attacker Intent	Attack Scale (ζ)	Fully homogeneous	Only model heterogeneity	Only drone heterogeneity	Fully heterogeneous
Save energy	$\zeta = 25\%$	646	750.3	859.58	759.4
	$\zeta = 37.5\%$	847.9	816.83	857.7	826.1
	$\zeta = 50\%$	644.5	755.85	759.95	753.5
Drain	$\zeta = 25\%$	950.7	413.5	-	-
	$\zeta = 37.5\%$	752.3	847.8	948.46	950.26
	$\zeta = 50\%$	152	554.65	-	-
Jeorpardise	$\zeta = 25\%$	-	-	-	-
	$\zeta = 37.5\%$	950.9	750.59	948.46	1047.65
	$\zeta = 50\%$	760.1	757	1354	1242.21
Operational	$\zeta = 50\%$	-	-	-	144.56

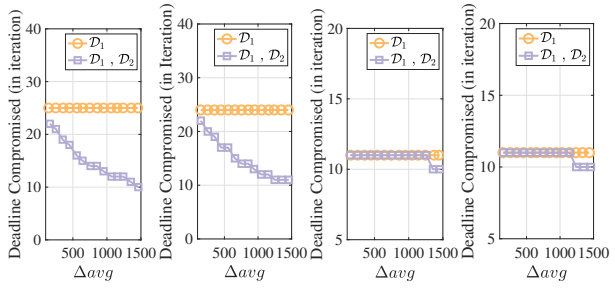


Fig. 6: *Jeopardize task deadlines* intent attack performance vs. Δ_{avg} across various scenarios: (a) fully homogeneous system, (b) only model heterogeneity, (c) only drone heterogeneity, (d) fully heterogeneous system

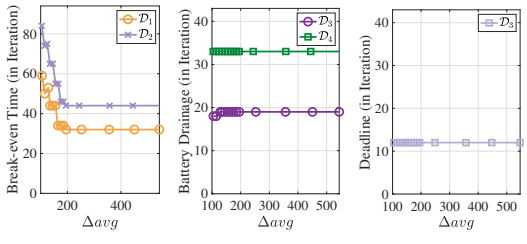


Fig. 7: *Operational impairment* intent attack performance vs. Δ_{avg} on (a) BET, (b) battery consumption of non compromised drones, and (c) deadline

explore a detailed analytical model for the attacks to support the experimental claims. Overall, the findings of this paper emphasize the need to address such consequences of data falsification attacks in collaborative IoT-edge environments in more detail by developing robust detection techniques and exploring resilient defense mechanisms.

REFERENCES

[1] X. Zhang and S. Debroy, "Energy efficient task offloading for compute-intensive mobile edge applications," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, 2020.

[2] M. Ahmed and A. S. Barkat Ullah, "False data injection attacks in healthcare," in *Data Mining: 15th Australasian Conference, AusDM 2017, Melbourne, VIC, Australia, August 19-20, 2017, Revised Selected Papers 15*, pp. 192–202, Springer, 2018.

[3] S. Bhattacharjee, S. Debroy, and M. Chatterjee, "Quantifying trust for robust fusion while spectrum sharing in distributed dsa networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 2, pp. 138–154, 2017.

[4] J. K. Stolaroff, C. Samaras, E. R. O'Neill, A. Lubers, A. S. Mitchell, and D. Ceperley, "Energy use and life cycle greenhouse gas emissions of drones for commercial package delivery," *Nature communications*, vol. 9, no. 1, p. 409, 2018.

[5] X. Zhang, M. Mounesan, and S. Debroy, "Effect-dnn: Energy-efficient edge framework for real-time dnn inference," in *2023 IEEE 24th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 10–20, IEEE, 2023.

[6] M. Mounesan, X. Zhang, and S. Debroy, "Edgerl: Reinforcement learning-driven deep learning model inference optimization at edge," *arXiv preprint arXiv:2410.12221*, 2024.

[7] Y. O. Sharrab and N. J. Sarhan, "Aggregate power consumption modeling of live video streaming systems," in *Proceedings of the 4th ACM Multimedia Systems Conference*, pp. 60–71, 2013.

[8] M. M. Hossain, M. Fotouhi, and R. Hasan, "Towards an analysis of security issues, challenges, and open problems in the internet of things," in *2015 ieee world congress on services*, pp. 21–28, IEEE, 2015.

[9] F. Kohnhäuser, N. Büscher, S. Gabmeyer, and S. Katzenbeisser, "Scapi: a scalable attestation protocol to detect software and physical attacks," in *Proceedings of the 10th ACM conference on security and privacy in wireless and mobile networks*, pp. 75–86, 2017.

[10] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, "Iot: Internet of threats? a survey of practical security vulnerabilities in real iot devices," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8182–8201, 2019.

[11] N. Nissim, R. Yahalom, and Y. Elovici, "Usb-based attacks," *Computers & Security*, vol. 70, pp. 675–688, 2017.

[12] B. Bhushan, G. Sahoo, and A. K. Rai, "Man-in-the-middle attack in wireless and computer networking—a review," in *2017 3rd International Conference on Advances in Computing, Communication & Automation (ICACCA)(Fall)*, pp. 1–6, IEEE, 2017.

[13] S. Bhattacharjee, A. Thakur, S. Silvestri, and S. K. Das, "Statistical security incident forensics against data falsification in smart grid advanced metering infrastructure," in *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy*, pp. 35–45, 2017.

Authorized licensed use limited to: CUNY Hunter College Library. Downloaded on February 04,2025 at 21:51:54 UTC from IEEE Xplore. Restrictions apply.