SNPSG Journal of Intelligent Unmanned Systems

2026, Vol. 1, No. 1, Article 113
DOI: https://doi.org/10.65218/jius.2026.113

RESEARCH ARTICLE

Deep Learning-Based Estimation of UAV-Swarm Communication Constraints

Laércio Lucchesi[1,*], Bruno José Olivieri de Souza[1,*], Paulo Ivson[1] and Markus Endler[1]

1 Department of Informatics, Pontifical Catholic University of Rio de Janeiro (PUC-Rio), Rio de Janeiro, Brazil

* Corresponding author(s): Laércio Lucchesi and Bruno José Olivieri de Souza

Email(s): llucchesi@inf.puc-rio.br, bolivieri@inf.puc-rio.br

Received: 05 March 2026 | Accepted: 01 June 2026 | Published Online: 3 July 2026

Abstract: Autonomous UAV swarms performing coordinated missions depend critically on communication reliability, but direct measurement of the underlying parameters—packet failure rate and maximum communication range—typically requires radio-level instrumentation that is unavailable in legacy or heterogeneous fleets. We address this gap with a non-intrusive, learning-based estimator that infers these two parameters from observables already produced by a consensus-based formation controller, with no access to the radio stack. The estimator combines a compact set of formation-level metrics (geometric deviation, coordination-state availability, member participation), temporal summarisation of their traces, and a small fully connected neural network. We validate the approach in simulation, using Raft-based decentralised control in GrADyS-SIM NextGen across a wide range of swarm sizes and communication conditions, and compare against a multiple linear regression baseline. The proposed estimator achieves $R^2 > 0.83$ for both targets, clearly outperforming the linear baseline ($R^2 = 0.73$ and $R^2 = 0.54$), which empirically supports the central claim that compact, control-aware observables are sufficient to recover hidden communication parameters. The contributions are: (i) a non-intrusive parameter-inference framework that does not rely on radio-level features; (ii) independent estimation of two coupled communication parameters from the same compact feature set; and (iii) a reproducible simulation-based evaluation across heterogeneous communication and mission conditions.

Keywords: ***Communication constraint estimation, Non-intrusive monitoring, Swarm robotics, Aerial systems, Deep learning methods, Consensus algorithms***

I. Introduction

Unmanned Aerial Vehicle (UAV) swarms in coordinated formations are increasingly used in mission-critical tasks such as surveillance, search and rescue, environmental monitoring, precision agriculture, and military operations [2, 6, 13, 27]. We focus on formation-control settings where preserving relative positioning (within a tolerance) is required during trajectory tracking. In such settings, mission success depends on the swarm's ability to maintain stable formations under adverse communication conditions, including limited range, delays, and intermittent loss [4, 5].

The problem. Formation stability is known to degrade quickly when packet loss grows or the inter-UAV distance approaches the maximum communication range [4, 5]. Yet, in operational deployments, the parameters that drive these failure modes—per-link packet failure rate and maximum effective communication range—are rarely visible to the controller. Direct measurement requires either physical-layer instrumentation (RSSI/SNR logging, channel sounding) or active probing, which is often unavailable in legacy hardware, heterogeneous fleets, or bandwidth-constrained networks. The mission, however, must continue, and decisions about formation tightness, fallback behaviour, or relay selection should arguably be informed by an estimate of these very parameters. This is the gap our work addresses.

Why a compact set of formation-level metrics? Existing learning-based approaches that touch on UAV-swarm communication tend to consume rich, radio-specific features—signal strength, power loss, routing overhead—and to focus on *adaptation* (closing the loop on action) rather than *estimation* (recovering the unobserved parameters). We pursue the opposite question: are the observables that a consensus-based formation controller already produces—geometric deviation, coordination-state availability, member participation—sufficient on their own to recover the hidden communication parameters? Our central hypothesis, which the rest of the paper validates, is that they are. The novelty is therefore not in inventing new metrics, but in showing that compact, control-aware observables are *sufficient*, and quantifying how much better than a linear baseline a small neural model can do with them.

We consider a mission in which the swarm follows a figure-eight trajectory composed of two tangent loops. Formations switch between circular and inline, which stresses real-time coordination (Fig. 1). Individual UAVs may also leave to recharge at a base station and later rejoin, increasing formation complexity.

To ground the inference framework, we adopt a decentralised consensus-based formation control architecture using Raft [20], which maintains leadership and state agreement under message loss, departures, or temporary disconnections. The choice of Raft is deliberate: beyond avoiding the single point of failure common in leader-follower models, it exposes control-state health signals (e.g., leader availability) that are observable to all participants without tapping into lower-level radio diagnostics, making it a natural platform for non-intrusive monitoring. The same idea has recently been explored for cooperative formation control [24], which reinforces the relevance of consensus-based coordination as a substrate for our estimator. While the present study focuses on Raft, the broader methodology—using consensus-coordination outcomes as proxies for communication quality—generalises to other decentralised control protocols.

All simulations were performed using the GrADyS-SIM NextGen framework[1], a network-oriented simulator for dis-

[1] https://project-gradys.github.io/gradys-sim-nextgen/

DOI: https://doi.org/10.65218/jius.2026.113

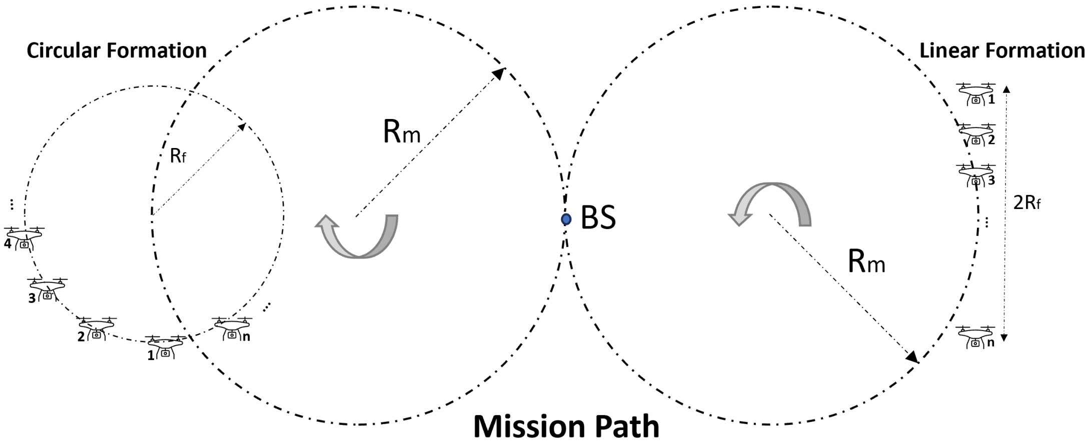

Fig. 1 Mission path with alternating circular and linear UAV formations.

tributed autonomous systems. Recent work has emphasised the importance of principled simulator selection and experiment design for UAV-swarm research [18], a methodological concern we share. It models communication delay, range limits, message loss, and UAV dropout. We vary swarm size, communication parameters, and formation types.

We define three evaluation metrics: (i) *shape_error*, deviation from ideal geometry; (ii) *leader_off*, fraction of time without a Raft leader; and (iii) *presence_error*, discrepancy between expected and actual participation. These metrics capture coordination quality and communication impact (Section III).

Despite advances in UAV communication and formation control, it is unclear whether a compact set of swarm metrics—*shape_error*, *leader_off*, *presence_error*, *size_formation*, and *comm_delay*—can infer two key communication parameters: failure rate and maximum range. Unlike delay, which can be measured via round-trip time (RTT), these parameters co-occur and are hard to disentangle.

We study whether a data-driven model can estimate these parameters from the compact feature set. We use a fully connected network that consumes temporally summarized stability metrics plus static features.

Our main contributions are:

- A non-intrusive inference approach that relies solely on measurable inputs (*shape_error*, *leader_off*, *presence_error*, *size_formation*, and *comm_delay*) to predict failure rate and communication range.
- A learning-based method to disentangle the joint effects of failure rate and maximum range, enabling independent estimation of both parameters.
- A simulation-based validation across diverse communication conditions and mission configurations, with a linear regression baseline for comparison.

Results show high predictive accuracy ($R^2 > 0.83$ for both targets) and low test loss, indicating that the proposed estimator can recover key communication parameters from minimal, conventional inputs.

The remainder of this paper is structured as follows. Section II reviews related research on UAV swarm communication and formation control. Section III defines the proposed performance metrics and problem formulation. Section IV details the neural network architectures and training strategy. Section V describes the simulation environment and experimental configuration. Section VI presents experimental results and analysis. Section VII discusses threats to validity. Section VIII outlines practical implications for deployment. Section IX concludes the paper and suggests directions for future work.

II. Related Work

Prior work applies machine learning to UAV swarm communication, including reinforcement learning, random forests, and federated learning. For example, [3] analyzes packet delivery ratios and throughput in linear formations, while [8] uses random forests to predict signal strength in triangular swarms. Deep reinforcement learning approaches are proposed in [11, 21] for coordination under variable communication conditions.

Most approaches rely on rich features beyond formation geometry, such as signal strength, power loss, or routing overhead [26, 28]. [19] proposes a multi-agent RL framework for relay selection under jamming. [10] combines DRL with LSTM to improve throughput and convergence. [15] uses deep learning to adjust UAV positions in circular and conical formations.

These studies largely address *adaptation* (choosing actions to improve communication or coordination) rather than *estimation* (inferring latent communication parameters). When estimation is included, it typically depends on features that require radio-level access or additional probing traffic, which limits applicability in constrained or heterogeneous hardware environments.

A complementary body of work in control engineering studies how UAV formation controllers should behave when communication is impaired. Vazquez Trejo et al. [1] design an event-triggered leader-following controller that remains stable under communication faults; Ji et al. [7] address communication delays and external disturbances via a dynamic event-triggered sliding-mode formulation; and Wu et al. [25] target multi-UAV systems with unstable topologies and hybrid delays. James et al. [5] and Hudak et al. [4] provide empirical evidence that packet loss and path loss directly degrade swarm coherency, motivating the choice of *comm_failure* and *comm_range* as the estimation targets in this work. Saulnier et al. [22] introduce a metric of resilience for mobile-robot teams, framing connectivity as

DOI: https://doi.org/10.65218/jius.2026.113

a controllable property of the team rather than as an external nuisance. These works share with our setting the assumption that communication quality is critical and difficult to instrument, but they aim to *control* or *characterise* the system rather than to *infer* unobserved parameters from coordination-level observables.

Our work is positioned at the intersection of formation monitoring and communication inference. By restricting inputs to high-level formation metrics, we aim to show that useful link-quality proxies can be derived without instrumenting the radio stack, enabling deployment in legacy systems or in mixed-fleet scenarios where low-level data is unavailable.

Table 1 summarizes recent studies by formation type, learning architecture, communication parameters, and metrics. In contrast to these approaches, we focus on estimating communication parameters from a compact set of formation-based metrics. The next section formalizes our scenario and the measurable variables used for inference.

Table 1 ML Approaches for UAV Swarm Communication

Ref.	Formation type	ML architecture	Performance metrics
[3]	Linear	N/A	Mission time, energy efficiency
[8]	Triangular	RF[a], K-means	Euclidean distance
[10]	N/A	DRL[b], LSTM[c]	Higher throughput
[11]	N/A	DRL[b]	Delay mitigation
[13]	Circular	N/A	Leadership error, formation error
[15]	Circular, conical	NN[d]	Position deviation
[19]	N/A	MARL[e], DRL[b]	BER[i], energy use, SINR[j]
[21]	Diamond (L-F[f])	DRL[b], DQN[g]	Convergence rate
[26]	N/A	DQN[g]	Waypoint tracking error
[28]	L-F[f]	Fed. learning[h]	Convergence rate, energy consumption

Notes: [a]Random Forest, [b]Deep Reinforcement Learning, [c]Long Short-Term Memory, [d]Neural Network, [e]Multi-Agent RL, [f]Leader-Follower, [g]Deep Q-Network, [h]Federated Learning, [i]Bit Error Rate, [j]Signal-to-interference and noise ratio.

Recent efforts have also strengthened the ecosystem around evaluation and validation. Lamenza et al. [16] describe GrADyS-SIM NextGen environments with varying realism levels, supporting externally parameterized delay, range, and loss rates during each run. Earlier distributed coordination strategies for dynamic UAV groups were explored by Olivieri and Endler [17], addressing node departures and re-entries during aerial data collection. Complementary real-world FANET experiments reported in [23] reinforce the practical relevance of simulation-based assessments.

Focusing on formation-level indicators under Raft-based coordination, Lucchesi et al. [14] show through extensive simulation that *shape_error*, *presence_error*, and leader-absence time capture communication-induced degradation; this supports our objective of inferring *comm_range* and *comm_failure* from compact, high-level observables.

Although these works demonstrate the potential of learning-based techniques for UAV swarm coordination, they generally employ rich input feature sets and focus on adaptation rather than inference. To the best of our knowledge, no prior research has investigated whether a limited set of measurable swarm metrics—such as *shape_error*, *leader_off*, *presence_error*, *size_formation*, and *comm_delay*—is sufficient to infer underlying communication characteristics such as failure rate and maximum communication range. This gap motivates the present study.

III. Problem Definition

This section formalizes the mission scenario, formation patterns, coordination mechanisms, and evaluation metrics used in our study. It also defines the learning objective: infer communication parameters from measurable swarm indicators.

A. Mission and Formations

The UAV swarm follows a planar figure-eight trajectory formed by two tangent circles of radius Rm (Fig. 1). This path requires adaptation during direction reversals and stresses coordinated maneuvering.

Two formation types are used: *circular*, with UAVs evenly spaced on a circumference, and *linear*, with UAVs aligned on a straight line.

UAVs may leave to recharge at a ground Base Station (BS) and later rejoin. Formations adapt to the number of active UAVs.

The mission parameters (trajectory geometry, speed, and formation spacing) are fixed within a run, while communication parameters vary across runs. This separation allows us to isolate communication effects from intentional changes in formation design.

B. Raft Implementation

Coordination is maintained through Raft-lite, a simplified Raft variant [20] adapted from [9]. Raft-lite maintains a shared status vector indicating UAV availability (active or recharging). Leader elections occur as needed, and heartbeat updates keep formation state consistent. *presence_error* and *leader_off* are derived from this status vector.

The Raft implementation exchanges periodic state updates that are already required for coordination. We treat these updates as a given communication load, and we do not introduce additional probing or diagnostic traffic during data collection.

C. Proposed Metrics

We define three metrics to quantify swarm stability:

- *shape_error*: Deviation from ideal inter-UAV and UAV-to-center distances, normalized between 0 and 1.
- *leader_off*: Fraction of time the swarm operates without an elected Raft leader.
- *presence_error*: Normalized Hamming distance between the expected and actual UAV participation.

 DOI: https://doi.org/10.65218/jius.2026.113

These metrics are intentionally heterogeneous: *shape_error* captures geometric deviation, *leader_off* captures control-state availability, and *presence_error* captures membership consistency. Together, they provide complementary signals about how communication constraints propagate into coordination. The heterogeneity is a deliberate design choice: rather than relying on a single connectivity index (such as the resilience-of-network-topology metric proposed for resilient flocking by Saulnier et al. [22]) or on the structural reliability indices surveyed by Zaitseva et al. [29], we use three metrics that target different layers of the coordination stack (geometry, control state, membership), which empirically helps the estimator disentangle the joint effects of *comm_failure* and *comm_range*.

Assume there are M key distances, i.e., distances between adjacent UAVs and between each UAV and the formation's center of mass. Let d_k^{ideal} and d_k^{real} denote the ideal and real distances. The unnormalized *shape_error* is first computed as:

$$shape\_error_{unnorm} = \frac{1}{M}\sum_{k=1}^{M}\frac{\left|d_k^{real} - d_k^{ideal}\right|}{d_k^{ideal}} \tag{1}$$

Then, it is normalized to the [0, 1] range:

$$shape\_error = \frac{shape\_error_{unnorm}}{1 + shape\_error_{unnorm}} \tag{2}$$

Let T_{leader_off} be the total time during which no Raft leader is elected, and T_{total} be the overall duration. Then, the already normalized metric is given by:

$$leader\_off = \frac{T_{leader\_off}}{T_{total}} \tag{3}$$

Let $\mathbf{s}^{ideal} \in \{0,1\}^n$ be the ideal status vector and $\mathbf{s}^{real} \in \{0,1\}^n$ be the measured status vector, where each element indicates whether a UAV is active (1) or recharging (0). The *presence_error* is the normalized Hamming distance:

$$presence\_error = \frac{1}{n}\sum_{i=1}^{n}\left|s_i^{ideal} - s_i^{real}\right| \tag{4}$$

D. Learning Objective

The goal is to estimate *comm_range* and *comm_failure* from high-level swarm metrics that the formation controller already produces. We choose these two parameters deliberately. Empirical studies of UAV swarm coherency identify them as the two communication-side properties that most directly drive coordination failures: increased per-link packet loss disrupts agreement and consensus [5], and shrinking maximum range fragments the communication graph as the formation manoeuvres [4]. *comm_delay*, by contrast, can be measured reliably from coordinator heartbeats via round-trip time and is therefore an *input* to the estimator rather than a target. *comm_range* and *comm_failure*, however, are jointly latent: their effects on formation observables are conflated (both can manifest as failed leader elections or geometric drift), which is precisely why disentangling them with a single, uniform input set is non-trivial.

Inputs are *size_formation*, *comm_delay*, *shape_error*, *leader_off*, and *presence_error*. We frame the task as supervised multi-output regression, learning a mapping $f(\mathbf{x}) \rightarrow (comm\_range, comm\_failure)$, where $\mathbf{x}$ is the compact feature vector. The deployment value of this mapping is operational rather than purely scientific: a system that estimates *comm_range* and *comm_failure* on the fly enables informed decisions about formation tightness, conservative fallback patterns, or relay-selection triggers, decisions that delay alone is not sufficient to support.

We treat *comm_range* as the maximum effective communication distance under the simulator's propagation model, and *comm_failure* as the probability of message loss. These parameters are fixed within a run, but the observed metrics evolve over time, motivating the temporal summarisation described next.

IV. Proposed Method

This section describes dataset generation, preprocessing, temporal summarization, and model architecture.

A. Scope and Observables

Beyond estimation accuracy, we emphasize deployability: the estimator only uses variables that already appear in nominal coordination loops, and it can run on-board or at the edge with minimal computational overhead.

We use *non-intrusive* to denote methods that avoid (i) direct wireless instrumentation (e.g., physical/link layer (PHY/MAC) measurements, received signal strength indicator/signal-to-noise ratio (RSSI/SNR) logging, channel sounding), (ii) access to internal protocol state beyond what formation control already needs, or (iii) probing traffic that changes network load. Our estimator uses only high-level observables from nominal state exchange.

The metric *leader_off* is Raft-specific and couples the method to this architecture. More generally, it represents a *control-state health* signal (e.g., controller availability or coordination validity). Future work will test protocol-agnostic substitutes and quantify the impact of removing it.

The relationship between communication quality and formation metrics is not strictly unidirectional. Metrics are also affected by control design, vehicle dynamics, and disturbances. Thus, the estimator learns statistical associations conditioned on the controller and scenario distribution, motivating validation with higher-fidelity dynamics and more realistic wireless models.

Scope and validity. The estimator is calibrated on Raft-lite and GrADyS-SIM NextGen. Reported performance is therefore conditional on this control stack, communication model, and scenario distribution; we treat the model as a practical monitor within the tested regime, not a universal link-quality estimator.

Our estimator is designed for continuous monitoring rather than one-off diagnosis. In this role, relative changes over time can be as useful as absolute accuracy, enabling early detection of degrading connectivity before formation control fails outright.

The input features are group-level but can be computed in a decentralized manner using the same state exchange required by formation control. Each UAV can compute a local geometric residual (contributing to *shape_error*) from neighbor-relative positions, and the swarm can aggregate values via low-rate consensus, a coordinator, or an edge observer. *presence_error* can be approximated from neighbor lists and timeouts, while *leader_off* comes from the coordination layer. Under severe disruption, observables may be missing; in that regime, the estimator is intended for early warning rather than perfect identification.

DOI: https://doi.org/10.65218/jius.2026.113

Training labels (*comm_failure* and *comm_range*) are available in simulation as controlled parameters. This follows standard simulation-to-reality practice. To make the dataset informative, the design should cover the operating envelope and use stratified splits to avoid leakage (e.g., disjoint seeds and parameter combinations across train/validation/test). The temporal summarization and FCN are not tied to a specific learner; recurrent or temporal convolutional models could replace the FCN when fine-grained dynamics are needed.

B. Dataset

The dataset was generated with GrADyS-SIM NextGen. Each run is parameterized by *size_formation*, *comm_delay*, *comm_range*, and *comm_failure*, and produces time series of *shape_error*, *presence_error*, and *leader_off*, recorded at 5 ms over 5 minutes.

Each simulation run yields a single labeled sample with fixed communication parameters and time-varying observables. This design keeps labeling unambiguous while still capturing transient effects caused by formation transitions and UAV departures.

C. Data Preprocessing

CSV files are parsed to extract static parameters and swarm metrics. Controlled noise is added to *comm_delay* to emulate measurement uncertainty. Static features are normalized to their maximum values. Time series are already in [0,1] and aligned to a shared timestamp. The dataset is stored in a compressed NumPy archive (NPZ) file.

D. Feature Importance via SHAP

SHAP (SHapley Additive exPlanations) [12] analysis indicates *size_formation* is negligible, so it is removed without loss of performance.

We use SHAP here as a lightweight diagnostic rather than a definitive causal analysis. The removal of *size_formation* reduces dimensionality and suggests that the remaining metrics already encode the dominant effects of swarm size for the scenarios considered.

E. Temporal Summary

To compress long time series into a fixed-size input, we apply a *Temporal Summary*. Each metric (*shape_error*, *leader_off*, *presence_error*) is summarized by windowed means, reducing 60,000 samples to a small set of multi-scale features. Figure 3 illustrates an example in which a 5-minute trace is reduced to four values corresponding to the last 30s, 120s, 210s, and 300s. The resulting summaries (four per metric) are concatenated with *comm_delay* and *size_formation*, yielding a 14-dimensional input.

We compute n variable-sized windows to capture short- and long-term patterns. Feeding full traces is impractical, while a single average discards dynamics.

This design trades temporal resolution for compactness, making it feasible to train on long runs without recurrent models. It also allows the model to emphasize recent behavior (short windows) while retaining longer-term context (large windows).

Let T be the total time, n the number of windows, and *min_ratio* the shortest window proportion. Window sizes w_i are computed as:

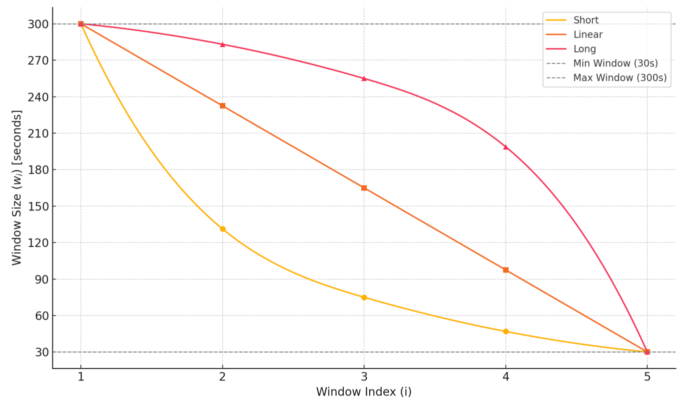

Fig. 2 Temporal Summary Windows ($n = 5, T = 300\text{s}, min\_ratio = 0.1$).

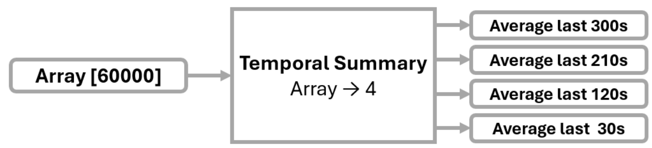

Fig. 3 Temporal Summary Architecture

(a) **Base scaling:**

$$\textit{short:}\quad \alpha_i^b = \frac{1}{i} \tag{5}$$

$$\textit{linear:}\quad \alpha_i^b = 1 - \frac{i-1}{n} \tag{6}$$

$$\textit{long:}\quad \alpha_i^b = 1 + \frac{1}{n} - \frac{1}{n-i+1} \tag{7}$$

(b) **Normalization:**

$$\alpha_i = min\_ratio + (1 - min\_ratio) \cdot \frac{\alpha_i^b - \frac{1}{n}}{1 - \frac{1}{n}} \tag{8}$$

(c) **Window size:**

$$w_i = \lfloor T \cdot \alpha_i \rfloor \tag{9}$$

The summarization mode (*short*, *linear*, or *long*) is selected via cross-validation (Section V). Figure 2 illustrates a window distribution for $T = 300$s, $n = 5$, and $min\_ratio = 0.1$.

F. Model Architecture

The model receives a 14-dimensional input composed of summarized temporal metrics and static features (Fig. 4). The three time-dependent metrics are each compressed into four values, and *comm_delay* and *size_formation* are appended. The vector feeds a four-layer fully connected network (FCN) with layer widths 256, 128, 64, and 32, using LeakyReLU and dropout. The output layer has two linear neurons predicting *comm_range* and *comm_failure*.

We train the two outputs jointly to exploit shared structure between the targets, while allowing the final linear heads to map to different scales. This multi-output regression reduces training overhead and encourages the network to learn a common representation of communication-induced coordination effects.

Figure 4 illustrates the complete architecture, from feature processing to output prediction.

 DOI: https://doi.org/10.65218/jius.2026.113

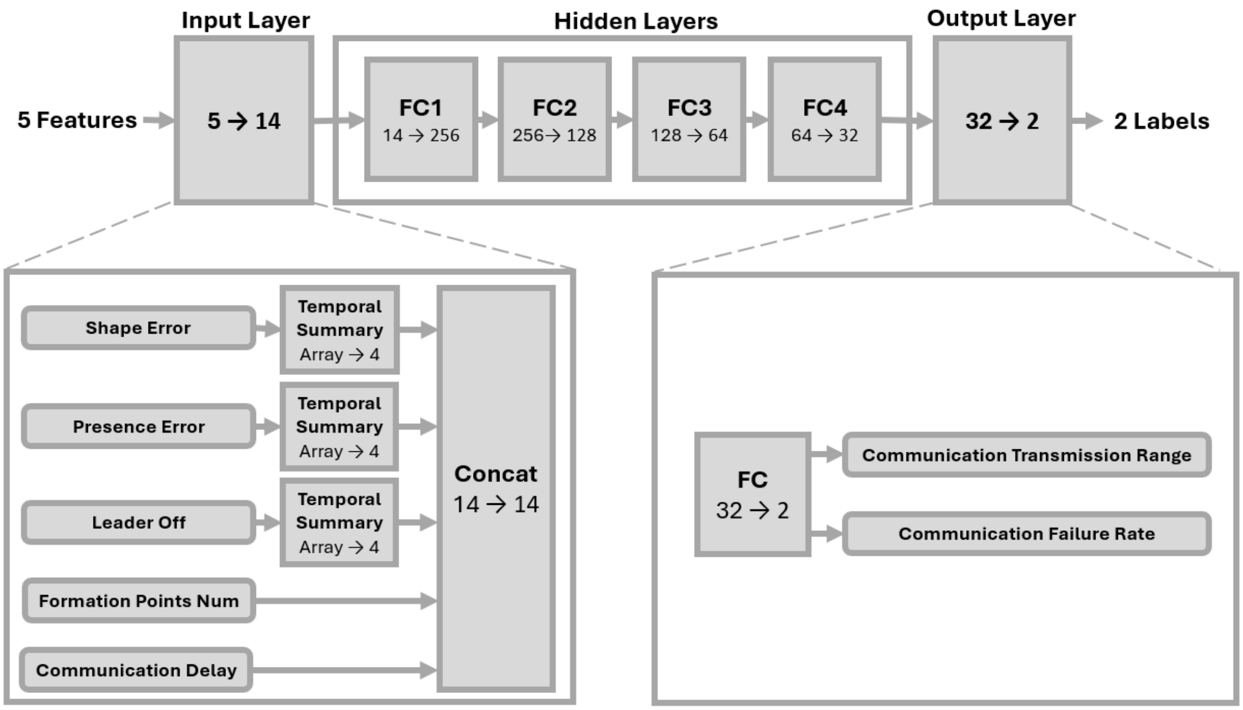

Fig. 4 Model architecture showing temporal summarization and FCN layers.

G. Baseline Model

We use a multiple linear regression baseline with the same 14 inputs. Each target is trained separately with an 80/20 split. Performance is reported with MSE and R^2, matching the deep model.

The baseline serves as a sanity check for linear separability and establishes a lower bound on performance when nonlinear interactions are ignored.

V. Experimental Setup

This section describes the simulation environment, data generation, preprocessing, temporal summarization, and training configuration. It also defines the evaluation metrics and reproducibility setup, bridging the methodology to the empirical results.

A. Simulation Environment and Data Generation

The dataset was generated with GrADyS-SIM NextGen. Each 5-minute run alternates circular and linear formations along a figure-eight path, with UAVs occasionally leaving to recharge.

Each simulation varies *size_formation*, *comm_delay*, *comm_range*, and *comm_failure*. We record *shape_error*, *leader_off*, and *presence_error* at 5 ms (60,000 samples per metric). Static inputs and ground-truth targets are stored per run.

A single run constitutes one dataset sample (Table 2). We generated 5,858 samples spanning *size_formation* 8–128, *comm_range* 20–60 m, and *comm_failure* 0.00–0.20. The dataset is split into train/validation/test (80/10/10).

B. Data Preprocessing

We follow Section IV.C, adding noise to *comm_delay*, normalizing static parameters, and consolidating aligned time series into NPZ. SHAP analysis shows *size_formation* has negligible importance for both targets, so it is removed without degrading performance, suggesting scalability with respect to swarm size.

C. Temporal Summarization Strategies

As detailed in Section IV.E, Temporal Summary compresses each long time series into compact features. Two hyperparameters control the mechanism: number of windows n and windowing mode. We evaluate *short*, *linear*, and *long*. Each metric is summarized into n averages.

We select n and windowing mode via 10-fold cross-validation, using mean validation loss and mean R^2. Figure 5 shows both metrics versus n. The best configuration is $n = 4$ with linear mode. Multiple windows outperform a single global average.

The cross-validation procedure is used strictly for hyperparameter selection of the Temporal Summary, not for final evaluation. This ensures the reported test results reflect a single, fixed configuration chosen without peeking at the test set.

Cross-validation is used only to select the Temporal Summary configuration; no fold models are reused. After selecting four linear windows, we retrain a final model on the 80/10/10 split. The next subsection details training and evaluation.

D. Model Training and Evaluation Metrics

The model is implemented in PyTorch and trained with Smooth L1 loss. We use AdamW with learning rate 0.001, weight decay 0.0001, and ReduceLROnPlateau scheduling. Early stopping with patience 100 prevents overfitting.

We report mean squared error (MSE) and the coefficient of determination (R^2) for each target, as well as mean R^2 across targets and the combined test loss. These metrics capture generalization and the strength of learned relationships.

VI. Results and Discussion

We evaluate whether the compact indicators—*shape_error*, *leader_off*, *presence_error*, and *comm_delay*—can predict *comm_range* and *comm_failure*.

 DOI: https://doi.org/10.65218/jius.2026.113

Table 2 Structure of a dataset sample

Role	Name	Data Type	Source
Features	Formation Points Num	int (scalar)	Design Parameter
	Shape Error	float [60000] ∈ [0, 1]	Measured data collected over 5 minutes, sampled at 5ms (total of 60,000 time steps)
	Leader Off	int [60000] ∈ {0, 1}	
	Presence Error	float [60000] ∈ [0, 1]	
	Comunication Delay	float (s)	Estimated (Round Trip Time)
Labels	Comunication Transmission Range	float (m)	Predicted
	Comunication Failure Rate	float (%)	

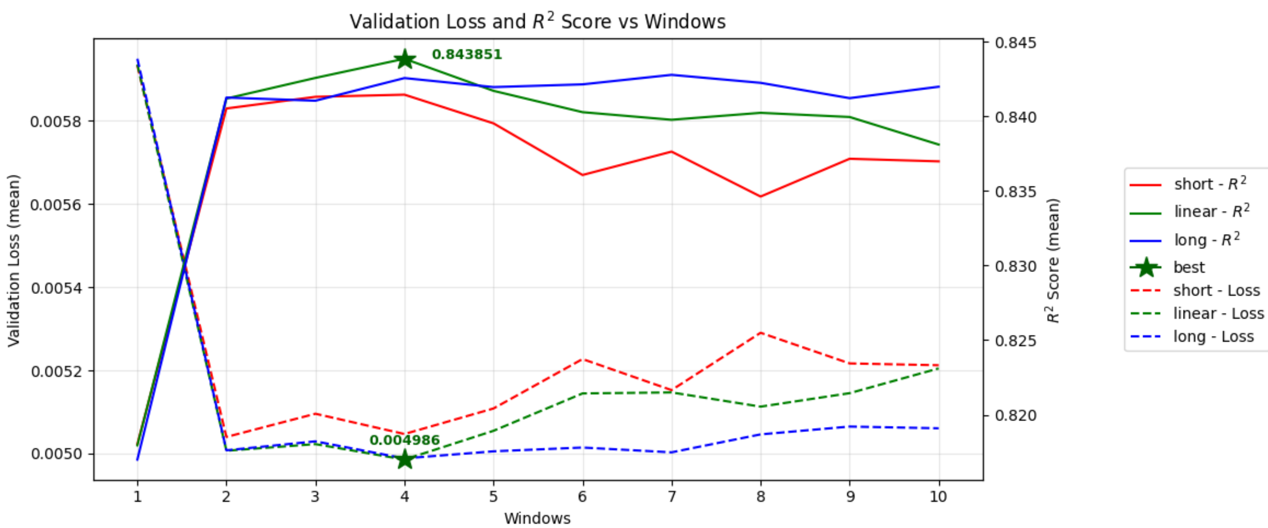

Fig. 5 Validation loss and R^2 score vs. number of summary windows for each summarization mode.

We report mean squared error (MSE) and R^2 per target, plus the global test loss. The final model uses the Temporal Summary configuration with four linear windows.

The final model achieved a test loss of 0.0043. Per target:
comm_range: $R^2 = 0.8315$, MSE = 0.0083 (5.47 m)
comm_failure: $R^2 = 0.8927$, MSE = 0.0101 (2.01%)

The linear regression baseline performs worse: MSEs of 0.0133 (6.92 m) for *comm_range* and 0.0436 (4.18%) for *comm_failure*, with R^2 scores of 0.727 and 0.544. This gap highlights the FCN's ability to capture nonlinear dependencies.

These results indicate low prediction error and strong explanatory power, suggesting the feature set is sufficient to estimate the communication parameters.

Qualitatively, the gap between the FCN and the linear baseline suggests that interactions among the metrics (e.g., simultaneous increases in *leader_off* and *shape_error*) carry information that cannot be captured by additive models. This supports the use of nonlinear estimators even when the input feature set is compact.

Figure 6 presents training and validation loss trends.

Loss curves show stable convergence and minimal overfitting. Slightly lower validation loss is consistent with dropout applied only during training.

Together, the results indicate the model captures the relationship between swarm metrics and communication parameters and supports estimating UAV communication characteristics from high-level observables without low-level diagnostics, enabling scalable and non-intrusive monitoring.

A. Comparison with Related Work and Novelty Validation

The natural follow-up question is how these results compare with the broader literature on machine-learning-based UAV-swarm communication studies. A direct quantitative comparison is constrained by the fact that the closest related works do not target the same estimation problem on the same observables; therefore, we compare both on *type of inputs required* and on *reported figures of merit*.

On the type of inputs. The works summarised in Table 1, including [10, 19, 21, 26, 28], rely on radio-level or routing-level features (e.g., signal-to-interference and noise ratio, bit-error rate, throughput, energy use) and frame the task as adaptation. Khalil et al. [8] use a richer feature set including geographical positions, formation distances, and signal strength to predict link quality with random forests. Lu and Li [15] use deep learn-

DOI: https://doi.org/10.65218/jius.2026.113

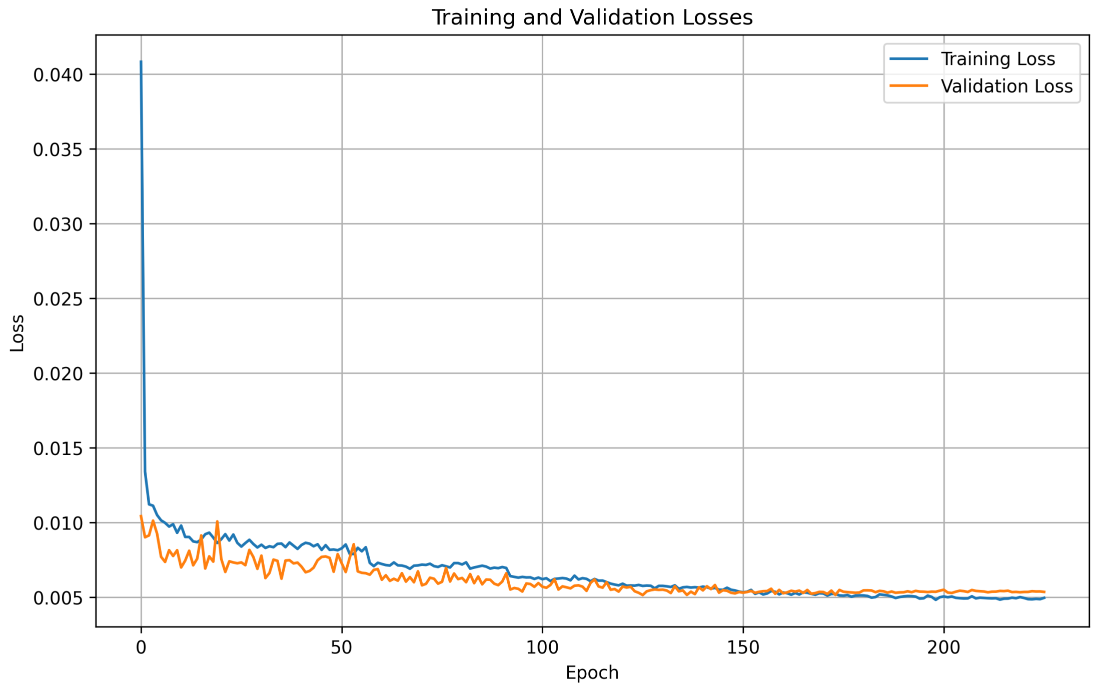

Fig. 6 Training and validation loss evolution

ing to adjust UAV positions based on geometric features. None of these works estimates *communication parameters* from the formation controller's own outputs only. Empirically, the experiments by James et al. [5] and Hudak et al. [4] demonstrate that packet loss and path loss are first-order drivers of swarm coherency loss, but they treat these effects as exogenous and do not infer the underlying parameters. Our compact feature set is therefore a strict subset of, or orthogonal to, what comparable approaches consume; the present results indicate that this subset is sufficient.

On the figures of merit. Achieving $R^2 > 0.83$ for both *comm_range* and *comm_failure*, with a test MSE corresponding to a 5.47 m error in range and a 2.01% error in failure rate, is, to our knowledge, the first quantitative report of joint estimation of these two parameters from formation-level observables alone. The closest comparable numbers are Khalil et al.'s machine-learning estimates of signal strength [8] and Lu and Li's deep-learning position adjustments [15], which, although qualitatively similar in methodology, target different output variables. We therefore frame our comparison as an existence proof rather than a head-to-head ranking: comparable accuracy is reachable without radio-level features.

On the FCN-vs-linear gap. A second, internal comparison provides direct empirical support for the novelty claim. The linear regression baseline, which uses the same 14-dimensional input, drops to $R^2 = 0.727$ for *comm_range* and $R^2 = 0.544$ for *comm_failure*. The substantial gap ($\Delta R^2 \approx 0.10$ and $\Delta R^2 \approx 0.35$, respectively) shows that the information needed to recover the communication parameters is present in the compact feature set, but it lives in nonlinear interactions among the metrics (e.g., the joint behaviour of *leader_off* and *shape_error*). This is the operational meaning of "compact, control-aware observables are sufficient": a small neural model can extract this information; an additive linear model cannot. We interpret this as empirical validation of the novelty claim stated in the Introduction.

VII. Threats to Validity

Simulation bias: The dataset is generated in GrADyS-SIM NextGen and reflects its wireless and mobility models. Real-world propagation, interference, and aerodynamic effects may differ.

Controller dependence: The use of Raft-lite couples the features (notably *leader_off*) to a specific consensus mechanism. Alternative coordination stacks may exhibit different observables or dynamics.

Feature availability: The method assumes access to high-level observables computed from nominal control messages. Under severe disruption, observables may become sparse or delayed.

Distribution shift: The model is trained on a bounded parameter space. Performance outside tested ranges of swarm size, delay, failure rate, or range is not guaranteed.

Temporal compression: The Temporal Summary trades temporal resolution for compactness. Rapid link changes shorter than the smallest window may be smoothed out, which can reduce responsiveness to transient failures.

Label fidelity: Ground-truth labels for *comm_range* and *comm_failure* are simulator parameters. In real deployments, these quantities may be only approximately measurable, which can introduce additional noise at deployment time. Established methods for assessing reliability and importance of components in drone swarms [29] could be combined with our approach to bound the residual uncertainty in operational settings.

Real-swarm validation: The study relies on simulation; real-swarm validation remains future work. One route is to log the same metrics during flight tests and obtain approximate labels via controlled experiments (e.g., varying transmit power, adding

DOI: https://doi.org/10.65218/jius.2026.113

artificial packet drops, or placing obstacles) or external instrumentation in a calibration phase. When exact ground truth is unavailable, validation can rely on relative ranking, cross-platform generalization, and consistency with independent link-quality proxies.

VIII. Practical Implications

The estimator can run online using the same telemetry exchanged for formation control. Inference consists of temporal summaries and a forward pass through a small FCN, which is lightweight enough for onboard or edge execution. Predicted *comm_range* and *comm_failure* can trigger formation reconfiguration (e.g., tighter spacing or a conservative fallback) before coordination degrades.

In a deployment pipeline, temporal summaries can be computed incrementally as new telemetry arrives, allowing near-real-time updates without buffering full traces. This supports low-latency decision making while keeping computation and memory bounded.

IX. Conclusion

We propose a deep learning approach to estimate failure rate and maximum communication range from high-level swarm metrics. The method avoids low-level communication measurements by using *shape_error*, *leader_off*, *presence_error*, and *comm_delay*. A Temporal Summary compresses long time series, and its configuration is selected via cross-validation.

Results show high predictive accuracy, with $R^2 > 0.83$ for both targets and low test loss, indicating good generalization.

Future work includes integrating estimation into formation control, generalizing features to reduce Raft-specific dependence, and improving responsiveness to transient communication changes. We also plan per-UAV estimation and model optimization for embedded hardware. A promising direction is to learn the Temporal Summary parameters jointly with the predictive task.

Overall, the study indicates that compact, formation-level observables can provide a practical proxy for communication health, enabling monitoring in systems where radio-level diagnostics are unavailable or undesirable.

Data Availability

The datasets and source code that support the findings of this study are publicly available at https://github.com/Project-GrADyS/DeepLearningEstimation.

Acknowledgment

This study was financed in part by Air Force Office of Scientific Research (AFOSR) grant FA8655-25-1-8035.

References

[1] J. A. Vazquez Trejo, A. Guenard, M. Adam-Medina, J.-C. Ponsart, L. Ciarletta, D. Rotondo, and D. Theilliol, "Event-triggered leader-following formation control for multi-agent systems under communication faults: application to a fleet of unmanned aerial vehicles," Journal of Systems Engineering and Electronics, vol. 32, no. 5, 2021. [Online]. Available: https://doi.org/10.23919/JSEE.2021.000086

[2] Y. Bu, Y. Yan, and Y. Yang, "Advancement Challenges in UAV Swarm Formation Control: A Comprehensive Review," Drones, vol. 8, no. 7, p. 320, Jul. 2024. [Online]. Available: https://doi.org/10.3390/drones8070320

[3] I. Chandran and K. Vipin, "Network analysis of decentralized fault-tolerant UAV swarm coordination in critical missions," Drone Systems and Applications, vol. 12, pp. 1–15, Jan. 2024. [Online]. Available: http://dx.doi.org/10.1139/dsa-2023-0101

[4] A. Hudak, S. James, and R. Raheb, "Impact of Communication Path Loss to Unmanned Aircraft Swarm Coherency," in *Proc. 2021 Integrated Communications Navigation and Surveillance Conference (ICNS)*, IEEE, 2021.

[5] S. James, R. Raheb, and A. Hudak, "Impact of Packet Loss to the Motion of Autonomous UAV Swarms," in *Proc. 2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*, IEEE, 2020.

[6] S. Javed, A. Hassan, R. Ahmad, W. Ahmed, R. Ahmed, A. Saadat, and M. Guizani, "State-of-the-Art and Future Research Challenges in UAV Swarms," IEEE Internet of Things Journal, vol. 11, no. 11, pp. 19023–19045, Jun. 2024.

[7] M. Ji, T. Li, J. Liao, and X. Yang, "Dynamic event-triggered formation control for UAVs under communication delay and outside disturbances," Transactions of the Institute of Measurement and Control, vol. 45, no. 12, pp. 2287–2298, 2023. [Online]. Available: https://doi.org/10.1177/01423312221151193

[8] H. Khalil, S. U. Rahman, I. Ullah, I. Khan, A. J. Alghadhban, M. H. Al-Adhaileh, G. Ali, and M. ElAffendi, "A UAV-Swarm-Communication Model Using a Machine-Learning Approach for Search-and-Rescue Applications," Drones, vol. 6, no. 12, p. 372, Nov. 2022. [Online]. Available: http://dx.doi.org/10.3390/drones6120372

[9] R. Liang, "raft-lite," GitHub repository, 2023. [Online]. Available: https://github.com/liangrunda/raft-lite [Accessed: Mar. 8, 2025].

[10] Y. Lin, M. Wang, X. Zhou, G. Ding, and S. Mao, "Dynamic Spectrum Interaction of UAV Flight Formation Communication With Priority: A Deep Reinforcement Learning Approach," IEEE Trans. Cogn. Commun. Netw., vol. 6, no. 3, pp. 892–903, Sep. 2020. [Online]. Available: http://dx.doi.org/10.1109/TCCN.2020.2973376

[11] L. Liu, M. Xu, Z. Wang, C. Fang, Z. Li, M. Li, Y. Sun, and H. Chen, "Delay-Informed Intelligent Formation Control for UAV-Assisted IoT Application," Sensors, vol. 23, no. 13, p. 6190, Jul. 2023. [Online]. Available: http://dx.doi.org/10.3390/s23136190

[12] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf

[13] L. Lucchesi, M. Endler, and B. J. O. de Souza, "Dynamic effects of communication delay, failure rates, and speed on UAV swarm formation," in Anais do XLII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2024), pp. 588–601, May 2024. [Online]. Available: http://dx.doi.org/10.5753/sbrc.2024.1445

[14] L. Lucchesi, M. Endler, and B. J. O. de Souza, "Evaluating Formation Metrics in Autonomous UAV Swarms Using Raft Under Communication Constraints," in *2025 13th Wireless Days Conference (WD)*, Niterói, Rio de Janeiro, Brazil, 2025, pp. 1–6. [Online]. Available: https://doi.org/10.1109/WD67713.2025.11302617

[15] J. Lu and Y. Li, "Research on optimal deep learning-based formation flight positioning of UAV," Concurrency and Computation: Practice and Experience, vol. 36, no. 14, Apr. 2024. [Online]. Available: http://dx.doi.org/10.1002/cpe.8099

DOI: https://doi.org/10.65218/jius.2026.113

[16] T. Lamenza, J. Kamysek, B. Souza, and M. Endler, "Developing Algorithms for the Internet of Flying Things Through Environments With Varying Degrees of Realism," in *Anais Estendidos do XLII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, Niterói/RJ, 2024, pp. 33–40. [Online]. Available: https://doi.org/10.5753/sbrc_estendido.2024.2305

[17] B. Olivieri and M. Endler, "DADCA: An Efficient Distributed Algorithm for Aerial Data Collection from Wireless Sensor Networks by UAVs," in *Proc. 20th ACM Int. Conf. on Modelling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM '17)*, Miami, FL, USA, 2017, pp. 129–136. [Online]. Available: https://doi.org/10.1145/3127540.3127553

[18] A. Phadke, F. A. Medrano, C. N. Sekharan, and T. Chu, "Designing UAV Swarm Experiments: A Simulator Selection and Experiment Design Process," Sensors, vol. 23, no. 17, p. 7359, Aug. 2023. [Online]. Available: https://doi.org/10.3390/s23177359

[19] Z. Lv, L. Xiao, Y. Du, G. Niu, C. Xing, and W. Xu, "Multi-Agent Reinforcement Learning Based UAV Swarm Communications Against Jamming," IEEE Trans. Wireless Commun., vol. 22, no. 12, pp. 9063–9075, Dec. 2023. [Online]. Available: http://dx.doi.org/10.1109/TWC.2023.3268082

[20] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in Proc. USENIX Annu. Tech. Conf. (USENIX ATC), Philadelphia, PA, USA, Jun. 2014, pp. 305–319.

[21] G. Raja, Y. Baskar, P. Dhanasekaran, R. Nawaz, and K. Yu, "An Efficient Formation Control mechanism for Multi-UAV Navigation in Remote Surveillance," in Proc. IEEE Globecom Workshops (GC Wkshps), pp. 1–6, Dec. 2021. [Online]. Available: http://dx.doi.org/10.1109/GCWkshps52748.2021.9682094

[22] K. Saulnier, D. Saldaña, A. Prorok, G. J. Pappas, and V. Kumar, "Resilient Flocking for Mobile Robot Teams," IEEE Robotics and Automation Letters, vol. 2, no. 2, pp. 1039–1046, Apr. 2017. [Online]. Available: https://doi.org/10.1109/LRA.2017.2654550

[23] B. J. Olivieri de Souza and M. Endler, "Ventures, Insights, and Ponderings from Real-World experiments with FANETs and UAVs," in *2024 IEEE Symposium on Computers and Communications (ISCC)*, 2024, pp. 1–7. [Online]. Available: https://doi.org/10.1109/ISCC61673.2024.10733659

[24] A. Tariverdi and J. Tørresen, "Rafting Towards Consensus: Formation Control of Distributed Dynamical Systems," arXiv preprint, 2023.

[25] J. Wu, C. Luo, G. Min, and S. McClean, "Formation Control Algorithms for Multi-UAV Systems With Unstable Topologies and Hybrid Delays," IEEE Transactions on Vehicular Technology, vol. 73, no. 9, pp. 12358–12373, Sep. 2024.

[26] C. Xu, K. Zhang, and H. Song, "UAV Swarm Communication Aware Formation Control via Deep Q Network," in Proc. IEEE 39th Int. Perf. Comput. Commun. Conf. (IPCCC), Nov. 2020. [Online]. Available: http://dx.doi.org/10.1109/IPCCC50635.2020.9391509

[27] D. Yu, J. Li, Z. Wang, and X. Li, "An Overview of Swarm Coordinated Control," IEEE Transactions on Artificial Intelligence, vol. 5, no. 5, pp. 1918–1937, May 2024.

[28] T. Zeng, O. Semiari, M. Mozaffari, M. Chen, W. Saad, and M. Bennis, "Federated Learning in the Sky: Joint Power Allocation and Scheduling with UAV Swarms," in Proc. IEEE Int. Conf. Commun. (ICC), pp. 1–6, Jun. 2020. [Online]. Available: http://dx.doi.org/10.1109/ICC40277.2020.9148776

[29] E. Zaitseva, V. Levashenko, R. Mukhamediev, N. Brinzei, A. Kovalenko, and A. Symagulov, "Review of Reliability Assessment Methods of Drone Swarm (Fleet) and a New Importance Evaluation Based Method of Drone Swarm Structure Analysis," Mathematics, vol. 11, no. 11, p. 2551, Jun. 2023. [Online]. Available: https://doi.org/10.3390/math11112551