Multi-Robot Exploration via Flocking Coordination and Machine Learning-Driven Connectivity Assessment

Hazem Chaabi, Nathalie Mitton

▶ To cite this version:

Hazem Chaabi, Nathalie Mitton. Multi-Robot Exploration via Flocking Coordination and Machine Learning-Driven Connectivity Assessment. 2025 23rd International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), May 2025, Linköping, Sweden. hal-05055785

HAL Id: hal-05055785
https://hal.science/hal-05055785v1

Submitted on 5 May 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Public Domain

Multi-Robot Exploration via Flocking Coordination and Machine Learning-Driven Connectivity Assessment

Hazem Chaabi
Inria, France
hazem.chaabi@inria.fr

Nathalie Mitton
Inria, France
nathalie.mitton@inria.fr

***Abstract*—This paper introduces a distributed multi-robot exploration system that combines bio-inspired flocking dynamics with machine learning-powered connectivity prediction to enable efficient unknown environment mapping while maintaining robust communication networks. Designed for emergency response scenarios where operational timelines and network reliability are critical, the approach enriches a previous work to implement a role-adaptive framework in which robots dynamically alternate between exploration-focused and connectivity reinforcement behaviors based on real-time signal strength assessments. Explorers employ frontier detection to explore the unknown areas enhanced with RSSI-aware frontier choosing mechanism, while supporters autonomously position themselves using Flocking inspired behavior, forming communication relays. A comparative evaluation of machine learning models trained on FIT-IoT-Lab-collected signal propagation data enables the system to predict RSSI values in different environments. The system's innovation lies in its integration of flocking-based swarm coordination with data-driven connectivity prediction, addressing the challenge of maintaining network stability while exploring unknown areas efficiently. In addition, the system is designed to operate within the strict computational constraints typical of embedded robotic platforms, leveraging lightweight machine learning models that enable real-time connectivity predictions.**

***Index Terms*—swarm robotics, distributed exploration, flocking algorithms, machine learning, RSSI prediction, fit-iot-lab**

I. Introduction

Modern disaster response operations require robotic systems that can rapidly map structurally compromised environments while maintaining continuous communication. This dual mandate exposes significant limitations in conventional swarm architectures. Frontier-based exploration excels at minimizing redundant coverage; however, centralized variants [1] are vulnerable to single-point failures, and decentralized implementations [2] can devolve into uncoordinated movements under signal attenuation. In contrast, pure flocking approaches [3] maintain swarm cohesion but tend to prioritize collective motion over targeted exploration, leaving critical areas unmapped.

To overcome these challenges, we propose a hybrid approach that integrates frontier-based exploration with bio-inspired flocking dynamics, enhanced by a machine learning–driven connectivity prediction module. In our system, robots dynamically switch between explorer and supporter roles to balance rapid area coverage with robust communication. Explorer robots employ established frontier-based methods [4], [5] to systematically map new territories, while supporter robots leverage bio-inspired positioning strategies [6] to form adaptive relay chains that maintain network connectivity. This dynamic role assignment optimizes both exploration efficiency and network stability, demonstrating the benefits of adaptive strategies in complex systems.

By distributing decision-making among individual robots and incorporating real-time connectivity assessments through machine learning, our approach mitigates the risks associated with single-point failures and uncoordinated behaviors observed in purely centralized or decentralized schemes. This seamless integration ensures continuous connectivity throughout the exploration process, enhancing operational efficiency in dynamically changing environments typical of disaster response scenarios.

More specifically, the specific contributions of this paper are twofold:

1) We propose a refined frontier selection process that uses an RSSI prediction model that evaluates candidate exploration targets based on predicted connectivity metrics, with the goal of maintaining the quality of the connection while efficiently exploring unknown areas.
2) We build upon DRBECM's [7] role-switching mechanism by incorporating an RSSI-based evaluation to more effectively preserve communication in practical environments.

The remainder of this paper is organized as follows: Section II reviews the literature on multi-robot exploration, machine learning for connectivity prediction and the existing role switching approaches, Section III introduces the system model and background, Section IV presents the proposed method, which implementation and experimental setup is presented in Section V. Section VI discusses the results and VII concludes and outlines directions for future research.

II. Related works

Maintaining robust communication while efficiently exploring unknown environments is a longstanding challenge in multi–robot systems. In what follows, we review the key contributions that have shaped research in distributed exploration, bio–inspired coordination, and connectivity prediction.

A. Multi–Robot exploration and frontier–based mapping

Frontier–based exploration was first introduced by Yamauchi [8], providing a simple yet effective method to guide robots toward the boundary between known and unknown areas. Building on this idea, Burgard et al. [9] developed coordinated multi–robot exploration strategies that allocate different frontiers to individual robots in order to minimize overlap and maximize coverage. The probabilistic framework detailed in Fox et al. [10] has become a cornerstone for integrating mapping and localization (SLAM) in uncertain environments. More recent surveys such as Quattrini Li [11] emphasize that distributed approaches have matured to address dynamic and large-scale settings, thus laying the foundation for our distributed system.

Our system builds on the seminal frontier–based exploration methods introduced by Yamauchi, Burgard et al., and Fox et al. We adopt these well–established exploration principles as a foundational baseline. In contrast to conventional frontier assignment approaches, we integrate bio–inspired flocking dynamics to drive decentralized coordination. This integration allows robots to assign exploration frontiers locally and adaptively, thereby reducing exploration overlap and enhancing coverage efficiency in unknown environments.

B. Machine learning for connectivity prediction

Ensuring continuous communication in distributed multi robot networks is critical, especially in emergency response scenarios. Traditional approaches rely on conservative heuristics. However, recent work has used data–driven methods for more accurate connectivity prediction. For example, the Latif and Parasuraman's CQLite framework [12] employs distributed Q learning based on coverage to reduce communication overhead during exploration. Complementary distributed approaches, such as DORA [13] and the recent work on autonomous swarm formation for dynamic network bridging by Galliera et al. [14], further illustrate how integrating connectivity prediction into the control loop can significantly enhance network robustness.

Inspired by recent advances in data–driven methods, such as those exemplified by the CQLite framework, our work leverages machine learning–powered connectivity prediction to inform robot motion planning. Rather than relying solely on conservative strategies based on heuristics, we embed an ML model that evaluates network quality in real time. This proactive approach helps adjust trajectories based on the anticipated quality of the link, thereby enhancing the overall resilience of the system.

Building on this seamless integration of exploration and connectivity, our framework further addresses dynamic operational demands in challenging scenarios.

C. Role–Adaptive frameworks for emergency response

In high–stakes applications like disaster response, maintaining network connectivity is as critical as covering unexplored territory. Role–adaptive frameworks enable robots to dynamically shift their behavior from pure exploration to acting as communication relays. Arslan et al. [15] introduced a hierarchical clustering method that supports dynamic role allocation, ensuring that certain robots reinforce connectivity while others extend the map. In parallel, Queralta et al. [16] provide strategies for collaborative multi–robot search and rescue that emphasize adaptive role distribution. More recently, Kashyap et al. [17] have highlighted the importance of such role–adaptivity in urban search and rescue, reinforcing the need to balance exploration with robust network maintenance.

Although earlier role-adaptive frameworks, as proposed by Arslan, Queralta, and Kashyap, typically allocate static roles or rely on hierarchical clustering, our approach takes a more fluid stance. By combining bio–inspired flocking dynamics with our ML–based connectivity prediction, the system inherently balances exploration with communication maintenance without enforcing rigid role assignments. This dynamic self-organizing mechanism allows robots to seamlessly shift between exploration and acting as communication relays, offering a flexible solution in rapidly evolving and mission–critical environments.

In summary, while prior work has individually addressed frontier-based exploration, bio–inspired coordination, and connectivity maintenance, our approach uniquely integrates these elements by embedding a machine learning–driven connectivity prediction module into a dynamic role–adaptive framework. This integration not only leverages the strengths of decentralized decision–making and robust swarm coordination but also provides a proactive mechanism to maintain network connectivity during rapid exploration. Such a comprehensive integration distinguishes our work from existing methods and lays the groundwork for the subsequent system model and proposed approach.

III. System Model and Background

A. System model

We consider a distributed multi–robot system operating in an unknown, grid–based environment. Each robot is equipped with limited onboard sensing, processing, and communication capabilities. The environment is discretized into a 2 dimensional grid, where each cell represents a spatial unit that can be explored. Robots maintain a local map, which is incrementally updated as they sense their surroundings, and share information with nearby agents via short–range wireless communication.

The communication network is modeled using a Relative Neighborhood Graph (RNG), where a link between two robots exists only if no other robot is closer to both. This mechanism ensures that redundant or weak links are minimized, thereby enhancing the reliability of multi–hop communication. Each robot's ability to communicate is quantified by the Received Signal Strength Indicator (RSSI), which is predicted based on the relative X and Y coordinates from the predicting robot.

Robots are assumed to be homogeneous in terms of mobility and sensor capabilities, but they dynamically assume different roles (explorer or supporter) based on local connectivity assessments. This role–adaptive behavior is critical for balancing rapid exploration with the need to sustain a connected network. The following subsection reviews the baseline DRBECM framework and details the modifications introduced in this work.

B. DBRECM

In the original DRBECM framework, robots autonomously explore unknown environments while preserving robust communication with a fixed base station. The algorithm dynamically assigns roles(explorer or supporter) based on local connectivity conditions and exploration needs. Robots employ a Relative Neighborhood Graph for efficient neighbor selection, use frontier-based exploration to identify the boundaries between known and unknown regions, and implement collision avoidance and stagnation detection mechanisms to ensure safe and continuous navigation. Each robot incrementally updates its local map and shares information with nearby agents, thereby maintaining a coherent, multi-hop communication network throughout the exploration process.

This work presents a variation and updated version of DRBECM with two principal modifications. First, to enhance connectivity maintenance, each robot now incorporates a RSSI prediction mechanism using a pre-trained machine learning model. In addition to the original geometric criteria, an explorer robot evaluates the predicted RSSI with respect to its nearest supporter or the base station. Should the predicted RSSI fall below a predefined threshold, the robot transitions to a supporter role to reinforce network connectivity.

Second, the frontier selection process has been refined with a post-processing step. After identifying frontier cells from the boundaries of the locally sensed map, each candidate is evaluated for its connectivity potential by predicting its RSSI relative to the robot's current position and the positions of connected supporters or the base. The frontier with the best predicted RSSI is then selected as the target, ensuring that the area to discover is within a safe connectivity range. In cases where no frontier satisfies the connectivity criteria, a fallback mechanism allows the robot to continue its exploration while still maintaining a minimal communication link.

The original DRBECM framework provides a decentralized method for multi–robot exploration by dynamically assigning roles based solely on geometric criteria. In our enhanced version, DBRECM-ML, the incorporation of machine learning–driven RSSI prediction and an improved frontier selection process allows for more informed and adaptive role switching, ensuring that communication reliability is maintained even in challenging signal environments.

IV. Proposed Approach

We propose an enhanced multi-robot exploration framework, termed *DBRECM-ML*, which extends the original DRBECM method by incorporating a machine learning–driven connectivity prediction module and refining the frontier selection process. Our approach retains the decentralized decision-making and flock-inspired connectivity maintenance of DRBECM, while leveraging real-time RSSI measurements to guide both dynamic role assignment and exploration target selection.

A. Dynamic Role Assignment

In our approach, each explorer robot continuously collects RSSI measurements from candidate communication links. During the role-update phase, an explorer evaluates the measured RSSI between its current position and that of its closest supporter (or the base). If the predicted RSSI falls below a designated threshold, the robot transitions to a supporter role to mitigate the risk of communication loss in areas with adverse signal propagation conditions.

Equation (1) formalizes the update rule for the role $\rho_i(t+1)$ of robot i based on its current state, the set of neighboring explorers $E_i(t)$, and local connectivity conditions. In the first case, if robot i is an explorer ($\rho_i(t)$ = explorer) and its set of neighboring explorers is non-empty, it switches to the supporter role if, for every robot r_j in its set of neighboring supporters $S_i(t)$ (augmented with the base B), the measured RSSI is less than or equal to a threshold T_{strong}. In the second case, if robot i is a supporter ($\rho_i(t)$ = supporter) and satisfies, being connected to the base (directly or via other supporters), having an empty set of neighboring explorers, and having exactly one supporter in its vicinity that is closer to the base, it transitions back to the explorer role when the RSSI is greater than or equal to a threshold T_{weak}. Otherwise, the robot retains its current role.

$$\rho_i(t+1) = \begin{cases} \text{supporter}, & \text{if } \rho_i(t) = \text{explorer},\ E_i(t) \neq \emptyset, \\ & \forall r_j \in S_i(t) \cup B \\ & \ \text{such that } \text{RSSI}_{ij}(t) \leq T_{\text{strong}}, \\ \text{explorer}, & \text{if } \rho_i(t) = \text{supporter}, \text{Cn}(i) = \text{True}, \\ & E_i(t) = \emptyset, |S_i(t)| = 1,\ \forall r_j \in L_i(t), \\ & \text{Cn}(j) = \text{True},\ \exists r_k \in L_i(t) \\ & \ \text{with } \rho_k(t) = \text{supporter}, \\ & d_{kB}(t) < d_{iB}(t) \\ & \ \text{and } \text{RSSI}_{ij}(t) \geq T_{\text{weak}}, \\ \rho_i(t), & \text{otherwise.} \end{cases} \tag{1}$$

Where:

- $\rho_i(t)$: The role of robot i at time t, where $\rho_i(t) \in \{\text{explorer}, \text{supporter}\}$.
- $E_i(t)$: The set of neighboring robots of i that are currently assigned the explorer role.
- $S_i(t)$: The set of neighboring robots of i that are currently assigned the supporter role.
- B: The base station, which serves as a fixed communication anchor.
- $\text{RSSI}_{ij}(t)$: The measured received signal strength indicator between robot i and robot j at time t.

- T_{strong} and T_{weak}: The RSSI thresholds used in the role switching mechanism.
- $\text{Cn}(i)$: A binary indicator denoting whether robot i is connected to the base (either directly or via other supporters).
- $L_i(t)$: The set of all neighboring robots of i as defined by the connectivity graph.
- $d_{iB}(t)$: The Euclidean distance from robot i to the base station at time t.

B. Frontier Selection

Once the local map is updated, frontier cells(the boundaries between explored and unexplored regions) are identified. The frontier selection process then evaluates each candidate based on its connectivity potential. In particular, Equation (2) defines the set of *safe frontiers* $F_{\text{safe}}(t)$ at time t. A frontier cell f is considered safe if the predicted RSSI between f and at least one supporter robot in $S_i(t)$ is at least T_{rssi}. Then, the frontier with the maximum predicted RSSI among all the predictions is is chosen as the next frontier to be explored. This criterion ensures that even under the worst-case connectivity condition, the communication link from the frontier to a supporter meets the minimum quality required for reliable operation.

$$F_{\text{safe}}(t) = \left\{ f \in F_i(t) \;\middle|\; \max_{r_j \in S_i(t)} \hat{\text{RSSI}}_{fj}(t) \geq T_{\text{rssi}} \right\}. \quad (2)$$

Where:

- $F_i(t)$: The set of frontier cells in the local map of robot i, representing the boundaries between explored and unexplored regions.
- $\hat{\text{RSSI}}_{fj}(t)$: The predicted RSSI value between a frontier cell f and supporter robot r_j at time t.
- T_{rssi}: The minimum acceptable predicted RSSI threshold required for a frontier cell to be considered safe.

V. Implementation and Experimental Setup

A. Data collection using the Fit IoT-Lab

This work leverages the large-scale IoT-LAB testbed [18], where we conducted experiments on 126 M3 nodes. These M3 nodes are equipped with a 32-bit ARM Cortex-M3 (STM32F103REY) microcontroller operating at up to 72 MHz, featuring 64 KB of RAM and 256 KB of ROM. The radio interface is an AT86RF231 chip that supports 2.4 GHz IEEE 802.15.4 communication at a maximum bandwidth of 256 kbit/s.

In our data collection scenario, we designated one node as the broadcaster (emitter) and the remaining 125 nodes as listeners. The broadcaster periodically sent messages at a given transmission power (0dBm), while the listeners recorded the Received Signal Strength Indicator (RSSI) along with additional metadata such as channel and message ID.

Each line contains a timestamp, node identifier, message ID, channel number, and the measured RSSI in dBm.

B. Data preprocessing

After extracting the relevant fields from the raw logs, we performed several preprocessing steps:

- **Cleaning and Parsing:** Lines without valid reception information or incomplete metadata were discarded. For each valid entry, the RSSI value was converted from a string (e.g., "-60dBm") into a numerical format (e.g., -60).
- **Outlier Removal:** We employed an Interquartile Range (IQR) filter [19] to remove extreme RSSI readings that could adversely affect model training. Concretely, we computed the first quartile Q_1 (the 25$^{\text{th}}$ percentile) and the third quartile Q_3 (the 75$^{\text{th}}$ percentile) of the RSSI distribution, then defined IQR $= Q_3 - Q_1$. Any data point whose RSSI value fell outside the interval $\left[Q_1 - 1.5 \times \text{IQR},\ Q_3 + 1.5 \times \text{IQR}\right]$ was deemed an outlier and excluded. This step removes sporadic signal fluctuations (e.g., due to hardware anomalies or environmental interference) and helps ensure that the resulting dataset better represents typical signal propagation patterns for robust model training.
- **Coordinate Normalization:** The raw (x, y) coordinates for each node (provided by FIT IoT-LAB's internal localization or by known node placements) were shifted and scaled relative to the broadcaster reference point to ensure consistency. This step helps the regression model learn distance-based RSSI patterns more effectively.

Following these steps, we split the refined dataset into training and test subsets, ensuring that the test data was withheld from any training process to provide an unbiased evaluation.

C. Model training and hyper-parameter tuning

We evaluated multiple regression algorithms (e.g., Decision Tree, Random Forest, Gradient Boosting, LightGBM, CatBoost, XGBoost, Support Vector Regressor (SVR), and K-Nearest Neighbors Regressor) to predict the RSSI based on the distance between transmitter and receiver nodes. Each model was tuned via a grid search with cross-validation (3-fold), optimizing hyperparameters to minimize the Root Mean Square Error (RMSE). We also tracked Mean Absolute Error (MAE) and R^2 for a comprehensive view of model performance. Inference time and memory usage were measured with Python's `time` and `tracemalloc` libraries, respectively, to ensure the selected model could run efficiently on resource-constrained robotic platforms. The weights of the final chosen model were frozen and saved using `pickle` library for integration into our multi-robot exploration framework.

D. Simulation environment and evaluation

The final system is implemented in Python on a 120×120 grid-based environment. Robots operate with predefined motion and communication parameters. The DRBECM algorithm, augmented with machine learning predictions, governs how robots explore new areas while preserving network connectivity. Each simulation run records metrics such as:

- *Exploration Time:* The number of steps required to cover a target percentage of the map.
- *Coverage:* The fraction of cells discovered in the environment.
- *Redundant Exploration:* The count of cells explored more than once.
- *Distance Traveled:* The cumulative distance traveled by all robots.

The results are aggregated over multiple runs with varying numbers of robots, enabling a robust evaluation of the trade-off between exploration efficiency and communication reliability.

VI. Results and Discussion

A. Machine learning model evaluation

1) Evaluation metrics for connectivity prediction model: To rigorously assess the performance of the connectivity prediction model, we employ several standard regression metrics along with some resource utilization metrics. These metrics evaluate both the accuracy of the RSSI predictions and the suitability of the model for real–time deployment on resource–constrained platforms. In our experiments, good model performance is characterized by low RMSE and MAE values (close to zero), an R^2 score near 1, as well as minimal inference time and memory usage. The metrics are defined as follows:

- **Root Mean Square Error (RMSE):** Defined as

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2}, \tag{3}$$

  where $\hat{y}_i$ is the predicted RSSI, y_i is the actual RSSI, and n is the number of samples. A low RMSE indicates that the average error is small, which is desirable.
- **Mean Absolute Error (MAE):** Given by

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i - y_i\right|, \tag{4}$$

  MAE provides the average absolute difference between the predicted and actual RSSI values. Values close to zero indicate high prediction accuracy.
- **Coefficient of Determination (R^2 Score):** This metric is calculated as

$$R2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \bar{y}\right)^2}, \tag{5}$$

  where $\bar{y}$ is the mean of the actual RSSI values. An R^2 score close to 1 signifies that the model explains most of the variability in the data.
- **Inference Time (s):** This is the average time required by the model to generate a RSSI predictions. Low inference time (on the order of milliseconds or lower) is critical for supporting real–time decision–making in dynamic robotic environments.
- **Memory Usage (bytes):** This metric quantifies the memory footprint of the deployed model. Lower memory usage is advantageous, especially when the model is to be run on embedded systems or platforms with limited computational resources.

2) Results of connectivity prediction model evaluation: Table I summarizes the performance of various regression models employed for RSSI prediction. The evaluation metrics include RMSE (Eq. (3)), MAE (Eq. (4)), $R2$ score (Eq. (5)), inference time, and memory usage. All experiments were conducted on a 64-bit operating system with a 12th Gen Intel® Core™ i5-12500H CPU (2.50 GHz) and 32 GB of RAM.

Accuracy: Among the evaluated models, tree–based methods (i.e., Decision Tree, Extra Trees, Gradient Boosting, LightGBM, CatBoost, Random Forest, and XGBoost) yield RMSE values around 1.1948 and MAE values close to 0.55, with R^2 scores approximately 0.9664. For example, the Decision Tree model achieves an RMSE of 1.194825, which represents a reduction of approximately 76.1% compared to AdaBoost (RMSE = 5.013599) and nearly 79.8% compared to Linear Regression (RMSE = 5.922185). In contrast, models such as AdaBoost, Elastic Net, Linear Regression, and Support Vector Regression exhibit significantly higher error values (with RMSEs ranging from 5.013599 to 5.922185 and R^2 scores below 0.41), indicating inferior performance in capturing the RSSI variability.

Inference Time: Inference time is a critical factor for real–time applications. Notably, the Decision Tree model demonstrates an inference time of 0.0025 s, which is approximately 99% lower than that of the Random Forest model (0.2513 s). Similarly, LightGBM and XGBoost offer competitive inference times (0.005 s and ∼0.008 s, respectively), making them suitable for real–time deployment.

Memory Usage: Memory footprint is equally important for resource–constrained platforms. XGBoost requires only 16,548 bytes of memory, which is about 42.6% lower than that of Random Forest (28,806 bytes). The Decision Tree model also shows favorable memory usage (24,953 bytes), and most high–accuracy models fall within a similar range (approximately 24,900 to 29,000 bytes), except for K-Nearest Neighbors which uses 29,979 bytes.

Summary: Overall, tree–based models such as Decision Tree and XGBoost deliver an excellent balance between prediction accuracy and computational efficiency. With RMSE and MAE values around 1.19 and 0.55, respectively, and R^2 scores exceeding 0.966, these models are able to capture the underlying RSSI dynamics effectively. Moreover, the Decision Tree model offers exceptionally low inference time (0.0025 s) and modest memory usage, while XGBoost provides the lowest memory footprint (16,548 bytes). These performance characteristics render them particularly well suited for integration into our distributed multi–robot exploration framework.

B. Simulations settings for models comparison

We performed a series of simulations to evaluate the performance of our proposed method compared to DRBECM and other multi-robot exploration algorithms. Table II summarizes

TABLE I
MODEL PERFORMANCE COMPARISON

Model	RMSE	MAE	R2 Score	Inference Time (s)	Memory Usage (bytes)
AdaBoost	5.013599	4.249386	0.407860	0.004002	25566
CatBoost	1.194825	0.549787	0.966369	0.007545	26706
Decision Tree	1.194825	0.549787	0.966369	0.002500	24953
Elastic Net	5.922161	4.839469	0.173799	0.001003	25063
Extra Trees	1.194825	0.549787	0.966369	0.062986	28694
Gradient Boosting	1.194830	0.549920	0.966369	0.112972	25863
K-Nearest Neighbors	1.262389	0.561584	0.962459	0.013499	29979
LightGBM	1.194848	0.550685	0.966368	0.005000	25016
Linear Regression	5.922185	4.839405	0.173792	0.002001	24894
Random Forest	1.194740	0.549707	0.966374	0.251301	28806
Support Vector Regression	5.105113	3.741817	0.386046	1.938107	25022
XGBoost	1.194824	0.550040	0.966370	0.007999	16548

the key simulation parameters. In our experiments, all simulations were conducted on a 120×120 grid map, with each robot possessing a sensing range of 4 units. The number of robots varied from 11 to 15, and each configuration was run 30 times. A maximum of 3000 simulation steps was allowed per run, after which the simulation was terminated if the target coverage was not achieved. In addition, we inspire from [20] the RSSI thresholds, for good and bad connectivity, for role switching were set as follows: T_{strong} was set to -65 dBm, while the T_{weak} was set to -75 dBm.

TABLE II
SIMULATION PARAMETERS

Parameter	Value
Grid Size	120×120
Sensing Range	4 units
Number of Robots	11–15
Maximum Simulation Steps	3000
T_{strong}	-65 dBm
T_{weak}	-75 dBm
Number of Runs per Configuration	30

The following methods were compared:

- **DRBECM:** A decentralized method inspired by flocking behavior, aiming to balance exploration and connectivity maintenance.
- **DRBECM-ML (Proposed Method):** A variate version of DRBECM that integrates a machine learning–driven connectivity prediction module. By predicting RSSI values in real time, DRBECM-ML dynamically adapts the role of each robot to ensure more robust communication, and make better choices while choosing the area to explore.
- **Random Walk** [21]: A baseline decentralized approach where each robot moves randomly. Robots independently choose a random direction and travel for a set number of steps before selecting a new random heading, resulting in uncoordinated coverage of the environment.
- **HCETIIC (Hybrid Cheetah Exploration Technique with Intelligent Initial Configuration)** [22]: Employs a central planner to coordinate exploration tasks and optimize initial robot placements. The algorithm's performance is evaluated across various starting configurations, such as uniform, centralized, random, perimeter, clustered, and strategic distributions, with the goal of maximizing exploration efficiency.
- **Frontier Exploration (No Map Sharing)** [4]: A decentralized method in which each robot independently maintains its own map and identifies frontier cells (unexplored regions at the boundary of known space). Robots select frontiers based on factors like distance, but do not share map data with other robots.
- **Frontier Exploration (Map Sharing)** [4]: A more centralized variant where robots periodically share their local maps with a central server or with one another. This shared global map informs more coordinated frontier selection, enabling robots to avoid duplicating efforts and to exploit collective knowledge of the environment.

C. Successful Exploration Rate

Table III reports the percentage of runs required for each method to successfully complete the exploration task among the 150 runs. We define a successful exploration as covering at least 99% of the environment within the 3000 steps. The results highlight the varying degrees of exploration effectiveness across different algorithms.

TABLE III
PERCENTAGE OF SUCCESSFUL EXPLORATION RUNS FOR EACH METHOD.

Method	Successful Exploration (%)
DRBECM	83.6
DRBECM-ML	99.3
Frontier (Map Sharing)	100.0
Frontier (No Map Sharing)	99.6
HCETIIC	45.6
Random Walk	61.4

We observe that only three methods can be qualified as successful : DRBECM-ML, and two variants of Frontier methods. *DRBECM-ML* achieves a notably higher successful exploration rate compared to *DRBECM*, suggesting that the integration of machine learning for connectivity prediction can significantly enhance overall coverage by eliminating hard geometric criteria. Additionally, *Frontier (Map Sharing)* achieves a success rate 100%, reflecting the advantages of collective knowledge when robots periodically exchange their local maps in spite of its high cost.

DRBECM achieves a low approximately 84% success rate, which is attributed to simulations with fewer robots being unable to fully extend coverage across the entire grid. By contrast, *Random Walk* displays comparatively lower success rates, indicating the limitations purely uncoordinated exploration.

These results underscore the importance of real-time connectivity assessment and cooperative map sharing in maintaining robust communication networks and achieving comprehensive coverage. Further experiments and analyses may explore the interplay between exploration speed, network reliability, and scalability to larger teams and environments.

D. Exploration time

We define *exploration time* as the total simulation steps required to achieve a target coverage of 99% of the environment. Figure 1 shows a box plot of exploration times for all methods.

Interestingly, *DRBECM* exhibits a slightly lower median exploration time than *DRBECM-ML*, indicating that its purely geometric connectivity criteria can lead to faster overall coverage. However, *DRBECM-ML* relies on a data-driven RSSI prediction model trained on real-world measurements, making its role switching and frontier selection more representative of actual signal conditions. Consequently, while there is a modest trade-off in exploration speed, *DRBECM-ML* is arguably more robust to realistic communication fluctuations.

Comparing the two DRBECM variants with other methods, *Frontier (Map Sharing)* demonstrates the lowest median exploration time by leveraging centralized, global map sharing—at the cost of heavier communication overhead. Meanwhile, purely decentralized approaches such as *Random Walk* and *Frontier (No Map Sharing)* exhibit notably longer exploration times and higher variance. *HCETIIC*, which relies on a central planner and specific initial configurations, generally yields moderate performance but displays a broader spread in exploration times.

Overall, these results highlight that *DRBECM-ML* offers a practical compromise between real-world connectivity modeling and efficient coverage. Although its exploration time is slightly higher than that of *DRBECM*, the improved realism in communication prediction may be advantageous in scenarios where signal quality is highly variable.

E. Redundant Exploration

Redundant exploration quantifies the degree of overlap among the areas covered by different robots during the mapping process. Figure 2 presents a box plot comparing the redundancy levels for several methods, including our proposed *DRBECM-ML*, the original *DRBECM*, *HCETIIC*, *Frontier (No Map Sharing)*, *Frontier (Map Sharing)*, and *Random Walk*.

Our experimental results indicate that *DRBECM* exhibits a slightly lower median redundancy than *DRBECM-ML*. This suggests that the purely geometric criteria employed by DRBECM are marginally more effective at minimizing overlap. However, *DRBECM-ML* incorporates real-world RSSI predictions to guide role switching and frontier selection, which introduces a modest increase in redundancy. Despite this trade-off, *DRBECM-ML* still shows significantly lower redundancy compared to other decentralized methods such as *Random Walk* and *Frontier (No Map Sharing)*.

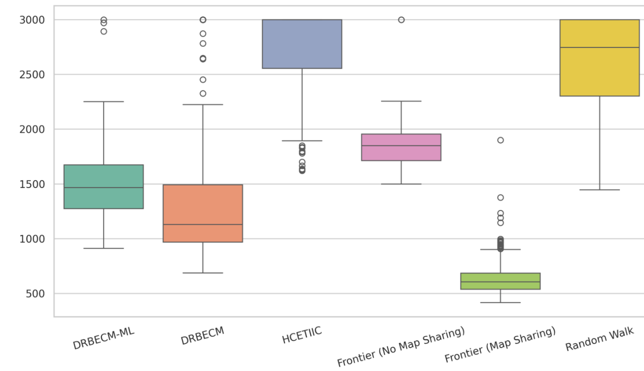

Fig. 1. Box plot of exploration times for the different exploration methods. The y-axis represents the steps needed to achieve the target coverage, while the x-axis lists the methods under comparison.

It is worth noting that the centralized *Frontier (Map Sharing)* approach achieves the lowest redundancy overall by leveraging a globally shared map, though this comes at the cost of increased communication (and thus consumption) overhead. Similarly, *HCETIIC*, which relies on centralized planning and is highly sensitive to initial configurations, displays greater variability in redundancy.

In summary, while *DRBECM-ML* may incur a slight penalty in terms of redundant exploration compared to *DRBECM*, its reliance on real-world RSSI data offers a more realistic and robust approach to maintaining connectivity in uncertain communication environments.

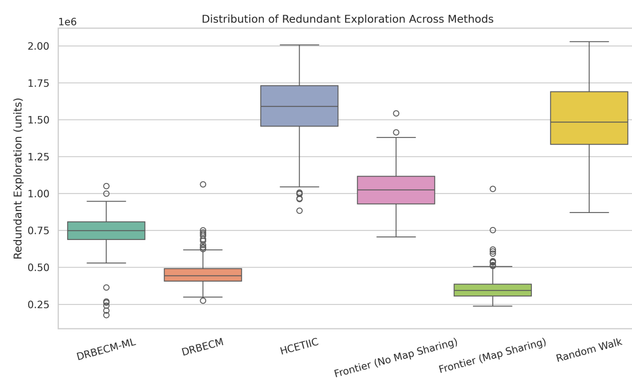

Fig. 2. Comparison of redundant exploration across the different methods.

VII. Conclusion and Future work

In this paper, we presented *DBRECM-ML*, an enhanced distributed multi-robot exploration framework that extends the original DRBECM method by integrating a machine learning–driven connectivity prediction module and refining the frontier selection process. Our approach leverages real-time RSSI measurements derived from real-world data collected via the Fit-IoT-Lab testbed, enabling robots to dynamically adjust their roles and select exploration targets based on realistic connectivity conditions.

Our experimental results indicate that while *DRBECM-ML* exhibits slightly longer exploration times compared to the original *DRBECM*—likely due to the additional overhead of processing real-world RSSI predictions—it achieves more robust connectivity under practical conditions. Furthermore, our comparative evaluation of various regression models revealed that tree-based methods, such as Decision Tree and XGBoost, provide superior prediction accuracy and computational efficiency, making them particularly well-suited for integration into resource-constrained multi-robot systems.

In addition, simulation studies show that *DRBECM-ML* attains high coverage with relatively low redundancy compared to decentralized baselines like Random Walk and Frontier Exploration without Map Sharing, while performing comparably to centralized approaches that require global map sharing. These findings underscore the potential benefits of incorporating real-world connectivity data into decentralized exploration strategies, despite a modest trade-off in exploration speed.

For future work, we plan to extend our approach to more complex and dynamic environments, where the presence of obstacles and varying terrain further challenge connectivity and exploration efficiency. We also intend to investigate the incorporation of additional Quality-of-Service (QoS) metrics, such as latency and packet loss, to enhance the robustness of our decision-making framework. Such extensions will provide deeper insights into the balance between exploration performance and communication reliability, particularly in emergency response and disaster relief operations where unpredictable conditions and communication failures can have critical consequences.

Overall, our work demonstrates a promising step towards more realistic and resilient multi-robot exploration systems for emergency scenarios, paving the way for further advancements in decentralized coordination and robust connectivity management.

Acknowledgment

This work has been partially supported by the European Union's Horizon Europe research and innovation project under grant agreement No. 101070487 (NEPHELE).

References

[1] W. Tang, C. Li, J. Wu, and Q. Zhu, "Decentralized communication-maintained coordination for multi-robot exploration: Achieving connectivity and adaptability," *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024.

[2] A. Ahmad, D. Bonilla Licea, G. Silano, T. Báča, and M. Saska, "Pacnav: a collective navigation approach for uav swarms deprived of communication and external localization," *Bioinspiration & Biomimetics*, 2022.

[3] L. Bayındır, "A review of swarm robotics tasks," *Neurocomputing*, 2016.

[4] U. Jain, R. Tiwari, and W. W. Godfrey, "Comparative study of frontier based exploration methods," in *Conference on Information and Communication Technology (CICT)*, 2017.

[5] N. Mahdoui, V. Fremont, and E. Natalizio, "Cooperative frontier-based exploration strategy for multi-robot system," in *IEEE 13th Annual Conference on System of Systems Engineering (SoSE)*, 2018.

[6] H. Zhang, J. Cheng, L. Zhang *et al.*, "H2gnn: Hierarchical-hops graph neural networks for multi-robot exploration in unknown environments," *IEEE Robotics and Automation Letters*, vol. PP, 2022.

[7] H. Chaabi and N. Mitton, "Distributed multi-robot exploration approach with connectivity maintenance," in *2025 23rd International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, 2025.

[8] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proc. IEEE Int. Symp. Comput. Intell. Robotics and Automation*, 1997.

[9] W. Burgard, M. Moors, C. Stachniss, and F. Schneider, "Coordinated multi-robot exploration," *IEEE Transactions on Robotics*, 2005.

[10] D. Fox, W. Burgard, and S. Thrun, *Probabilistic Robotics*. MIT Press, 2005.

[11] A. Quattrini Li, "Exploration and mapping with groups of robots: Recent trends," *Current Robotics Reports*, 2020.

[12] E. Latif and R. Parasuraman, "Communication-efficient multi-robot exploration using coverage-biased distributed q-learning," *IEEE Robotics and Automation Letters*, 2023.

[13] D. Vielfaure, S. Arseneault, P.-Y. Lajoie, and G. Beltrame, "Dora: Distributed online risk-aware explorer," *2022 International Conference on Robotics and Automation (ICRA)*, 2021.

[14] R. Galliera, T. Möhlenhof, A. Amato, D. Duran, K. B. Venable, and N. Suri, "Distributed autonomous swarm formation for dynamic network bridging," *CoRR*, 2024.

[15] O. Arslan, D. P. Guralnik, and D. E. Koditschek, "Coordinated robot navigation via hierarchical clustering," *IEEE Transactions on Robotics*, 2016.

[16] J. P. Queralta, J. Taipalmaa, B. C. Pullinen, V. K. Sarker, T. N. Gia, H. Tenhunen, M. Gabbouj, J. Raitoharju, and T. Westerlund, "Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision," *IEEE Access*, 2020.

[17] G. S. Kashyap, D. Mahajan, O. C. Phukan, A. Kumar, A. E. I. Brownlee, and J. Gao, "From simulations to reality: Enhancing multi-robot exploration for urban search and rescue," *arXiv preprint*, 2023.

[18] C. Adjih, E. Baccelli, E. Fleury, G. Harter, N. Mitton, T. Noel, R. Pissard-Gibollet, F. Saint-Marcel, G. Schreiner, J. Vandaele, and T. Watteyne, "Fit iot-lab: A large scale open experimental iot testbed," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, 2015.

[19] K. D. Buch, "Decision based non-linear filtering using interquartile range estimator for gaussian signals," in *2014 Annual IEEE India Conference (INDICON)*, 2014.

[20] S. Zacharias, T. Newe, S. O'Keeffe, and E. Lewis, "2.4 ghz ieee 802.15.4 channel interference classification algorithm running live on a sensor node," in *SENSORS, 2012 IEEE*, 2012.

[21] F. Xia, J. Liu, H. Nie, Y. Fu, L. Wan, and X. Kong, "Random walks: A review of algorithms and applications," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 2, p. 95–107, Apr. 2020. [Online]. Available: http://dx.doi.org/10.1109/TETCI.2019.2952908

[22] A. El Romeh and S. Mirjalili, "Theoretical framework and practical considerations for achieving superior multi-robot exploration: Hybrid cheetah optimization with intelligent initial configurations," *MDPI Mathematics*, vol. 11, 2023.