



OPEN

## Human activity-aware coverage path planning for robot-based mosquito control

Ash Yaw Sang Wan<sup>1</sup>, Prabakaran Veerajagadheswar<sup>1</sup>, Mohan Rajesh Elara<sup>1</sup> & Anh Vu Le<sup>2</sup>✉

Automating mosquito control is a pivotal advancement in the pest control industry with the primary objective of mitigating the prevalence of vector-borne diseases. Recent progress in pest control robotics has enabled the automation of mosquito activity restrictions. However, existing robotic solutions have exhibited limitations in effectively addressing mosquito control while lacking a sensitive strategy for maximizing area coverage with crowded areas as a priority. In response to these challenges, this article proposes a novel human-first approach for complete coverage path planning (HFA-CCPP) that leverages the Glasius Bio-inspired Neural Network (GBNN) to cover areas that simulate and consider human activity patterns systematically. In this study, a mosquito-capturing robot, Dragonfly, is demonstrated with HFA-CCPP. This article provides an in-depth exploration of the technical intricacies of the proposed solution. The efficacy of the proposed technique is evaluated in terms of total area coverage and times taken to cover the high human activity region in simulation and real-world environments by comparing results with traditional GBNN. Across all scenarios, the proposed HFA-CCPP surpasses the traditional method by delivering efficient area coverage with minimal time for human-dense area coverage and efficiency in mosquito trapping. This finding stands as a newfound direction in automated mosquito control, holding great potential for curbing vector-borne diseases.

**Keywords** Automating mosquito control, Coverage path planning, Human activity-aware, Bio-inspired neural network, Mosquito-capturing robot

Mosquito-borne diseases constitute a significant focal point of health apprehension in various nations, encompassing those developed and developing. The European Centre for Disease Prevention and Control documented a pinnacle in vector-borne malady occurrences due to mosquitoes in 2020<sup>1</sup>. As a consequence of this augmented incidence, nations are directing escalated efforts towards the control of mosquito activity in public settings. Several conventional techniques have been employed for mosquito breeding regulation, comprising environmental manipulation<sup>2</sup>, chemical-based interventions<sup>3</sup>, and biological strategies<sup>4</sup>. Products of mosquito lure-and-catch mechanisms are also widely available<sup>5–7</sup> using ultraviolet light, pheromones, and electrical zappers or sticky traps. The majority of such methodologies are labour-intensive, repetitive, and engender ecological contamination. A prospective avenue to transcend these monotonous strategies involves the mechanization of processes in an ecologically benign manner. Recent developments in the field of robotics have made this mechanism attainable.

Mosquito detection in current state-of-the-art technologies is limited. Imaging detection methods are rapidly growing<sup>8,9</sup>, but are limited to the quality of images. While Motion blur, background clutter, and lighting conditions play an important factor in inference, it often requires mosquitoes to be immobilized or trapped. At present, it is extremely challenging to capture flying mosquitoes without a high-speed camera and without a prior informed location of the mosquitoes. Another interesting method for detection is via wingbeat sound detections<sup>10</sup> developed by Yin et al. This method is also challenging to be deployed in human-dense areas with noise generated by people and background noises. The presence of mosquitoes is well known to be predictable by environmental factors such as the presence of human<sup>11</sup> or natural habitat, humidity<sup>12</sup>, temperature<sup>13</sup>, and time of day. Deploying mosquito control robots based on known risk factors is significantly more efficient and effective than relying on mosquito-seeking methods.

In the prior work<sup>14</sup>, a mobile robot was developed for mosquito trapping; however, it lacked an optimized, human-aware path planning strategy during mosquito capturing. Equipped with bait modules to attract

<sup>1</sup>ROAR Lab, Engineering Product Development, Singapore University of Technology and Design, Singapore 487372, Singapore. <sup>2</sup>Advanced Intelligent Technology Research Group, Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City, Vietnam. ✉email: leanhvu@tdtu.edu.vn

mosquitoes and a trapping mechanism to ensnare them, the robot aimed to classify trapped mosquitoes, particularly targeting the *Aedes Aegypti* species responsible for dengue transmission. The study centred on the robot's navigation strategy, employing manually designated waypoints to ensure comprehensive coverage of the area. However, despite this meticulous approach, the robot encountered limitations in capturing mosquitoes at these waypoints due to minimal mosquito activity in those areas. As highlighted in a WHO article<sup>15</sup>, mosquitoes tend to exhibit heightened activity in regions with high human density, as these areas offer abundant feeding opportunities. Consequently, it becomes imperative for the robot's coverage path to align with areas of elevated human activity to optimize mosquito capture rates and mitigate disease spread. Given the dynamic nature of human density across different locales, it is essential for the robot to autonomously track and incorporate human density information into its coverage path planning process, obviating the need for manual intervention. Crafting such a path presents a multifaceted challenge, necessitating a nuanced balance between various considerations, including overall area coverage, re-covering previously visited areas, and maximizing mosquito capture rates relative to human activity levels. One promising avenue to address this challenge involves employing a complete coverage path planning (CCPP) technique, capable of dynamically adjusting the robot's path to accommodate fluctuations in human density across the environment.

The present study introduces a pioneering approach termed the Human-First Approach Complete Coverage Path Planning (HFA-CCPP) scheme, designed specifically for a mosquito-capturing robot operating in environments with high human activity levels. The proposed methodology involves striking a balance between prioritizing areas of human activity and ensuring comprehensive coverage, thereby optimizing the robot's ability to capture mosquitoes while effectively completing the coverage task. This paper covers the system architecture of the mosquito-capturing robot, delineates the translation of theoretical formulations into practical implementation of the HFA-CCPP scheme, presents simulation results, and validates the approach through real-world experiments. Furthermore, the following points outline the primary contributions of this research article.

- A novel human activity-based complete coverage path planning for a mosquito-capturing robot is proposed, human-first approach CCPP (HFA-CCPP).
- A modified Glasius bio-inspired neural network has been developed to perform human-dense area coverage for improved mosquito capturing efficiency.
- Development of an autonomous mobile robot with the mosquito bait and catching mechanism.
- Validation of the proposed CCPP scheme on the developed physical robotic platform in a real-world scenario.

Therefore, this article presents a novel mosquito-capturing robot to achieve an autonomous maintenance system for the pest care domain. Note that this paper does not aim to provide a mosquito-seeking algorithm but to deploy the robot in a way where it can prevent people from mosquito harassment if one is present using the presence of a lure-and-catch mechanism onboard a robot.

The structure of the paper is as follows. Section 2 reviews the recent robot development carried out on pest care and vector control, then discusses the recent research on CCPP based on GBNN. Section 3 describes a mosquito-catching robot and its mechanisms. The following Sect. 4 introduces the robotic system and describes the proposed CCPP scheme. Section 5 provides the simulated result of the proposed scheme in a controlled setting, and 6 describes the experimental design, setup, results, and related discussions on the performance of the proposed solution on the robot in various real-world scenarios. Section 7 summarises the paper and discusses future works.

## Related work

There are several approaches used for path planners. The types of path planners control are namely for shortest path<sup>16,17</sup>, area coverage<sup>18,19</sup>, smoothness path<sup>20</sup>, safe path<sup>21,22</sup>, rescue. These problems formulated path planners can be modeled with several methods such as using artificial potential field<sup>23,24</sup>, bio-inspired<sup>25,26</sup>, nature-inspired<sup>27,28</sup> or classical search algorithms<sup>29</sup>. Each type of path planner is suitable for different types of contextual applications, and each model has a specific disadvantage.

Amongst the path planners, coverage planners are the most appropriate path planners. It grants the robot the ability to perform full coverage in most of the cases. The most typical path planners used are spiral<sup>30</sup> or zig-zag<sup>31</sup>, reinforcement learning<sup>32,33</sup>, evolutionary<sup>34</sup>, genetic<sup>35</sup>, vision<sup>36</sup> algorithms. Artificial potential offers a simple model with known deadlocks highlighted in some developers<sup>37,38</sup>. With the rise of neural networks, bio-inspiration has been used for path planning. Neural network solutions often package themselves with training requirements before usage, which can be expensive to deploy in businesses or with a lack of technical support. Glasius bio-inspired neural network (GBNN)<sup>39</sup> and its modified version<sup>40</sup> demonstrate a rule-based neural network that does not require training but, like most path planners, requires a map to operate. There are similar state-of-the-art solutions, such as  $\epsilon^*$ <sup>41</sup> algorithm and predator-prey coverage path planner (PPCPP)<sup>42</sup>. These path planners require additional models to complement the primary model when it encounters a natural deadlock. GBNN does not have a natural deadlock, and the computational complexity is at the peak of the state-of-the-art. With a suitable optimization model, it can be used to enhance the algorithm further for practical use. A complete coverage path planner (CCPP) is not new to the robotics community. There have been numerous other CCPP techniques proposed in the literature for various robotic applications, including cleaning<sup>40</sup>, inspection<sup>43</sup>, agriculture<sup>44</sup>, and so on. Also, there are studies in the literature that considered the tradeoff models on CCPP. For instance, Katharin et al proposed an energy cost-based Voronoi-based path generation (VPG) algorithm for a multipurpose UAV robot<sup>45</sup>. In another work, a CCPP-based autonomous radiation mapping scheme has been proposed for a mobile that trades the total area covered with its coverage cost<sup>46</sup> and energy efficiency<sup>47</sup>. Van-Thach and Quang-Cuong proposed a new approach to establish the CCPP with minimal overlapping on 3D surfaces using mesh models<sup>48</sup>. Even though there are numerous novel techniques proposed in the literature, none

of the existing approaches focuses on performing CCPB based on human activity. To the best of our knowledge, the previous approaches have minimal connection to pest care robots and no relation towards autonomous mosquito-capturing robots.

A limited number of path planning approaches are used for pest-control<sup>49,50</sup>. The main development of pest control revolved around crop management. There are no path-planning robots for pest control to the best of our knowledge. Path planning is essential for the robot to draw the catching mechanism closer to humans. When lures perform best at close proximity, such as carbon dioxide<sup>51</sup>, ultra-violet lamps<sup>52</sup> and pheromones<sup>53</sup> in early research works. Sustained position might increase efficacy in allowing pheromones to draw more mosquitoes<sup>54</sup>. The proposed lure-and-catch mechanism will serve as a distracting element in people in a targeted area.

Hence, the current development in mosquito-catching automation presents a gap in development. Automating mosquito catching solution plays a critical role as it does not require environmental installations of ultraviolet light and scent attractants. More importantly, it provides semantic control for adequate use of equipment in a human environment.

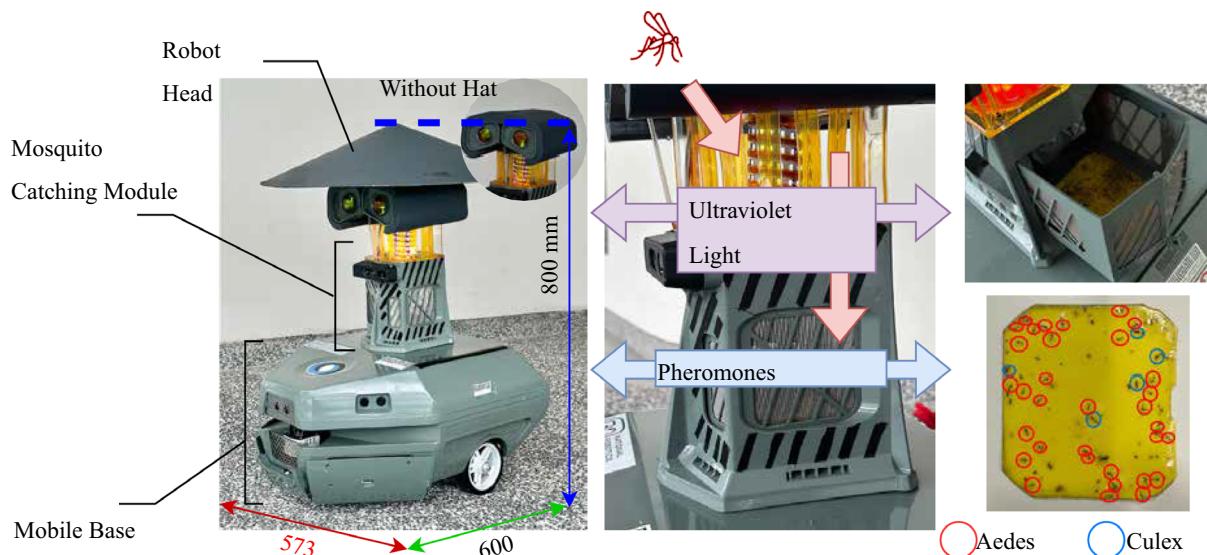
## Mosquito-capturing dragon fly robot system architecture

### Mechanical design and system architecture

The section focuses on the mechanical and electrical aspects of the developed mosquito-capturing robot, *Dragonfly*. *Dragonfly* employs a differential drive mechanism that is designed to navigate challenging semi-outdoor environments, propelled by two robust 24V BLDC motors seen in Fig.1. The mechanical structure is built for resilience, integrating aluminum and 3D-printed parts that draw inspiration from the iconic character Wall-E. Notably, the robot's dimensions are 600 mm × 573 mm × 800 mm (length × width × height), ensuring a compact yet sturdy form factor. The locomotion system is enhanced with a front caster wheel that provides a three-point configuration, bolstering stability during navigation.

The robot's electrical architecture encompasses an array of processing components and sensing peripherals. At the heart of this system is a formidable industrial PC equipped with an i7 processor, 512 GB SSD memory, and 16 GB RAM. This processing powerhouse serves as the central hub for collecting and processing data from a multitude of sensors, driving crucial decision-making processes. Microcontroller units play pivotal roles in data collection, communication, and safety. The intricate dance of data orchestration is supported by a mix of sensors, including a 2D lidar (Sick Tim 581) that facilitates navigation through SLAM and AMCL algorithms, a depth camera (Intel RealSense D415i) for rich 3D environmental perception, and an inertial sensor (VectorNav IMU) that contributes to global orientation awareness, coupled with encoder information for dead reckoning. Additional sensors such as cliff detection, ultrasonic, and bump switch sensors furnish the robot with a comprehensive understanding of its surroundings.

Powering this intricate symphony is a robust 24V 10AH capacity battery, supplemented by strategic step-down voltage regulators. With full charge, the robot can operate for 8 to 10 hours in patrol mode. It is also reconfigurable with a hat attachment to allow it to operate in drizzling consider. This comprehensive mechanical and electrical design enables the robot to autonomously navigate and fulfill its mission of mosquito capturing with precision and resilience.



**Fig. 1.** *Dragonfly* is made up of 3 key component, the head, mosquito-catching mechanism, movebase. The catching mechanism emits ultraviolet light and pheromones from the upper and lower chambers, respectively. The lower chamber also contains the sticky pad as the final stage of the mosquito-catching process. Aedes and Culex mosquito breeds are identified in the sticky pad sample shown.

### Mosquito trapping design

A sophisticated mosquito-trapping unit is at the heart of the *Dragonfly*'s mission to capture mosquitoes. This component resides within the robot's neck and embodies two core operations: mosquito attraction and capturing. The intricately designed unit employs a combination of stimuli to lure mosquitoes. Ultraviolet (UV) light, with a wavelength of 368 nm for public safety, and olfactory cues derived from a blend of octanol and lactic acid, simulating human sweat, create a compelling allure for mosquitoes, drawing them toward the robot.

The trapping process is a carefully orchestrated interplay of components, as seen in Fig. 1. There are 3 lures used for attraction: ultraviolet light, formulated pheromones, and strip patterns on the catching mechanism. The mosquito is able to enter the upper chamber, which contains the UV light source. In the upper chamber, a high-speed suction fan, rotating at 1000 RPM, generates a controlled airflow that pulls mosquitoes downward into the lower chamber of the catching mechanism. Within this lower chamber, a yellow sticky tape serves as an effective adhesive surface, which completes the capturing process. The hat helps to provide adequate shelter if the robot is expected to operate in the rain. This captured mosquito collection method not only ensures efficacy but also provides a tangible resource for future studies and analysis.

To facilitate the functioning of this unit, a microcontroller serves as the conductor, orchestrating the operation of the UV light, olfactory cues, and suction fan. The unit draws power from the same robust 24V battery that fuels the robot's various functions. This meticulously engineered mosquito trapping unit positions the *Dragonfly* as a powerful tool in the pursuit of mosquito population management.

### Human-first complete coverage path planning scheme

For a mosquito-catching robot deployment, the proposed solution will be described with detailed elaboration in this section. The structure of this section first introduces the proposed method with an overview in 4.1, then details the operating concepts that enable complete coverage in 4.3 and added social behavior in 4.4. Finally, in 4.5 describes the combined effect and integration of the two layers.

#### Preliminaries

In Fig. 2, the robot predominantly requires 3 inputs for the solution to be executed. A map of the working environment( $M$ ), the location of the robot location ( $L$ ), and the identified area where people are interacting the most "Human-Dense Area" ( $HDA$ ) Mapping<sup>55,56</sup>, localization<sup>57,58</sup>, and human identification<sup>59,60</sup> software solutions are widely established in numerous development. Alternatively, classification training and mapping can be operationally expensive, and manual production of this information may be recommended. From a typical navigation stack, the localized position of the robot will be used as the starting point of the algorithm, as most path planners do. Unlike point-to-point path planners, goal poses are not required.

The proposed work assumes that the users of the robot deployed with the proposed solution that deploys known information of human-dense areas. While it can be performed with human-in-the-loop, the identification can be automated by camera or other technology as well. This can be determined by visual observations by users or machine learning-trained policies. However, the purpose of work does aim to enable robots to roam areas with more people but does not claim that it necessarily means more mosquitoes. Rather, the solution aims to protect human-populated areas from mosquitoes, if any. The scope and aim of this paper is not to predict or hunt for mosquito-rich areas but to attend to areas where humans frequent. The key assumptions of our paper are as follows:

1. human-in-the-loop deployment or AI-assisted technologies for determining risk boundaries. The work assumes that the expert or technology-informed input is reliable and accurate,
2. mosquito-harassment risk only arises when it is a mosquito and human present areas. The risk is reduced in areas where people are less frequent.

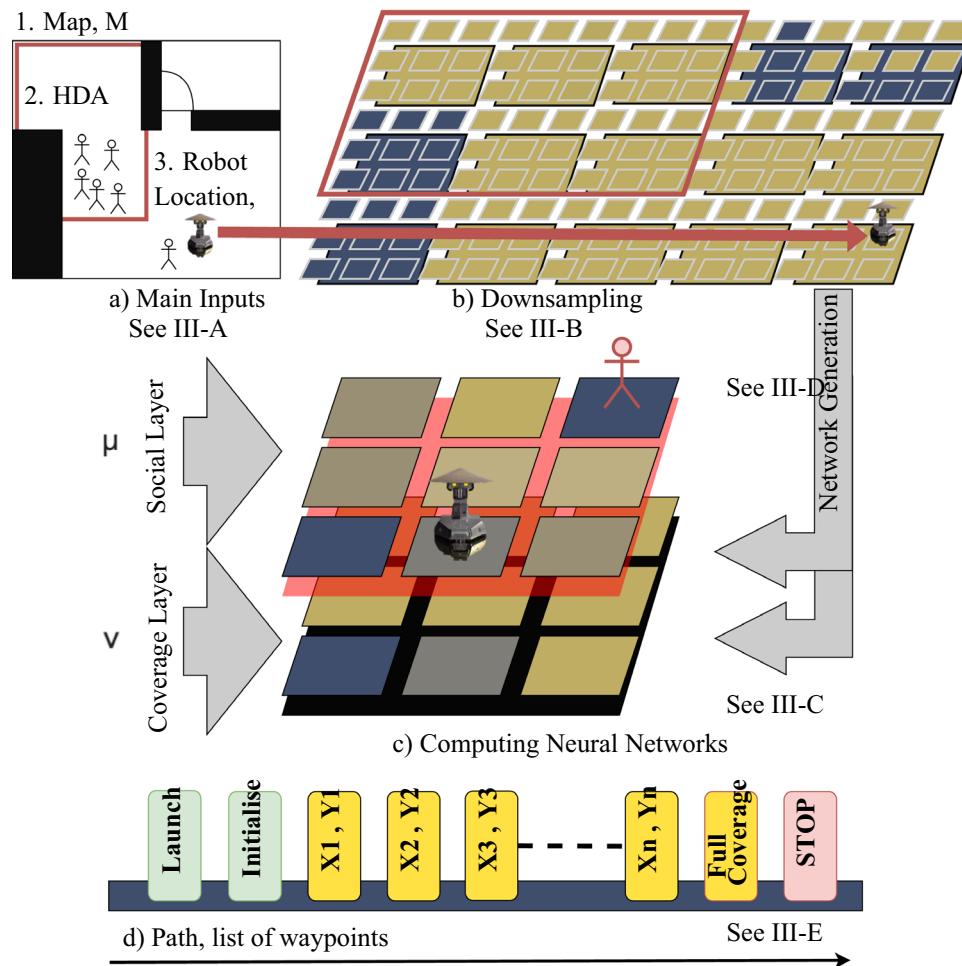
#### Map resolution

The process path computation will be processed using the provided information. The map will be downsampled and translated into two versions of neural networks: the coverage and the social layer. The downsampling process aims to reduce the resolution of the map as the coverage of mosquito catching does not require the robot to cover every inch of the area. Moreover, coverage for spatially precise coverage is computationally expensive. For all the nodes combined during the reduction, any number of occupied grids found will be considered occupied in the final grid. The optimal threshold of the resolution sensitivity to the occupancy will not be part of the scope of this paper. The  $HDA$  will translate into bounds of simulation of human activities moving around the same space. Areas outside the indicated bound can also include simulated human activity but with a significantly smaller number (i.e., inside  $HDA$ : 10, outside  $HDA$ : 2). These numbers can be set as the parameters for area bias that models the actual scenario at deployment.

#### Coverage layer

The coverage layer aims to provide complete coverage for the intended operating space. The algorithm is modeled from Glasius Bioinspired Neural Network (GBNN) found in several developmental studies<sup>61,62</sup>. GBNN remains a useful algorithm for guaranteed complete area coverage. The main equation of the algorithm is as follows:

$$N_{i(t+1)} = f \left( \sum_{j=1}^m w_{ij} N_k^+ + \alpha_i \right) \quad (1)$$



**Fig. 2.** (a) Required inputs: map of operating space (M), labeled human-dense area (HDA), and robot location, (b) down sampling the digital map into a pair of neural networks, (c) pair of neural networks, one for social attraction and the other for complete coverage, (d) produced array containing the list of step positions for complete coverage. Note: All parts only display examples for visualization of the solution concept. The photo of *Dragonfly* is shown on the right.

where  $N_i$  is the  $i$ th neuron on the neural network,  $k$  is the  $k$ th neighbor of the indexed neuron,  $w$  is the weight of the specific neighbor of a neuron and, lastly,  $\alpha$  is the neural activity level associated with the state of the corresponding point in the map, as seen in (2).

The neural network models the state of the discrete point on the map. The states of the points numerically labeled ranged from -1 to 1 in neural activity levels. The equation is as follows:

$$\alpha_i = \begin{cases} +E, & i = \text{Uncovered} \\ -E, & i = \text{Obstacle} \\ 0, & i = \text{Covered} \end{cases} \quad (2)$$

where the  $\alpha$  is the neural activity level associated with the state of the corresponding point in the map,  $i$  is the  $i$ th neuron of the neural network, and  $E$  is a large positive constant in order for that occupied areas maintain as a negative value during path planning.

The weight of each neuron determines the network's sensitivity from (1). The derivation of the weight is computed from (3) as follows:

$$w_{ij} = \begin{cases} \exp(-\beta(i-j)^2), & 0 < i - j \leq R \\ 0, & i - j > R \end{cases} \quad (3)$$

where  $w$  is the weight of the specific neighbor of a neuron on the neural network,  $i$  neuron,  $j$  is the  $j$ th neighbor of the  $i$ th neuron. The variable  $\beta$  is a positive constant.

Then, finally, the value of each neuron of the neural network will be redefined using (4) after computing (1). The equation is presented as follows:

$$f(x) = \begin{cases} -1 & , x < 0 \\ \epsilon x & , 0 \leq x < 1, \text{ where } \epsilon > 0 \\ 1 & , x > 1 \end{cases} \quad (4)$$

where  $x$  is the computed value from (1), and  $\epsilon$  is a small positive constant.

This process iterates in each step update. Hence, the neural activity map in this network changes as the robot plans its move on the corresponding map.

### Social layer

In the same manner, the social layer operates similarly. However, the interest of this layer is different. Instead of labeling covered and uncovered spaces, it labels the neighbors of simulated people walking inside and outside the HDA.

Human activity is simulated with a random walk algorithm<sup>63</sup>. Each human label is bound to move either inside or outside the HDA. Human location is denoted by the value  $-2$  in this layer. The space around the human location will be a mosquito risk zone. The risk zone is the interest of coverage for the robot, denoted with  $1$ . The model of the state of the corresponding point on the map varies from the description in 4.3, (2). The remodelled Eqs. (2) and (4), are as follow:

$$\alpha_i = \begin{cases} +E & , i = \text{Potential Risk of Mosquito's bite space} \\ -E & , i = \text{Prohibited} \\ 0 & , i = \text{Mitigated Risk space} \end{cases} \quad (5)$$

where the  $\alpha$  is the neural activity level associated with the state of the corresponding point in the map,  $i$  is the  $i$ th neuron of the neural network, and  $E$  is a large positive constant in order for that occupied areas maintain as a negative value during path planning.

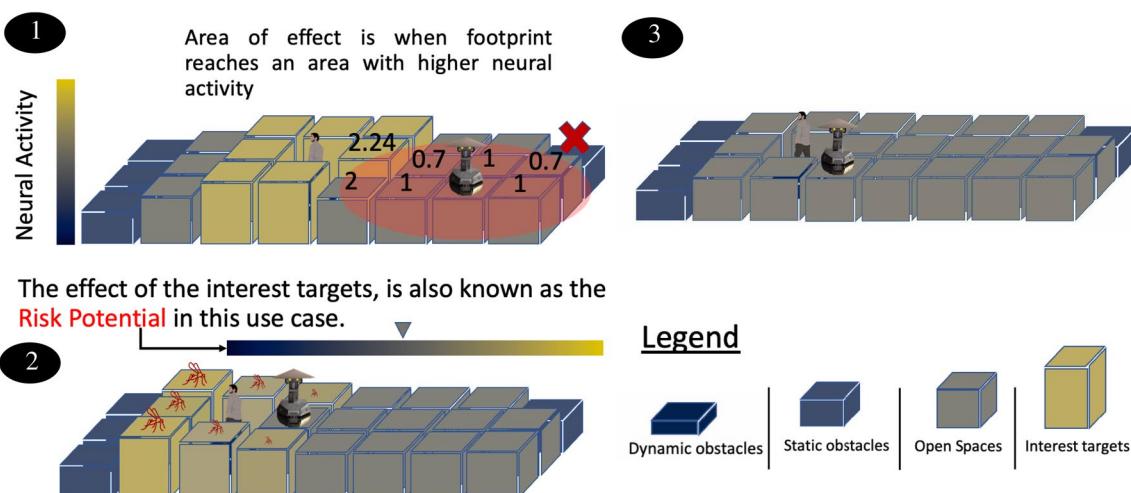
$$f(x) = \begin{cases} -2 & , x = -2 \\ -1 & , -2 < x < 0 \\ \epsilon x & , 0 \leq x < 1, \text{ where } \epsilon > 0 \\ 1 & , x \geq 1 \end{cases} \quad (6)$$

where  $x$  is the computed value from (1), and  $\epsilon$  is a small positive constant.

As the robot approaches the human position, as seen in Fig. 3, the area of effect (AOE) affects the neural activity of the robot. The closer the robot's proximity to the simulated human position, the lower the risk of mosquito attacks. When the robot is on a neighbor node from a simulated human, the risk potential of the simulation is suppressed to  $0$ .

This effect, seen in Fig. 3, is modelled using two equations, namely the AOE and risk mitigation  $r^*$  as follows:

$$r_r^* = \begin{cases} \max(d \times f(x), 0) / \phi, & d < \phi \\ 1 & , d \geq \phi \end{cases} \quad (7)$$



**Fig. 3.** 1 - Area of effect (AOE) can be described by the radial length,  $\phi$ , of the grid with respect to robot position (computing waypoint), 2 - As the robot approach closer to the human simulation, the risk potential (residual risk,  $r_r^*$ ) within the AOE boundary reduces, 3 - If the robot is next to the human pose, the risk potential (carrying risk,  $r_c^*$ ) surrounding the specific human pose reduces to 0. Interest targets refer to human surround (neighboring neurons around a simulated human position).

$$r_c^* = \begin{cases} \max(d - 1, 0)/\phi & , d < \phi \\ 1 & , d \geq \phi \end{cases} \quad (8)$$

where,  $d$  refers to the Manhattan distance of the spatial position or human position relative to the robot,  $\phi$  refers to the radius of the area of effect, and  $f(x)$  refers to the function in (4).  $r^*$  is a generalized risk variable that can be further divided into two risk parameters, namely, residual risk( $r_r^*$ ) and carrying risk( $r_c^*$ ).

The residual risk is position-specific, where a simulation of human positioning can influence it. The risk will not decrease unless it is acted upon by the robot AOE as described in (8). The carrying risk is the maximum risk potential that can be influenced by a simulation of human per neighbor position.

When a simulated human position shifts to another position, the risk potential updates according to the new set of neighbors from (6). But when the previous set of neighbor nodes is no longer a neighbor, the risk potentials reduce modeled using (7) and (8) put into (9):

$$N_{i(t+1)} = f(x) \times r_r^* \times r_c^* \quad (9)$$

where  $N_i$  is the  $i$ th neuron on the neural network.  $r_c^*$  and  $r_r^*$  is the carrying risk and residual risk, respectively.

### Waypoint generation via combined layer

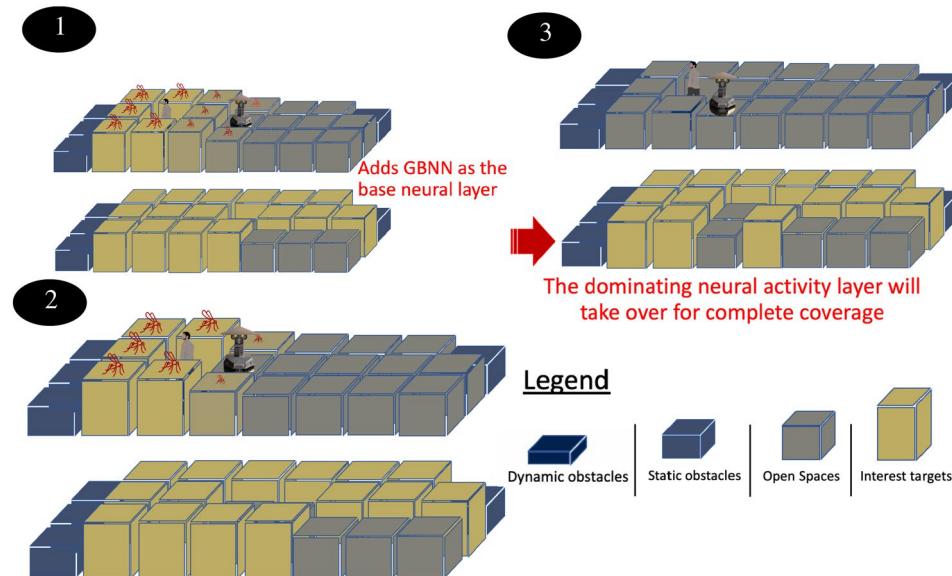
With both neural network models, the networks can be simultaneously used to determine the waypoints of the path plans. In each iteration, a new waypoint is determined in each iteration of the algorithm as seen:

$$wp_{i+1} = \max(N_{k\mu} + N_{k\nu}) \quad (10)$$

where  $wp_{i+1}$  denotes the next waypoint, and where  $\max(x)$  returns the neighbor position with the highest neural activity from its corresponding neuron in both networks. Let the social graph be denoted as  $\mu$  and the coverage graph as  $\nu$ .

At each point in time, the robot is assigned to a neuron that corresponds to its spatial position from its latest waypoint position. Both  $\mu$  and  $\nu$  will be computed to the equations described in 4.3 and 4.4, as seen in Fig. 4. The algorithm flow is shown in Algorithm 1.

From Algorithm 1,  $E$ ,  $\alpha$ ,  $\beta$ ,  $R$ ,  $\epsilon$  are coefficients used in GBNN. The variables  $d$ ,  $\phi$ ,  $r_r^*$  and  $r_c^*$  are variables used in social layer.  $N_k$  is a list containing the neighbors for the selection of the next waypoint. The next waypoint can be computed using (10) as seen in line 8 of the algorithm. The map,  $M$ , can be updated represented by  $\mu$  and  $\nu$  neuron network, along with  $r_r^*$  and  $r_c^*$ . The algorithm will continuously iterate until all opened grids are covered( $\nu \notin \mathbb{R}^+$ ). Opened grids refer to the neurons with neutral or positive values.



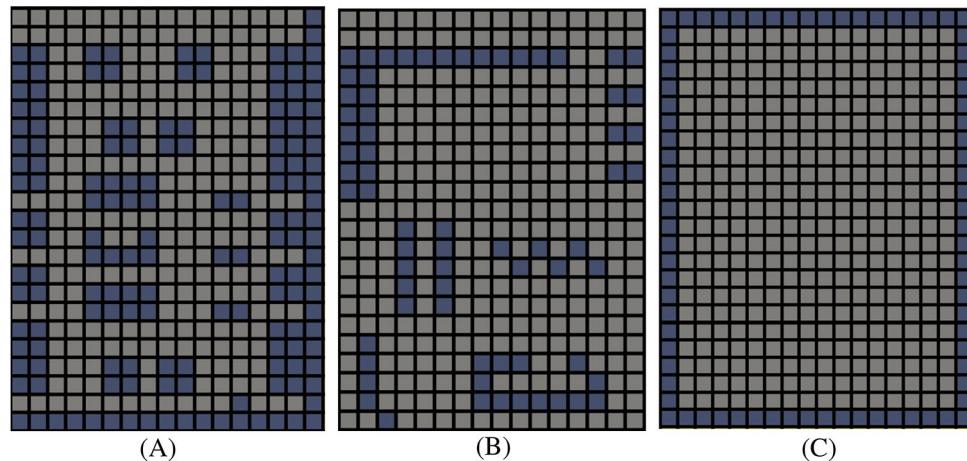
**Fig. 4.** 1 - Social layer (upper layer,  $\mu$ ) and coverage layer (lower layer,  $\nu$ ) can be stacked to represent the same space on the actual map ( $M$ ), 2 - both corresponding network neurons will react corresponding to the path plan position as described in (1) - (10), 3 - The social layer will result in complete open space neurons with no interest targets. Interest targets refer to human surround (neighboring neurons around a simulated human position for the social layer. For the uncovered layer, the open space refers to covered areas, while interest targets refer to uncovered areas.

**Input: Map with Crowd-Bound label****Output: Path Plan**

```

1: Init GBNN Constants  $\leftarrow E, \alpha, \beta, R, \epsilon, d, \phi, r_r^*, r_c^*$ ;
2:  $\mu, \nu \leftarrow M$ 
3:  $HDA \leftarrow$  polygon edge positions
4: While
5:   Simulate  $n_1$  people via random walk
6:   Simulate  $n_2$  outsiders via random walk
7:    $N_k \leftarrow$  Get neighbors
8:   Select next waypoint from  $arr_k$  using (10)
9:   Update  $\mu, \nu$ 
10:  Update  $r_r^*, r_c^*$ 
11:   $iterations += 1$ 
12:  If  $\nu \notin R+$ 
13:    Return Path Plan

```

**Fig. a.** HFA-CCPP AlgorithmInput**Fig. 5.** Various simulated environments used. The provided environments of (A), (B), and (C) are provided to demonstrate repeatability of the HFA-CCPP performance against GBNN.**Simulation**

The proposed method is validated with simulations in path plan generations. The simulation setup and results are discussed in this section. The aim of the proposed algorithm is to show evidence of a human-first approach and relative insights into its applications.

**System set-up**

The simulation is tested on a Python script running on Sypder 4.1.5, a scientific Python Development Environment. The software is operated on a Macbook Pro, M1 chip with 16 GB RAM. The simulation experiment is iterated by two-fold. The operations are run twice in the same initial conditions (same simulated environment and initial positions of the robot and simulated humans). This process is repeated 1000 times for quantification and repeatability. In each iteration, the running time taken to achieve 1 - human-first behavior, 2 - complete coverage, and lastly, 3 - number of waypoints generated in the path plan.

The running time for human-first behavior is qualified in the instance when all simulated humans surrounding risk potential are reduced to 0 ( $\alpha_i = +E \notin \mu$ ). The complete coverage is qualified in the instance when all uncovered areas are converted to covered areas (where  $N_i = 1 \notin \nu$ ). The path plan generates a waypoint from one node to another, and the total number of waypoints is logged during the simulations.

In Fig. 5, environments A, B, and C are used to evaluate the performance of the path planner. The simulation will be used with 3 simulated human in the top-left quadrant of the graph and 1 simulated human among other quadrants. In the neural network variables,  $E, r, \beta, \epsilon$  are 100, 2, 2, and 0.7, respectively.

### Simulation results

In Fig. 6, GBNN and HFA-CCPP have taken 3.22 and 1.15 seconds for completing human-first behavior, and 4.85 and 5.52 seconds for complete coverage on average, respectively. The average percentage of time taken for human-first compared to complete coverage is 66.38% and 20.68%.

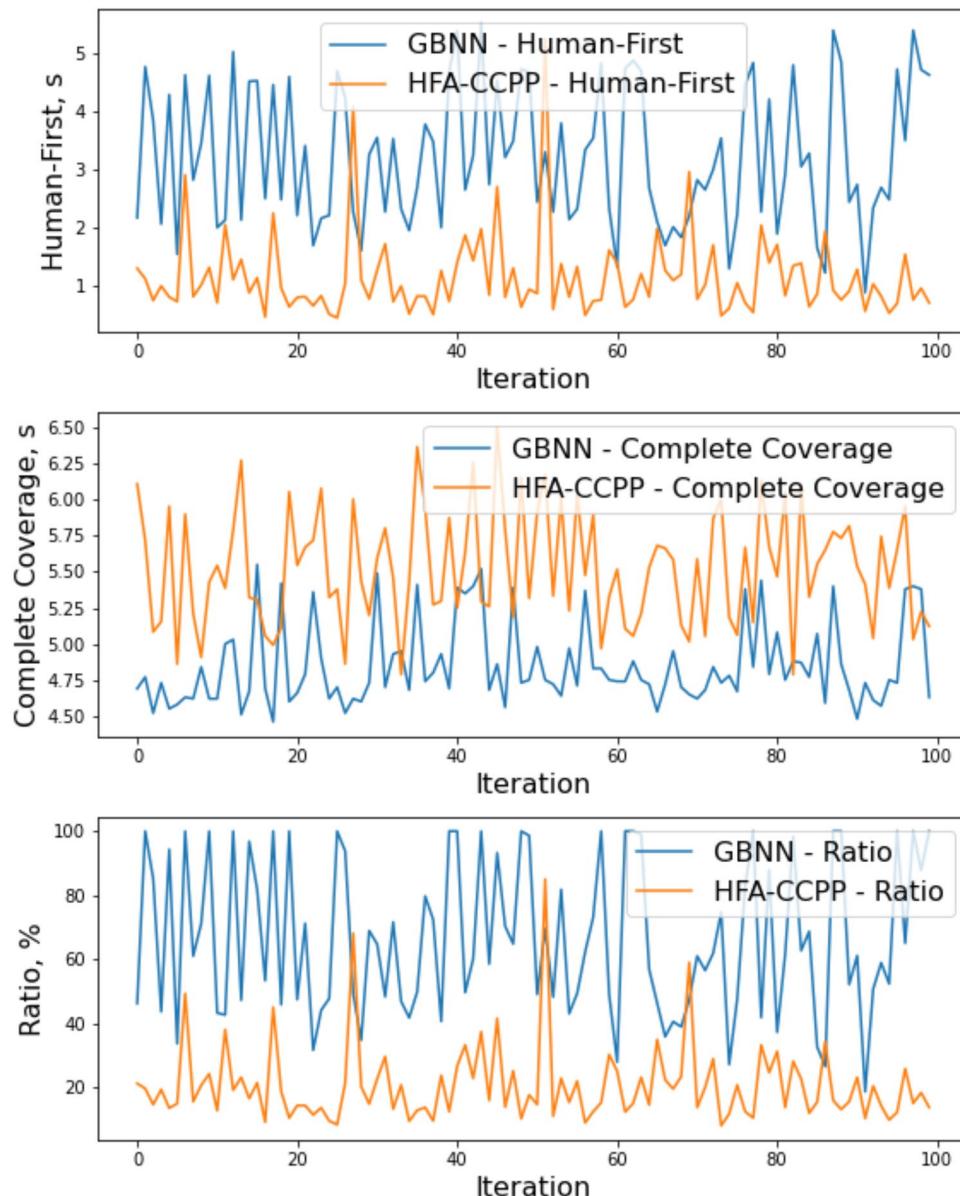
In Fig. 7, GBNN and HFA-CCPP have taken 4.00 and 1.88 seconds for completing human-first behavior, 6.43 and 7.00 seconds for complete coverage on average, respectively. The average percentage of time taken for human-first behavior compared to complete coverage is 62.77% and 26.44%.

In Fig. 8, GBNN and HFA-CCPP have taken 5.20 and 1.17 seconds for completing human-first behavior, 8.15 and 8.64 seconds for complete coverage on average, respectively. The average percentage of time taken for human-first behavior compared to complete coverage is 63.92% and 13.62%.

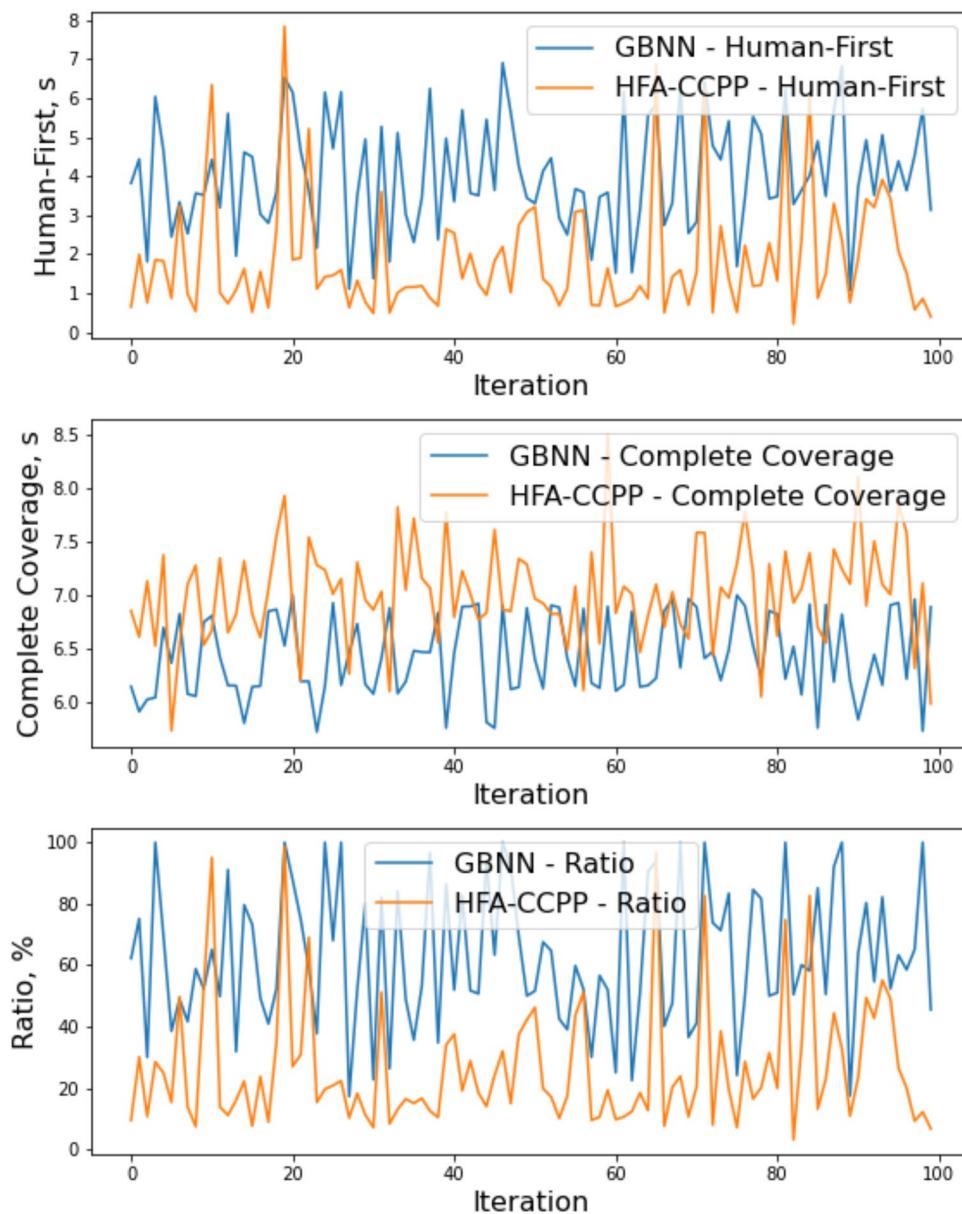
### Discussion

In Figs. 6, 7 and 8, the HFA-CCPP has shown that, in most cases, it is quicker than GBNN but is also mostly slower in terms of complete coverage. In all simulations, both GBNN and HFA-CCPP have successfully provided complete coverage in all cases.

Statistically, via Z-Test, the human-first and complete coverage performances exhibited by HFA-CCPP are not the same as GBNN at 99% confidence in all simulations. There are some vivid observations in the standard



**Fig. 6.** Simulation results for human-first behaviors and complete coverage performances shown in time, and human-first to complete coverage proportionality by percent. The comparison is between the HFA-CCPP and GBNN algorithm.

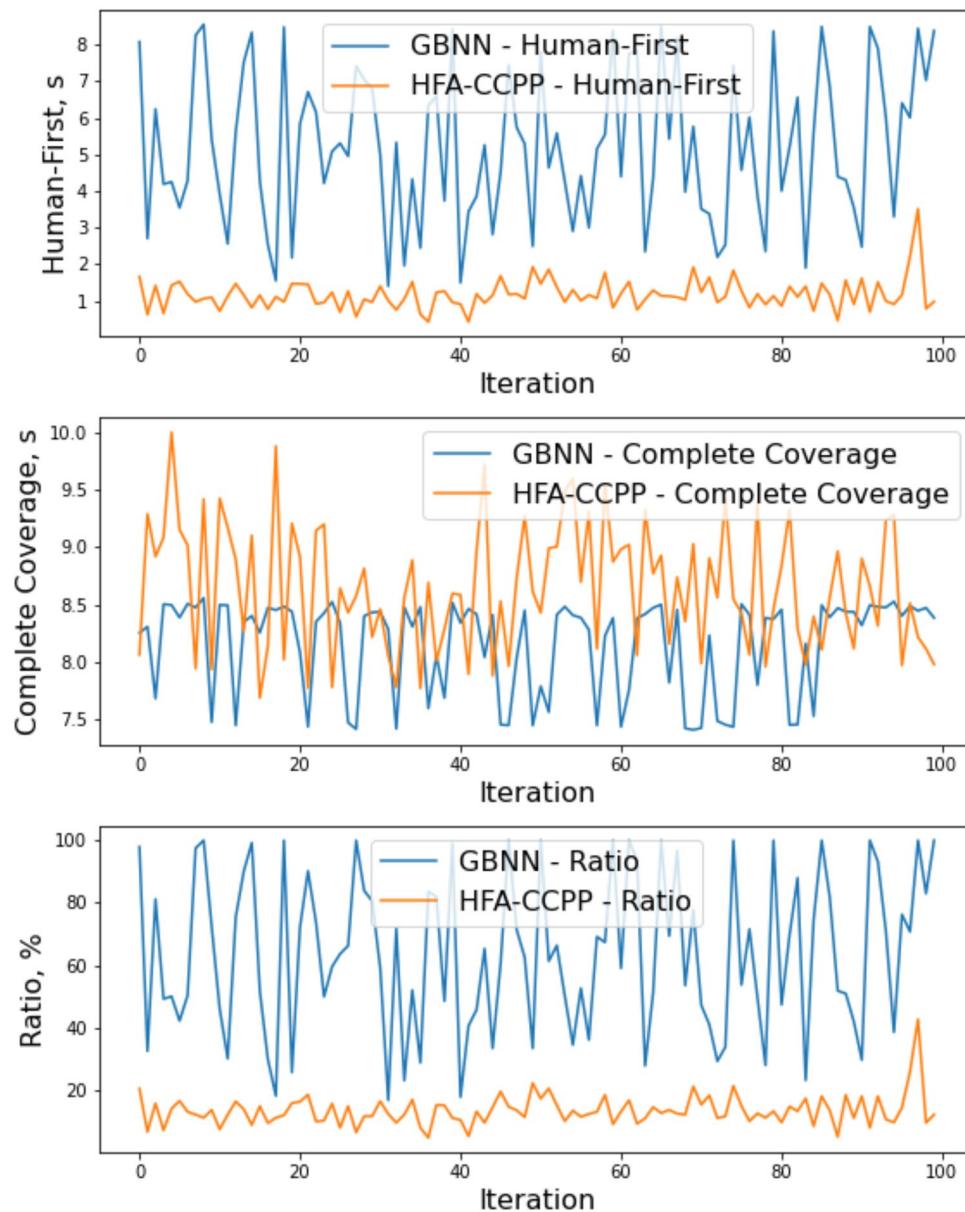


**Fig. 7.** Simulation results for human-first behaviors and complete coverage performances shown in time, and human-first to complete coverage proportionality by percent. The comparison is between the HFA-CCPP and GBNN algorithms.

deviations in the human-first approach compared against GBNN across all simulated environments from Tables 1, 2 and 3. HFA-CCPP has a smaller standard deviation in human-first approach performance, while GBNN has a smaller standard deviation in complete coverage. One exception in this observation is from the second test environment, where the data set had experienced several data spikes in the human-first approach with HFA-CCPP. In this exception, HFA-CCPP is seen with a high percentage of the time for complete coverage used for approaching intended human-dense zones.

The numerical gradient of the combined neural network layer determines the behavior of the path plan. Fig. 9 shows the gradient of neural network values for enabling complete coverage behavior and a human-first approach. Each waypoint is determined by its previous position and its surrounding neuron values. These values are derived solely from Eqn. 1. From between iterations, the neuron network always selects the neighboring neuron with the highest neural activity. The gradient of values across the neural network is attributed to covered, uncovered, and obstacle representations on each neuron. In a case where all possible neighbours share the same magnitude, the selected next waypoint is based on the search-based bias. The emergent behavior forms completed coverage with priority human-dense areas.

It is important to note that GBNN is mostly deterministic in its step performance as compared to HFA-CCPP. HFA-CCPP is largely probabilistic due to the randomly walking human simulation. In some cases, the human simulation is also an obstacle for the planner on the  $\mu$  layer and affects the neural activity propagation. Both



**Fig. 8.** Simulation results for human-first behaviors and complete coverage performances shown in time, and human-first to complete coverage proportionality by percent. The comparison is between the HFA-CCP and GBNN algorithms.

Metric	Environment A		
	Human-first	Complete coverage	Ratio
Mean (GBNN)	3.22	4.85	66.38
Mean (HFA-CCPP)	1.15	5.52	20.68
Std (GBNN)	1.19	0.29	24.12
Std (HFA-CCPP)	0.73	0.40	12.28

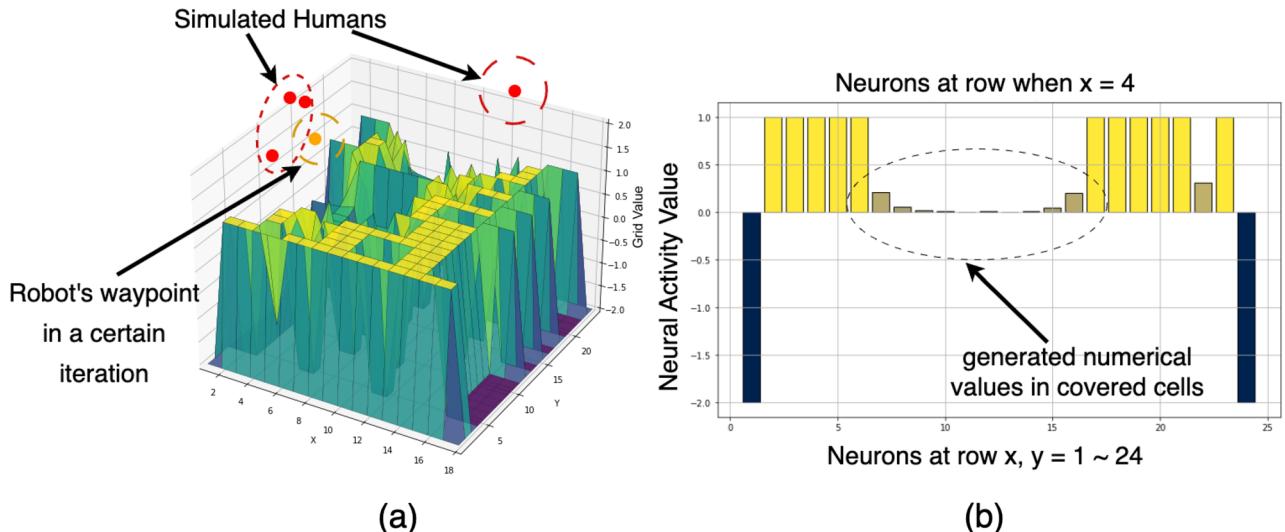
**Table 1.** Summary Statistics.

planners' performances vary across different environments, as the obstacles in  $\nu$  layer affect the propagation in neural activity as well. The elongated path from the top-left-hand corner of the simulated environment - B, may have made neural activity influenced the path planner to complete slower than expected on the occasions that the HFA-CCPP performed badly in the human-first approach. The HFA-CCP has the smallest standard deviation in an open space for the human-first approach, which vividly demonstrates the effects of the environment.

Metric	Environment B		
	Human-first	Complete coverage	Ratio
Mean (GBNN)	4.01	6.43	62.77
Mean (HFA-CCPP)	1.88	7.01	26.44
Std (GBNN)	1.42	0.38	23.06
Std (HFA-CCPP)	1.52	0.47	20.78

**Table 2.** Summary Statistics.

Metric	Environment C		
	Human-first	Complete coverage	Ratio
Mean (GBNN)	5.20	8.15	63.93
Mean (HFA-CCPP)	1.17	8.65	13.62
Std (GBNN)	2.05	0.41	25.17
Std (HFA-CCPP)	0.42	0.55	4.94

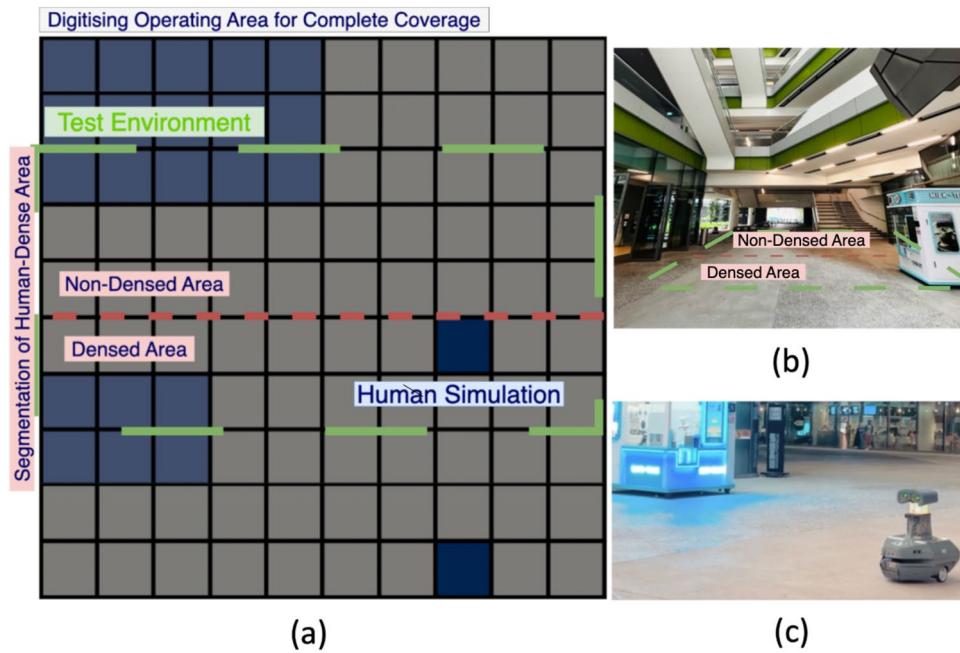
**Table 3.** Summary Statistics.**Fig. 9.** A 3-dimensional plot of the neural network of the combined layer in (a) and its illustrated numerical gradient in (b).

### Recommendations

In practice, a shorter amount of computation time is preferred; however, the bias of the path plan is required towards human-activity areas to ensure sensible implementation of the use case. Complete coverage would be achieved with purposeless if the bias is not ensured. The time lost thus becomes insignificant as it is approximately 1 to 2 seconds on average. This can be improved by reducing the resolution of the coverage map and avoiding implementation for coverage in long and narrow environments, as discussed.

From the simulations, it takes GBNN up to 8.15 seconds for complete coverage based on its intended deployment method. The proposed human-first approach takes up to 1.88 seconds maximum. The approach can be computationally divided into *humanfirstcoverage* and *remaining coverage* phases in execution to maintain quick responsiveness within 1 to 2 seconds. The *remaining coverage* will not be iterative updated until human first coverage is completed. Onboard computational resources can be reduced real-time deployment.

Hence, for optimal application of the proposed algorithm, it has to be deployed in an area with the absence of an isolated passageway. In most scenarios, the proposed method would perform better than a typical GBNN despite its dependency on the applied environment. The duration of approaching human-first bias is shown by achieving approximately 3 times in reduction compared to the duration for complete coverage.



**Fig. 10.** (a) Virtual representation of the map space and neuron states, (b) corresponding experiment space, (c) Example of *Dragonfly* operating in the experiment space.

Metric	Definition
Path	Constructed by the list of poses. While the robot moves from one pose to the next, all poses are considered as waypoints that define the path.
Path keeping	For all waypoint poses listed by the path planner except the last waypoint should be maintained at 0.4 m.
Docking	The final step is where the system positions itself at the final waypoint for the docking procedure. For the waypoint pose listed by the path planner should be maintained at 0.2 m.

**Table 4.** Performance Metric Definition Table.

### Real-world experiment

The proposed method is demonstrated with experimental studies by path planner capabilities in real life and discusses implications involved with the integration of the proposed path planner. The aim of this section is to validate its purpose in real-life implementation for complete coverage with the human-first approach.

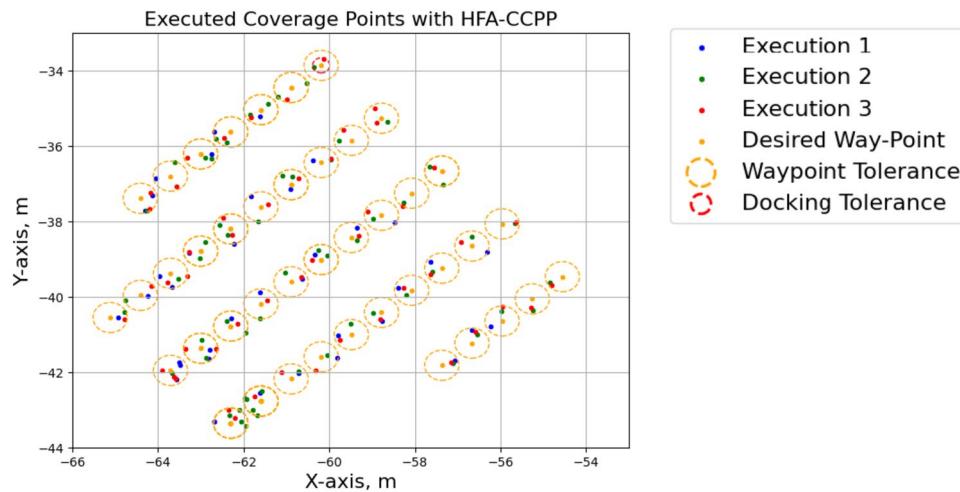
#### System set-up

The system is implemented on a mosquito-catching robot platform, *Dragonfly*, as introduced in Sect. 3.1 - 3.2. The platform operates on Robot Operating System (ROS, melodic) navigation stack with Oriental motor control.

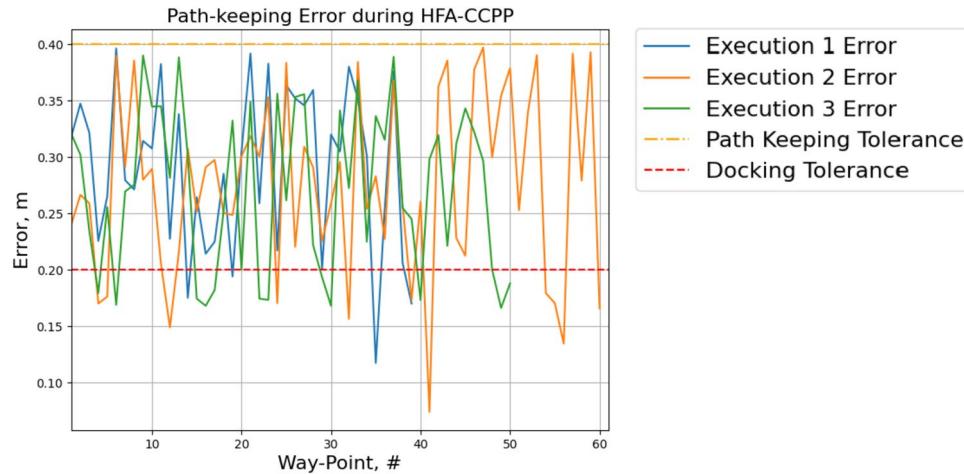
The experimental test is conducted 3 times per algorithm. Two algorithms will be used, with HFA-CCPP and GBNN. The test environment is shown in the given map shown in Fig. 10. The map is generated by the GMapping algorithm found on the Robot Operating System (ROS). The map is post-processed to reduce its resolution while preserving its occupancy. The post-processing logic simply takes a correspondingly larger area (i.e., resolution changed to 0.5 meters per pixel). The robot system operates in a 3-level navigation stack, task planning, global path planning, and local path planning. The proposed algorithm will operate at the task planning level; conventional global and local path planners from ROS are used in the navigation stack, namely the Astar algorithm and Dynamic Window Approach, respectively.

In all execution of each test, the robot will be initialized at a fixed position for fairness in each test. The HDA is manually determined and provided to the robot. The experiments are also operated in the evening period in Singapore, 1700hrs to 2000 hrs SGT, where most mosquitoes are found in this facility. Additionally, a fresh sticky pad will be inserted into the lower chamber of the lure-and-catch mechanism at the start of each method experimentation and removed when the experiment iteration cycle ends.

The metric used to evaluate the performances of the state-of-the-art and the proposed algorithm is defined in Table 4.



**Fig. 11.** Scatter plot of 3 rounds of execution with the pose of opened spaced as reference for HFA-CCPP way-keeping and docking performance. The orange circle represents way-point keeping tolerance while the red circle represent docking tolerance at specified location.



**Fig. 12.** Error plot of 3 rounds of execution with the pose of opened spaced as reference for HFA-CCPP way-keeping and docking performance. The upper band limit represents way-keeping tolerance, while the lower band limit represents docking tolerance.

## Results

The results of a complete coverage are shown in Fig. 11 and its accuracy Fig. 12. The set path-keeping accuracy is to be within 0.4 meters for accuracy and 0.2 meters for docking accuracy. The robot had kept its path and docking to satisfactory levels. The path plan is deterministic, as the path plan is in a fixed environment, and the initial position is fixed. The running time for complete coverage of the 3 executions by HFA-CCPP is 11 minutes and 59.735 seconds, 17 minutes and 58.124 seconds, and 15 minutes and 52.978 seconds. The respective average time per step is 18.455 seconds/step, 17.969 seconds/step, and 19.060 seconds/step. The path planner took 1 minute and 29.089 seconds (5 stpng), 1 minute and 20.083 seconds (6 stpng), and 1 minute 29.089 seconds (5 stpng) to clear out the mosquito bite risk potential in its social layer.

## Discussion

From the experiment, the results have shown that the path planner is capable of complete coverage in both path planners, human-first approach by showing highly deterministic behavior of reducing risk potential from human-dense areas by moving towards specific areas. The area of the experiment zone is reduced as it takes a long time for complete coverage. The reduced area is sufficient to provide clarity and high repeatability for proof-of-concept.

In the experiments, the robot has shown less significance between the two algorithms when it comes to complete coverage performances, while the human-first performance of HFA-CCPP maintains largely significance comparing 1 minute 29 seconds (worst case in HFA-CCPP) to 1 minute 36 seconds (best case in

GBNN). This is approximately 32.2% quicker than traditional GBNN. The variations of robot running time have shown that the running time of the algorithm plays a less significant role in the physical demand. The human-first approach would benefit the human-dense area as it should remain the key priority in most practical cases.

In a large space, the robot might not have sufficient battery power to enable the robot to fulfil its role in complete coverage. The impact of such robots would be reduced if it does not provide coverage in human-dense areas. Applicable scenarios can be found in areas such as a public park area or a complicated facility with the risk of mosquito threats. HFA-CCPP would benefit the application more than just a complete coverage algorithm. This benefit applies to any existing complete coverage algorithm.

For the interest of understanding the proposed algorithm with the navigation stack, the lost recovery behavior of the ROS navigation stack will be triggered if localization is poor. There is no estimation of human-dense area or human activity performed by the proposed work. The human-dense areas can be provided manually by the user, while the moving humans are simulated during path planning. The reason for such implementation is to provide a natural weaving path plan in human-dense areas while not needing to rely on live-human detection, which is disruptive in motion execution, computationally intensive and reduces the actual quality of localization. Hence, tracking live humans is avoided to negate implementation, making it easier to adopt while preserving the proposed algorithm purpose for human-first approach path planning.

While strategic location improves the capture rate, there are other factors that affect capture rates, such as weather and time of the day. Mosquito-catching success rate depends on the presence of mosquitoes. However, the aim of the proposed work is to validate that the algorithm does not hinder or limit the lure-and-catch mechanism's ability to capture mosquitoes. The captured mosquito reflects the protection provided by the robot during our experiment. The proposed framework does not adopt mosquito-seeking capabilities but to lure-and-catch it if one is available. Despite the availability of a lure-and-catch mechanism, the study presents limitations in proper entomological controls for conclusive biological effects of the mosquito capturing rate. The scope of this study is unable to extend to the factors of capturing success rate.

Lastly, in the deploy spaces, the experiment do not have to consider the actual presence of humans. This allows the proposed path planner to be deployed without complex visual classification of humans and pose predictions being active. Without active these functions to be constantly activated, computational resources and energy consumption is reduced for the robot service to be deployed sensibly. Live human presence can be trivialized by having simulated human activities in the designated *HDA* by the deployer. The aim of the algorithm is to reduce mosquito population in human populated areas instead of actual chasing or tailing a human movement.

The *Dragonfly* has demonstrated deterministic ability in prioritizing human-dense area first in contrast to the original state-of-the-art GBNN with probabilistic clearance of the human-dense area. In other practical considerations, such as deploying in a large area for maximum return of interest of robot owners, the human-dense area must first be covered before robot energy is fully drained in the large compound.

## Conclusion

This paper presents a novel complete coverage path planning scheme (HFA-CCPP) leveraging on a modified Glasius bio-inspired neural network algorithm (GBNN) for a mosquito-capturing robot named *Dragonfly* that can generate a coverage path closer to a crowded area. The proposed method had an extra social layer on top of the existing GBNN to simulate social behaviours between the human activity zone and the robot. This paper also introduced the mechanical and system architecture of the developed mosquito-capturing robotic platform. The proposed algorithm has been systematically benchmarked with the traditional GBNN algorithm in both simulated and real-world environments. The proposed algorithm performs about 3 times better in terms of bringing the robot closer to humans as a priority, which is critical since the mosquitoes tend to be closer to human-dense areas. Even though the proposed method is slower by 1 second in terms of performing the complete coverage, it can perform 2.5 times faster in covering the crowded area. Such a behaviour will help in capturing more mosquitoes in the real-world operation, which is also validated in our real-time experiments. The methods can be extended to other applications, such as drinks-serving waiter robots and event or facility management, where humans are of key interest to the specified tasks. In both simulations of the algorithm running time and experimented robot running time, HFA-CCPP has demonstrated its priority to the human-dense environment and complete coverage as its second priority.

## Future work

There are several potential directions for future development of the HFA-CCPP. First, it can be extended to validate across a variety of application settings beyond mosquito control. Second, the human-first approach could be enhanced by incorporating a dynamic model in which the human-dense area evolves over time, supported by live human tracking for real-time local planning. Finally, integrating lightweight behavioral proxies, such as ambient CO<sub>2</sub> concentration levels to align the coverage strategy closely with entomological modelling, thereby enhancing ecological realism and bridging sim2real gap from current standpoint.

## Data availability

The datasets used and analyzed during the current study are available from the corresponding author on reasonable request.

Received: 20 April 2025; Accepted: 13 August 2025

Published online: 23 August 2025

## References

1. Dengue worldwide overview european centre for disease prevention and control. <https://www.ecdc.europa.eu/en/dengue-monthly>. Accessed: 2024-10-06.
2. Buhler, C., Winkler, V., Runge-Ranzinger, S., Boyce, R. & Horstick, O. Environmental methods for dengue vector control-a systematic review and meta-analysis. *PLoS Negl. Trop. Dis.* **13**, e0007420 (2019).
3. Ogunlade, S. T. et al. A review: Aedes-borne arboviral infections, controls and wolbachia-based strategies. *Vaccines* **9**, 32 (2021).
4. Rawal, D. A review on different strategies used for biological control of mosquitoes. *Int. J. Mosquito Res.* **6**, 41–43 (2019).
5. Electric flying insect trap kit, sticky catcher traps for fruit fly, flying insect, mosquito, gnats, moths. [https://www.lazada.sg/product/s/electric-flying-insect-trap-kit-sticky-catcher-traps-for-fruit-fly-flying-insect-mosquito-gnats-moths-i3161042797-s21505005766.html?from\\_gmc=1&fl\\_tag=1](https://www.lazada.sg/product/s/electric-flying-insect-trap-kit-sticky-catcher-traps-for-fruit-fly-flying-insect-mosquito-gnats-moths-i3161042797-s21505005766.html?from_gmc=1&fl_tag=1). Accessed: 2025-04-15.
6. Qualikey mosquito attractant compatible with dynatrap. [https://www.amazon.sg/Qualikey-Mosquito-Attractant-Compatible-DynaTrap/dp/B0D4DLQHX4?source=ps-sl-shoppingads-lpcontext&ref\\_=plfs\\_&psc=1&smid=ARPIJN329XQ0D](https://www.amazon.sg/Qualikey-Mosquito-Attractant-Compatible-DynaTrap/dp/B0D4DLQHX4?source=ps-sl-shoppingads-lpcontext&ref_=plfs_&psc=1&smid=ARPIJN329XQ0D). Accessed: 2025-04-15.
7. Foldable electric mosquito killer, wall mounted, handheld, 2-in-1 usb charging, home fly swatter, lightwave lure fly for camping. <https://www.aliexpress.com/item/1005006500124757.html>. Accessed: 2025-04-15.
8. Siddiqua, R., Rahman, S. & Uddin, J. A deep learning-based dengue mosquito detection method using faster R-CNN and image processing techniques. *Ann. Emerg. Technol. Comput. (AETiC)* **5**, 11–23 (2021).
9. Goodwin, A. et al. Mosquito species identification using convolutional neural networks with a multitiered ensemble model for novel species detection. *Sci. Rep.* **11**, 13656 (2021).
10. Yin, M. S. et al. A deep learning-based pipeline for mosquito detection and classification from wingbeat sounds. *Multimed. Tools Appl.* **82**, 5189–5205 (2023).
11. Zhao, Z. et al. Mosquito brains encode unique features of human Odour to drive host seeking. *Nature* **605**, 706–712 (2022).
12. Laursen, W. J. et al. Humidity sensors that alert mosquitoes to nearby hosts and egg-laying sites. *Neuron* **111**, 874–887 (2023).
13. Greppi, C. et al. Mosquito heat seeking is driven by an ancestral cooling receptor. *Science* **367**, 681–684 (2020).
14. Semwal, A., Melvin, L. M. J., Mohan, R. E., Ramalingam, B. & Pathmakumar, T. Ai-enabled mosquito surveillance and population mapping using dragonfly robot. *Sensors* **22**, 4921 (2022).
15. Greenfieldboyce, N. How mosquitoes sniff out human sweat to find us (2019).
16. Kyaw, P. T. et al. Energy-efficient path planning of reconfigurable robots in complex environments. *IEEE Trans. Rob.* **38**, 2481–2494 (2022).
17. Le, A. V. et al. Inter-reconfigurable robot by modified informed sampling-based shortest path planning in cleaning and maintenance. *Robotica* **43**, 1–22 (2025).
18. Le, A. V., Veerajagadheswar, P., Thiiha Kyaw, P., Elara, M. R. & Nhan, N. H. K. Coverage path planning using reinforcement learning-based tsp for Htetran-a polyabolo-inspired self-reconfigurable tiling robot. *Sensors* **21**, 2577 (2021).
19. Le, A. V. et al. Long-term trials for improvement of autonomous area coverage with a Tetris inspired tiling self-reconfigurable system. *Expert Syst. Appl.* **206**, 117810 (2022).
20. Song, B., Wang, Z. & Zou, L. An improved PSO algorithm for smooth path planning of mobile robots using continuous high-degree Bezier curve. *Appl. Soft Comput.* **100**, 106960 (2021).
21. Do, H. et al. Heat conduction combined grid-based optimization method for reconfigurable pavement sweeping robot path planning. *Robot. Auton. Syst.* **152**, 104063 (2022).
22. Le, A. V. et al. Autonomous floor and staircase cleaning framework by reconfigurable stetro robot with perception sensors. *J. Intell. Robot. Syst.* **101**, 1–19 (2021).
23. Orozco-Rosas, U., Montiel, O. & Sepulveda, R. Mobile robot path planning using membrane evolutionary artificial potential field. *Appl. Soft Comput.* **77**, 236–251 (2019).
24. Le, A. V., Vo, D. T., Dat, N. T., Vu, M. B. & Elara, M. R. Complete coverage planning using deep reinforcement learning for polyiamonds-based reconfigurable robot. *Eng. Appl. Artif. Intell.* **138**, 109424 (2024).
25. Yang, L., He, Q., Yang, L. & Luo, S. A fusion multi-strategy marine predator algorithm for mobile robot path planning. *Appl. Sci.* **12**, 9170 (2022).
26. Muthugala, M. V. J. et al. A self-organizing fuzzy logic classifier for benchmarking robot-aided blasting of ship hulls. *Sensors* **20**, 3215 (2020).
27. Prabakaran, V. et al. Hornbill: A self-evaluating hydro-blasting reconfigurable robot for ship hull maintenance. *IEEE Access* **8**, 193790–193800 (2020).
28. Le, A. V. et al. Towards staircase navigation and maintenance using self-reconfigurable service robot. *Expert Syst. Appl.* **266**, 125969 (2025).
29. Tang, Q. et al. Complete coverage path planning strategy for reconfigurable robot with variable footprint. *IEEE Access* (2025).
30. Gonzalez, E., Alvarez, O., Diaz, Y., Parra, C. & Bustacara, C. Bsa: A complete coverage algorithm. In *proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2040–2044 (IEEE, 2005).
31. Wang, P., Xiang, Q., Hu, Y. & Tian, M. Contour-guided zigzag path planning for the automation of pothole spray repair based on the depth image. *Int. J. Pavement Eng.* **24**, 1–16 (2022).
32. Le, A. V. et al. Reinforcement learning-based energy-aware area coverage for reconfigurable Hrombo tiling robot. *IEEE Access* **8**, 209750–209761 (2020).
33. Vo, D. T. et al. Toward complete coverage planning using deep reinforcement learning by trapezoid-based transformable robot. *Eng. Appl. Artif. Intell.* **122**, 105999 (2023).
34. Le, A. V., Parween, R., Elara Mohan, R., Nhan, N. H. K. & Enjikalayil Abdulkader, R. Optimization complete area coverage by reconfigurable htrihept tiling robot. *Sensors* **20**, 3170 (2020).
35. Sriniketh, K. et al. Robot-aided human evacuation optimal path planning for fire drill in buildings. *J. Build. Eng.* **72**, 106512 (2023).
36. Yi, L. et al. Reconfiguration during locomotion by pavement sweeping robot with feedback control from vision system. *IEEE Access* **8**, 113355–113370 (2020).
37. Weerakoon, T., Ishii, K. & Nassiraei, A. A. F. An artificial potential field based mobile robot navigation method to prevent from deadlock. *J. Artif. Intell. Soft Comput. Res.* **5**, 189–203 (2015).
38. Gao, Y., Hu, T., Wang, Y. & Zhang, Y. Research on the path planning algorithm of mobile robot. In *2021 13th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, 447–450 (IEEE, 2021).
39. Zhu, D., Tian, C., Sun, B. & Luo, C. Complete coverage path planning of autonomous underwater vehicle based on GBNN algorithm. *J. Intell. Robot. Syst.* **94**, 237–249 (2019).
40. Yi, L. et al. Complete coverage path planning for omnidirectional self-reconfigurable cleaning robot using AGBNN. *IEEE Trans. Automat. Sci. Eng.*
41. Shen, Z., Song, J., Mittal, K. & Gupta, S. Ct-CPP: Coverage path planning for 3d terrain reconstruction using dynamic coverage trees. *IEEE Robot. Automat. Lett.* **7**, 135–142 (2021).
42. Hassan, M. & Liu, D. Ppcpp: A predator-prey-based approach to adaptive coverage path planning. *IEEE Trans. Rob.* **36**, 284–301 (2019).
43. Mansouri, S. S., Kanellakis, C., Fresk, E., Kominik, D. & Nikolakopoulos, G. Cooperative coverage path planning for visual inspection. *Control. Eng. Pract.* **74**, 118–131 (2018).

44. Wang, L. et al. Full coverage path planning methods of harvesting robot with multi-objective constraints. *J. Intell. Robot. Syst.* **106**, 17 (2022).
45. Jensen-Nau, K. R., Hermans, T. & Leang, K. K. Near-optimal area-coverage path planning of energy-constrained aerial robots with application in autonomous environmental monitoring. *IEEE Trans. Autom. Sci. Eng.* **18**, 1453–1468 (2020).
46. Abd Rahman, N. A., Sahari, K. S. M., Hamid, N. A. & Hou, Y. C. A coverage path planning approach for autonomous radiation mapping with a mobile robot. *Int. J. Adv. Rob. Syst.* **19**, 17298806221116484 (2022).
47. Manimuthu, A. et al. Energy consumption estimation model for complete coverage of a tetromino inspired reconfigurable surface tiling robot. *Energies* **12**, 2257 (2019).
48. Do, V. T. & Pham, Q. C. Geometry-aware coverage path planning for depowdering on complex 3D surfaces. *IEEE Robot. Automat. Lett.* **8**(9), 5552–9 (2023).
49. Ghosh, D., John, E. A. & Wilkinson, A. Clever pest control? the role of cognition in biological pest regulation. *Anim. Cogn.* **26**, 189–197 (2023).
50. Hajjaj, S. S. H. & Sahari, K. S. M. Review of agriculture robotics: Practicality and feasibility. In *2016 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)*, 194–198 (IEEE, 2016).
51. Gillies, M. & Wilkes, T. A comparison of the range of attraction of animal baits and of carbon dioxide for some west African mosquitoes. *Bull. Entomol. Res.* **59**, 441–456 (1969).
52. Moore, S. et al. The efficacy of different mosquito trapping methods in a forest-fringe village, Yunnan province, Southern China. *Southeast Asian J. Trop. Med. Public Health* **32**, 282–289 (2001).
53. Bernier, U. R., Kline, D. L., Allan, S. A. & Barnard, D. R. Laboratory studies of Aedes aegypti attraction to ketones, sulfides, and primary chloroalkanes tested alone and in combination with l-lactic acid. *J. Am. Mosq. Control Assoc.* **31**, 63–70 (2015).
54. Davis, E. E. Development of lactic acid-receptor sensitivity and host-seeking behaviour in newly emerged female aedes aegypti mosquitoes. *J. Insect Physiol.* **30**, 211–215 (1984).
55. Sivanantham, V., Le, A. V., Shi, Y., Elara, M. R. & Sheu, B. J. Adaptive floor cleaning strategy by human density surveillance mapping with a reconfigurable multi-purpose service robot. *Sensors* **21**, 2965 (2021).
56. Xu, X. et al. A review of multi-sensor fusion slam systems based on 3D lidar. *R. Sens.* **14**, 2835 (2022).
57. Le, A. V. et al. Social density monitoring toward selective cleaning by human support robot with 3D based perception system. *IEEE Access* **9**, 41407–41416 (2021).
58. Peavy, M., Kim, P., Oyediran, H. & Kim, K. Integration of real-time semantic building map updating with adaptive monte Carlo localization (AMCL) for robust indoor mobile robot localization. *Appl. Sci.* **13**, 909 (2023).
59. Le, A. V. & Choi, J. Robust tracking occluded human in group by perception sensors network system. *J. Intell. Robot. Syst.* **90**, 349–361 (2018).
60. Wensel, J., Ullah, H. & Munir, A. Vit-ret: Vision and recurrent transformer neural networks for human activity recognition in videos. *IEEE Access* **11**, 72227–49 (2023).
61. Sun, B., Zhu, D., Tian, C. & Luo, C. Complete coverage autonomous underwater vehicles path planning based on Glasius bio-inspired neural network algorithm for discrete and centralized programming. *IEEE Trans. Cognit. Dev. Syst.* **11**, 73–84 (2018).
62. Yi, L. et al. Complete coverage path planning for reconfigurable omni-directional mobile robots with varying width using GBNN (n). *Expert Syst. Appl.* **228**, 120349 (2023).
63. Mahapatra, D. & Triambak, S. Towards predicting COVID-19 infection waves: A random-walk monte Carlo simulation approach. *Chaos, Solit. Fract.* **156**, 111785 (2022).

## Acknowledgements

This research is supported by the National Robotics Programme under its National Robotics Programme (NRP) BAU, Ermine III: Deployable Reconfigurable Robots, Award No. M22NBK0054 and also supported by A\*STAR under its “RIE2025 IAF-PP Advanced ROS2-native Platform Technologies for Cross sectorial Robotics Adoption (M21K1a0104)” programme.

## Author contributions

A.Y.S.W, P.V, A.V. L : Conceptualization, Methodology, Validation, Writing - Original Draft, Writing—Review and Editing. A.Y.S.W: Software. R.E .M: Supervision, Project administration, Funding acquisition, Writing—Review and Editing. All authors have read and agreed to the published version of the manuscript.

## Declarations

### Competing interests

The authors declare no competing interests.

### Additional information

Correspondence and requests for materials should be addressed to A.V.L.

Reprints and permissions information is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025