IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 31 May 2023, accepted 3 August 2023, date of publication 14 August 2023, date of current version 24 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3304682

RESEARCH ARTICLE

Automatic Recognition of Collective Emergent Behaviors Using Behavioral Metrics

SHUO YANG, DILINI SAMARASINGHE, ANUPAMA ARUKGODA, SHADI ABPEIKAR, (Member, IEEE), ERANDI LAKSHIKA, (Senior Member, IEEE), AND MICHAEL BARLOW

School of Engineering and Information Technology, University of New South Wales, Campbell, ACT 2600, Australia

Corresponding authors: Shuo Yang (shuo.yang5@adfa.edu.au), Dilini Samarasinghe (d.samarasinghe@adfa.edu.au), Anupama Arukgoda (a.arukgoda@adfa.edu.au), and Shadi Abpeikar (s.abpeikar@adfa.edu.au)

ABSTRACT Collective emergent behaviours are commonly seen in nature such as in flocks of birds and schools of fish. These behaviours are the results of years of evolution and have been studied in artificial agent systems in a wide range of application areas such as robotics, serious games, and crowd simulations. Automatic recognition of such collective behaviours is imperative in such application areas in order to measure and improve the effectiveness and efficiency of the artificial agent systems, especially when it involves machine learning approaches where human labelling is not feasible. While it is easy for the human eye to recognise collective behaviours, this is an extremely challenging task for a machine to automatically recognise them as such emergent behaviours cannot be captured by a simple mathematical equation. This paper investigates how emergent behaviours can be automatically recognised through capturing the behavioural aspects of the collective nature of the agents' performance. We identify seven metrics such as grouping, order, and flock density that can capture diverse and distinct emergent characteristics of agent behaviours. Five machine learning models that use a combination of these metrics as features of a range of representative behaviours were trained to investigate the potential of automatic recognition of collective emergent behaviours. The evaluation results show that training the machine learning models with the proposed approach enables automatic recognition of a range of diverse emergent collective behaviours. Further, we conducted leave-one-behaviour-out experiments on the representative behaviours and the metrics used. The results confirmed that each behaviour and metric have a unique impact on accurate recognition of emergent behaviours in collective agent systems.

INDEX TERMS Emergent behaviours, collective behaviour metrics, machine learning, automatic collective behaviour recognition, boids system, leave-one-out.

I. INTRODUCTION

Collective behaviours in nature are found in the way flocks of birds, swarms of insects, herds of land animals or schools of fish, move, aggregate, and disperse [1]. In simulating such collective motion, individual agents are often codified with simple rules. Therefore, they act according to their own discrete perceptions without centralised control. The local interactions among them give rise to emergent fluid motions at the group level which replicate the collective behaviours in nature [2]. Emergent behaviours in an agent system refer to the kind of collective behaviours that are not explicitly programmed but are the result of local interactions among individual components in the system [3]. Those kind of emergent behaviours with an embedded pattern in their motion are also referred to as collective structured behaviours in the literature [4]. Hence, we use the terms emergent behaviour and structured behaviour interchangeably in this paper. The first automated collective motion simulation was introduced by Craig Reynolds in 1987 [5],

The associate editor coordinating the review of this manuscript and approving it for publication was Li Zhang.

which is widely known as Reynolds' Boids (bird-android) model. Reynolds proposed three simple rules: cohesion, avoidance, and alignment which in combination can result in collective motions similar to those observed in the nature.

Collective structured behaviours are useful in multiple application fields including military swarm attacks [6], civil aviation [7], search and rescue robotics [8], and hazardous material localisation [9]. Being able to recognise and analyse collective structured behaviours is helpful in imitating and reproducing these behaviours [10] as well as generating new behaviours [3], [11]. Consequently, they help make informed responses and decisions based on the understanding of the more profound nature of these behaviours. Although humans could easily recognise such behaviours, limitations related to efficiency lead to the requirement of automatic recognition models [12]. Nevertheless, automatic recognition of collective structured behaviours is still an open question due to the constraints associated with inherent unpredictability and sensitivity to control parameters [13], [14]. With the latest development of new types of artificial collective behaviours, the recognition problem becomes increasingly difficult. This is because the collective behaviour configuration of robots, and other multi-agent systems might result in behaviours which are not exhibited in natural systems.

Current automatic recognition models for collective structured behaviours usually require the temporal parameter space of each individual agent to make an accurate prediction [10]. These methods focus on internal interactions between agents across a set of discrete time steps. However, due to these complex interactions in such multi-dimensional systems, the parameter space can become extremely dynamic. Calculations within such a space on the individual level introduce exponential complexity to the model [15]. Therefore, this paper investigates how collective structured behaviours can be recognised through capturing the collective nature of the agents' performance in the spatial parameter space. To focus on the movement of the whole group of agents rather than individual behaviours, we identify a set of metrics that capture their collective behavioural aspects in terms of direction of motion, collisions among agents, and ability to balance convergence and dispersal as a group, among others. These metrics abstract the spatial features of the group that can be extracted without looking into each individual agent's temporal behaviour.

Existing works have shown the effectiveness of using individual or a simple combination of such metrics in determining the presence of one specific behaviour as each of these metrics can capture a specific characteristic of structured behaviours [16], [17], [18]. However, each individual metric is not versatile enough to identify many structured behaviours as they can only capture a limited number of characteristics independently [19]. This paper proposes a unique combination of seven metrics as discussed in Section III-B that can capture the high-dimensional characteristics of collective structured behaviours, and in turn, can provide an accurate recognition for a wide range of collective structured behaviours.

We propose a machine learning (ML) model which encapsulates the seven metrics mentioned above and the features extracted from a pool of eight structured behaviours as discussed in Section III-A that are recognised by the literature as a representative set of diverse agent behavioural characteristics [4] as a means to automatically recognise emergent agent behaviours. A point-mass boid simulation system is used as the experimental platform to generate the agent behavioural dataset for the training of the ML models. Using the eight structured behaviours together with a range of random unstructured behaviours generated with the point-mass simulator as the trainig set, we evaluate five ML models on their ability to capture a wide variety of collective structured behaviours. The ML classifiers: Decision Tree, Naïve Bayes, K-Nearest Neighbour (KNN), Support Vector Machines (SVM), and Multi-Layer Perceptron (MLP); were selected to cover a diverse range of learning approaches to investigate the full capacity of our proposed metric combination and the representative behaviours in collective structured behaviour recognition. The trained models are tested on a different set of structured and unstructured behaviours to investigate their potential in recognising the agent behavioural characteristics. These evaluations demonstrate that this approach could identify collective structured behaviours previously unseen during training in contrast to the existing approaches [10]. We also investigate redundancies and quantify the importance of each of the seven metrics and the eight structured behaviours used in training through a leave-one-out statistical analysis [20] to testify the comprehensiveness of our framework. As such, the contributions of this paper are as follows:

- Identifying a comprehensive set of seven metrics that capture the behavioural aspects of collective motions.
- Introducing a novel automatic recognition model of collective behaviours by employing the combination of these seven metrics and eight structured behaviours that are recognised by the literature as a representative set of diverse agent behavioural characteristics as the training attributes.
- Illustrating the applicability of the model to automatically recognise unseen structured and unstructured collective behaviours.
- Demonstrating the significance and comprehensiveness of the seven metrics and the representative eight structured behaviours in automatic collective behaviour recognition.

The rest of the paper is organised as follows. Section II summarises the existing literature related to collective behaviour recognition. The proposed methodology including details on the seven metrics and the eight structured behaviours is presented in Section III. The experimental evaluations of the applicability of the model are discussed in

detail in Section IV. Finally, Section V concludes the paper with reference to future extensions to the proposed work.

II. BACKGROUND

This paper evaluates the potential of machine recognition of collective behaviours using a set of collective behaviour metrics as the training attributes. This section includes discussions on the relevant background, as follows:

- Section II-A provides discussions related to the artificial collective behaviour methods.
- Section II-B illustrates the existing categories of collective behaviour of boids, defined in the literature.
- Section II-C indicates the available metrics in the literature for evaluating simulated collective behaviours.

A. ARTIFICIAL COLLECTIVE BEHAVIOUR MODELS

Collective behaviours in nature refer to the behaviours of flocks of birds, schools of fish, and herds of land animals [1]. The collective behaviour motions illustrate the way that these organisms move close to each other, and in the same direction, without running into each other, while doing a specific mission. The missions include but are not limited to searching for a food source, transporting the food, hunting and migrating [21]. These behaviours are also considered productive and efficient in multi-agent systems and multi-robot systems [21]. The seminal computer system which simulated collective behaviours was inspired by flocks of birds. This computer-based system introduced by Reynolds is known as the Reynolds boids model [5]. A boid (short for bird-android) is a small, simulated agent, which imitates the collective behaviour of birds. Three boids rules were introduced by Reynolds as follows:

- Cohesion ($\vec{c_i^t}$): This is the force which makes boids move close to the other boids in their neighbourhoods (N_c) within an area called cohesion radius R_c.
- Alignment ($\vec{a_i^t}$): This is the force which makes boids move in the same direction as the other boids in their neighbourhoods (N_a) within an area called alignment radius R_a.
- Separation ($\vec{s_i^t}$): This is the force which makes boids avoid collision with the other boids in their neighbourhoods (N_s) within an area called separation radius R_s.

To make boids form collective structured behaviours, these rules are required to be managed by applying the cohesion weight (W_c), alignment weight (W_a), and separation weight (W_s) in each corresponding radius, respectively. Applying these weights on each of the N boids of $B^i \in \{B^1, B^2, \ldots, B^N\}$ and for each time step $t \leq T$ will result in the temporal state values of cohesion, separation, and alignment, as per (1), (2), and (3), respectively. In these equations, x_t^i, and v_t^i refer to the position point and velocity vector of boid i at time step t, respectively.

$$W_c \times \vec{c_i^t} = (c_{x_t}^i, c_{y_t}^i); \quad \vec{c_i^t} = \frac{\sum_i x_t^i}{|(N_c)_t^i|} \tag{1}$$

$$W_a \times \vec{a_i^t} = (a_{x_t}^i, a_{y_t}^i); \quad \vec{a_i^t} = \frac{\sum_i v_t^i}{|(N_a)_t^i|} \tag{2}$$

$$W_s \times \vec{s_i^t} = (s_{x_t}^i, s_{y_t}^i); \quad \vec{s_i^t} = \frac{\sum_i x_t^i}{|(N_s)_t^i|} \tag{3}$$

Moreover, the velocity vector of each boid i at time step t, will be updated using (4).

$$v_{t+1}^i = v_t^i + W_c\vec{c_i^t} + W_s\vec{s_i^t} + W_a\vec{a_i^t} \tag{4}$$

Using the updated velocity, the position of each boid i at time step t, will be updated using (5).

$$x_{t+1}^i = x_t^i + V_{t+1}^i \tag{5}$$

For more information on Reynolds' boid model please see [5].

Since Reynolds' model, there have been several extensions to this model and other initiatives to describe artificial collective behaviours. Saber and Murray [22] mention that although Reynolds' model could imitate flocks of birds using the three rules, the convergence of this model is not proved. They mention that the main issue of such multi-agent systems is the group agreement or consensus problem. Therefore, they propose a consensus protocol for a multi-agent systems that allows the agents to agree in a distributed and cooperative fashion. Further, John Conway proposes the idea of the Game of Life [23] which is a simple simulation of the dynamic evolution of a society of living organisms. It is a cellular automation defined on a square grid where the state of each cell is determined by a set of local rules. It is a zero-player game where the evolution is only determined by the initial state with no further input. Each cell can be in one of the two states representing the presence (live) or absence (dead) of a living individual. The rules that apply to the evolution process are based on the eight nearest neighbours of each individual in the grid:

- An individual will die at the next time step if there are less than two (under-population) or more than three (overpopulation) live neighbours; else, it will remain alive.
- At any dead cell, a new individual will be born at the next time step only if there are exactly three live neighbours (reproduction).

This model is considered as a useful tool to understand and represent a variety of complex and stable societies with numerous local stationary configurations generated by only a few simple rules. In our paper, we use Reynolds' model as the basis to explore the means of automatically detecting artificial emergent behaviours.

B. CATEGORIES OF COLLECTIVE BEHAVIOUR

Different taxonomies have been proposed for collective beahviours in the literature [21], [24]. Brambilla et al. [25] classified collective behaviours into four main groups of spatially organising behaviours, navigation behaviours, collective decision making, and other collective behaviours. In another review article by Kolling et al. [1] a task-dependent

classification of bio-inspired collective behaviours is provided, which includes four categories: aggregation and rendezvous, flocking and formation control, deployment and area coverage, and foraging. Following this, Bayındır [26] provided broader categories of collective behaviour of robots, in terms of the specific mission and tasks they are performing. Specifically, the collective (swarm) robotics behaviours are categorised into nine behaviours regarding the robots' tasks including aggregation, foraging, flocking, and path formation. None of the categories discussed above is defined based on machine-recognisable behaviours. Recently, Khan et al. [4] categorised collective behaviours into two large detectable classes of ''structured'' and ''unstructured'' behaviours in a point-mass simulator. These two categories mainly belong to the formation category, identified by Kolling et al. [1], Bayındır [26], and Brambilla et al. [25]. Structured behaviour refers to the behaviour of boids with an embedded recognisable pattern in their motion [4]. Eight sub-classes were identified including flocking, line, gravity, and firefly motion formations [4]. Unstructured behaviour refers to any random motion with no detectable embedded pattern. These categories identified in [4] show a good performance of being recognised by machine learning models in recent literature including evolutionary approach- [4], supervised models [27], and reinforcement learning [15].

Hence, this paper uses these behaviours identified in [4] for the purpose of automatic collective behaviour recognition. The eight structured behaviours were chosen based on their capacity to capture characteristics of collective motion. Each behaviour has a distinct emphasis on a different subset of Reynolds' three boid rules as further discussed in Section III-A. In contrast to the existing methods, this paper uses collective behaviour metrics as the attributes of the training data. The existing literature related to the collective behaviour metrics is discussed in Section II-C.

C. COLLECTIVE BEHAVIOUR METRICS

As mentioned earlier, many research articles have proposed computer-based collective behaviours inspired by natural swarms. To evaluate the quality of the designed computer-based collective behaviours, many articles proposed different metrics, known as collective behaviour (swarm behaviour) metrics. Vicsek et al. [28] defined a metric to evaluate if the agents in a collective behaviour move in the same direction, which is known as the order metric. Genter et al. [29] proposed a set of metrics for evaluating the influencing agent positions in a computer-based flock behaviour. An influencing agent is defined as the agent which leads the other flock members within a specific neighbouring radius. Four metrics are identified regarding the three rules of Reynolds' model to evaluate the influencing agents. They evaluate 1) the number of flocking agents which are not connected to the influencing agent, 2) the number of connections between the connected agents and the influencing agent, 3) the number of direct connections between the connected agents and the influencing agent, and 4) the number of flocking agents which are not directly connected to the influencing agent. These metrics are dependent on the influencing agent. However, in another article, two more global metrics: group angular and group polarisation, are used to evaluate two specific collective structured behaviours of flocking and touring [17]. Aggregating these two metrics then led to identifying the general group metric for collective behaviour evaluation proposed by Ferrante et al. [30]. In another work, Harvey et al. [18] used the order and group metrics to classify collective behaviours. Then, by providing a human study, Harvey et al. [12] investigated which collective behaviour parameters defined by Reynolds' three boid rules, could help humans to judge if the behaviour is ordered or grouped. Hence, due to the proven applicability of group and order metrics to evaluate the quality of structured behaviours, they are used as two of the collective behaviour metrics of the training attributes in this work. More details on these two metrics are indicated in Section III-B.

In another approach, Barlow and Lakshika [31] proposed three concepts for evaluating collective behaviours, including measuring the flocking performance, quantifying situational awareness of agents, and quantifying the computational cost. These three approaches could be used to evaluate ''teaming'' systems. Szabo et al. [32] propose three metrics: Hausdorff distance, active Hausdorff distance, and statistical complexity; to identify and analyze the potential emergent behaviours. The emergent behaviours studied in this paper are flocks of birds, game of life, and predator-prey. Further, Birdsey et al. [33] introduce an observation tool that includes several metrics for identifying the self-organised behaviours. In this work the metrics are analysed in terms of effectiveness under same experimental setting. A few of the metrics identified here include working/adaptivity time (WAT), availability, and situation performance. Moreover, in recent work, Alharthi et al. [14] proposed the use of four metrics for evaluating the quality of collective behaviours. These metrics are motion metrics, sparsity metrics, diversity metrics, and connectivity metrics. Recently, by extending the metrics identified by Alharthi et al. [14] and in a broader research study, Hussein et al. [16] proposed a divergent set of metrics to evaluate collective behaviours. This article identified ten metrics for evaluating collective behaviours, by investigating the literature. Some of these metrics are directly related to teaming interaction performance among artificial agents that present collective behaviour (swarm behaviour) and humans. Rather than the concept of human-swarm teaming, we articulate that some of the metrics identified by Hussein et al. [16] are common in collective behaviour evaluation. These metrics are collision count, flock density, number of stragglers, subgroup number, and diffusion. Based on the above discussion, we identify that these five metrics in combination with group and order measures mentioned above are promising in representing the behaviour level characteristics of collective motion.

Furthermore, these metrics can be measured solely based on spatial observations which is an unobtrusive mode of collecting behavioural features. Therefore, these seven metrics were chosen as the attributes of the training set for the ML methods used in this paper. These models are then trained for automatic collective behaviour recognition. The computation procedure of these seven metrics is discussed in more detail in Section III-B.

III. PROPOSED METHODOLOGY

In this section, we introduce the different collective behaviours we adopted to create the datasets for our experiments, the features selected, and the ML models employed to automatically classify structured and unstructured collective behaviours. Fig. 1 illustrates the experimental framework used in this regard. The details of the dataset, metrics, and the ML models depicted in the framework diagram are discussed in the following sections.

A. DATASET

Our initial step was to create a substantially large dataset with structured and unstructured behaviours, to be trained using the ML models. We created two different datasets which from here onwards will be referred to as ''training'' and ''testing'' datasets. The parameter space of structured and unstructured behaviours used for training and testing are available under Appendix A. Video recordings of all behaviours involved can be found as supplementary materials.

1) TRAINING DATASET

The training dataset contains metrics pertaining to both structured and unstructured behaviours. By using a point-mass simulator with 200 boids, in a wrap-around 1400 $\times$ 1000 arena, we recreated the eight behaviours stated in the works of Khan et al. [4] by tuning the same parameters. Below, we discuss these eight structured behaviours, with respect to the three boids rules cohesion, separation, and alignment.[1]

1) Flocking: This behaviour reflects the movement of a flock of birds, school of fish, insects etc. They move in a group in the same direction as their neighbours while avoiding collisions.
2) Lines: In this behaviour we can observe boids queuing one after the other and forming a line. They may split into several lines or multiple lines could merge together over time.
3) Spermatozoa: This behaviour is simulated with a very high separation between boids which will lead to a collective motion where boids move in the same direction while avoiding collisions. However, they do not move together in a small group, rather, they are spread across the world while maximising the distance between each other.
4) Old Man River: This behaviour is similar to grouped flocking behaviour but a noticeable distance between the boids can be observed largely due to the high separation.
5) Gravity Wells: In this behaviour we can observe that with time, boids get attracted to other boids in the vicinity, forming tightly grouped clusters of boids due to high cohesion.
6) Firefly: In this behaviour, the movement of the boids is rather fast and chaotic. However, a repetitive pattern of boids moving away and then back towards different gravitational points can be observed.
7) Brownian: This behaviour can be achieved with a high separation between boids. Hence, the boids show no grouping behaviour nor any collisions, and they move freely in random directions. However, Brownian behaviours are different from unstructured behaviours as Brownian actively avoids collisions.
8) Ink in Water: This behaviour shows characteristics of both ''Flocking'' and ''Lines''. It has an undulating, smooth, flowing behaviour similar to lines but unlike lines, boids do not move straight; rather, in a wiggly manner.

The eight unstructured behaviours used for training have a combination of random cohesion, alignment, and separation weights, which makes boids move freely without considering collision avoidance, and with no pattern of motion in the arena [4]. Their unstructured nature has also been confirmed through human observations [4], [10]. In the point-mass simulator, we tuned the parameters for each behaviour and obtained the metrics discussed in Section III-B within the time window of $[1000-1500]$ time-steps and averaged them. These averaged values for the seven metrics were extracted in order to create the features of the datasets. Since we generated data for 50 samples of each behaviour, our training dataset consists of $50 \times 16 = 800$ rows of data.

2) TESTING DATASET

The next dataset, which is referred to as the testing dataset, contains eight more structured behaviours and 20 unstructured behaviours obtained from the work of Abpeikar et al. [27]. This data is not used in training. Hence, these are unseen samples that will be tested on the trained models. To ensure that the testing structured behaviours are statistically different from those of the training set, a voting mechanism which involve human perspective is applied. The voting mechanism is discussed in details in Section IV.

Similar to the training data generation, the average over the time window of $[1000 - 1500]$ time-steps is generated for these behaviours in the point-mass simulator. Then, the seven metrics are computed to generate the feature set. Again, the data is generated for 50 samples of each behaviour, and consequently, the testing dataset contained $50 \times 28 = 1400$ rows of data.

[1]The exact configuration values are provided under Appendix A.

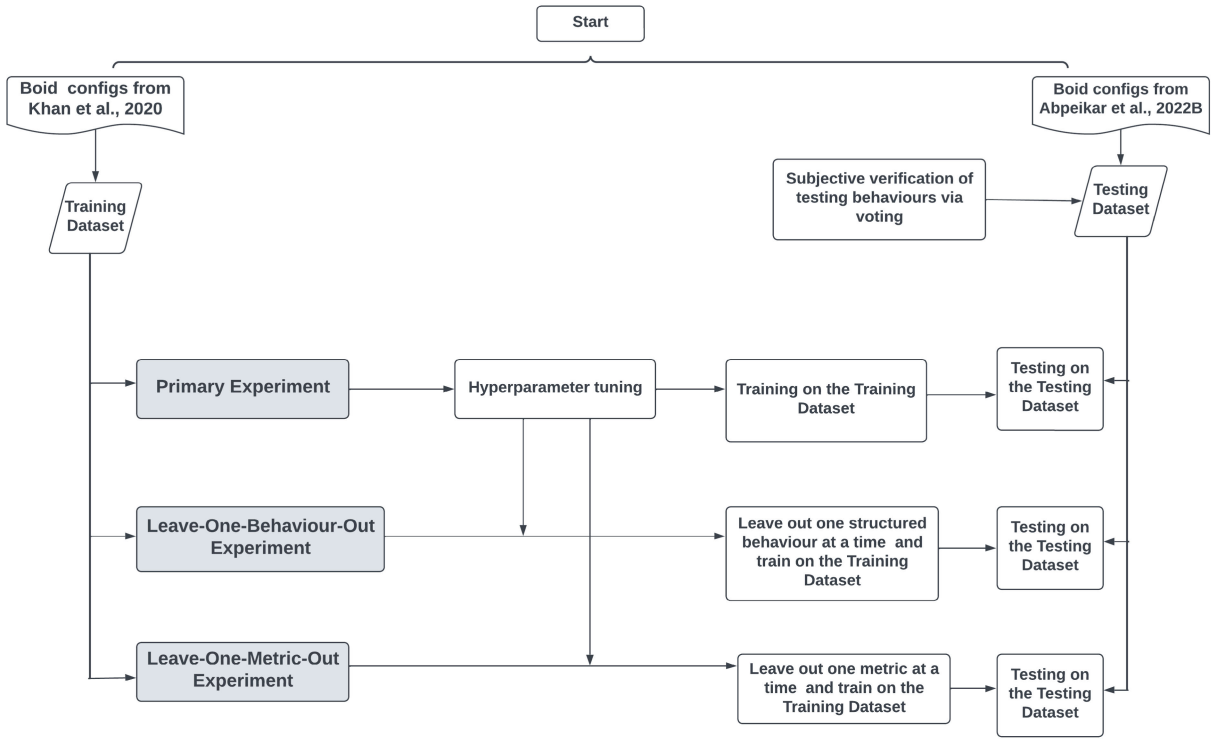


FIGURE 1. A diagrammatic representation of the experimental framework. Three experimental evaluations of the proposed model and a verification of the testing dataset are included in the framework to investigate the applicability of the model.

B. FEATURE GENERATION USING COLLECTIVE BEHAVIOUR METRICS

We selected the following seven collective behaviour metrics based on the discussion presented in Section II, as our features. These metrics in combination are used to measure the behavioural characteristics of collective motions in boids systems. The combination includes characteristics of agents across their performance in collision avoidance, behaviour within groups, and directional motion. Hence, this combination could be considered as a good representation of recognising collective behaviours. Each of these metrics captures a different aspect of the behaviours. Further evaluations to investigate the redundancies and impact of each of the metrics in recognition of collective behaviours are presented in Section IV.

1) Collision Count: Collision count is the number of boid collisions per time-step [34], where a collision is accounted when the distance between two boids is less than half the vision range of boids.
2) Flock density: Also referred to as "flock thickness", this measure calculates the number of boids in a defined unit of area Ar_n [34]. If the total number of boids is n, flock density fd is given by,

$$fd = \frac{n}{Ar_n} \tag{6}$$

3) Grouping: Also referred to as "cohesion", grouping calculates how connected a swarm is [16], [30]. Hence, we obtained the separation distance of each boid from the rest of the flock and averaged it across the entire flock. When n is the number of boids, position of boid i is b_i, separation of boid i from the rest of the flock s_i is calculated as follows.

$$s_i = \frac{1}{n-1} \sum_{j=1}^{n} \|b_i, b_j\| \tag{7}$$

4) Straggler Count: Stragglers are the boids that do not belong to boid groups [34]. By tuning a threshold distance, the number of boids at a distance further than this threshold from a cluster of boids was accounted as the straggler count. We identified half the value of the vision range of boids as an appropriate threshold distance after a sensitivity analysis with different values.
5) Order: Order is the averaged normalised velocities of the boids [35]. It can be calculated as per (8), where n is the total number of boids and v_i is the velocity of the i^{th} boids [28].

$$Order = \frac{1}{n} \left| \sum_{i=1}^{n} v_i \right| \tag{8}$$

6) Subgroup count: It is beneficial for flocks of animals/agents to split while moving especially to avoid obstacles. Under subgroup count, we calculate the number of groups the entire swarm has been split into,

using the algorithm proposed to calculate grouping in the works of Navarro and Matía [36].

7) Diffusion: Diffusion calculates the convergence and dispersal of boids. The combination of these two aspects will evaluate how the movement of each individual boid could lead to a visualizable collective motion within the whole system [37].

C. COLLECTIVE BEHAVIOUR RECOGNITION USING MACHINE LEARNING MODELS

Once the training and testing datasets with structured and unstructured behaviours were created with their metrics as explained above, we conducted three experiments to evaluate the potential of five ML models: Decision Tree, Naïve Bayes, MLP, KNN, and SVM. The behavioural space being examined is complex and unpredictable. Therefore, the suitability of different ML models in addressing the problem is unknown. These five models were chosen to evaluate the impact of different properties of the dataset and the approach including being linearly separable, size of the dataset, number of behaviours, and supervised versus unsupervised learning techniques. The experiments including the subjective verification of the testing dataset and the three evaluation approaches of the proposed models are as follows:

1) Subjective Verification of the Testing Dataset: The purpose of testing on an unseen dataset is to investigate the potential of the model to recognise collective behaviour characteristics that may not have been present in the training dataset. The testing dataset was derived using the boid configurations proposed by Abpeikar et al. [15]. As those boid parameters were generated via an unsupervised learning technique, we needed to ensure that the behaviours generated through the parameters were visually different from the 8 structured behaviours in the training set. Hence, we conducted a verification experiment to get the human perception of the similarity between the behaviours in the testing set and the training set. This provides further evidence to support the applicability of the model in recognising a wider range of diverse collective agent behaviours.
2) Primary Experiment: By using the training dataset of 400 data points, we first conducted 5-fold cross validation for each of the ML models. The models with the optimal hyperparameters are the model configurations used for all the experiments. The chosen values of the tuned parameters can be found in the Appendix B. As the primary experiment, we trained the optimal models on the complete training dataset and tested on the testing dataset. This experiment evaluates the accuracy of each of the ML models and their capacity to recognise collective behaviours.
3) Leave-one-behaviour-out Experiment: With the intention of evaluating the impact of each of eight structured behaviours on the performance of the ML models, we trained the five ML models by leaving out one structured behaviour at a time from the training dataset. These models were then tested on the testing dataset and the left-out behaviours.
4) Leave-one-metric-out Experiment: The third set of experiments were conducted to verify the significance of the metrics to sufficiently classify structured and unstructured behaviours. We trained the models by leaving a metric out at a time from the training dataset and tested on the testing dataset.

TABLE 1. Voting results for comparing the possible similarity between to testing behaviours and the training behaviours. The rank is inversely proportional to the similarity of the behaviours to the original training behaviours.

Testing Behaviours	Average Rating	Rank
Testing Behaviour 1	1.4	3
Testing Behaviour 2	2.2	5
Testing Behaviour 3	1.2	2
Testing Behaviour 4	1.8	4
Testing Behaviour 5	2.2	5
Testing Behaviour 6	1.2	2
Testing Behaviour 7	0.8	1
Testing Behaviour 8	1.4	3

IV. EXPERIMENTAL EVALUATIONS

As mentioned in the methodology section, we first conducted a subjective verification of the testing behaviour set. Next, three experiments were conducted to evaluate the potential of the approach to distinguish between the structured versus unstructured behaviours. These four experiments are discussed in the following subsections, to evaluate the accuracy of machine learning methods.

A. SUBJECTIVE VERIFICATION OF THE TESTING BEHAVIOURS

As mentioned in Section III-A, the set of parameters of the testing behaviours were derived from an unsupervised learning approach proposed by [15]. In order to ensure these behaviours are different collective behaviours from the original training set, we conducted a subjective verification. Five members of the research team who have knowledge on identifying collective behaviours participated in the verification process. The participant visually monitored each of these behaviours. Then, they were asked to vote for the similarity between the eight testing and the original eight training behaviours. For voting, the participants had to consider the most similar behaviour from the training set for each testing behaviour. Then a similarity rating in the range [0-4] was assigned by each participant such that the similarity is proportionate to the rating. The average rating for each testing behaviour was considered to analyse the similarity of the testing behaviour set to the training set.

Table 1 illustrates the average rating received by each testing behaviour and their rank according to how dissimilar they are to the original training set. These results were then used to investigate the impact of their similarity rank on the performance of the ML models.

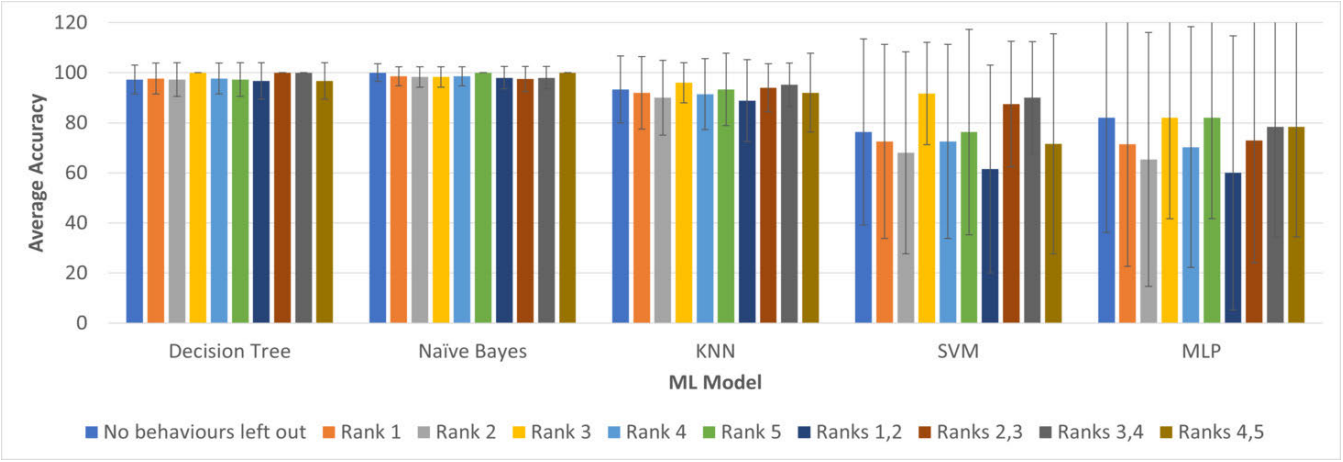


FIGURE 2. Accuracies of the machine learning models as a percentage in distinguishing predicting results, excluding testing behaviours with different rank combinations.

We considered leaving out different combinations of behaviours according to their similarity ranks (Table 1) and compared the results against the entire testing set. We used the five ML models: Decision Tree, Naïve Bayes, MLP, KNN, and SVM which were evaluated with 5-fold cross validation for this comparison. Fig. 2 demonstrates the average accuracy of each ML model tested on the testing behaviours after leaving out each combination.

According to the figure, there is no statistically significant impact on accuracy ($p < 0.05$) for any of the ML models when any combination is left-out when compared with the accuracy of the entire testing test. Since the accuracy does not change when more similar behaviours compared to the training set are removed, it can be concluded that no behaviour in the testing set is disproportionately similar to any behaviour in the training set. The rest of the experiments were therefore continued with the entire testing set.

B. PRIMARY EXPERIMENT

The first set of experimental evaluations were conducted to investigate the potential of the chosen ML models to distinguish between structured versus unstructured behaviours. As the first step, the five machine learning models: Decision Tree, Naïve Bayes, MLP, KNN, and SVM were evaluated with 5-fold cross validation as discussed under Section III. After performing cross validation, the ML models were then trained with the entire set of 800 behaviours and tested on a set of 400 new structured behaviours (structured-testing) that were not available for training and were verified with the voting mechanism presented in Section IV-A, and 1000 new unstructured behaviours which were also not available for training (unstructured-testing). Evaluations were conducted based on prediction accuracies observed with the five ML models and Kruskal-Wallis H test was used with a confidence level set at 95% for comparisons across models.

Fig. 3 illustrates the accuracies of the models averaged across the 5 folds, and the accuracy received when the models were tested on the structured-testing and unstructured-testing behaviours. The error bars in the 5-fold cross validation results depict the standard deviation across the five experiments. According to the results, the average cross validation results for all five models remain above 93%, with KNN having the highest prediction accuracy followed closely by the Decision Tree-based model (99% and 98% respectively). The best prediction accuracy when tested with structured-testing behaviours was observed with the Naïve Bayes model (99%) followed by KNN (92%) whereas the worst accuracy was observed with SVM (70%). When the models were tested for unstructured-testing behaviour set, the prediction accuracies were generally better across all models except with Decision Tree and Naïve Bayes compared to the results with structured-testing behaviours. KNN achieved a 100% accuracy whereas the worst accuracy was observed with Decision Tree which was at 83%. The overall results suggest that KNN was the best ML model when all three experiments are taken into consideration, whereas, SVM had the poorest prediction accuracy. However, the differences in the prediction accuracies were not statistically significant ($p = 0.327 \gg 0.05$). Therefore, it can be concluded that all five models are capable of distinguishing between the structured and unstructured behaviours using the identified collective behaviour metrics. They are capable of recognising a substantial set of new structured behaviours based on the characteristics learned from the original eight structured behaviours.

The above results suggest that despite the ML model used, the originally identified eight structured behaviours and the seven metrics are capable of capturing the unique attributes that distinguish between a significant pool of structured versus unstructured behaviours. However, a concrete conclusion cannot be made with regard to the redundancy of the eight structured behaviours or the seven metrics without

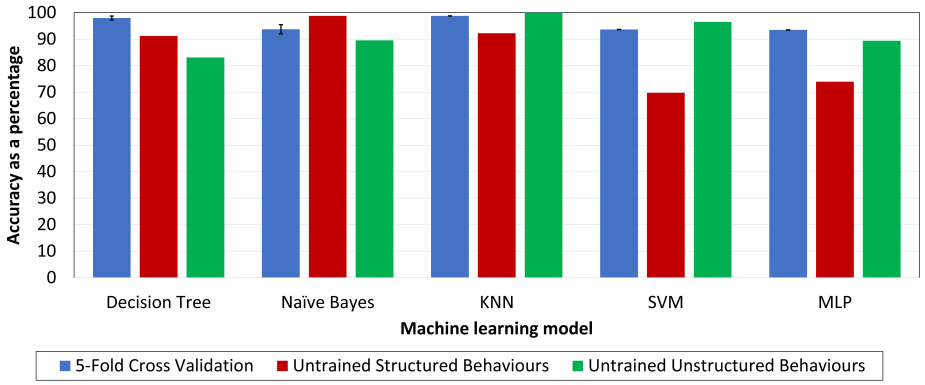
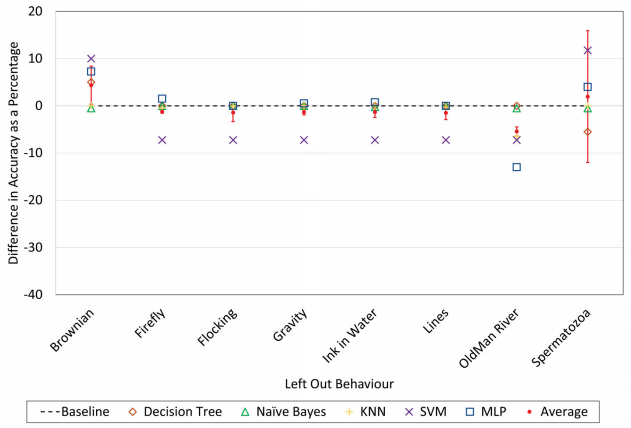
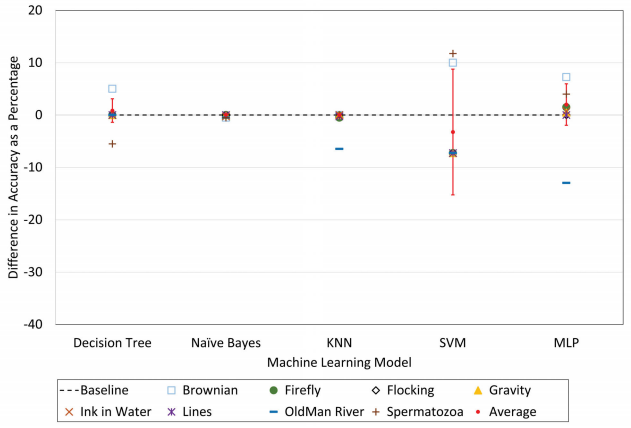


FIGURE 3. Accuracies of the machine learning models as a percentage in distinguishing structured vs unstructured behaviours. 5-Fold cross validation bar represents the average accuracy across the 5 folds tested with error bars depicting the standard deviation.

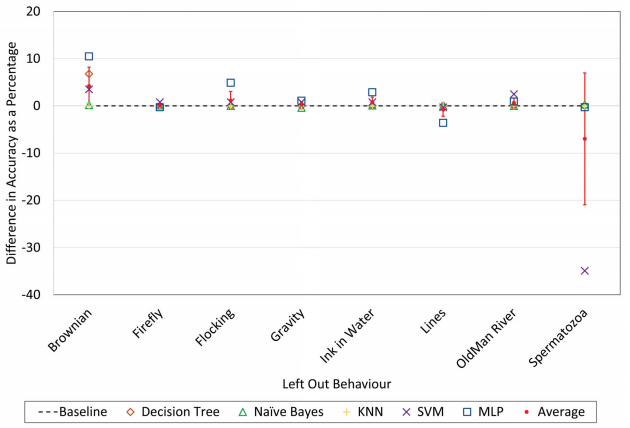


(a) Grouped by the behaviours left out.

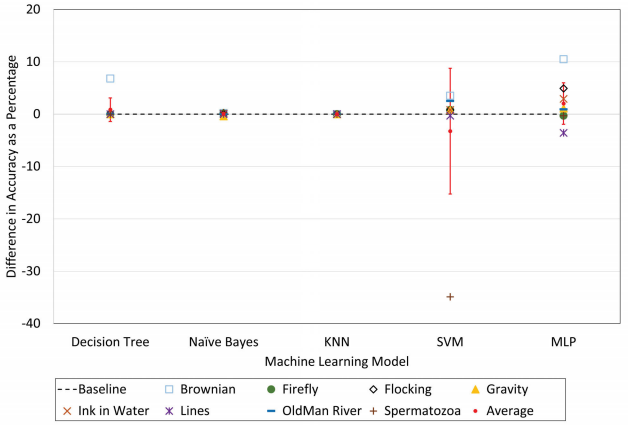


(b) Grouped by the machine learning models used.

FIGURE 4. Difference in accuracy for the leave-one-behaviour-out evaluations tested on structured-testing behaviours compared to the accuracy obtained by the original models trained with all eight behaviours. The averages across all ML models for each experiment are shown in red with the error bars representing the standard deviation across the multiple experiments.



(a) Grouped by the behaviours left out.



(b) Grouped by the machine learning models used.

FIGURE 5. Difference in accuracy for the leave-one-behaviour-out evaluations tested on unstructured-testing behaviours compared to the accuracy obtained by the original models trained with all eight behaviours. The averages across all ML models for each experiment are shown in red with the error bars representing the standard deviation across the multiple experiments.

an evaluation of their performance in a leave-one-out setting. The next two experiments were designed to further confirm the importance of each structured behaviour and each metric in recognising the collective characteristics of behaviours in autonomous virtual environments.

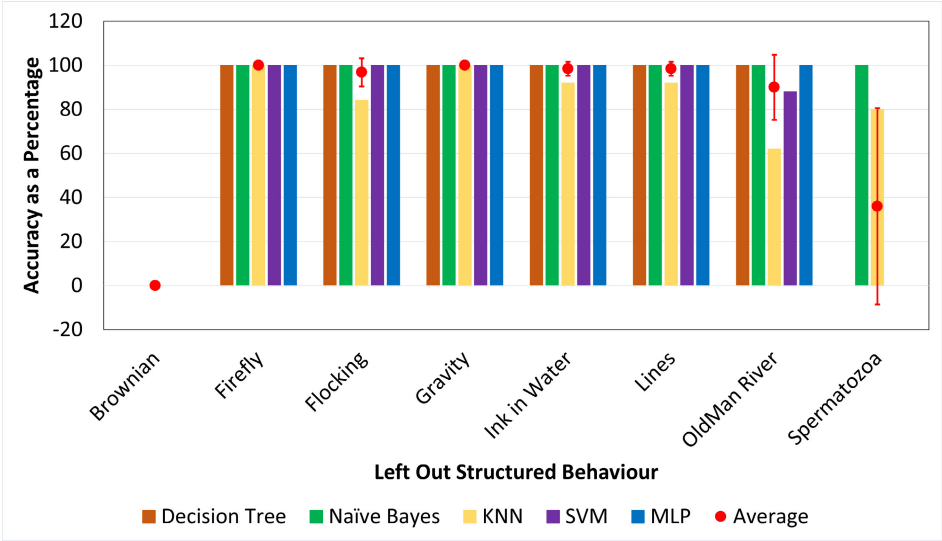


FIGURE 6. Accuracy of each machine learning model for the leave-one-behaviour-out evaluations tested on each left out behaviour. The averages across all machine learning models for each leave-one-out experiment are shown in red with error bars representing the standard deviation across the multiple experiments.

C. LEAVE-ONE-BEHAVIOUR-OUT EXPERIMENT

In order to understand the significance of the original eight structured behaviours that were used in the training of the above models, the second set of experiments focus on a leave-one-behaviour-out strategy. Figures 4, 5, and 6 illustrate the results when the five ML models trained with each behaviour left out are tested on the structured-testing behaviours, unstructured-testing behaviours, and the original behaviour that was left out, respectively. The difference in the accuracy obtained compared to the accuracy of the original model when tested with structured-testing and unstructured-testing behaviours are shown in Figures 4 and 4b.

Based on these figures, it can be noticed that the performance of most ML models increases when the Brownian behaviour is left out. Decision Tree (5%), MPL (7%), and SVM (10%) see an improvement in accuracy, whereas KNN remains at the same accuracy level and Naïve Bayes sees a reduction of 0.5% in accuracy. Similarly, accuracy of the models also increases on average when the Spermatozoa behaviour is left out of training, which is largely impacted by the significant increase in accuracy in the SVM model with an improvement of 12% in the accuracy. Leaving out other behaviours from the training set causes a slightly negative impact on accuracy of the model. This suggests that these behaviours capture certain unique attributes of the structured behaviours that are necessary to recognise such behaviours. When the models were tested on unstructured-testing behaviours, the performance of the model with leave-out-brownian behaviour appears to be similar with an average of 4% improvement in accuracy across the five ML models. Average performance difference of the leave-out-spermatozoa model is skewed towards a negative value due to the significant accuracy drop with the SVM model (35% reduction compared to the original model), whereas the other models remain at a negligible level of difference.

At a glance, this may suggest that the Brownian and Spermatozoa behaviours can be safely removed from the training set, given that their removal generally increases the chance of the machine learners in recognising previously unseen structured behaviours. The results presented in Fig. 6 illustrate the accuracy of the leave-one-out models when they are tested on the behaviours left out. Based on these results, it can be observed that none of the models can identify Brownian behaviour at all without having it seen during training. The average accuracy of the models also decreases when Spermatozoa is tested when it was left out of the training set (36%) with Decision Tree, MLP, and SVM not being able to recognise the behaviour at all. These observations prove that Brownian and Spermatozoa have unique characteristics that are not commonly observed in (at least) the other tested structured behaviours but are specific to these behaviours themselves. Further, they have some common attributes with unstructured behaviours such as lower cohesion which make it more difficult for the machine learners to distinguish them from unstructured behaviours.From a human subjective perspective, it can also be argued that it is closer to an unstructured behaviour rather than a structured behaviour, given the more chaotic formations observable. This explains the improvement in accuracy when the models trained without these behaviours are tested with structured-testing behaviours. The distinction between unstructured versus structured behaviours is made clearer for the models by removing these structured behaviours which have characteristics that lean towards unstructured behaviours.

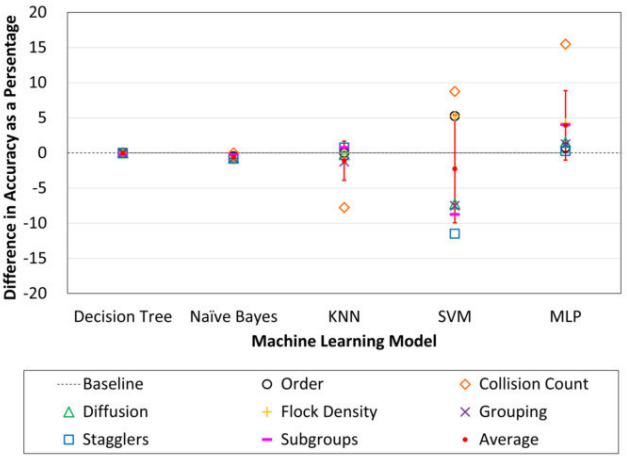
Further, the Figures 4b and 5b illustrate that SVM is the most impacted machine learner across all leave-one-behaviour-out experiments. Leaving out Brownian improves the accuracy in testing on both structured-testing and unstructured-testing (but not on the left out Brownian behaviour) behaviours. However, leaving all other behaviours have a significant impact on either one or both test cases. The impact on other machine learners is not as significant and generally lies within a negligible range on average across all leave-one-behaviour-out experiments.

In conclusion, the leave-one-behaviour-out experiment indicates that all eight original behaviours can be ascribed to a significant characteristic(s) of structured behaviours that are eminent in collective behaviours. Therefore, the respective attributes captured by these eight behaviours can be identified as essential in determining the nature of the behaviours at least within the scope of structured and unstructured behaviours tested in the context of this paper.

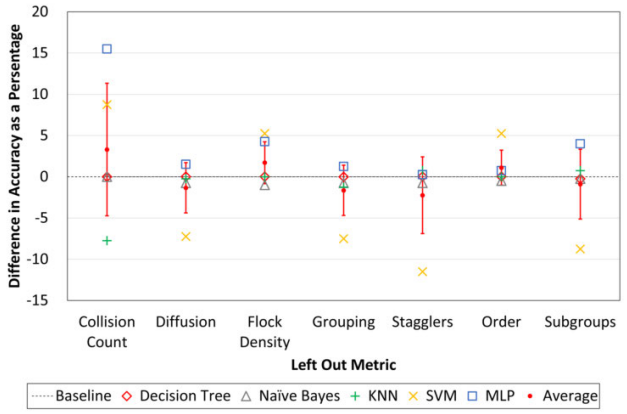
D. LEAVE-ONE-METRIC-OUT EXPERIMENT

Leave-one-metric-out approach was used as a measure of investigating the significance of each of the metrics identified as features for predicting structured behaviours. The significance of a metric can be measured by comparing the performance of the model trained with all the metrics with the performance of the model trained without the metric. Like the process of demonstrating the performance of the complete set of metrics, the performance of the leave-one-metric-out models was demonstrated by predicting new behaviours that are not involved in the training. The new behaviours include the two sets of behaviours, i.e., the structured-testing behaviours and the unstructured-testing behaviours.

For each of the ML models, when predicting structured-testing behaviours, the prediction accuracy after excluding each metric was compared with the accuracy achieved with the complete set of metrics (see in Fig. 7a). For Decision Tree, Naïve Bayes, and MLP, leaving either metric out does not result in a significant decrease in accuracy when predicting structured-testing behaviours compared to the original accuracy achieved with the complete set of metrics (91%, 98% and 74%, respectively). For KNN, only removing Collision Count results in a clear decrease by 8% compared to the original accuracy (92%). The prediction accuracy of SVM on unstructured-testing behaviours is impacted by a range of metrics, including Stragglers, Subgroups, Grouping, and Diffusion. Leaving either of these metrics out will result in a significant decrease in accuracy (12%, 9%, 8% and 7%, respectively) compared to the original accuracy (70%). Relatively speaking, SVM is more sensitive to the completeness of the seven metrics when predicting the unstructured-testing behaviours. The performance difference of each machine learning model after leaving each metric out is illustrated in Fig. 7b. Naïve Bayes performs better when predicting structured-testing behaviours even after leaving any metric out. KNN is the second best unless



(a) Grouped by the machine learning models used.

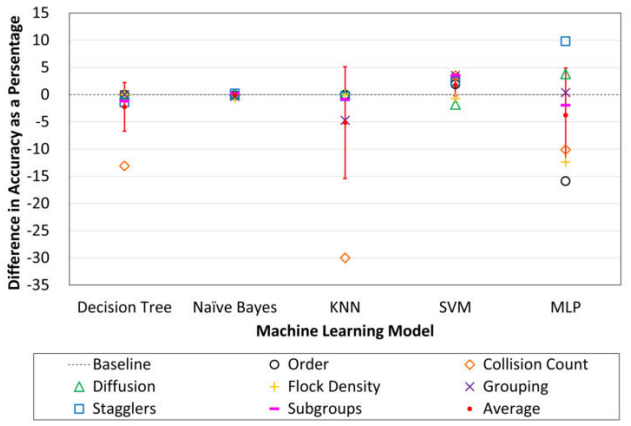


(b) Grouped by the metrics left out.

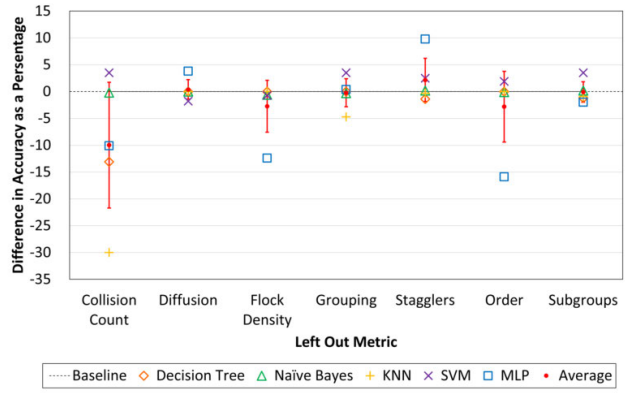
FIGURE 7. Difference in accuracy for the leave-one-metric-out evaluations tested on structured-testing behaviours compared to the accuracy obtained by the original models trained with all seven metrics. The averages across all ML models for each experiment are shown in red with the error bars representing the standard deviation across the multiple experiments.

Collision Count is not available. Although the accuracy of Decision Tree is slightly lower than KNN in the ideal cases, the unavailability of either metric does not impact its performance.

When predicting unstructured-testing behaviours, the prediction accuracy of each ML model after leaving each metric out is compared with the accuracy achieved with the complete set of metrics (see in Fig. 8a). For Naïve Bayes and SVM, leaving either metric out does not result in a decrease of accuracy when predicting on unstructured-testing behaviours compared to the original accuracy achieved with the complete set of metrics (90% and 100%, respectively). MLP is the most vulnerable to the exclusion of metrics. Compared to the original accuracy (89%), the prediction accuracy of MLP on unstructured-testing behaviours is decreased when either one of Order, Flock Density, and Collision count is left out (by 16%, 12% and 10%, respectively). Leaving Collision Count out results in a high impact on accuracy for KNN, Decision



(a) Grouped by the machine learning models used.



(b) Grouped by the metrics left out.

FIGURE 8. Difference in accuracy for the leave-one-metric-out evaluations tested on unstructured-testing behaviours compared to the accuracy obtained by the original models trained with all seven metrics. The averages across all ML models for each experiment are shown in red with the error bars representing the standard deviation across the multiple experiments.

Tree and MLP as their accuracies are decreased by 30%, 13% and 10%, respectively (as shown in Fig. 8b).

KNN can ensure a high accuracy (>95%) in predicting unstructured-testing behaviours using all seven metrics or six of them as long as Collision Count is available. Although the full potential of SVM (97%) is not as high as KNN (100%) in predicting unstructured-testing behaviours, the accuracy is barely compromised by leaving either metric out (at least 95%).

By considering both results of the structured-testing and unstructured-testing behaviours, every metric has shown an impact on at least one machine learning model in predicting either structured-testing or unstructured-testing behaviours. Among all the machine learning models, only Naïve Bayes is not impacted by excluding either one of the metrics in terms of accuracy in predicting the structured-testing and unstructured-testing behaviours. However, it is not the best performing machine learner out of the five ML models considering all tested behaviours. SVM and MLP are more sensitive to the completeness of the set of metrics. Based on the results, it can be concluded that all metrics are essential for accurate detection of behaviours.

TABLE 2. Basic parameters of the boid simulation environment.

Parameter	Value
World size	1400×1000 units with a wrap-around
Number of boids, N	200
Boid speed, v	1-20 units per tick

Summarising the performance of the ML models in the three experiments, although our primary experiments didn't show statistical differences among the ML models, certain models exhibited sensitivity when specific behaviors or metrics were excluded. SVMs showed sensitivity in leave-one-behavior-out and leave-one-metric-out experiments. This sensitivity could be attributed to SVMs struggling with imbalanced classes, impacting their ability to find a balanced decision boundary and leading to poorer performance on the minority class. MLPs were notably susceptible to metric exclusion due to their intricate architecture and numerous hyperparameters. The modest dataset size exacerbated this sensitivity, potentially hindering noise tolerance and generalization. Naive Bayes, despite assuming conditional independence among features given the class label, showcased resilience against noise and irrelevant features. Excluding specific metrics had minimal impact on Naive Bayes' accuracy, owing to this inherent property. KNN and Decision Trees similarly exhibited sensitivity to omitted metrics and behaviors. A small k-value for KNN supported smoother decision boundaries, bolstering robustness to individual data points. Decision Trees maintained stability despite minor data distribution changes, contributing to their resistance against metric exclusion in the leave-one-metric-out experiments.

V. CONCLUSION AND FUTURE WORK

Collective behaviours inspired by flocks of birds, schools of fish, and herds of land animals are widely applied to autonomous agents for enhancing efficiency, speed, and accuracy. Automatic recognition of these behaviours observed in simulated and real agent systems is very challenging as repeatedly observed in literature [38]. This paper proposes to use a set of collective behaviour metrics to capture the high-dimensional characteristics of structured and unstructured behaviours. These metrics are derived from simulated boids moving in a point-mass boid simulator. Then they are used in combination as the attributes to train five machine learning models: Decision Tree, Naïve Bayes, KNN, MLP, and SVM. The main contribution of this paper is to distinguish between structured versus unstructured behaviours through training ML models with the identified collective behaviour metrics. A dataset of new structured and unstructured behaviours are selected for testing the ML models, which were not used in the training set. A voting procedure is applied to analyse how different they are from the original training set based on human perspective. This procedure provides verification that the testing data

TABLE 3. Boid behaviour configurations [4].

Parameter	Description	Value/Range
W_s	Weight of separation rule	0-4
W_a	Weight for alignment rule	0-4
W_c	Weight for cohesion rule	0-4
V_{max}	Maximum speed: Maximum distance a boid will cover per tick	5-20
V_{min}	Minimum speed: Minimum distance a boid will cover per tick	2-10
R_s	Separation radius: Boids under this distance qualify for separation rule	2-62
R_c	Cohesion radius: Boids under this distance qualify for cohesion rule	5-1005
R_a	Alignment radius: Boids under this distance qualify for alignment rule	5-1005
θ_v	Vision angle: Agents within this angle relative to theagent's heading may be visible, if they are in range	17-360
P_{sa}	SA likelihood: Agents will update SA each tick with this probability	0-1
F_{sa}	SA frequency: Number of simulation ticks between agents updating situational awareness	1-10
$P_{fullscan}$	Full scan likelihood: The probability with which agents will look all around and do a 360-degree scan	0-1
F_{rule}	Rule frequency: Number of ticks between successive applications of rules to calculate change of velocity	1-10
P_{rule}	Rule likelihood: Probability agent will update velocity each tick	0-1
F_s	Separation frequency: Number of ticks between successive applications of the separation rule	1-10
P_s	Separation likelihood: Probability agent will apply separation rule each tick	0-1
F_a	Alignment frequency: Number of ticks between successive applications of the alignment rule	1-10
P_a	Alignment likelihood: Probability agent will apply alignment rule each tick	0-1
F_c	Cohesion frequency: Number of ticks between successive applications of the cohesion rule	1-10
P_c	Cohesion likelihood: Probability agent will apply cohesion rule each tick	0-1

TABLE 4. Parameter configurations of structured-testing behaviours.

Parameter	TS1	TS2	TS3	TS4	TS5	TS6	TS7	TS8
W_s	2.97	1.99	3.43	1.99	1.63	0.87	0.48	2.13
W_a	0.98	2.90	3.28	2.29	2.37	2.33	3.45	2.97
W_c	2.60	2.15	2.32	2.63	0.36	3.63	1.84	0.60
V_{max}	6.27	16.73	11.72	16.54	15.76	6.66	18.91	14.65
V_{min}	3.63	7.80	5.31	8.75	5.14	3.90	6.23	6.83
R_s	52.09	9.66	35.88	7.89	49.44	18.62	18.39	48.98
R_c, R_a	850.49	627.10	183.52	834.93	820.44	182.28	771.89	818.35
θ	1.48	1.48	5.38	2.34	1.61	5.27	2.18	2.11
P_{rull}	0.33	0.32	0.81	0.23	0.88	0.47	0.55	0.79
P_{sa}	0.95	0.36	0.59	0.82	0.65	0.53	0.23	0.15
$P_{fullscan}$	0.25	0.25	0.45	0.54	0.41	0.16	0.56	0.12
P_s	0.70	0.79	0.40	0.85	0.86	0.34	0.32	0.31
P_a	0.19	0.50	0.15	0.33	0.53	0.47	0.18	0.53
P_c	0.39	0.90	0.38	0.98	0.81	0.69	0.98	0.29

represents a significantly different set of samples of structured behaviours. This lends evidence to support that the proposed model can recognise a significant pool of structured collective behaviours without any pre-knowledge on them. Three experiments are conducted to investigate the aim of this contribution. In the first experiment, all the collective behaviour metrics and behaviours are made available for training. In the second and third experiments, the leave-one-behaviour-out, and leave-one-metric-out techniques, are investigated, respectively, to investigate any redundancies that exist within the metrics and behavioural attributes used in training. In conclusion:

- Five ML models were tested with the proposed approach of combining the eight behaviours characterised with the seven metrics in recognising structured and unstructured collective behaviours.
- All eight behaviours and seven metrics are impactful in training the ML models for collective behaviour recognition. Removal of each metric and behaviour from the training set demonstrated an impact on the performance of one or more ML models which signifies the importance of the proposed combination.
- Evaluation results show that training the ML models with the proposed approach enables accurate recognition of a significant set of diverse unseen structured and unstructured behaviours. Whilst an exhaustive exploration of all existing collective behaviours is not conducted (nor possible), this provides evidence that the proposed model is capable of characterising a large pool of collective behaviours to support automatic recognition of them.

The mentioned achievements also open avenues for future works as follows:

- In this paper, we only propose structured vs unstructured recognition. In future, means of recognising characteristics of structured behaviours rather than a binary distinction between structured and unstructured behaviours could be helpful. This will assist the automatic generation of specific behaviours that are applicable in real-world scenarios. Future research could be directed towards labelling unseen structured behaviours with terms meaningful to humans that capture why a human would declare such behaviours as structured. This has implications towards generating a framework for

TABLE 5. Parameter configurations of unstructured-testing behaviours.

Parameter	TU1	TU2	TU3	TU4	TU5	TU6	TU7	TU8	TU9	TU10
W_s	2.30	1.96	2.55	3.72	2.99	0.87	0.39	2.57	3.73	0.97
W_a	0.24	0.67	0.13	2.79	0.04	2.29	3.63	0.01	3.34	0.52
W_c	0.94	3.91	0.28	2.33	0.19	0.49	0.43	0.12	3.58	0.90
V_{max}	10.30	15.69	9.79	17.23	15.02	15.07	12.75	8.13	13.74	10.25
V_{min}	8.57	6.00	6.25	9.03	6.83	6.80	3.15	5.64	6.66	4.30
R_s	2.92	30.27	41.27	61.33	33.57	5.36	35.56	9.64	53.30	57.65
R_c, R_a	48.02	64.62	412.62	5.52	734.71	61.34	9.58	13.65	39.87	56.31
θ	1.31	4.39	5.22	5.49	4.54	1.22	4.90	4.66	5.61	3.86
P_{rule}	0.65	0.04	0.72	0.61	0.78	0.02	0.85	0.35	0.41	0.16
P_{sa}	0.73	0.07	0.97	0.99	0.29	0.44	0.92	0.78	0.04	0.84
$P_{fullscan}$	0.65	0.52	0.53	0.53	0.69	0.83	0.99	0.44	0.75	0.17
P_s	0.45	0.10	0.33	0.48	0.56	0.62	0.51	0.44	0.15	0.50
P_a	0.55	0.82	0.11	0.80	0.40	0.52	0.27	0.05	0.14	1.00
P_c	0.30	0.82	0.61	0.23	0.06	0.86	0.10	0.05	0.61	0.36

TABLE 6. Parameter configurations of unstructured-testing behaviours (continued).

Parameter	TU11	TU12	TU13	TU14	TU15	TU16	TU17	TU18	TU19	TU20
W_s	3.95	3.64	2.27	1.93	0.50	3.18	3.89	2.74	2.08	3.66
W_a	0.88	0.23	1.51	1.68	3.01	3.82	0.87	1.90	0.90	1.73
W_c	1.42	1.75	0.85	1.53	3.31	1.78	2.82	0.56	2.27	1.16
V_{max}	8.99	13.58	16.88	18.30	16.72	11.85	5.59	19.26	19.97	14.48
V_{min}	4.33	6.52	3.16	5.36	3.53	6.80	6.93	9.06	3.05	4.36
R_s	13.30	51.43	31.35	19.03	27.72	52.56	42.16	28.25	59.28	39.32
R_c, R_a	27.86	131.10	17.85	53.18	19.46	36.20	42.20	839.96	128.88	52.53
θ	3.00	2.10	1.42	1.61	2.25	1.42	0.32	2.25	1.42	6.27
P_{rull}	0.24	0.00	0.49	0.24	0.13	0.94	0.14	0.37	0.65	0.21
P_{sa}	0.87	0.95	0.84	0.03	0.45	0.95	0.86	0.79	0.13	0.61
$P_{fullscan}$	0.53	0.77	0.14	0.70	0.57	0.45	0.28	0.10	0.08	0.35
P_s	0.91	0.75	0.73	0.01	0.79	0.81	0.53	0.95	0.66	0.72
P_a	0.97	0.14	0.69	0.61	0.42	0.93	0.52	0.00	0.03	0.03
P_c	0.59	0.35	0.03	0.41	0.53	0.67	0.57	0.30	0.99	0.07

automatic determination of the characteristics of structured behaviours, rather than focusing on an exhaustive list of specific behaviours.

- We propose a combination of collective behaviour metrics to explore the structured behaviour space. However, we do not analyse the spatial parameter space that impacts the nature of the behaviours, which is still a black box. This work could be extended in future to explore this parameter space to understand where order meets chaos.
- This paper explores a significant set of collective behaviours only within the capabilities of the boids system used. We recognise that more unexplored structured behaviours exist in nature, of which the characteristics may not have been captured through the proposed model. Also, with the progress in applying collective behaviours in real-world problems including swarm robots, there is potential to expand this model towards such domains.

CONFLICT OF INTEREST

All authors declare that they have no conflicts of interest.

APPENDIX A
ENVIRONMENT AND BEHAVIOURS CONFIGURATIONS

Tables 2, 3, 4, 5 and 6 illustrate the parameter configurations of the point-mass simulation environment, the parameter value ranges of the behaviours used in the simulation (adapted from [4]) and the exact configurations used for the testing behaviours (which include both structured and unstructured test sets).

APPENDIX B
PARAMETERS OF MACHINE LEARNING MODELS

In the case of the Multi-Layer Perceptron (MLP), we utilised a solitary fully connected (FC) hidden layer and experimented with varying numbers of nodes within this hidden layer (8, 16, 32, 64, 128, and 256). We also explored different mini batch sizes (16, 32, 64, 128, and 256) and settled on 128 nodes for the hidden layer and a mini batch size of 128. Other hyperparameters included the use of the 'adam' solver and a maximum number of epochs set to 2000. For the K-Nearest Neighbours (KNN) algorithm, we explored k-values ranging from 1 to 10 and ultimately selected k=1 as the optimal choice. In the context of Support Vector Machines (SVM), we conducted tests employing various kernel functions including linear, radial basis function (RBF), polynomial, and sigmoid. Following evaluation, we opted for the linear kernel due to its ability to achieve the highest accuracy. Additionally, the Regularization Parameter C, which we fine-tuned using 10 values, was set to C=1 as it yielded the highest accuracy on the training dataset when coupled with a linear kernel.

ACKNOWLEDGMENT

(Shuo Yang, Dilini Samarasinghe, Anupama Arukgoda, and Shadi Abpeikar are co-first authors.)

REFERENCES

[1] A. Kolling, P. Walker, N. Chakraborty, K. Sycara, and M. Lewis, "Human interaction with robot swarms: A survey," *IEEE Trans. Human-Mach. Syst.*, vol. 46, no. 1, pp. 9–26, Feb. 2016.

[2] H. Duan, M. Huo, and Y. Fan, "From animal collective behaviors to swarm robotic cooperation," *Nat. Sci. Rev.*, vol. 10, no. 5, Apr. 2023, Art. no. nwad040.

[3] D. Samarasinghe, M. Barlow, E. Lakshika, and K. Kasmarik, "Grammar-based cooperative learning for evolving collective behaviours in multi-agent systems," *Swarm Evol. Comput.*, vol. 69, Mar. 2022, Art. no. 101017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210650221001796

[4] M. M. Khan, K. Kasmarik, and M. Barlow, "Autonomous detection of collective behaviours in swarms," *Swarm Evol. Comput.*, vol. 57, Sep. 2020, Art. no. 100715.

[5] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proc. SIGGRAPH*, Jul. 1987, pp. 25–34.

[6] J. H. Yang, M. Kapolka, and T. H. Chung, "Autonomy balancing in a manned-unmanned teaming (MUT) swarm attack," in *Robot Intelligence Technology and Applications 2012*. Berlin, Germany: Springer, 2013, pp. 561–569.

[7] E. A. Bolelov, B. V. Lezhankin, V. V. Erokhin, and S. A. Zyabkin, "Using a MLAT surveillance system to locate unmanned aerial vehicles flying as a swarm," in *Proc. TSCZh*, 2022, pp. 67–70.

[8] U. Dah-Achinanon, S. E. Marjani Bajestani, P.-Y. Lajoie, and G. Beltrame, "Search and rescue with sparsely connected swarms," *Auton. Robots*, vol. 47, pp. 1–15, Jan. 2023.

[9] K. M. Batoo, S. Pandiaraj, M. Muthuramamoorthy, E. H. Raslan, and S. Krishnamoorthy, "Behavior-based swarm model using fuzzy controller for route planning and e-waste collection," *Environ. Sci. Pollut. Res.*, vol. 29, no. 14, pp. 19940–19954, 2022.

[10] K. Kasmarik, S. Abpeikar, M. M. Khan, N. Khattab, M. Barlow, and M. Garratt, "Autonomous recognition of collective behaviour in robot swarms," in *Proc. Australas. Joint Conf. Artif. Intell. (AJCAI)*, M. Gallagher, N. Moustafa, and E. Lakshika, Eds. Cham, Switzerland: Springer, 2020, pp. 281–293.

[11] D. Samarasinghe, M. Barlow, E. Lakshika, and K. Kasmarik, "Task allocation in multi-agent systems with grammar-based evolution," in *Proc. Int. Conf. Intell. Virtual Agents (IVA)*. New York, NY, USA: Association for Computing Machinery, 2021, pp. 175–182, doi: 10.1145/3472306.3478337.

[12] J. Harvey, K. E. Merrick, and H. A. Abbass, "Assessing human judgment of computationally generated swarming behavior," *Frontiers Robot. AI*, vol. 5, p. 13, Feb. 2018.

[13] D. Samarasinghe, M. Barlow, E. Lakshika, and K. Kasmarik, "Grammar-based autonomous discovery of abstractions for evolution of complex multi-agent behaviours," *Swarm Evol. Comput.*, vol. 73, Aug. 2022, Art. no. 101106. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210650222000761

[14] K. Alharthi, Z. S. Abdallah, and S. Hauert, "Automatic extraction of understandable controllers from video observations of swarm behaviors," in *Proc. Int. Conf. Swarm Intell. (ICSI)*. Cham, Switzerland: Springer, 2022, pp. 41–53.

[15] S. Abpeikar, K. Kasmarik, P. V. Tran, M. Garratt, S. Anavatti, and M. M. Khan, "Tuning swarm behavior for environmental sensing tasks represented as coverage problems," in *Artificial Intelligence and Data Science in Environmental Sensing*. Amsterdam, The Netherlands: Elsevier, 2022, pp. 155–178.

[16] A. Hussein, L. Ghignone, T. Nguyen, N. Salimi, H. Nguyen, M. Wang, and H. A. Abbass, "Characterization of indicators for adaptive human-swarm teaming," *Frontiers Robot. AI*, vol. 9, Feb. 2022, Art. no. 745958.

[17] D. S. Brown and M. A. Goodrich, "Limited bandwidth recognition of collective behaviors in bio-inspired swarms," Air Force Research Laboratory/RISC, Rome, NY, USA, Tech. Rep. AFRL-RI-RS-TP-2015-001, 2014.

[18] J. Harvey, K. Merrick, and H. Abbass, "Quantifying swarming behaviour," in *Proc. Int. Conf. Swarm Intell. (ICSI)*. Cham, Switzerland: Springer, 2016, pp. 119–130.

[19] M. M. Khan, "Autonomous generation and recognition of collective behaviour in swarms," Ph.D. dissertation, School Eng. IT, UNSW, Sydney, NSW, Australia, 2020.

[20] C. Sammut and G. I. Webb, *Leave-One-Out Cross-Validation*. Boston, MA, USA: Springer, 2010, pp. 600–601, doi: 10.1007/978-0-387-30164-8_469.

[21] I. Navarro and F. Matía, "A survey of collective movement of mobile robots," *Int. J. Adv. Robotic Syst.*, vol. 10, no. 1, p. 73, Jan. 2013.

[22] R. O. Saber and R. M. Murray, "Consensus protocols for networks of dynamic agents," in *Proc. Amer. Control Conf.*, vol. 2, Jun. 2003, pp. 951–956.

[23] M. Gardner, "The fantastic combinations of John Conway's new solitaire game 'life,'" *Sci. Amer.*, vol. 223, pp. 120–123, Oct. 1970.

[24] M. Dorigo, G. Theraulaz, and V. Trianni, "Swarm robotics: Past, present, and future [point of view]," *Proc. IEEE*, vol. 109, no. 7, pp. 1152–1165, Jul. 2021. [Online]. Available: https://hal.archives-ouvertes.fr/hal-03362874

[25] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: A review from the swarm engineering perspective," *Swarm Intell.*, vol. 7, no. 1, pp. 1–41, Mar. 2013.

[26] L. Bayındır, "A review of swarm robotics tasks," *Neurocomputing*, vol. 172, pp. 292–321, Jan. 2016.

[27] S. Abpeikar, K. Kasmarik, M. Garratt, R. Hunjet, M. M. Khan, and H. Qiu, "Automatic collective motion tuning using actor-critic deep reinforcement learning," *Swarm Evol. Comput.*, vol. 72, Jul. 2022, Art. no. 101085.

[28] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, "Novel type of phase transition in a system of self-driven particles," *Phys. Rev. Lett.*, vol. 75, no. 6, pp. 1226–1229, Aug. 1995.

[29] K. Genter, S. Zhang, and P. Stone, "Determining placements of influencing agents in a flock," in *Proc. Int. Conf. Auton. Agents Multiagent Syst. (AAMAS)*, May 2015, pp. 247–255.

[30] E. Ferrante, A. E. Turgut, E. Duéñez-Guzmán, M. Dorigo, and T. Wenseleers, "Evolution of self-organized task specialization in robot swarms," *PLoS Comput. Biol.*, vol. 11, no. 8, Aug. 2015, Art. no. e1004273.

[31] M. Barlow and E. Lakshika, "What cost teamwork: Quantifying situational awareness and computational requirements in a proto-team via multi-objective evolution," in *Proc. Congr. Evol. Comput. (CEC)*, Jul. 2016, pp. 3525–3532.

[32] C. Szabo and L. Birdsey, "Toward the automated detection of emergent behavior," *Emergent Behavior in Complex Systems Engineering: A Modeling and Simulation Approach*. Hoboken, NJ, USA: Wiley, 2018, pp. 228–261.

[33] L. Birdsey, C. Szabo, and K. Falkner, "Identifying self-organization and adaptability in complex adaptive systems," in *Proc. IEEE 11th Int. Conf. Self-Adapt. Self-Organizing Syst. (SASO)*, Sep. 2017, pp. 131–140.

[34] J. K. Parrish, S. V. Viscido, and D. Grünbaum, "Self-organized fish schools: An examination of emergent properties," *Biol. Bull.*, vol. 202, no. 3, pp. 296–305, Jun. 2002.

[35] J. Tang, G. Leu, and H. A. Abbass, "Networking the boids is more robust against adversarial learning," *IEEE Trans. Netw. Sci. Eng.*, vol. 5, no. 2, pp. 141–155, Apr. 2018.

[36] I. Navarro and F. Matía, "A proposal of a set of metrics for collective movement of robots," in *Proc. Workshop Good Exp. Methodol. Robot.*, 2009, pp. 1–6.

[37] M. D. Manning, C. E. Harriott, S. T. Hayes, J. A. Adams, and A. E. Seiffert, "Heuristic evaluation of swarm metrics' effectiveness," in *Proc. 10th Annu. ACM/IEEE Int. Conf. Human-Robot Interact. Extended Abstr. (HRI)*, 2015, pp. 17–18.

[38] X. Wang, S. Liu, Y. Yu, S. Yue, Y. Liu, F. Zhang, and Y. Lin, "Modeling collective motion for fish schooling via multi-agent reinforcement learning," *Ecol. Model.*, vol. 477, Mar. 2023, Art. no. 110259. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S030438002200357X

SHUO YANG received the Ph.D. degree in computer science from the University of New South Wales, Australia, in 2022. He is currently a Postdoctoral Research Associate with the School of Engineering and Information Technology, University of New South Wales. His research interests include trustworthy AI, multi-agent systems, machine learning, human–machine interaction, and serious games.

DILINI SAMARASINGHE received the Ph.D. degree in computer science from the University of New South Wales, Australia, in 2021. She is currently a Postdoctoral Research Associate with the School of Engineering and Information Technology, University of New South Wales. Her research interests include artificial intelligence, serious games, autonomous agent systems, and machine learning.

ANUPAMA ARUKGODA received the B.Sc. degree (Hons.) in computer science from the School of Computing, University of Colombo, Sri Lanka. She is currently pursuing the Ph.D. degree in multi-agent systems with the University of New South Wales, Canberra, Australia. With her current work, she strives to simulate human psychology and sociology theories on artificial agents, and thereby create socially intelligent, explainable agents that can work alongside humans. Her research interests include machine learning, data science, and multi-agent systems.

SHADI ABPEIKAR (Member, IEEE) received the B.S. degree in applied mathematics from the Iran University of Science and Technology, Tehran, Iran, in 2011, and the M.S. and Ph.D. degrees in computer science from the Amirkabir University of Technology, Tehran, in 2013 and 2018, respectively. From 2019 to 2022, she was a Research Associate with the University of New South Wales, Canberra, Australia, where she is currently a Senior Research Associate. Her research interests include machine learning, swarm robotics, swarm intelligence algorithms, big data, feature selection, decision support systems, neural networks, neural trees, and intelligent transportation systems.

ERANDI LAKSHIKA (Senior Member, IEEE) received the B.Sc. degree (Hons.) in computer science from the University of Colombo, Sri Lanka, and the Ph.D. degree in computer science from the University of New South Wales, Canberra (UNSW Canberra), in 2014. In 2009, she joined the School of Computing, University of Colombo, as an Assistant Lecturer. She is currently a Senior Lecturer with UNSW Canberra. Her research interests include human–computer interfaces, multi-agent systems, computational intelligence, multi-objective optimization, serious games, and games for health.

MICHAEL BARLOW received the Ph.D. degree in computer science from the University of New South Wales, Australia, in 1991. Then, he joined The University of Queensland, Australia, as a Postdoctoral Researcher and thereafter Nippon Telegraph and Telephone's Human Communication Laboratories, Japan. In 1996, he joined the University of New South Wales, where he is currently a Professor. His research interests include simulation, virtual environments, machine learning, serious games, and human–computer interaction.