Estimating Dynamics of Honeybee Population Densities with Machine Learning Algorithms

✉ Ziad Salem[1] and Gerald Radspieler[1] and Karlo Griparić[2] and Thomas Schmickl[1]

[1] Artificial Life Lab at the Institute for Zoology
Karl-Franzens-University Graz
Universitätsplatz 2, A-8010 Graz, Austria
{ziad.salem,gerald.radspieler,thomas.schmickl}@uni-graz.at
[2] LARICS Lab at the Faculty of Electrical Engineering and Computing
University of Zagreb
Unska 3, 10000 Zagreb, Croatia
karlo.griparic@fer.hr

Abstract. The estimation of the density of a population of behaviourally diverse agents based on limited sensor data is a challenging task. We employed different machine learning algorithms and assessed their suitability for solving the task of finding the approximate number of honeybees in a circular arena based on data from an autonomous stationary robot's short range proximity sensors that can only detect a small proportion of a group of bees at any given time. We investigate the application of different machine learning algorithms to classify datasets of pre-processed, highly variable sensor data. We present a new method for the estimation of the density of bees in an arena based on a set of rules generated by the algorithms and demonstrate that the algorithm can classify the density with good accuracy. This enabled us to create a robot society that is able to develop communication channels (heat, vibration and airflow stimuli) to an animal society (honeybees) on its own.

Keywords: machine learning, data mining, classification algorithms, density estimation, robots, honeybees

1 Introduction

The availability of a precise estimation of the population density is an important prerequisite for the establishment of a mixed society of interacting bees and robots, which is the aim of the ASSISI$|_{bf}$ (*Animal and robot Societies Self-organise and Integrate by Social Interaction (bees and fish)*) project. The main concept of the project is to generate a mixed society of honeybees and autonomous robots, which aims to establishing a robotic society that is able to develop communication channels to animal societies (bees and fish swarms) on its own [20]. The robots will adapt by evolutionary algorithms until they have learned to interact with animals in a desired way. Honeybees are an established and widely used model organism in the field of collective behaviour and swarm intelligence due to their social nature [23]. The project's long-term objective is to integrate bio-inspired robots and biological agents to form an interactive mixed society. The experiments focus on limited interaction between stationary robots and bees in an experimental arena. These robots are implemented as *CASU*s (Combined Actuator Sensor Units) that communicate with the bees via their actuators (heat, vibration and airflow) [21]. The bees communicate with the CASUs via their mere presence (detected by the CASUs' six proximity sensors). The task of estimating the population density of a group of agents, based on information gathered by sensors with severely limited surveillance range, poses a great challenge for the development of automated realtime solutions. This is especially true for agents such as honeybees with highly variable and unpredictable behaviour. Supervised machine learning methods are the most promising candidates for a reliable and time efficient solution for density estimation. The new agent density estimation approach poses a new challenge when selecting a good learning algorithm and its parametrization. Therefore, this aspect had high priority in our work. To our knowledge, agents density estimation based on machine learning has not been applied before to solve such a problem.

1.1 State of the Art

Researchers use machine learning classification algorithms to generate decsion trees or rulesets as descriptors of datasets, which represent the problem to be solved [18]. The algorithm is trained by the training dataset where the generated rules are tested with a test dataset. The algorithms either split the dataset into separate sets of test and training data, or use cross-validation for generating the classifiers. The produced classifiers can be evaluated by different measures such as accuracy, robustness, speed and scalability [11]. For this work, we used accuracy, calculated as the number of correct predictions per total number of predictions made while using bees as agents. In prior works, Salem and Schmickl [19] used *bristle-bots*, simple micro robots propelled by vibrating the slanted bristles they rest on [7], as substitutes for groups of bees. In this case, an algorithm was used to learn how to derive the number of bristle bots in a circular arena from the sensor activities of a CASU at the center of the arena. The resulting rules were induced by an algorithm trained with datasets collected during the experiments. The work showed that the set of rules was able to predict the number of bristle-bots with satisfying accuracy. While this study was valuable for the development of the project, there are important differences between bristle-bots and bees that required an extended study with bees. For instance, the bees' locomotion patterns are modulated by the environment and dependent on communication between individuals. The work reported in this paper aims to enable the CASUs to determine the number of bees in an arena with a good accuracy by employing different algorithms. By determining the best suited algorithm and its parametrization we extend the scope of machine learning applications to the field of bio-hybrid societies, where they will be implemented on different layers of control and evaluation software. In additional related work an artificial neural network based on LSTM architecture was designed and trained for bee density estimation [16].

1.2 Research Questions

The work presented in this paper was devoted to answering the following research questions:

1. Does the number and type of attributes correlate with the classification accuracy?
2. Does the number of classes correlate with the accuracy and number of generated rules?
3. Does the number of examples affect the accuracy and number of generated rules?
4. Does the number of generated rules correlate with the accuracy?
5. Is there a specific experimental setup that allows any or all of the algorithms to perform exceptionally well?
6. Are there differences between the three tested algorithms regarding the achieved accuracy and number of generated rules?
7. Is there an algorithm that performs consistently better or worse than the others in all experimental setups?

2 Material and Methods

The data is generated by the sensors mounted on the hexagonal top part of a CASU. The CASU is located at the center of a circular arena ($d = 12.5cm$), which is equipped with a wax floor and a plastic wall. The infra-red proximity sensors are triggered by objects at a distance of up to $1.5cm$. The values of each sensor are logged at a rate of $10s^{-1}$, thus producing data with six features. For each animal experiment, single or groups of young (up to one day old) European honeybees (*Apis mellifera* sp. [10]) were released into the arena and left to walk freely for a specified time. The experiments were conducted in an infrared lit environment which is essentially dark for the bees. The number of bees in the arena was varied in different steps between experiments. The resulting log files were processed to extract various attributes relevant for the learning process, which we combined to constitute the actual datasets. We conducted several series of experiments on these datasets with different selections of attributes and evaluated the accuracy achieved with each combination along with the corresponding Kappa coefficient [4]. The aim of the learning process was to assign to each group size one of several group size classes, which were also subject to change between iterations of the series of experiments. In order to prevent the adverse effects of an excessive number of classes on the accuracy, we grouped the population sizes into larger and fewer bins for some experiments and assessed the impact of this measure on the accuracy by repeating the experiment four times with different numbers of classes. In this work we used different algorithms from the *Weka package*[3] (*Waikato Environment for Knowledge Analysis*) [22]. We focussed on

[3] Available for download at http://www.cs.waikato.ac.nz/ml/weka/

three different methods which process combined training and test datasets that store pre-existing knowledge about the classification (the output, in our case the group size) inline with the attributes and their values used for the training (the input) [12].

2.1 Hardware Description

The hardware system that delivers the data to learn from is a custom stationary robot (CASU) developed to facilitate interactions with nearby bees (*Bee-CASU system*) [8]. The CASU is mounted beneath the arena, with its hexagonal top part ($d = 1.5cm$) above the aluminium ring protruding through the arena floor into the arena. This part hosts six lateral proximity sensors (VCNL4010) with an I2C communication interface to detect nearby honeybees. The sensors are fully integrated and implement an independent distance measurement procedure that resorts to a built-in infra-red emitter and a photo diode to detect reflected infrared light. The sensors are able to detect bees at a distance of up to $1.5cm$ and thus do not allow to directly determine the total number of honeybees in the arena. During the experiments, the values reported by the proximity sensors and other relevant status information of the CASU were logged by the control software at a rate of $10s^{-1}$. One or more bees are detected by a sensor whenever its value reaches or exceeds its threshold value. The sensor specific threshold is assumed at 3% above the minimum value encountered by the sensor during the entire experiment. The short detection range of the sensors is the primary reason why we had to develop a method to (spatially and temporally) integrate the sparse information retrieved from the individual sensors in order to get an estimation of the total number of honeybees in the arena.

2.2 Learning Algorithms

All learning experiments were performed using three algorithms, one of them based on decision trees (J48) and two based on classification rules (JRip and PART).

J48 Decision Tree J48 is a Java implementation of the C4.5 decision tree algorithm [17]. J48 is an extension of ID3 algorithm and is often referred to as a statistical classifier.

JRip Rules Classifier JRip is a fast and efficient RIPPER algorithm [2]. Classes are examined in increasing size and an initial set of rules for each class is generated using incremental reduced error pruning [5]. JRip proceeds by treating all the examples of a particular judgement in the training data as a class, and finding a set of rules that cover all the members of that class.

PART Algorithm PART is a partial decision tree algorithm, which is the developed version of C4.5 and RIPPER algorithms [15]. However, decision trees are sometimes more problematic due to the larger size of the tree which could be oversized and might perform badly for classification problems [3].

2.3 Dataset Description

In order to create the datasets for or our learning experiments, we pre-processed the data logs for the sensors including six integer values that reflect the status of each proximity sensor, and extracted a number of attributes that integrate the information of one or several sensors. Depending on the experiment series, we averaged the raw data over a period of $3s$ (30 values) or $1min$ (600 values) and employed different sub-sets of the defined attributes[4].

2.4 Learning Experiments

The classical supervised learning problem is to construct a classifier that can correctly predict the classes of new objects given training examples of old objects [14]. If the classifier classifies most cases in the test examples correctly, we can assume that it works accurately also on the future data. If the classifier makes too many errors (misclassifications) in the test examples, we can assume that it was a wrong model. A better model can be searched after modifying the data, changing the settings of the learning algorithm, or by using another classification method [9]. In order to identify the best combination of algorithm and dataset setup, we conducted several series experiments to test the variability of rules generated using different setups. This approach resulted in five main experiment series (s1 - s5) with different population sizes and different population size granularity. s1 was complemented with four additional sub-series (s1.1 - s1.4). The main reason to iterate over different experiment series was to determine the response to altering the number of bees, definition of group sizes, attributes or experiment duration on the learning algorithm for the aim of concluding to the best setup. For this purpose, we processed the averaged or summed attributes with each of the algorithms to test. The dataset based on sums was tested in its original version (examples sorted by population size) and in a randomly shuffled version (examples in random order). In **s1**, we used a very simple dataset setup with 14 attributes and 14 classes, where every population size of bees is considered as a class. We derived the attributes from both averaged (real) and summed (integer) value in order to test for a possible impact of the data type on the performance of the algorithm, which proved to be marginal. In s1.1 - s1.4 we used different grouping schemes with different population numbers in every sub-series while sticking with the same attributes. In **s2**, we introduced new attributes based on the original sensor readings in addition to the attributes used earlier. We also added a new class for *no bees*. In this case the algorithm can learn from a mixture of real and logical values with more information, which can be beneficial

[4] A sample dataset is available at https://doi.org/10.5281/zenodo.824923

for generating better results. In this and all subsequent series, we averaged the sensor values over 600 sensor log records, which correspond to $1min$ of recordings. This method reduces the number of records in the dataset and also reduces the impact of the large variance in the instantaneous data. In **s3**, we introduced normalized versions of several attributes (sum and standard deviation of sensor values) in order to better compensate for differences between sensors, to reduce the data range and to complement the logical and integer values with real values, thus creating a more diverse dataset, which is expected to have a positive impact on the algorithm's search space. In **s4**, the duration of the bee experiments was shortened to $10min$. In **s5** we modified the setup of the experiments to decrease the population size granularity. We cycled through all population sizes from 1 to 32 bees and repeated each experiment twice. For details on experiment setups in the different series of learning experiments, consult tab. 1.

Table 1: Setups of experiment series. The number of attributes, examples and classes and the duration (τ) and number of repetitions (n_{runs}) of the bee experiments are shown for the different series. Each example consists of the indicated number of attributes and resolves to the respective number of classes.

Series	$n_{examples}$	$n_{attributes}$	$n_{classes}$	$\tau[min]$	n_{runs}
1	1008	14	14	30	2
1.1	1008	14	2	30	2
1.2	1008	14	3	30	2
1.3	1008	14	4	30	2
1.4	1008	14	4	30	2
2	1080	25	5	30	2
3	1080	25	5	30	2
4	360	25	5	10	2
5	320	25	4	10	1

3 Results

The J48 and PART algorithms achieved a similar accuracy (differences between 3-6%) while JRip typically performed slightly lower (see fig. 1). For all algorithms, the lowest accuracy (47.4% and 46.7% for J48 and PART) was achieved in the s1 experiments (no categorization) and the highest (97.8% and 98.3% for J48 and PART) in s1.1 experiments (categorization into the lowest number of classes). The Kappa statistics demonstrate a sufficient (s1) to excellent (s2, s3) agreement between predicted and actual classes (see tab. 3). The accuracy achieved while learning from the different types of dataset (based on averages, sums and shuffled sums), which were provided in the s1 experiments, was almost identical for the three dataset types (see fig. 1). While the number of rules generated by JRip slightly changed depending on the dataset (25, 22 and 24 rules for averages, sums and shuffled sums), the PART algorithm generated the same number of rules (90) with all datasets (see tab. 2). During the average based learning of s1 experiments, missing values (zeros) occurred in 68% of all examples for 4 of the 14 attributes. During the s2 experiments, missing values occurred in 10% of all examples for 3 out of 25 attributes. While the states were evenly distributed in the s1 datasets (72 examples per state), the introduction of classes resulted in an imbalanced distribution of the dataset examples over the different states. For example, of the five classes in the datasets used for the s2 and s3 experiments, two were covered by 72, two by 360 and one by 216 examples. The results of categorization into different numbers of nominal classes (see fig. 2a) corroborate the negative correlation between number of classes and achieved accuracy for all three algorithms (compare s1 experiments of fig. 1). However, for all tested numbers of classes the achieved accuracy was higher than for learning from uncategorized (continuous) states. Over all numbers of classes and with uncategorized states, J48 and PART achieved a higher accuracy than JRip. There is a direct correlation between the number of classes and the number of rules generated by the JRip and PART algorithms (see fig. 2b and tab. 2). This correlation is approximately linear in the investigated range of 2-15 classes. PART produced more rules than JRip over the entire range. In contrast to the number of classes, the number of attributes does not have an influence on the number of rules generated by JRip or PART (see tab. 2). It follows from the correlation between number of classes and number of generated rules and the negative correlation between number of classes and accuracy, that there is also a negative correlation between the number of generated rules and the achieved accuracy (see fig. 3).

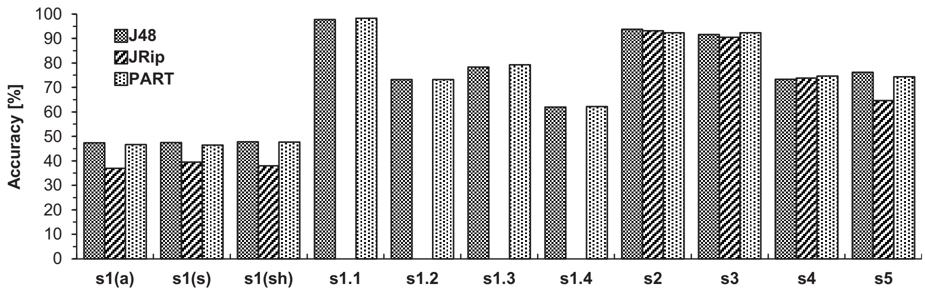
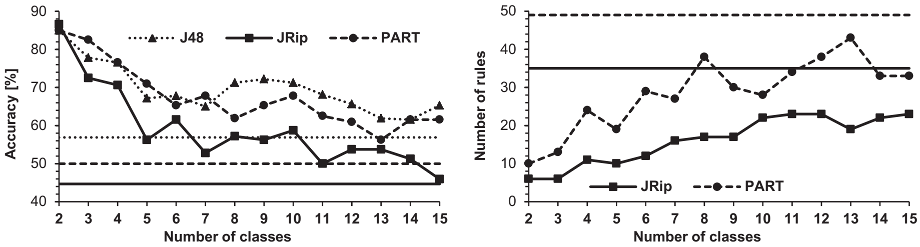


Fig. 1: Overview over the accuracy achieved in the different learning experiment series. The setups of different series differ in data reduction method and number of states (classes). The dataset for s1(a) is based on averages, for s1(s) on sums and for s1(sh) on shuffled sums. For s5, (evenly) redistributed classes were used. Consult tab. 1 for numbers of classes and attributes used in each series. The JRip algorithm was not used in experiment series s1.1 - s1.4.



(a) Accuracy (b) Number of rules

Fig. 2: The dependence of accuracy and number of generated rules on the number of classes. The learning experiments were conducted for different numbers of distinguished classes (lines with markers; group sizes of 1-32 honeybees categorized into 2-15 nominal classes) and for uncategorized group sizes (markerless lines). The learning experiments were conducted with the setup of the s5 experiments. (a) The accuracy achieved by all three algorithms for different numbers of classes. (b) The number of rules generated by the JRip and PART algorithms for different numbers of classes

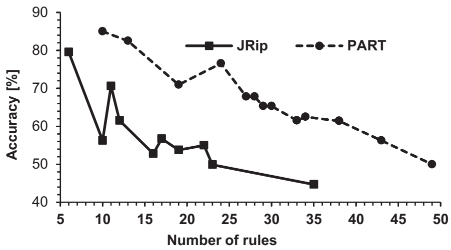


Fig. 3: The accuracy achieved by the JRip and PART algorithms for different numbers of generated rules

4 Discussion

In this paper, the primary benchmark for the quality of the algorithms is the accuracy of the predictions made by a model trained on the test dataset. The lowest accuracy was achieved in the s1 experiments, which were based on datasets with uncategorized states. This finding is explicable by the scattering of the dataset over 14 possible states (3 to 42 bees in steps of 3), which has a negative effect on the ability of the searching mechanism to find the best rules to represent the whole dataset. A considerable improvement was achieved by categorizing the states into classes (i.e. by grouping populations; compare experiment series s1.1 - s1.4), which reduces the amount of scattering. In accordance with the negative relationship between number of classes and accuracy, the highest accuracy was achieved in the s1.1 experiments, in which only two classes were distinguished. The method of calculating the attribute values (summation or averaging) and the

Table 2: The number of rules generated by the JRip and PART algorithm in all learning experiment series. For s1, the values are given for attributes constructed of averages (a), sums (s) and shuffled sums (sh).

Algorithm	s1(a)	s1(s)	s1(sh)	s1.1	s1.2	s1.3	s1.4	s2	s3	s4	s5
JRip	25	22	24	-	-	-	-	16	18	8	9
PART	90	90	90	9	8	30	46	20	25	26	15

Table 3: Kappa statistics. The kappa coefficients calculated during the learning process of the three algorithms are shown for the major experiment series.

Algorithm	s1(a)	s1(s)	s1(sh)	s2	s3	s4	s5
PART	0.4263	0.4241	0.437	0.8957	0.8945	0.6693	0.738
Jrip	0.3216	0.3494	0.3323	0.9057	0.8701	0.6595	0.6847
J48	0.4338	0.4348	0.438	0.9148	0.8859	0.6517	0.6728

randomization of the order of examples did not have any impact on the accuracy. Some of the 14 attributes used in the s1 and s1.1-s1.4 experiments assumed zero-values in many dataset examples. Frequent occurrences of such missing values can affect the quality of the classifiers trained on these datasets [1] as they tend to inflate the ruleset. Due to the interdependencies between number of classes, number of rules and accuracy, it is not possible to imply from these experiments a direct impact of zero values on the accuracy. The new attributes and data reduction methods introduced with s2 had a dampening effect on the number of zero values encountered in the datasets, as both the number of occurrences and the number of affected attributes decreased. On the other hand, the introduction of classes with s1.1 brought along the problem of uneven class distribution, which can seriously affect the performance of the learning algorithms both during the training and the evaluation phase [6]. For example, in the s1.1 experiments, one of the two classes comprised populations of 3 bees while the other comprised populations of 6-42 bees, thus causing an extremely imbalanced class coverage. Despite this factor, the learning from categorized states, which was employed in s1.1 and higher, proved to be beneficial to the accuracy. Therefore, the effect of uneven class distribution appears to be outbalanced by the negative toll taken by the large number of states to differentiate in datasets without categorization. In experiment series s2-s5, additional features (attributes) were introduced to the dataset, bringing their number from 14 to 25 and thus providing the algorithms with more information to learn from (see tab. 1). Additionally, some of the newly introduced attributes directly reflect low level sensor activities rather than the values derived from logical combinations of sensor values. Both of these changes had a beneficial effect on the performance of the algorithms. The setup of the s2 experiments proved to be especially beneficial to the accuracy of the learning algorithms. In this series with categorization into 5 classes, the J48 and PART algorithms performed better than in the s1.2, s1.3 and s1.4 experiments that only differentiated 3, 4 and 4 classes, respectively, and the JRip algorithm performed on level with J48 and PART. However, J48 and PART didn't achieve the same accuracy as in s1.1 experiments, in which only two classes were differentiated. Given the persistence of other parameters, this improvement can only be ascribed to the increase in number and qualitative changes of the attributes and the resulting improvement of the algorithms' database. The changes introduced in s3-s5 experiments did not improve the performance of the algorithms compared to s2. While they performed comparably well in s3 experiments, the setup changes in s4 and s5 had a detrimental effect on the classification accuracy of all three algorithms. Since the number of classes is the same (5) in s4 as in s2 and s3 and even lower (4) in s5 and the attributes are identical as well, this effect appears to be due to the lower number of examples in the respective datasets (360) compared to those used for s2 and s3 (1080). In this case, the shorter duration of the biological experiments was reflected by smaller datasets, which in turn provided less information for the algorithms to learn from. The comparison of the accuracy achieved by the three algorithms shows that JRip typically has a lower performance than the other two algorithms, which perform approximately equal. This is especially obvious when considering the accuracy in correlation with the number of distinct classes. However, in experiment series s2-s4, all three algorithms have a comparable performance. The number and quality of generated rules are an important factor that determines the classification accuracy both during the learning process and upon application of the ruleset to a classification problem, where a lower number of rules is beneficial for the applicability of the classifier. In this regard, JRip clearly outperforms PART as it manages to achieve a comparable accuracy with fewer rules in all tested setups. Generally it can be said that the setups of the experiments played a much greater role for the accuracy than the selected algorithm.

5 Conclusions

Our approach of detecting the approximate number of bees in an arena by employing machine learning algorithms to evaluate highly variable data from a limited detection system has shown promising results. The algorithms applied to the problem were able to classify the bees population in the arena with good accuracy and a reasonable number of generated rules (where applicable). It is obvious that the choice of the algorithm and a proper configuration of dataset and algorithm is crucial to the quality of the classification results. However, finding the optimal parameters for all applied algorithms and datasets requires a large amount of resources [13] as each algorithm has to be tuned individually for each dataset. Although there are automatic methods for setting parameters, we adhered to the default parametrization for all machine learning algorithms employed in our experiments, so that some potential remains to further improve the accuracy of our classification results. The comparison of the different algorithms employed in our experiments showed that PART is the best suited algorithm for our specific problem as it achieves the highest classification accuracy with the smallest rulesets. However, all three algorithms achieved a comparable accuracy when processing datasets configured according to the setup of the s2 experiments. This setup proved to be the best suited among the different setups we tested due to the high accuracy it allowed the algorithms to achieve while differentiating a reasonable number of classes. Due to this compromise between accuracy and a useful number of classes, this setup will continue to be used in further experiments. Therefore, the research questions on which the work presented in this paper was focused, can be answered as follows:

1. The number of attributes and the procedure of deriving their values from the raw sensor data have an influence on the classification accuracy. However, the method of calculating their values (averages, sums or shuffled sums) does not influence the accuracy.
2. The classification accuracy correlates negatively and the number of generated rules positively with the number of classes.
3. A low number of examples can have a negative effect on the classification accuracy if the learning algorithm is trained on too little data to achieve its potential.
4. The number of generated rules is interdependent with the achieved accuracy, but it is not clear from our results whether this is a real or spurious correlation.
5. All three algorithms showed their best performance when trained on datasets configured according to the s2 and s3 setup. However, there is potential for further improvements.
6. J48 and PART perform similarly in terms of classification accuracy while JRip falls behind. PART also generates fewer rules than JRip.
7. Due to its prevalence over the competing algorithms in terms of accuracy and ruleset size, PART shows the best overall performance during all experiments. Judged by the same criteria, JRip underperformed consistently.

Therefore, our approach to determine the approximate number of bees in an arena using machine learning algorithms was successful. The proportion of correct classifications was excellent, so that the method can be implemented in future experimentation, where it will for example provide stationary robots with the information required to take control over a swarm of bees or other animals. At the same time, the flexibility of the method and its implementation will allow for easy adaptation to different scenarios or environments. This wide applicability gives the method relevance for various fields of research.

6 Future Work

The density detection mechanisms developed in this work will be implemented in future experiments conducted in the framework of the $ASSISI_{bf}$ project. The pre-generated rulesets will be used by the CASU control program to estimate the number of bees in the arena. The control program will be able to integrate information from several CASUs and thus classify the group sizes with higher accuracy. We will complement this top-down approach by the implementation of a learning algorithm directly in each CASUs' control logic. This will enable the CASUs to independently estimate the number of bees in the arena and to adapt to new scenarios by dynamically modifying the classification rulesets. The methods developed in our work could be used to monitor bees in their hive in order to continuously assess the colony size or to detect abnormal behaviour. This could provide beekeepers with a valuable new tool to survey the health of their colonies.

Acknowledgments This study was supported by the EU FP7 FET-Proactive project $ASSISI_{bf}$, grant no. 601074.

References

1. Acuña, E., Rodriguez, C.: The treatment of missing values and its effect on classifier accuracy. In: Classification, clustering, and data mining applications, pp. 639–647. Springer (2004)
2. Cohen, W.W.: Fast effective rule induction. In: Proceedings of the twelfth international conference on machine learning. pp. 115–123. Morgan Kaufmann (1995)
3. Duin, R.P.: A note on comparing classifiers. Pattern Recognition Letters 17(5), 529–536 (1996)
4. Foody, G.M.: Thematic map comparison: Evaluating the statistical significance of differences in classification accuracy. Photogrammetric Engineering and Remote Sensing 70(5), 627–633 (2004)
5. Fürnkranz, J., Widmer, G.: Incremental reduced error pruning. In: Proceedings of the 11th International Conference on Machine Learning (ML-94). pp. 70–77 (1994)
6. Ganganwar, V.: An overview of classification algorithms for imbalanced datasets. International Journal of Emerging Technology and Advanced Engineering 2(4), 42–47 (2012)
7. Giomi, L., Hawley-Weld, N., Mahadevan, L.: Swarming, swirling and stasis in sequestered bristle-bots. In: Proceedings of the Royal Society a Mathematical Physical and Engineering. vol. 469, p. 20120637. The Royal Society (2013)
8. Griparić, K., Haus, T., Miklić, D., Bogdan, S.: Combined actuator sensor unit for interaction with honeybees. In: Sensors Applications Symposium (SAS), 2015 IEEE. pp. 1–5. IEEE (2015)
9. Hämäläinen, W., Vinni, M.: Classifiers for educational data mining. Handbook of educational data mining pp. 57–74 (2010)
10. Heinrich, B.: The hot-blooded insects: strategies and mechanisms of thermoregulation. Springer Science & Business Media (2013)
11. Jordan, M., Mitchell, T.: Machine learning: Trends, perspectives, and prospects. Science 349(6245), 255–260 (2015)
12. Kotsiantis, S.B.: Supervised machine learning: A review of classification techniques. informatica - an International Journal for Computing and Informatics 31(3), 249–268 (2007)
13. Kotthoff, L., Gent, I.P., Miguel, I.: A preliminary evaluation of machine learning in algorithm selection for search problems. In: Fourth Annual Symposium on Combinatorial Search (2011)

14. Mitchell, T.M.: Machine Learning. McGraw-Hill, Inc., New York, NY, USA, 1 edn. (1997)
15. Mohamed, W.N.H.W., Salleh, M.N.M., Omar, A.H.: A comparative study of reduced error pruning method in decision tree algorithms. In: IEEE International Conference on Control System, Computing and Engineering (ICCSCE). pp. 392–397. IEEE (2012)
16. Polić, M., Salem, Z., Griparić, K., Bogdan, S., Schmickl, T.: Estimation of moving agents density in 2d space based on lstm neural network. In: IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS). pp. 1–8. IEEE (2017)
17. Quinlan, J.R.: C4.5: Programs for Machine Learning. elsevier (2014)
18. Salem, Z.: Enhanced computer algorithms for machine learning. Ph.D. thesis, Intelligent system Research laboratory, Cardiff University, Wales, UK (2002)
19. Salem, Z., Schmickl, T.: The efficiency of the rules-4 classification learning algorithm in predicting the density of agents. Cogent Engineering 1(1), 986262 (2014)
20. Schmickl, T., Bogdan, S., Correia, L., Kernbach, S., Mondada, F., Bodi, M., Gribovskiy, A., Hahshold, S., Miklić, D., Szopek, M., Thenius, R., Halloy, J.: Assisi: Mixing animals with robots in a hybrid society. In: Biomimetic and Biohybrid Systems Lecture Notes in Computer Science 8064, pp. 441–443. Springer (2013)
21. Schmickl, T., Szopek, M., Bodi, M., Hahshold, S., Radspieler, G., Thenius, R., Bogdan, S., Miklić, D., Griparić, K., Haus, T., Kernbach, S., Kernbach, O.: Assisi: Charged hot bees shakin'in the spotlight. In: 2013 IEEE 7th International Conference on Self-Adaptive and Self-Organizing Systems. pp. 259–260. IEEE (2013)
22. Witten, I.H., Frank, E., Hall, M.A., Pal, C.J.: Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, 4 edn. (2016)
23. Zahadat, P., Hahshold, S., Thenius, R., Crailsheim, K., Schmickl, T.: From honeybees to robots and back: Division of labor based on partitioning social inhibition. Bioinspiration & Biomimetics 10(6), 066005 (2015)