International Partnership on Innovation

SAMS - Smart Apiculture Management Services

Deliverable N°4.2

Report on Data Analysis and Interpretation

Work package 4 Decision Support System

Horizon 2020 (H2020-ICT-39-2017)

Project N°780755

This project has received funding from the European Union´s Horizon 2020 research and innovation programme under **grant agreement N° 780755.** The sole responsibility for the content of this document lies with the authors. It does not necessarily reflect the opinion of the EU.

Project information		
Lead partner for the deliverable	Latvia University of Life Sciences and Technologies, Faculty of Information Technologies, Department of Computer Systems. Lead researcher and Project lead: Asoc.prof. Dr. Aleksejs Zacepins	
Document type	Report	
Dissemination level	Public	
Due date and status of the deliverable	*31.10.2019*	*31.10.2019*
Author(s)	Dr. Aleksejs Zacepins, Dr. Vitalijs Komasilovs, Dr. Armands Kviesis, Olvija Komasilova	
Reviewer(s)	Magdalena Sperl (GIZ), Stefanie Schädlich (GIZ), Sascha Fielder (UNIKAS), Fakhri Rido Muhammad (CV.PI)	

This document is issued by the consortium formed for the implementation of the SAMS project under Grant Agreement N° 780755.

SAMS consortium partners

Logo	Partner name	Short	Country
giz Deutsche Gesellschaft für Internationale Zusammenarbeit (GIZ) GmbH	Deutsche Gesellschaft für internationale Zusammen-arbeit (GIZ) GmBH (Coordinator)	GIZ	Germany
UNIKASSEL VERSITÄT	University of Kassel	UNIKAS	Germany
KARL-FRANZENS-UNIVERSITÄT GRAZ UNIVERSITY OF GRAZ	University of Graz (Institute for Biology)	UNIGRA	Austria
Latvia University of Life Sciences and Technologies	Latvia University of Life Sciences and Technologies	UNILV	Latvia
iceaddis	ICEADDIS – IT-Consultancy PLC	ICEADDIS	Ethiopia
IQQO Oromia Agricultural Research Institute	Oromia Agricultural Research Institute, Holeta Bee Research Center	HOLETA	Ethiopia

Universitas Padjadjaran	University Padjadjaran	UNPAD	Indonesia
PRIMARY TRAINING & CONSULTING	Commanditaire Vennootschap (CV.) Primary Indonesia	CV.PI	Indonesia

List of Abbreviations

ANN	Artificial Neural Network
DW	Data Warehouse
FCL	Fuzzy Control Language
FIS	Fuzzy Inference System
FLC	Fuzzy Logic Controller
HIVE	HIVE measurement system (WP3)
ID3	Iterative Dichotomiser 3
ITAPIC	ERA-Net ICT-Agri project “Application of Information Technologies in Precision Apiculture”
SGD	Stochastic gradient descent
Lasso	Least Absolute Shrinkage and Selection Operator

Summary of the project

SAMS is a service offer for beekeepers that allows active monitoring and remote sensing of bee colonies by an appropriate and adapted ICT solution. This system supports the beekeeper in ensuring bee health and bee productivity, since bees play a key role in the preservation of our ecosystem, the global fight against hunger and in ensuring our existence. The high potentials to foster sustainable development in different sectors of the partner regions are they are often used inefficient.

Three continents - three scenarios

(1) In Europe, consumption and trading of honey products are increasing whereas the production is stagnating. Beside honey production, pollination services are less developed. Nevertheless, within the EU 35% of human food consumption depend directly or indirectly on pollination activities.

(2) In Ethiopia, beekeepers have a limited access to modern beehive equipment and bee management systems. Due to these constraints, the apicultural sector is far behind his potential.

(3) The apiculture sector in Indonesia is developing slowly and beekeeping is not a priority in the governmental program. These aspects lead to a low beekeeper rate, a low rate of professional processing of bee products, support and marketing and a lack of professional interconnection with bee products processing companies.

Based on the User Centered Design the core activities of SAMS include the development of marketable SAMS Business Services, the adaption of a hive monitoring system for local needs and usability as well as the adaption of a Decision Support System (DSS) based on an open source system. As a key factor of success SAMS uses a multi stakeholder approach on an international and national level to foster the involvement and active participation of beekeepers and all relevant stakeholders along the whole value chain of bees.

The aim of SAMS is to:

- enhance international cooperation of ICT and sustainable agriculture between EU and developing countries in pursuit of the EU commitment to the UN Sustainable Development Goal (SDG N°2) “End hunger, achieve food security and improved nutrition and promote sustainable agriculture”
- increases production of bee products
- creates jobs (particularly youths/ women)
- triggers investments and establishes knowledge exchange through networks..

Project objectives

The overall objective of SAMS is to strengthen international cooperation of the EU with developing countries in ICT, concentrating on the field of sustainable agriculture as a vehicle for rural areas. The SAMS Project aims to develop and refine an open source remote sensing technology and user interaction interface to support small-hold beekeepers in managing and monitoring the health and productivity in their own bee colonies. Highlighted will be especially the production of bee products and the strengthening of resilience to environmental factors.

- Specific objectives to achieve the aim:
- Addressing requirements of communities and stakeholder
- Adapted monitoring and support technology
- Bee related partnership and cooperation
- International and interregional knowledge and technology transfer
- Training and behavioral response
- Implementation SAMS Business cooperation

Contents

List of tables / figures

1. Background

The Latvia University of Life Sciences and Technologies developed the Data Warehouse (DW) system within the SAMS project. The DW is an important part of the project as it acts as a data storage and analysis unit for all beehive data collected by the SAMS HIVE measurement systems developed by the University of Kassel (UNIKAS). All collected data is sent from the HIVE systems via Wi-Fi/ Internet to the data warehouse system for storage and further processing. After data is stored in the DW it is possible to apply different data analysis algorithms, methods and models to detect abnormalities in the bee colony behavior or identify different states of the colony, including death of the colony, swarming, etc.

Scope of the Deliverable

This report describes the development process of the algorithms and models, which can be used for bee colony state detection. Some algorithms are applied to data, received from SAMS HIVE measurement system. As well basic data analysis is demonstrated based on SAMS data. Some models need real-time bee colony data, while some can operate with historical data obtained and stored. All presented models are part of the global Decision Support System, which will be used to inform the beekeeper about the state and situation within the bee colonies.

2. Report on collected data

The following chapter summarizes the amount and quality of the data, collected by the SAMS HIVE measurement systems installed in Ethiopia and Indonesia until 01.11.2019. All systems are equipped with sensors for temperature, humidity, weight and sound monitoring.

In Ethiopia, three apiaries are equipped with the SAMS HIVE system: Bako (three systems), Gedo (five systems) and Holeta (five systems). Due to challenges in maintaining stable connectivity and a lack of IT expertise near the installation sites of Holeta and Gedo for applying a quick solution/ fixing the problems only three systems deployed in Bako are active at the moment. In Indonesia, five measurement systems are installed in different geographical locations (two in Ciwidey, one in Tanikota, two in Ciburial).

Until today, the data collection procedure was not working properly (due to unstable internet connection and problems in audio data recordings that in some cases caused the whole system to fault. That is why the proposed and developed algorithms presented in this report are based on historically collected bee colony data from the ERA-NET ICT-Agri ITAPIC project as well as on temperature data, which was gathered by UNILV for scientific purposes from 2010.

The EU funded (within ICT-AGRI 2012 FP 7) ERA-NET project "Application of Information Technologies in Precision Apiculture" (ITApic) was implemented between 2013 and 2016. The project was mainly focused on adapting precision agriculture methods and principles in beekeeping by implementing existing and newest technologies in the field of information and communication technologies in order to identify different honeybee colony states. The project objectives included the development of a beehive monitoring system and a web service system for data access (with measurement storage). As a result, the beehive monitoring systems (wireless and wired) were developed together with a web-based data system and integrated decision support. The main measured parameters included temperature, sound and video recordings to detect bee activity at the entrance of the hive. Temperature monitoring was one of the main target parameters (due to thermoregulation processes inside the hive) that was measured in several hives located in Strazdu iela 1, Jelgava, Latvia. Colonies were placed in Norwegian-type hive bodies (with outer size of 47x47x27cm and 38x38x27cm inner size). Measurements were taken during the period of 2014 and 2016 and temperature changes inside the hive were recorded during various seasons (winter, spring, summer, autumn). The open source technologies developed during the ITAPIC project are the fundamental blocks and forms for the technological background of the SAMS project.

One of the most important parameters that can give insight into what is happening in the beehive is the temperature. Honeybees perform thermoregulation inside the hive, e.g. during the active brood rearing period it is necessary to keep a very steady temperature, so the new bees can develop properly. Therefore, changes in temperature can indicate abnormalities in quite an early stage.

Previously collected honeybee data (during ITAPIC project) contains temperature dynamics forming patterns specific for different bee colony states. A detailed inspection of these temperature changes and patterns lead to several definitions of rules. It is also important to note, that algorithms developed during ITAPIC project were taken into account and improved/ adjusted for the SAMS project and peculiarities of target countries – season differences, weather conditions – therefore analysis and state detection methods were chosen to modify them in a convenient way, e.g. application of fuzzy logic instead of static conditional statements to uncomplicated the coding. So we can apply them to SAMS specifics, for example, not using, where the coding would be complicated for each country (temperature levels during seasons) but rather using fuzzy logic, were we can define such things in rules. This is described in detail under section 7.

3. Rules for bee colony state detection

Proposed bee colony state detection rules are divided into two groups:

- individual bee colony rules, when the analysis is based on a single colony data;
- differential rules, when the analysis is based on comparison between several colonies.

Within the SAMS project, it is evaluated that detection of several bee colony states are essential for the end users (beekeepers):

- active foraging state (start of the nectar flow);
- broodless;
- absconding;
- colony death;
- swarming.

In addition to mentioned states, it is important to emphasize the identification of unknown states as well - to detect abnormalities.

3.1 Active foraging state (start of the nectar flow)

For beekeepers, it is important to know when the nectar flow has started. Such information indicates that beekeepers won´t need to prepare extra suppers for the hives. Theoretically, during the active foraging a distinctive pattern in weight data should be noticeable, but more data is needed to derive an algorithm.

It should be noted that weight was not monitored during the ITAPIC project, hence historical data about weight change were not available.

3.2 Broodless state

If a colony is in a broodless state, it is an indicator that this colony needs special attention. Probably it has a non-laying queen that needs to be replaced, or does not have a queen at all. Detection of this state could be completely opposite to detection of brood rearing. During brood rearing bees try to maintain stable temperature (34-36°C), but in broodless state temperature inside the hive tends to depend on ambient temperature[1].

3.3 Absconding state

Absconding is a state when all adult bees leave the hive. Causes for such behavior includes lack of food, mites, poor microclimate inside the hive (heat, moisture) etc. Absconding is still not researched enough. Therefore, more information is needed to properly identify it so that applicable algorithms (or modifications to proposed solutions) could be designed. Theoretically, by the assumptions of the SAMS project, it should be determined by temperature and weight data. After absconding there are no “living beings” that could perform thermoregulation inside the hive, therefore a noticeable weight reduction should also be observed (the unknown still remains – what is the pattern of such reduction?).

3.4 Bee colony death detection

For the beekeeper it is very important to detect bee colony death on time to be prepared for future loss of honey production, to get a new colony instead of a dead one and basically to effectively manage his apiary. Death of the colony can be detected by the temperature measurement or/ and by sound. Proposed model for the death detection is to compare real-time colony temperature with the environmental temperature and if the temperature difference is not significant, then it can be concluded that the colony is dead. Such a situation can be observed in the following figure:

[1] Egils Stalidzans and Almars Berzonis, “Temperature Changes above the Upper Hive Body Reveal the Annual Development Periods of Honey Bee Colonies,” *Computers and Electronics in Agriculture* 90 (2013): 1–6.

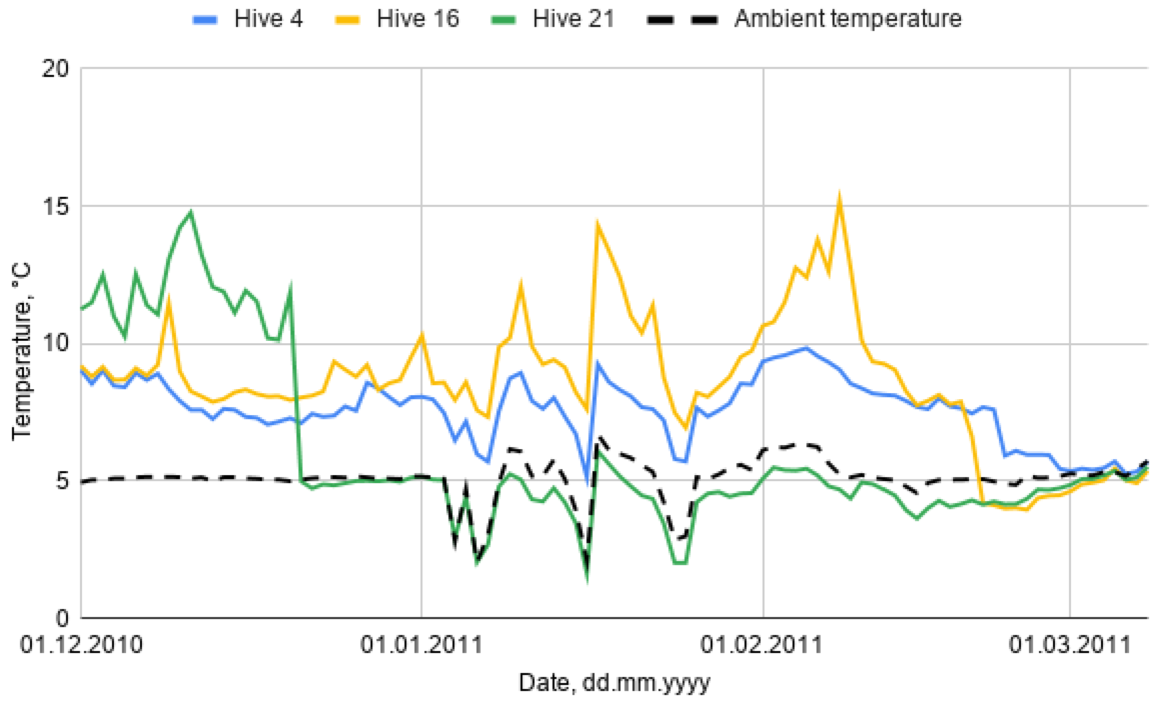

Figure 1. Examples of colony death identified by temperature data

Figure 1 depicts a situation where honeybee colonies were kept in closed bee wintering building with controlled microclimate. And in some moments, it can be seen, that bee colony temperature decreased till it reached the value of environmental temperature, thus afterwards both temperatures followed the same pattern.

At this point it is not possible to predict the death of the colony, but only detect that the colony has started to decline or has died. But as discussed previously this also gives valuable information for the beekeeper.

3.5 Brood rearing state detection

Brood rearing detection can help the beekeeper to evaluate the strength of the colony. In some cases, a too early start of the brood rearing can be unworthy, in other cases, the beekeeper definitely has to inspect the colony if the colony did not start this process. If the colony starts the brood rearing during the winter, it might be alarming, because food resources may not be sufficient.

From previous experience and literature it is known that during the active brood rearing period, honeybees keep their brood temperature at 34-36°C. In some periods temperature increase in the hive is more observable, in some periods – not so much, but as a result temperature should be more than 30°C.

Below (Figure 2) some examples of observed bee colony temperature increase is demonstrated.

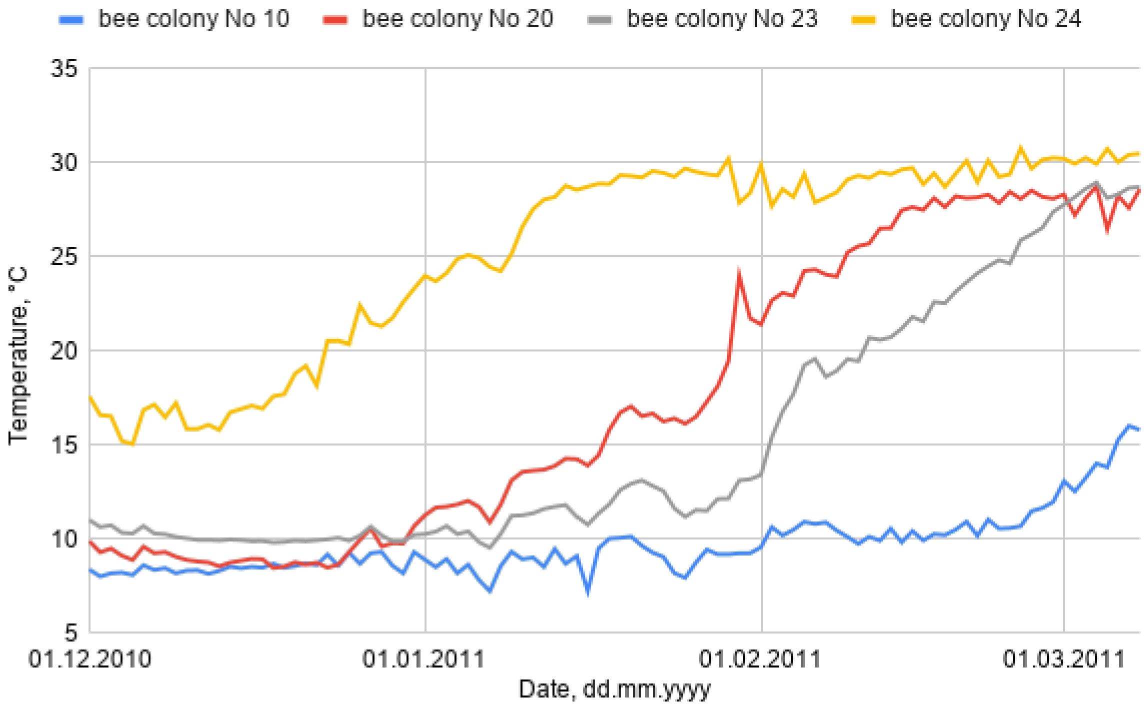

Figure 2. Examples of the start of brood rearing process

3.6 Swarming state detection

Swarming is a natural honeybee reproduction process, when a large amount of bees leave the hive to form a new colony (or in some cases more than one). The main disadvantage of natural swarming for commercial beekeeping is its spontaneous character, as in some years there could be more swarms than in others. The detection of such state is important, because the original colony is weakened, this leads to the decrease of honey production from this colony. The mother colony becomes weaker, experiences a gap in emerging adult honey bees and is not able to collect nectar at previous strength and speed. In some cases weak colonies can even die afterwards – so beekeepers will lose a part of their honey related income, which could potentially be collected by a non-swarming colony.

If a colony has swarmed, beekeepers still have some time from few hours to more than a day to catch the swarm and place it back into the hive or a new box. By catching the swarm, beekeepers can minimize the financial losses caused by unwanted bee colony swarming.

Since bees prepare their flight muscles (by heating them) before leaving the hive, this can be observed in hive temperature data. Figure 3 depicts some of such examples.

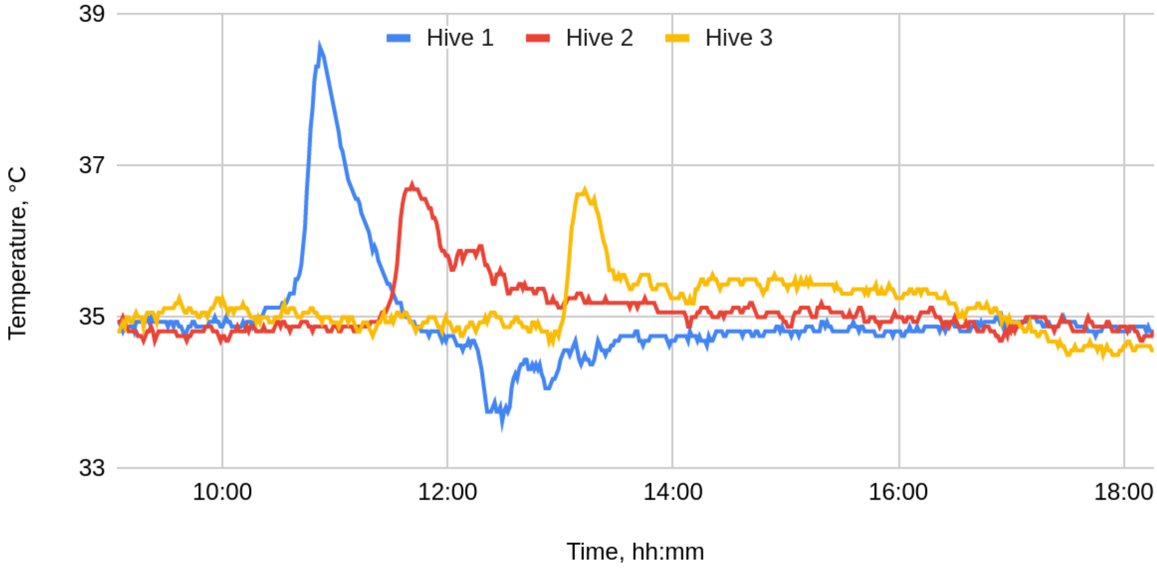

Figure 3. Temperature changes during swarming

Historical data that was used as basis for data analysis consists of temperature data recorded every minute (during the ITAPIC project). Such high measurement frequency was set to better understand annual temperature dynamics inside a beehive. This allowed to detect temperature patterns during swarming events.

Importance of measurement frequency is closely linked with the nature of temperature pattern during swarming. It was found that usually the preparation for swarming (hence the increase in temperature) only takes 8-20min.

It was also concluded that, with less frequent measurements, there is a possibility that such patterns could not be detected, due to lack of data points. As an example, below are presented several charts (Figure 4) that show swarming pattern when measuring with different time intervals.

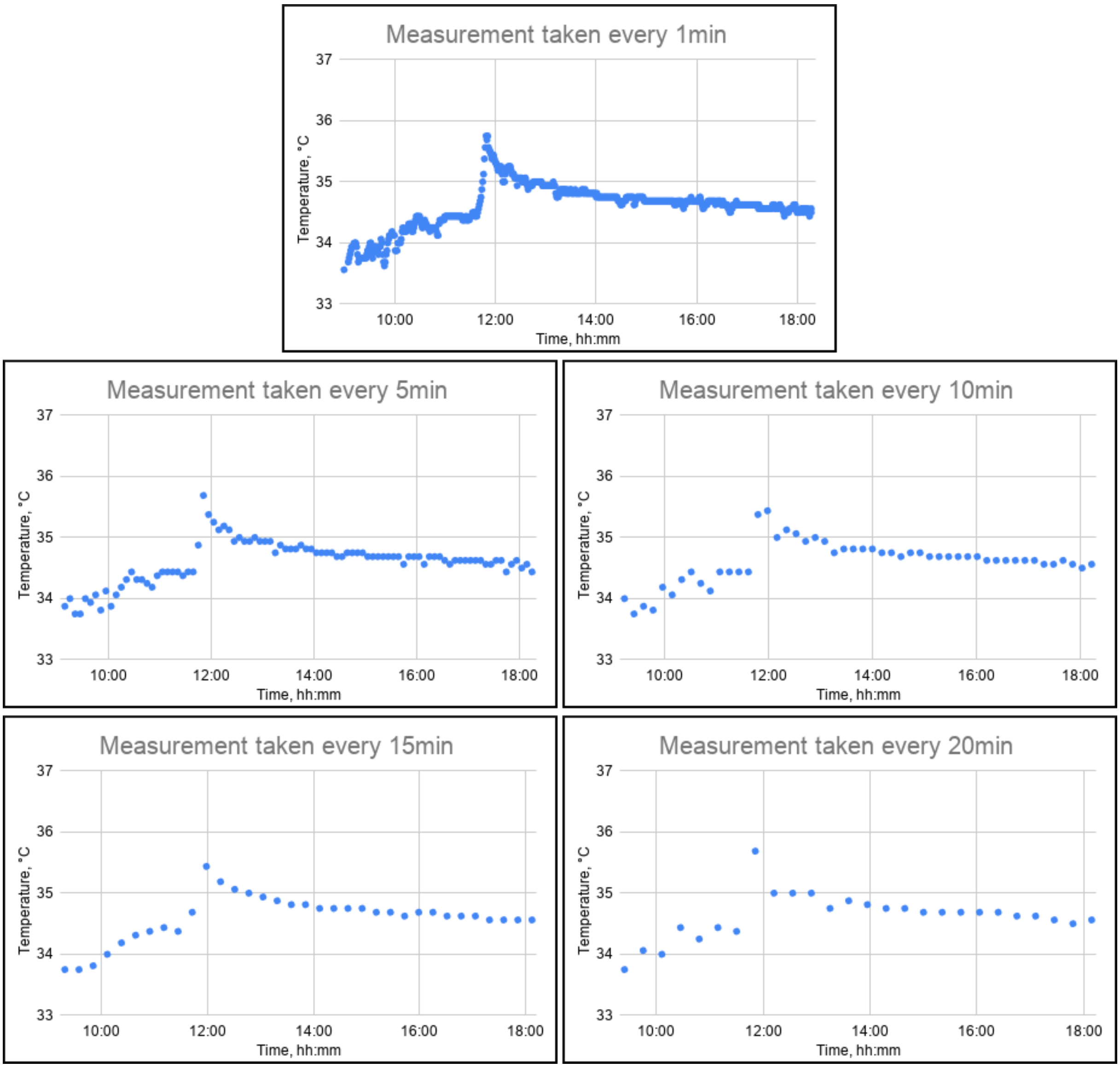

Figure 4. Temperature measurements with different time intervals

The charts clearly show that there is a high probability that with less data points (measurements taken every 15, 20min) the temperature peak (and values close to peak) during swarming can be “missed” since only a slight increase in temperature is recorded. Measurements taken every 1, 5 or even 10min could be more sufficient.

3.7 Abnormal behavior of the colony

Some abnormal situation of the colony can be detected by comparing individual bee colony temperature with average temperature in the apiary. If the temperature difference between a single colony and the whole average is greater by a specific number (determined by the beekeeper), then the beekeeper is informed that this specific colony should be inspected on-site.

4. Methods to detect colony death

One method to detect if the colony is declining, is to compare current (last) temperature value with ambient temperature value. If the values are almost the same or the difference is minimal, it can be concluded that the colony has died. Such a comparison can be expressed as shown below (1):

$$f(x) = \begin{cases} 1, & x \leq (T_{ambient} + \Delta) \\ 0, & x > (T_{ambient} + \Delta) \end{cases} \tag{1}$$

where:

x – current temperature;
$T_{ambient}$ – ambient temperature;
Δ – defined threshold.

Nevertheless, such method may not be enough to detect the death of a colony, because seasons should also be taken into account and this method does not specify the trend of the temperature (increasing/ decreasing, similarity with outside temperature). Therefore, a time series trend analysis between ambient temperature and the temperature inside the hive was considered as a possible method to indicate the possibility of colony declining, nevertheless, this method has some drawbacks, to be specific, some authors[2] state that during autumn (broodless stage) difference between ambient and in-hive temperature is very small. Furthermore, during this stage the temperature trend inside the hive will most likely be similar to ambient temperature. Unlike that honeybees are performing thermoregulation during other seasons, allowing temperature fluctuations within an extremely narrow temperature range.

The proposed method involves a combination of different methods of calculating the correlation coefficient (https://www.statisticssolutions.com/correlation-pearson-kendall-spearman/):

- Pearson correlation – measures the linear relationship between linearly bound variables;
- Spearman rank correlation – a nonparametric test to determine the degree of association between two variables. This correlation test does not provide any assumptions about data distribution, but indicates the strength and direction of the relationship between the two variables;
- Kendall rank correlation – a nonparametric test that determines how dependent are the two variables under consideration;
- Theil-Sen evaluation – usually used in the case of linear trend to estimate slope. The line of Theil-Sen is a nonparametric alternative to linear regression. It models how the median changes linearly with time.

These methods together can provide enough information about the time series trend and linearity. To test these methods, a time series was selected that shows a gradual drop in temperature (see Figure 5). Programmatically this was done using *Python* as the programming

[2] Stalidzans and Berzonis.

language together with *scipy* (https://www.scipy.org/index.html) and *numpy* (http://www.numpy.org/) libraries.

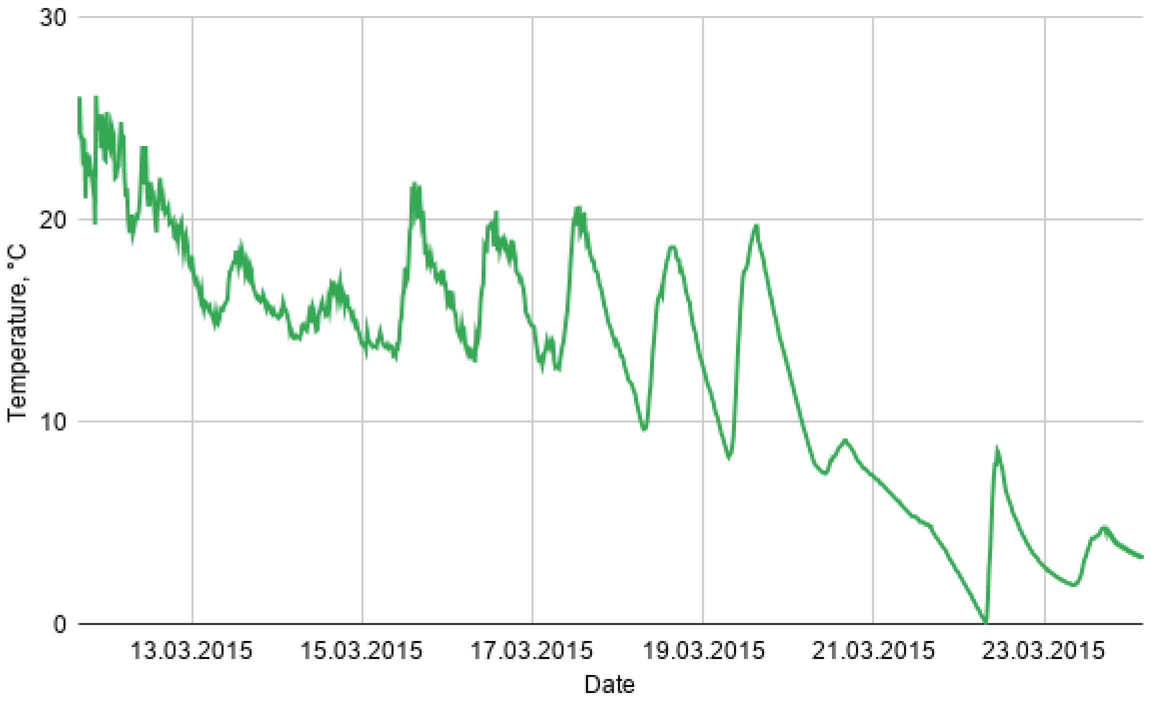

Figure 5. Example of a time series representing colony's decline

Such data example was tested with developed *Fuzzy Inference System* (FIS), described in chapter 6. The first case, when FIS detected abnormality, is shown in Figure 6. As it can be seen, the decrease in temperature is very slow.

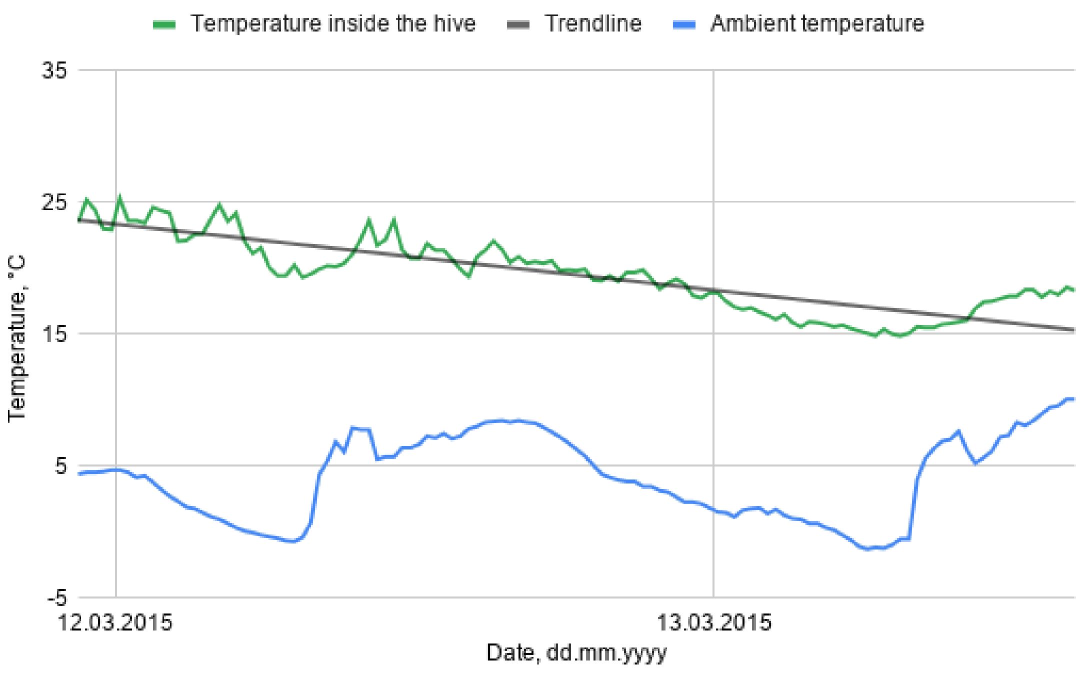

Figure 6. Slow temperature decrease inside the hive

After applying the above mentioned methods (Pearson, Spearman and Kendall) results for a slow temperature decrease analysis are represented in Table 1. Correlation coefficients were calculated between hive temperature and ambient temperature, as well as within the hive temperature.

Table 1. Results of slow temperature decrease analysis

Method	Correlation coefficient		p-value	
	Hive temp. / Ambient temp.	**Within hive**	**Hive temp. / Ambient temp.**	**Within hive**
Pearson	0.22	-0.87	0.012	$2.26*10^{-37}$
Spearman	0.26	-0.87	0.004	$7.27*10^{-39}$
Kendall	0.19	-0.68	0.002	$3.11*10^{-28}$

Correlation by its meaning is a statistical relationship that allows to determine how close a pair of variables are related. The higher the value of coefficient (range [-1;1]), the closest the relationship (dependence), meaning that change in one variable highly impacts the other. In short, *p-value* represents a probability and allows to determine statistical significance. Its value lets to determine, if the result would be obtained if the correlation coefficient would be 0. This value is usually compared against significance value $\alpha = 0.05$ – if *p-value* is less than α, the result (correlation coefficient) is statistically significant.

The values show, that there are no similarities between hive and ambient temperatures (correlation is < 0.3), but that the temperature inside the hive has a linearly negative trend (-0.87). This case just shows that the temperature tends to decrease (also important to know), but there is no reason to believe that in this case the colony has died.

The next case shows the dynamics of similar changes (see Figure 7) in both temperatures.

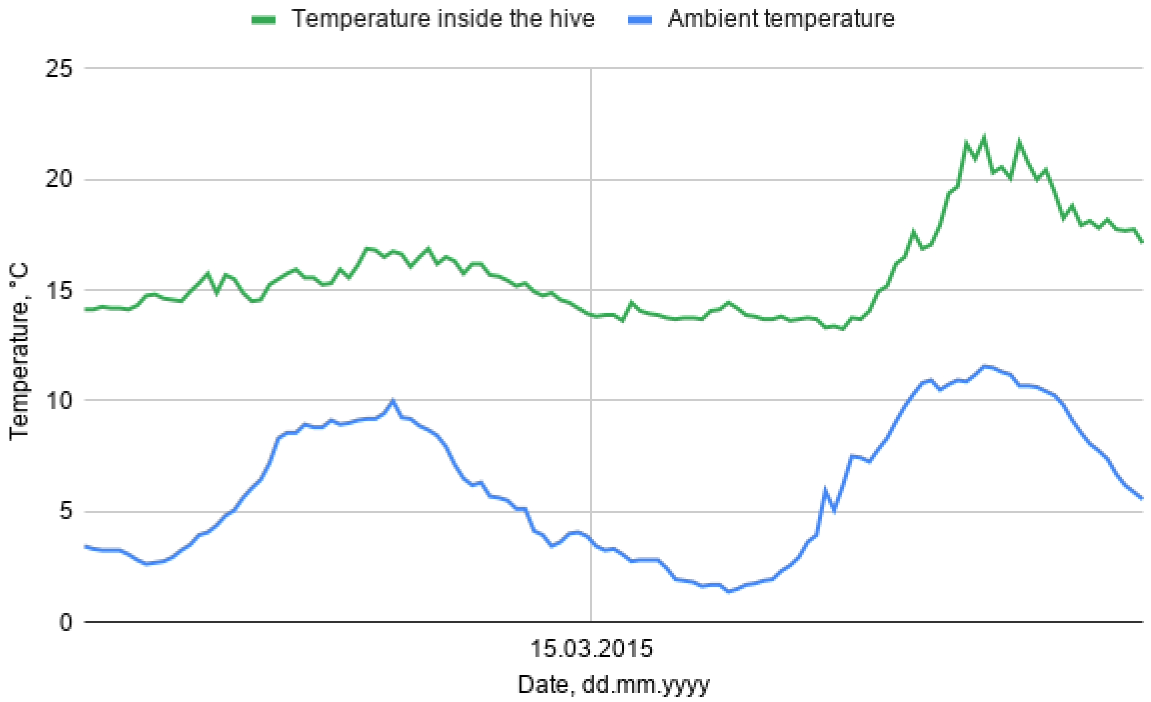

Figure 7. Representation of a close relationship between hive temperature and ambient temperature

The correlation results between hive and ambient temperature are shown in Table 2.

Table 2. Results of close relationship between hive and ambient temperature

Method	Correlation coefficient		p-value	
	Hive temp. / Ambient temp.	**Within hive**	**Hive temp. / Ambient temp.**	**Within hive**
Pearson	0.80	0.48	$7.43*10^{-28}$	$2.87*10^{-08}$
Spearman	0.81	0.31	$1.06*10^{-29}$	0.001
Kendall	0.63	0.15	$6.12*10^{-24}$	0.013

The results clearly show a close relationship between two temperatures (~0.8 with statistical significance way below 0.05). Moreover, the temperature in the hive is not linear over the selected period. Such a close relationship leads to a reasonable assumption that the temperature inside the hive is highly dependent on the ambient temperature and the colony under observation needs to be inspected.

The moment, when it is possible to announce the death of a certain colony is represented in Figure 8:

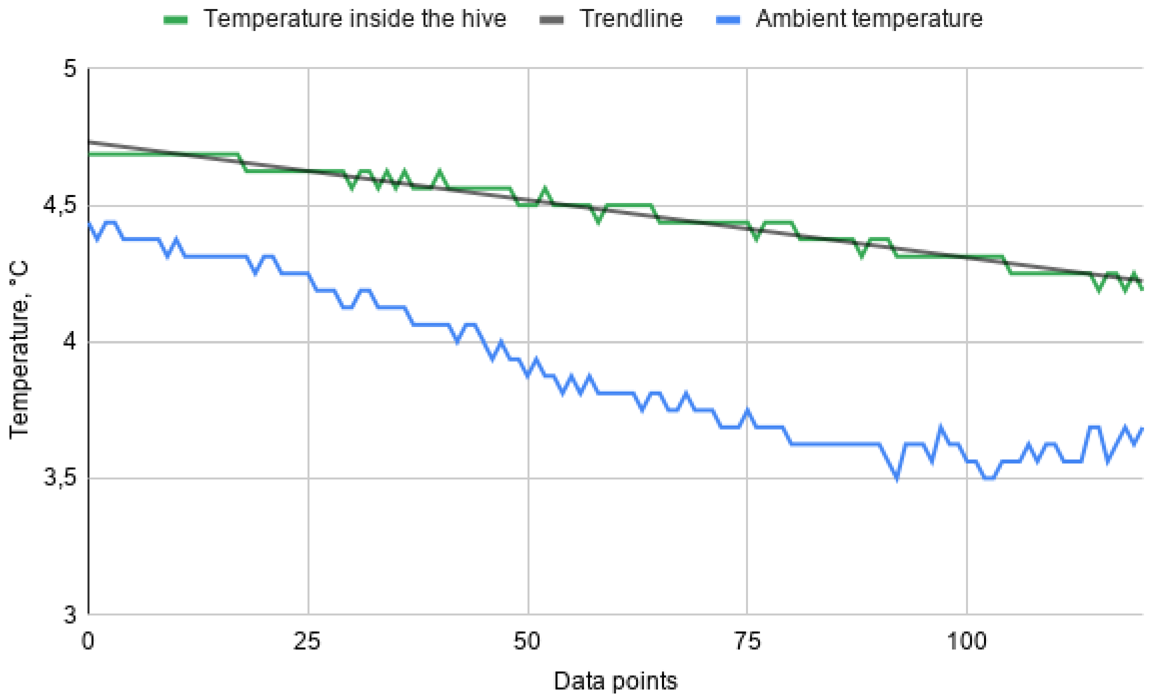

Figure 8. Temperature dynamics in the case of a dead colony

In this case the difference in temperatures is very small, furthermore both curves have a close relationship (dependence) that is represented by the results of analysis in Table 3.

Table 3. Results of temperature data in the case of a dead colony

Method	Correlation coefficient		p-value	
	Hive temp. / Ambient temp.	Within hive	Hive temp. / Ambient temp.	Within hive
Pearson	0.93	-0.99	$1.95*10^{-52}$	$9.83*10^{-97}$
Spearman	0.94	-0.98	$9.28*10^{-59}$	$1.33*10^{-100}$
Kendall	0.86	-0.93	$3.77*10^{-37}$	$1.21*10^{-46}$

The demonstrated cases showed a high potential to use such methods to determine the decline of a colony.

5. Potential of acoustic data for colony state detection

Scientific studies on bee colonies deal with various problems. Naturally, bee colonies have very different characteristics. Examples of this are the genetics and the strength of the colony, which in turn influence the collection and breeding behavior. Furthermore, regular intensive controls to assess the status of the respective bee colonies can be detrimental to the health of the bee colonies. Each opening of a hive influences the colony and thus the research results.

Therefore, it is usually not possible to draw simple causal conclusions from such experiments. The internal validity, as essential condition, cannot be fulfilled. The effects measured with the dependent variable would have to be clearly attributable to the manipulation of the independent variable. Therefore, other influences must be excluded or kept constant. In order to meet these sensible requirements, at least a large number of bee colonies of a relatively pure genetics should be considered. Additionally, methods of neural networks for colony state detection could be used to evaluate the acoustics.

Several current studies show the potential of acoustic data for colony state detection. Some authors[3] were able to determine potentials for the detection of the queenless state with analysis of acoustic data using a multiclass classification. Methods of feature extraction by Mel Frequency Cepstral Coefficients (MFCCs) were used. Feature selection and regularization was performed using a Lasso logistic regression model. It is assumed that the method can also be used to determine further conditions in bee colonies. Some researchers[4] were also able to demonstrate the potential of neural network-based machine learning methods for queenless state detection using acoustics.

At the experimental site of the University of Kassel in Witzenhausen, Germany, seven bee colonies of the sub species Western Honey Bee Buckfast (Apis mellifera buckfast) are permanently monitored acoustically. The data are currently being checked for their interpretability to contribute to the further development of the DSS.

6. Application of neural networks for colony swarming state detection

Just before the swarming, bees prepare for take-off and produce extra heat, thereby a distinctive temperature pattern can be observed – a temperature increase which lasts for about 20 min. During this time, the temperature can increase for more than 3°C. Regarding detection of such cases, it is not enough to set a threshold value in order to detect such temperature increase, since there can be cases, when rapid temperature increase does not point to swarming as it was observed during experiments in ITAPIC project – a high rise in temperature was observed during the month of August. Beekeepers on site did not record any swarming activities that day. Furthermore the temperature pattern had a high peak in a very fast time (3-4 min), which is unlike any other case. Unfortunately, causes for such behavior were not identified.

Furthermore, it needs to be pointed out that temperature values highly depend on the sensor placement inside the hive. If the sensor is not placed right, the threshold method “suffers” and the threshold value needs to be adjusted for each hive. In case that the temperature dynamics are looked as a pattern, the sensor placement will not as much as in threshold’s case but has still an impact. In most cases the pattern would still be formed, but with different peak values. Therefore, pattern classification would perform better.

---

[3] Antonio Robles-Guerrero et al., “Analysis of a Multiclass Classification Problem by Lasso Logistic Regression and Singular Value Decomposition to Identify Sound Patterns in Queenless Bee Colonies,” *Computers and Electronics in Agriculture* 159 (2019): 69–74.

[4] Inês Nolasco et al., “Audio-Based Identification of Beehive States,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, 8256–60.

Such temperature pattern can be recognized/ classified by using artificial neural networks (ANN). ANN or connectionist systems are computing systems that are inspired by biological neural networks, but not identical, that constitute animal brains. Such systems "learn" to perform tasks by considering examples, generally without being programmed with task-specific rules. An ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal to other neurons. An artificial neuron that receives a signal will process it and can signal neurons connected to it.

First implementation of such a method (in order to identify swarming) was done during the ITAPIC project, where the network consisted of 60 inputs – each input neuron corresponds to a data point from an hourly intervals in a time series (where data were recorded each minute). The neural network (further ref. as *Model_1*) structure consisted of three layers: input (60 neurons + bias), hidden (41 neurons + bias) and output (1 neuron giving the probability – how much the pattern is similar to the swarming one) (see Figure 9):

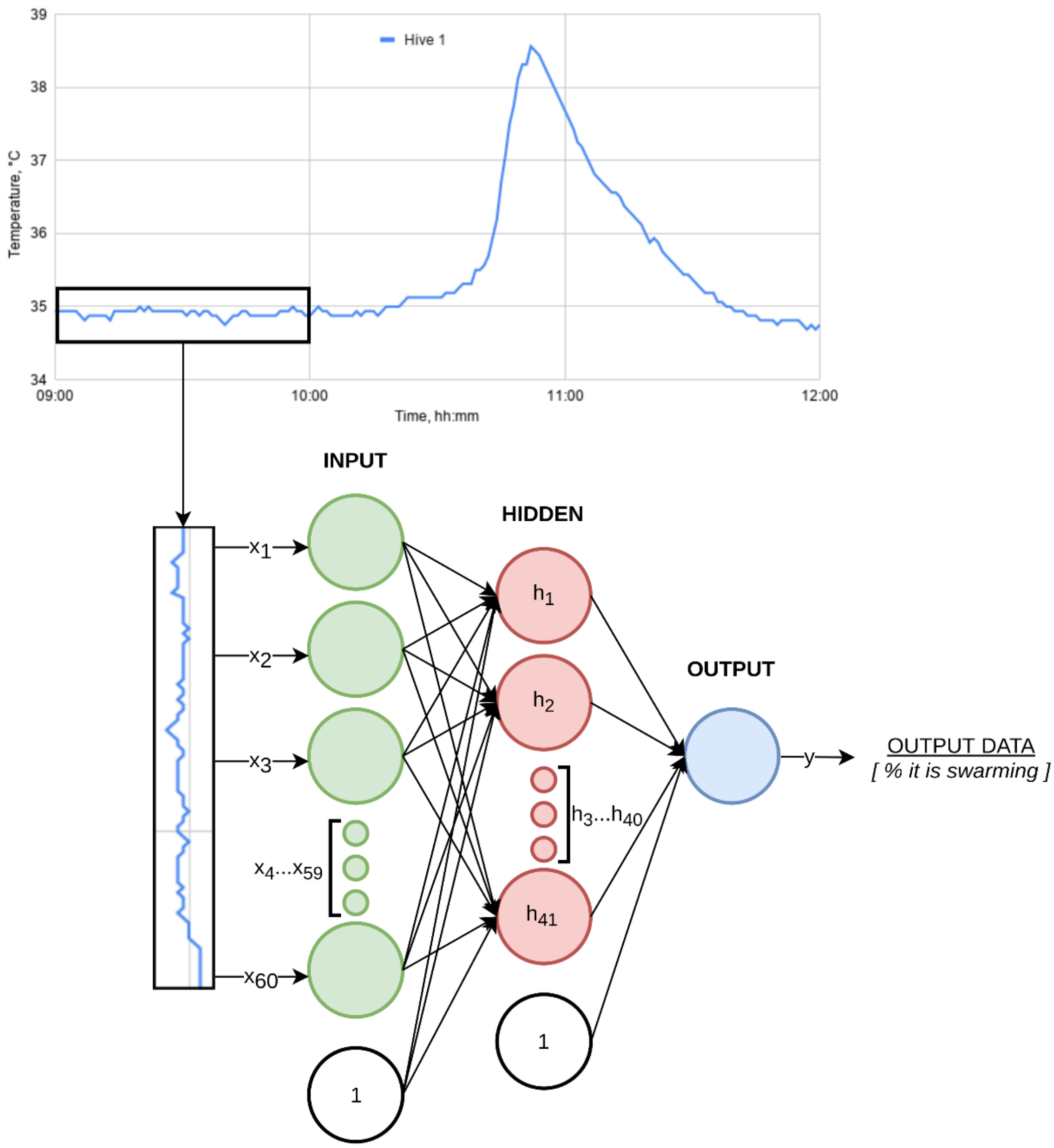

Figure 9. Neural network architecture with time series as inputs

One of the main issues with training a model for such problem area is the lack of swarming data, mainly because of the fact that swarming usually happens in a very short annual period - during spring (occasionally swarms can also happen during the whole foraging season). This should also be pointed out regarding bee swarming activities in Ethiopia and Indonesia. As it was found out from beekeepers in Ethiopia, active swarming is usually observed during September and October, and April through June and most common in traditional hives. Regarding swarming in Indonesia, it was found that most of the traditional beekeepers are not concerned about the colony state. Furthermore, some of them are not even aware of such bee behavior. They are aware of colony absconding, thus detecting such state is important for them. Therefore, there are no information from the local beekeepers during what months the bees do swarm. Swarming detection for *Apis Cerana* species should be similar as it is for *Apis Mellifera* – by detecting rise in temperature also suggested by researchers[5]. Theoretically SAMS can bring great value, by raising awareness on such bee behavior (natural reproduction), when bee population in the colony declines, as a result less productivity from the colony is expected.

Described model (Model_1) was tested with a test set consisting of 90 samples, but it is worth mentioning that the test set was unevenly distributed, meaning that there were more non-swarming samples than swarming. Model_1 proved to work quite effectively. Its performance was evaluated using confusion matrix:

		Predicted	
		YES	**NO**
Actual	**YES**	TP	FN
	NO	FP	TN

where:

TP – true positive (an actual swarming event is predicted as swarming),
FN – false negative (an actual swarming event is predicted as non-swarming),
FP – false positive (an actual non-swarming event is predicted as swarming),
TN – true negative (an actual non-swarming event is predicted as non-swarming).

Five performance measures[6,7,8] were calculated (2)-(6):

[5] Xiangjie Zhu et al., "The Temperature Increase at One Position in the Colony Can Predict Honey Bee Swarming (Apis Cerana)," *Journal of Apicultural Research* 58, no. 4 (2019): 489–91.
[6] Renuka Joshi, "Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures," 2016, https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/.
[7] David Martin Powers, "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation," 2011.
[8] Shruti Saxena, "Precision vs Recall," 2018, https://towardsdatascience.com/precision-vs-recall-386cf9f89488.

1. Accuracy (ACC): the ratio of correct observations to all observations. But it is wrong to assume that this measure is the best performance indicator (especially when the test set is unevenly distributed).

$$ACC = \frac{TP + TN}{TP + FP + FN + TN} \quad (2)$$

2. Recall (REC): the ratio of correctly predicted positive observations to all positive observations. It can be interpreted as: "*From all observed swarming events, how many did the model recognize?*"

$$REC = \frac{TP}{TP + FN} \quad (3)$$

3. Precision (PRC): the ratio of correctly predicted positive observations to the total number of positive observations. This measure is interpreted as: *"From x predicted swarming events, how many were actual swarming events?*"

$$PRC = \frac{TP}{TP + FP} \quad (4)$$

4. Specificity (SPC): the ratio of actual negative cases to the correctly predicted negative cases. In other words it represents "the number of true negatives".

$$SPC = \frac{TN}{FP + TN} \quad (5)$$

5. F score (F): measure that represents both recall and precision. It is assumed that if the model performs very well the F number is larger. The calculation of this measure is based on harmonic mean. In cases when test set is unevenly distributed, it is best to use F score.

$$F = \frac{2 * REC * PRC}{REC + PRC} \quad (6)$$

Model_1 performed as follows:

- accuracy ~100%,
- precision 100%,
- recall ~82%,
- specificity 100%,
- F score ~90%.

During the test phase, it was observed that Model_1 performed poorly when the swarming pattern did not have a very high peak value (max temperature increase ~1.5°C). Therefore, it was decided to improve the swarming detection that resulted in a development of a second model (Model_2).

Model_2 for swarming detection was built using *Tensorflow 1.5* - an open source machine learning platform (https://www.tensorflow.org/) and *Keras* open source neural network library. The whole network training and testing was done using *Python3* programming language. *Tensorflow* framework provides various APIs for software development such as desktop, mobile, web or cloud applications. *Keras* is a high level API to develop (deep) machine learning models.

Model_2 was based on different approach than Model_1. In Model_2 case, four features (statistical measures) were determined from the swarming temperature pattern:

- standard deviation – measures the dispersion of a given data set;
- variance – a measure of distance, representing the mean square deviation of the given data sets mean value;
- kurtosis – a measure that determines the distribution of data around the mean value of a given data set;
- skew – a measure of symmetry.

It was found that those features characterize the temperature dynamics best.

The whole model design process (when using *Keras*) consists of three main phases:

- building the structure of the model (model type, choosing layers),
- model compilation (declaring loss functions and setting optimizers, choosing metrics for evaluation),
- model adjusting/ training (training set, number of epochs, validation conditions).

Structure of Model_2 consists of 4 sequentially connected layers (*Dense* layer type in *Keras* library) that are linearly arranged:

- one input (4 neurons) layer,
- two hidden layers (18 and 12 neurons, respectively),
- one output layer (1 neuron).

Model_2 architecture is shown in Figure 10:

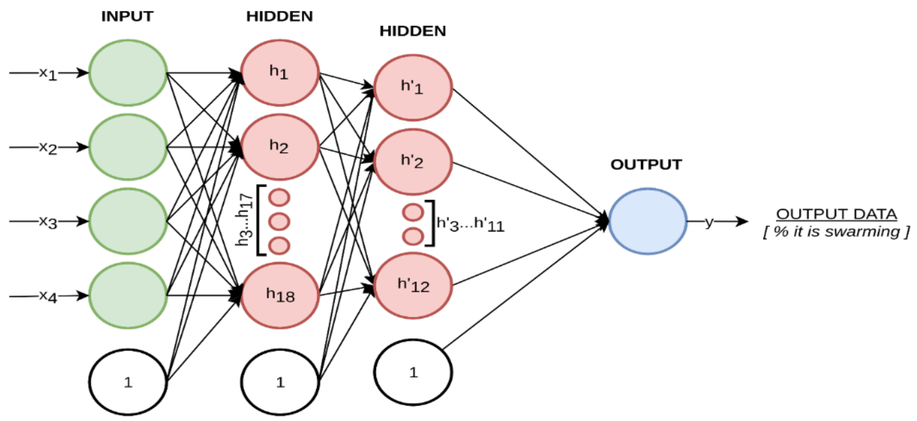

Figure 10. Model_2 architecture

Each layer (except output) is supplemented with a bias neuron as well. For every hidden layer *Rectified Linear Unit* was chosen as activation function that is represented by the following formula (7):

$$f(x) = \max(0, x) \tag{7}$$

Regarding the activation function and output layer, *Sigmoid* was chosen, since its return value falls between [0;1], which is ideal in classification problems, where probability is being predicted. The formula for such an activation function is shown below (8):

$$S(x) = \frac{1}{(1 + e^{-x})} \tag{8}$$

Binary cross entropy was selected for loss calculation. Loss functions evaluates how good/ bad the predictions are. *Binary cross entropy* method is also suggested to use in cases when there are only two label classes (as it is in swarming case) (https://www.tensorflow.org/api_docs/python/tf/keras/losses/BinaryCrossentropy).

An optimizer is required to train a neural network model. Optimizers are closely linked to loss functions and help to adjust the model by tweaking the network weights. As a result, the model is formed to perform at its best. There are a lot of optimizers to choose from, like, *Stochastic gradient descent* (*SGD*), *Adagrad*, *Adam* etc. In this case *Adam* was used. One of the parameters that is important when using optimizers is learning rate (η). Learning rate is a coefficient that impacts the training speed in every epoch. By default *Adam* optimizer has η=0.001.

Adam optimizer is considered as an extension to classical *SGD*. The main difference between *Adam* and *SGD* is the way, how learning rate is being used. If η is constant in *SGD*, then in *Adam* it is adapting during the training process (https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/).

The code to construct described neural network, expressed in *Python3*, is represented below:

```
def getModelArchitecture(self):
      model = keras.models.Sequential()
      model.add(keras.layers.Dense(18, input_dim=self.inputCount,
            activation='relu'))
      model.add(keras.layers.Dense(12, activation='relu'))
      model.add(keras.layers.Dense(1, activation='sigmoid'))
      model.compile(loss='binary_crossentropy', optimizer='adam',
            metrics=['accuracy'])
      return model
```

The compiled model runs with the following settings:

- epochs = 150,
- batch_size = 10

During training process a validation set was also introduced, containing 10% of training data set. The validation set contains data that helps to evaluate the model, but the model never learns from this set. It is used to tune the model's hyperparameters. After the neural network is trained, the model is saved for further usage. The code sample for model training/ fitting is shown below:

```
def runModel(self):
      model = self.getModelArchitecture()
      x_train, y_train, x_val, y_val = self.loadData()
      model.fit(x_train, y_train, epochs=150, batch_size=10, verbose=1,
validation_data=(x_val, y_val))
      scores = model.evaluate(x_train, y_train)
      print("\n%s: %.2f%%" % (model.metrics_names[1], scores[1] * 100))
      model.save(self.modelPath)
```

The same test dataset that was used to evaluate Model_1, was also used to evaluate Model_2. As a result, Model_2 performed better. The performance of both models is summarized in the table below (see Table 4).

Table 4. Model performance comparison

Name	**Accuracy, %**	**Precision, %**	**Recall, %**	**Specificity, %**	**F score, %**
Model_1	100	100	~82	100	~90
Model_2	100	~91	100	100	**~95**

As the table shows, both models have the accuracy measure of 100%, but precision and recall differs. As it was mentioned and suggested before by the resource[9],it is better to take into account both precision and recall when dealing with test datasets that are unevenly distributed. This is represented by the F score, proving that Model_2 performs better than Model_1.

7. Application of Fuzzy Logic for colony state detection

Not all honeybee states have known temperature patterns; therefore, it is crucial to observe any abnormal bee behavior, which can be detected by temperature changes inside the hive, at an early stage. Furthermore, honeybee behavior has non-linear characteristics, which means, for a better understanding of their wellbeing, the application of traditional linear mathematical methods may not be sufficient. That is why the fuzzy logic was proposed to be used to detect early abnormality and honeybee colony states.

Fuzzy logic can be described as logic that was designed to express human knowledge and reasoning. It is a generalization of standard logic. If we compare fuzzy logic with Boolean logic, the main difference lies in how the values are expressed. Boolean logic can be applicable to concepts with complete values, like 0 or 1, True or False. Fuzzy logic uses what is called "degree of truth", therefore it is dealing with values between 0 and 1 (referring to certain things that are not completely clear ("fuzzy")). This is similar to how, for example, customers would provide product feedback by answering not only with "Agree" or "Disagree", but also "Partly agree", "Rather agree than disagree" etc. To determine the "degree of truth", fuzzy logic uses

[9] Joshi, "Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures."

membership functions (represented as μ(x)). Mathematically membership function can be expressed as (9):

$$\mu_A : X \rightarrow [0,1] \tag{9}$$

Membership functions are the most important part in a fuzzy logic system, because these functions define the fuzziness of the given fuzzy set. Graphically a membership function represents the given fuzzy set, where *x* axis represents the universe of discourse, y – degree of membership in a range [0,1]. These functions can take many forms (shapes), yet there are no strict rules, when to choose one or another, but this choice should be made wisely considering the particular problem and its specifics. By conducting several information resources[10,11,12] the most popular membership functions are triangular, trapezoidal, singleton, R- and L-, and Gaussian.

Another difference from classical mathematical methods is that fuzzy logic systems uses linguistic variables, meaning rules that make up the knowledge base, do not contain numerical values, but rather are represented by words describing the numerical value. For example, when referring to room temperature that is 30°C, fuzzy rule would include values as "hot" or "high" instead of real, specific numbers. There are mainly two fuzzy rule types – *Mamdani* and *Takagi-Sugeno*, where the difference lies in how the consequent part of a rule is constructed. If in *Mamdani* type the consequent also consists of linguistic variables, then in *Takagi-Sugeno* type consequent is a function or a constant (this is ideal for non-linear, adaptive control with fuzzy controllers). Rules are defined by IF..THEN statements:

`IF room_temp IS high THEN window IS open.` (*Mamdani* type)

`IF room_temp IS high THEN window_angle IS f(x,y).` (*Takagi-Sugeno* type)

Since Mamdani fuzzy rule type was chosen as it better suits the problem, further description is related to only such type of systems.

Linguistic variables are determined during fuzzification process, where input crisp values are mapped into membership functions. Fuzzification is an important step in the *Fuzzy Inference System* (FIS).

Fuzzy Inference System and its components

FIS is a system that takes crisp (numeric) input values, applies fuzzy logic and generates crisp output values. Such a system usually consists of several main units (*input interface*,

[10] Sanjay Krishnankutty Alonso, "FUZZY OPERATORS," 2015, http://www.dma.fi.upm.es/recursos/aplicaciones/logica_borrosa/web/fuzzy_inferencia/fuzzyop_en.htm.
[11] Qadri Hamarsheh, "Neural Networks and Fuzzy Logic, Lecture 18," n.d., http://www.philadelphia.edu.jo/academics/qhamarsheh/uploads/Lecture 18_Different Types of Membership Functions 1.pdf.
[12] Mojtaba Rajabi, Bahman Bohloli, and Esmaeil Gholampour Ahangar, "Intelligent Approaches for Prediction of Compressional, Shear and Stoneley Wave Velocities from Conventional Well Log Data: A Case Study from the Sarvak Carbonate Reservoir in the Abadan Plain (Southwestern Iran)," *Computers & Geosciences* 36 (2010): 647–64, https://doi.org/10.1016/j.cageo.2009.09.008.

knowledge base, *inference engine*, *output interface*) and in general involves three steps. A typical FIS is demonstrated in Figure 11:

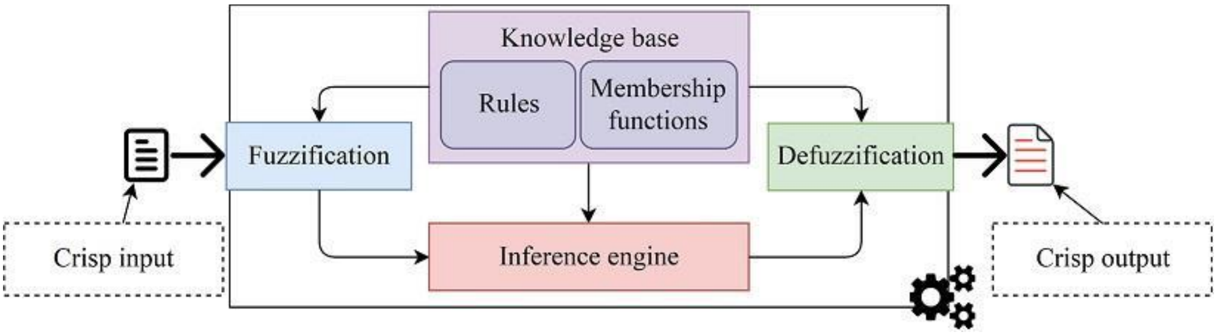

Figure 11. General structure of FIS architecture[13]

Description of FIS components:

- input interface – system accepts crisp (numeric) input values and applies fuzzification;
- knowledge base – contains membership functions and rule base;
- inference engine – performs processing of defined rules;
- output interface – result value (fuzzy result) is defuzzified into crisp (numeric) output value.

Three main steps are necessary for a functional FIS:

- fuzzification – crisp inputs are turned into fuzzy data or membership functions;
- fuzzy inference process – combines membership functions with defined rules to obtain fuzzy output data. This step involves interaction between inference engine and knowledge base;
- defuzzification – transforms output fuzzy data into crisp values.

Besides the non-linearity, fuzzy logic can help to solve another problem, the actual sensor location that arises when monitoring the temperature inside the hive. The placement inside the hive directly affects the values recorded (previous experience from the ITAPIC project suggests that, if the temperature sensor is placed in the middle of the brood frame, temperatures can be close to 36°C, if placed aside – even less than 34°C during spring/ summer, active brood rearing period). These variations in temperature (during normal honeybee activity) can be solved by carefully designed membership functions.

FIS membership functions for honeybee colony state identification

Since data analysis and interpretation was mostly based on historical data gathered in European region, the following FIS and its components were built for colony state detection for this region. Nevertheless, in this report, advantages of using FIS will be demonstrated and it will be described how and why such a system is adjustable for various regions and its specifics, requiring minimal efforts.

[13] Kamyar Mehran, "Takagi-Sugeno Fuzzy Modeling for Process Control," *Industrial Automation, Robotics and Artificial Intelligence (EEE8005)* 262 (2008).

Membership functions for FIS input parameters were designed by conducting multiple information sources – consultations with beekeepers, literature (books and scientific publications) and personal experience (gained during ITAPIC project). As a result a *Mamdani* type FIS with five input parameters was selected:

- temperature inside the hive (*th:{verylow, low, moderate, normal, high}*) – this input parameter describes the temperature in the beehive during all seasons, thus several membership functions were derived (five in total). For example, during winter temperature values that fall between 5°C < x < 20°C range are considered to interpret that bee colony are in normal state, however during summer this "normal" state is considered when temperature values are between 30°C < x < 36°C. This is the reason for defining multiple membership functions;
- ambient temperature (*tout:{verylow, low, normal, high, veryhigh}*) – temperature changes in different seasons were taken into account when defining memberships for this input parameter;
- difference between *th* and *tout* (*tdiff:{small, large}*) – this input parameter represents the difference between the ambient temperature and the temperature inside the beehive, which is important, e.g., for the detection of possible colony decline;
- difference between temperatures inside the same hive (current and an hour before) (*tdiff_hive:{small, large}*) – input parameter that denotes the difference in temperature, by comparing the current temperature value with an hour ago;
- month (*month:{winter, spring, summer, autumn}*) – membership functions for this parameter maps the input (month) to a specific season.

FIS output membership functions for honeybee state – *state:{death, extreme, normal}*. Output has three membership functions representing death, normal and extreme states. Since the output of developed FIS is a crisp value it is interpreted as an "assessment of the colony in %". The application of FIS in beekeeping is a special case, because usually the output of FIS is being used as a feedback to some kind of process controller[14,15].

To elaborate more on the output membership functions, it is important to point out that state "death" can also be called (included in) "extreme" state, but there are cases, rules that can describe, distinguish honeybee colony decline, hence detecting a more specific state. But since the colony death may vary from one season to another (and there are still not enough data about such cases) it is acceptable to count it as "Extreme" state.

"Extreme" also includes swarming state. Swarming is not distinguished as a separate state due to the fact, that a specific pattern needs to be identified, but FIS can only point to some abnormal temperature deviations (high temperature values does not always mean that it is a swarming event. Therefore, detection of "Extreme" state needs to be verified by described ANN). "Extreme" state may include some diseases whose pattern is not known (for example, a distinctive temperature increase during winter) or an early brood rearing state.

[14] Amit Salunkhe, "Compound—Fuzzy Inference System for Temperature Controller," *International Journal of Electronics Engineering* 2, no. 2 (2010): 341–44.
[15] Piyush Singhala, Dhrumil Shah, and Bhavikkumar Patel, "Temperature Control Using Fuzzy Logic," *ArXiv Preprint ArXiv:1402.3654*, 2014.

To conclude – there is still not enough data to separate several states (as minimum at least 10 cases per colony state could give a reasonable insight to define rules or develop specific algorithms), but counting also the unknown ones as “Extreme” is a really good indicator to the beekeeper that there is something wrong with his colony(-ies).

Graphically these membership functions are represented below (Figure 12 - Figure 17):

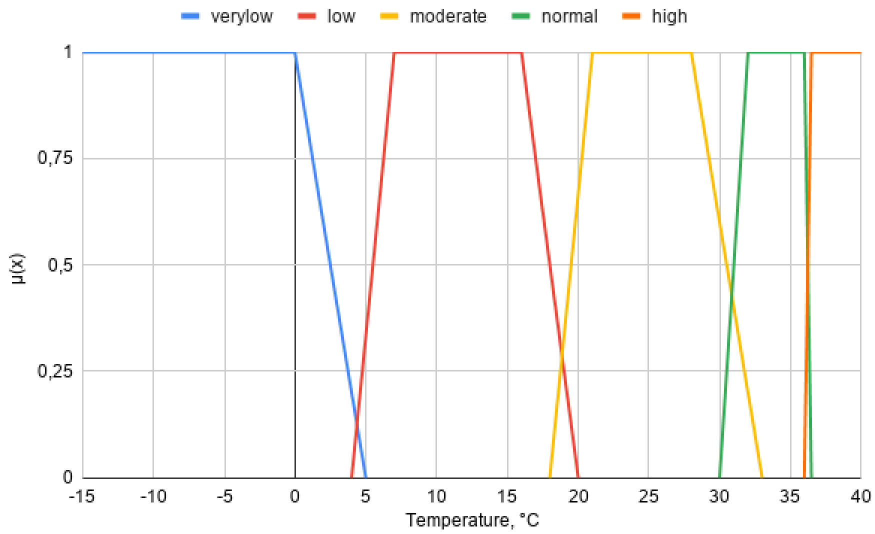

Figure 12. Membership functions for temperature inside the hive parameter (th)

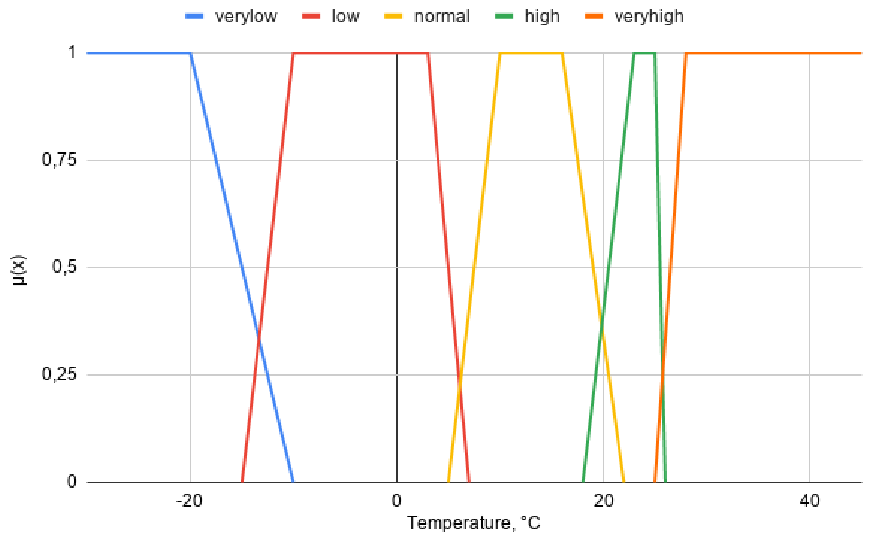

Figure 13. Membership functions for ambient temperature parameter (tout)

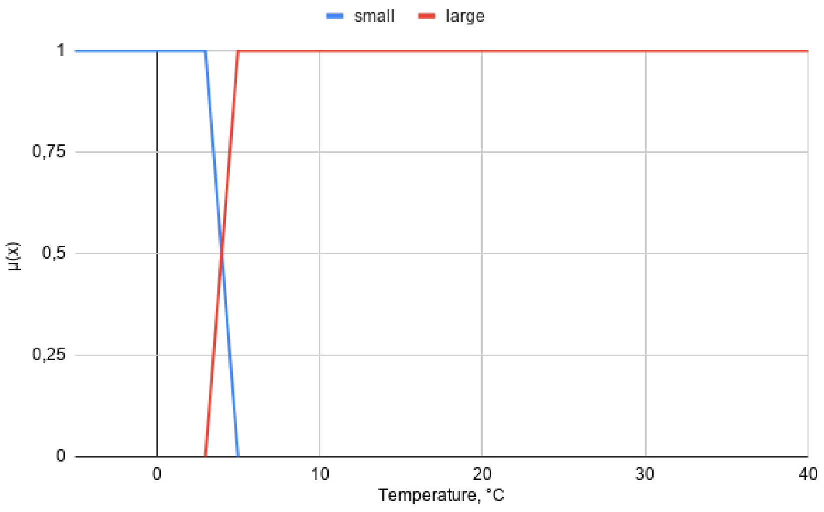

Figure 14. Membership functions representing difference between th and tout (tdiff)

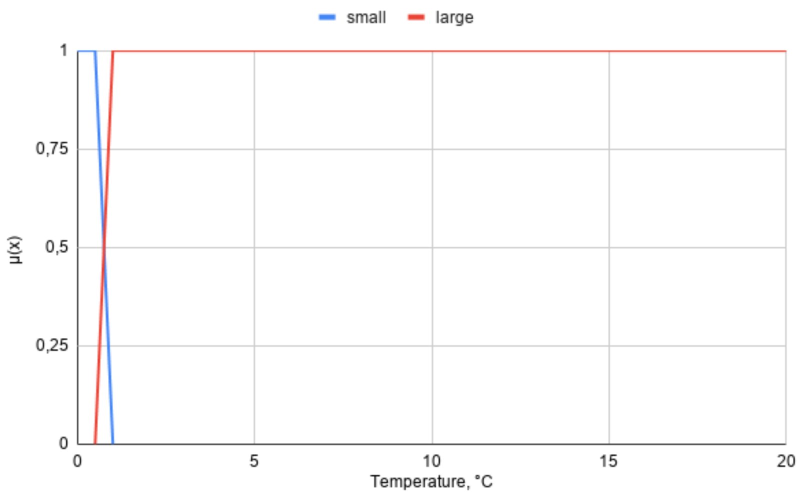

Figure 15. Membership functions representing difference in temperature inside the same hive (thdiff)

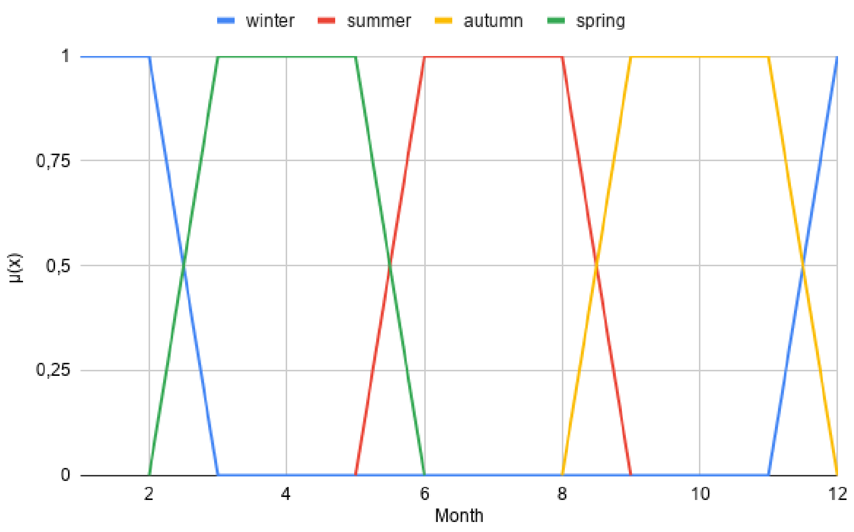

Figure 16. Membership functions representing month parameter (month)

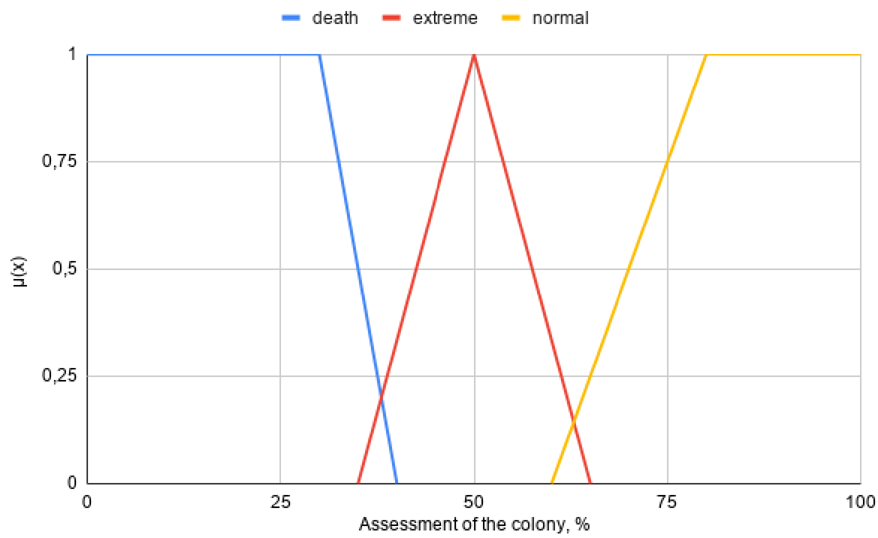

Figure 17. Membership functions for output (state)

Data set and decision tree for defining a rule set

During the FIS development a data set was defined that describes the occurrence of various states at defined input parameters. The data set consisted of 156 records. An example of such a data set (3 records) is shown in Table 5.

Table 5. An example of a bee colony state identification data set

#	Inputs (crisp values)					Output
	th	***tout***	***month***	***tdiff***	***thdiff***	***state***
1.	33.0	15.0	5.0	18.0	0.5	normal
2.	32.0	8.0	12.0	20.0	0.2	extreme
3.	23.0	12.5	10.0	10.5	0.8	normal
#	**Inputs (fuzzified)**					**Output**
	th	***tout***	***month***	***tdiff***	***thdiff***	***state***
1.	normal	normal	spring	large	small	normal
2.	normal	normal	winter	large	small	extreme
3.	moderate	normal	autumn	large	small	normal

The created data set served as the basis for the final rule set. It is important to determine which rules have very less impact or have no impact at all. Therefore, a decision tree algorithm was applied – one of the most popular decision tree algorithms is ID3 (*Iterative Dichotomiser 3*). Such an approach makes it possible to identify the attribute with the highest information gain (*root* node) and construct a tree by determining the next attributes under respective branches.

As a result, a final rule set was developed and based on the constructed decision tree where *thdiff* was identified as the root node (see Figure 18, Figure 19):

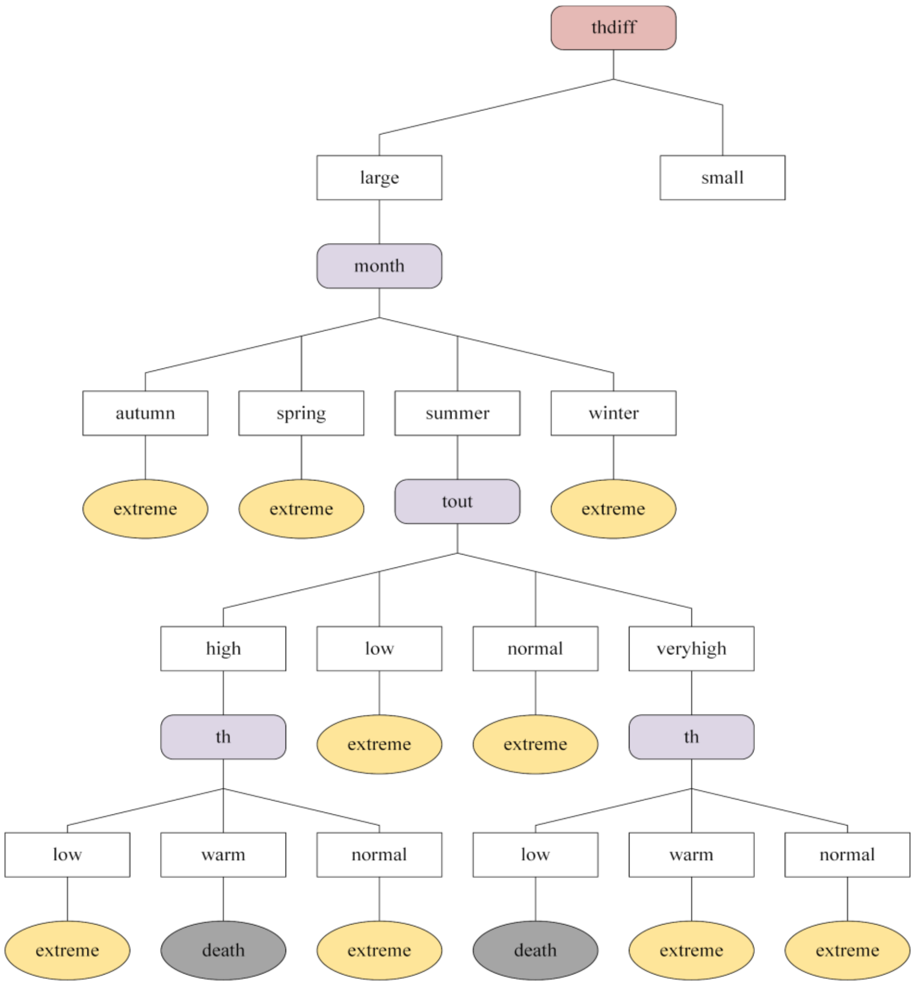

Figure 18. Left branch of developed decision tree

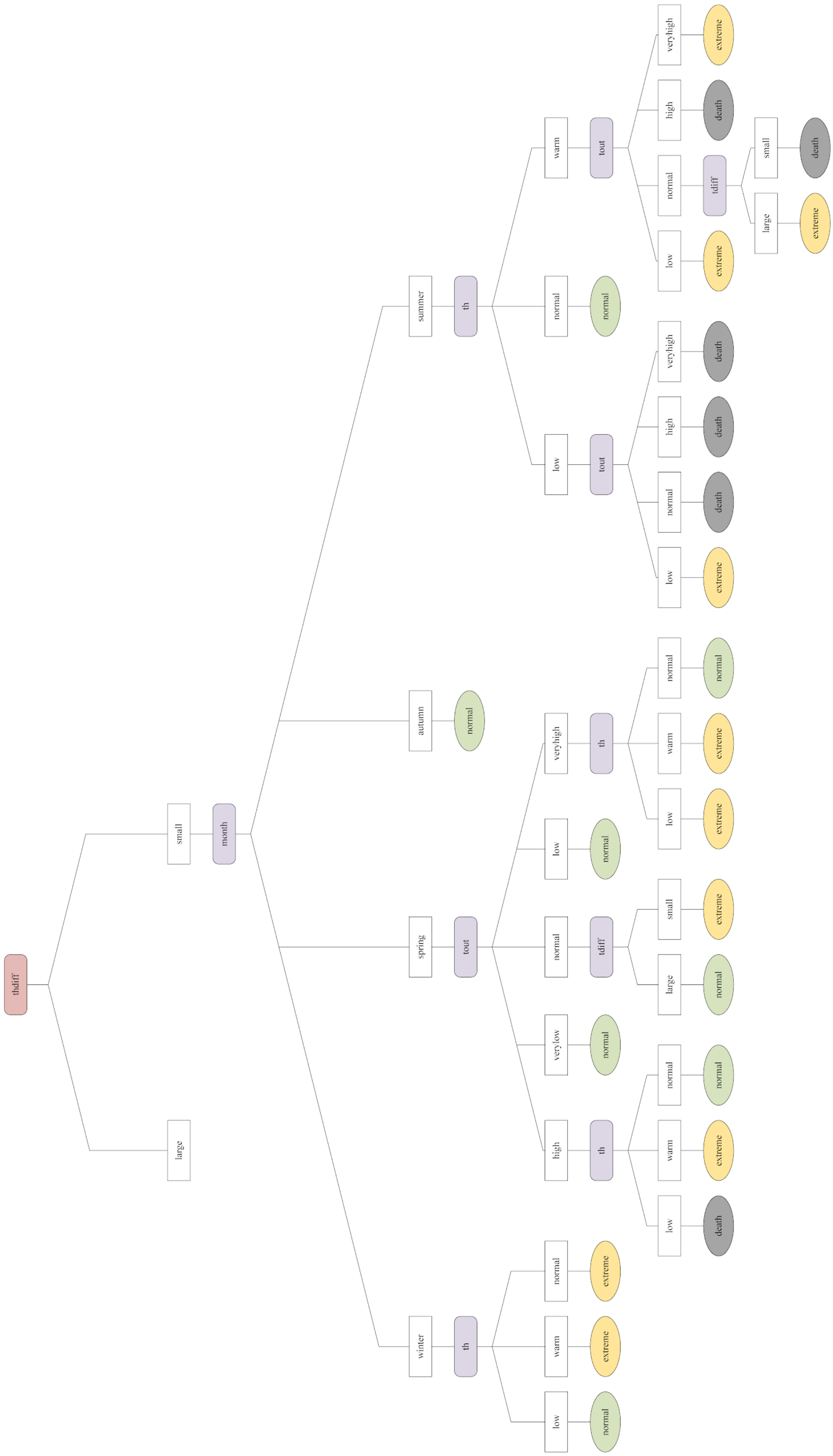

Figure 19. Right branch of developed decision tree

Final rule set consisted of 37 rules. An example of the rule set is given below:

```
RULE 1: IF thdiff IS large AND month IS autumn THEN state IS extreme;
RULE 2: IF thdiff IS large AND month IS spring THEN state IS extreme;
RULE 3: IF thdiff IS large AND month IS winter THEN state IS extreme;
RULE 4: IF thdiff IS large AND month IS summer AND tout IS high AND th IS
low THEN state IS extreme;
RULE 5: IF thdiff IS large AND month IS summer AND tout IS high AND th IS
moderate THEN state IS death;
...
```

FIS development – programming language and libraries required

Different languages (*Java*, *C#*, *Python* etc.) and thus libraries and frameworks can be used to develop a fuzzy logic based systems, such as *AForge*, *jFuzzyLogic*, *fuzzylitex*, *FuzzyLite* etc. Since the SAMS DW back-end is developed using the *Java* programming language, it was decided to use *Java* and hence *jFuzzyLogic* library to better integrate the FIS into the existing data management system.

jFuzzyLogic library[16,17] is an open source library for fuzzy system development that allows to design and develop fuzzy logic controllers (FLC) following a certain standard (IEC 61131-7). This library offers implementation for a fully functioning FIS and API for writing and testing Fuzzy Control Language (FCL) code.

jFuzzyLogic uses a separate file to define one or many function blocks for FIS development. This file follows a specific structure. There are sections that define inputs, outputs, membership functions and a rule block. Rules are written as previously shown in this report, by using linguistic approach.

Below is an example of how the input as well as output parameters and membership functions for *th* are defined:

```
FUNCTION_BLOCK honeybees

VAR_INPUT
    th : REAL;
    tout : REAL;
    month : REAL;
    tdiff : REAL;
    thdiff : REAL;
END_VAR

VAR_OUTPUT
    state : REAL;
END_VAR

FUZZIFY th
    TERM verylow :=(-15, 1) (0, 1) (5, 0);
    TERM low := (4, 0) (7,1) (16,1) (20,0);
```

[16] Pablo Cingolani and Jesus Alcala-Fdez, "JFuzzyLogic: A Robust and Flexible Fuzzy-Logic Inference System Language Implementation," in *Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference On*, 2012, 1–8.
[17] Pablo Cingolani and Jesús Alcalá-Fdez, "JFuzzyLogic: A Java Library to Design Fuzzy Logic Controllers According to the Standard for Fuzzy Control Programming," *International Journal of Computational Intelligence Systems* 6, no. sup1 (2013): 61–75.

```
    TERM moderate := (18, 0) (21,1) (28,1) (33,0);
    TERM normal := (30, 0) (32,1) (36,1) (36.5,0);
    TERM high := (36, 0) (36.5, 1) (40, 1);
END_FUZZIFY

FUZZIFY tout
    TERM verylow := (-30, 1) (-20,1) (-10, 0);
...
```

The code sample to initialize FIS and evaluate the result is demonstrated below:

```
FIS fis;
public FuzzyInferenceSystem() {

      String fileName = "fcl/honeybees.fcl";
      this.fis = FIS.load(fileName, true);
      if (fis == null) {
            System.err.println("Error loading file: '" + fileName + "'");
            return;
      }
 }

public double eval (double th, double tout, int month, double thdiff) {

      fis.getVariable("th").setValue(th);
      fis.getVariable("tout").setValue(tout);
      fis.getVariable("month").setValue(month);
      fis.getVariable("tdiff").setValue(Math.abs(th - tout));
      fis.getVariable("thdiff").setValue(thdiff);

      fis.evaluate();
      Variable resState = fis.getVariable("state");

      return resState.getLatestDefuzzifiedValue();
}
```

FIS testing and evaluation

FIS was tested with a test set consisting of 90 samples. Test samples included various bee colony states (swarming, death, high temperature increases) in different seasons. The test set contained 20 swarming cases in total. One of the interesting test samples was a swarming event. Usually the peak temperature during a swarming is close or above 37°C but during the test it occurred with only 36,3°C and even so it is not the typical case the FIS detected the swarming, as it is shown in Figure 20.

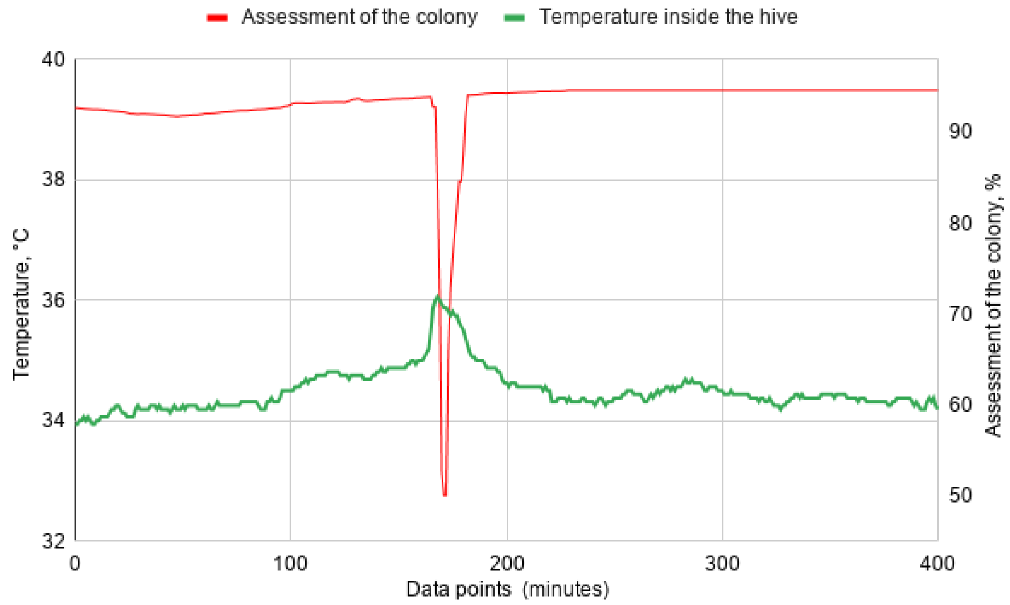

Figure 20. FIS detects swarming event

Developed FIS was able to detect slow decrease in temperature (4°C in total) (Figure 21). Such temperature change is not typical during summer. As Figure 21 shows, temperature was kept stable at the beginning, but then sudden changes were observed and detected by FIS.

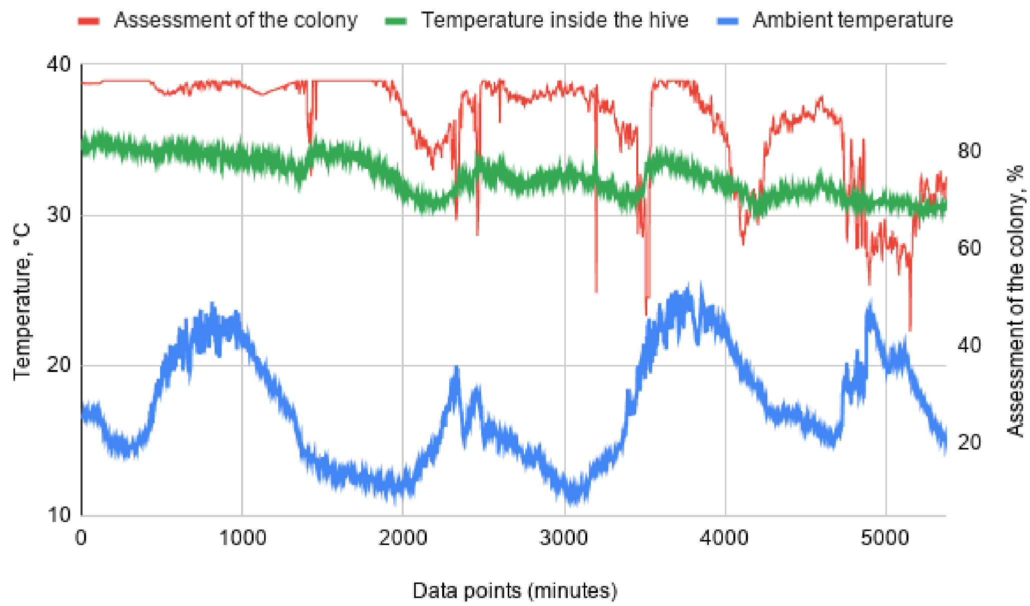

Figure 21. Detection of temperature decrease (and fluctuations) during summer season

In case when honeybee colony declined, FIS was able to detect such an abnormal behavior by signaling the occurrence of “Extreme” state, and since the temperature had a decreasing trend, after a while FIS showed the occurrence of “Death” state (Figure 22):

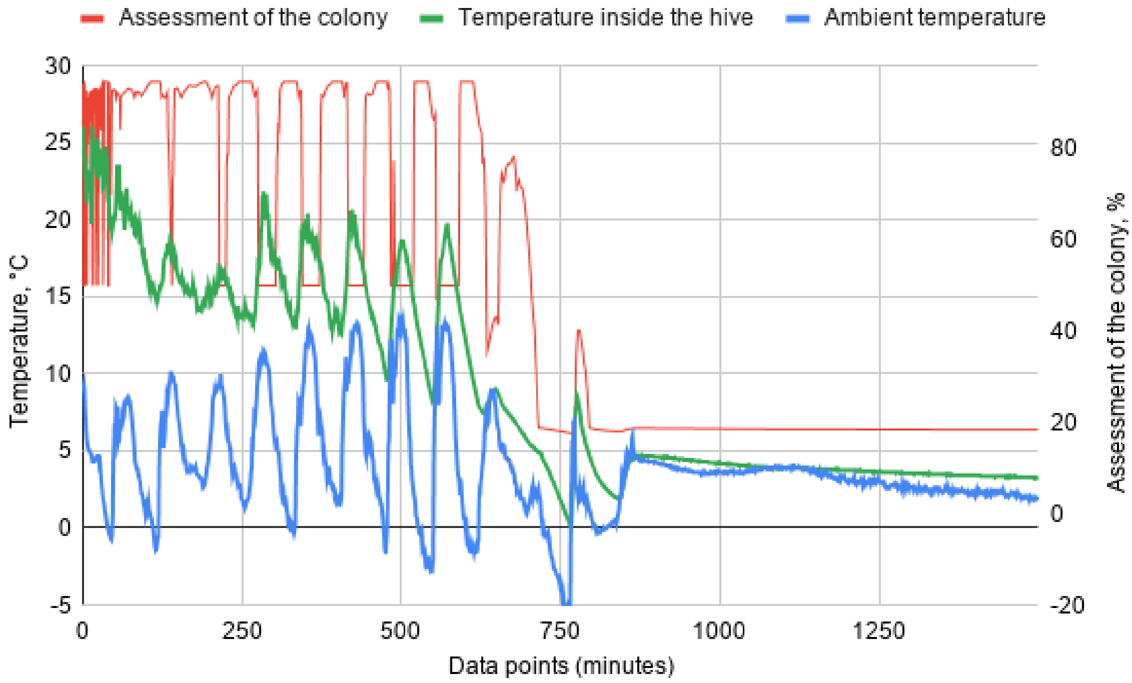

Figure 22. Detection of a honey bee colony death

Uncharacteristic increases in temperature during winter was also successfully detected by FIS, as it is shown by Figure 23:

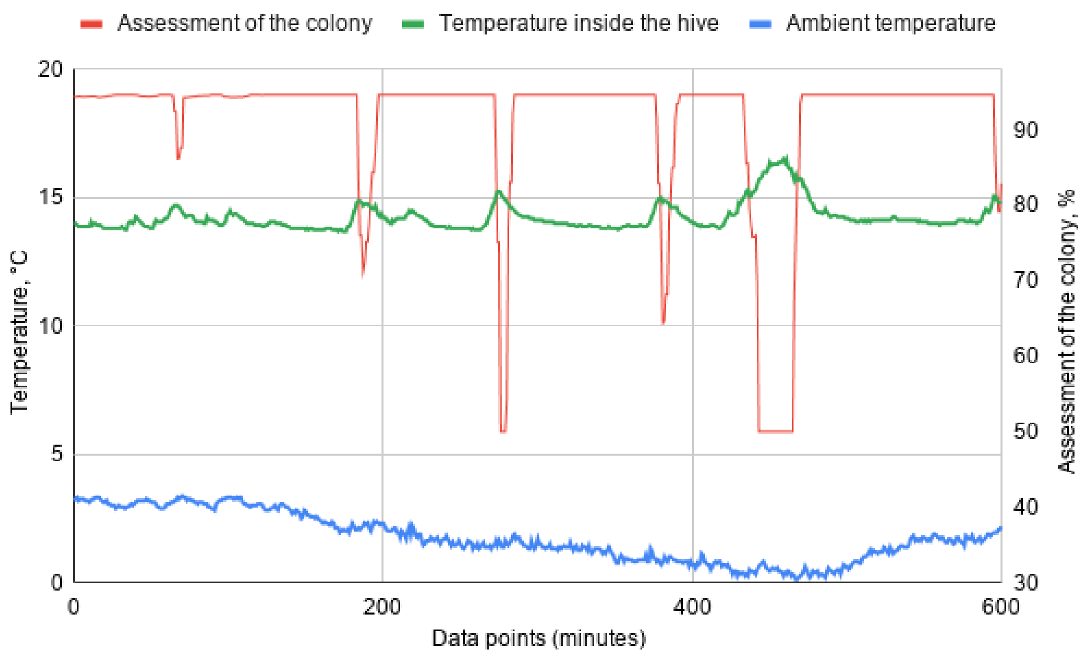

Figure 23. Temperature spike detection during winter season

Overall, the developed FIS demonstrated a robust performance that was proven also by confusion matrix:

- accuracy ~98%,

- precision 100%,
- specificity 100%,
- recall ~97%,
- F score ~98%.

FIS instability was only observed in cases where specific and much deeper investigation is needed.

As it was mentioned earlier, the demonstrated state detection was tested on cases in European climatic regions, but the principle how the FIS is built, how the memberships are defined, shows that it is adjustable and modifiable in a very convenient way, to adapt such a system for Ethiopian and Indonesian regions, either by introducing new input parameters (like “region”) or by defining new function blocks. An example of such additional input parameter is given below:

```
IF th IS normal AND month IS winter AND region IS Latvia THEN state IS
extreme
IF th IS normal AND month IS winter AND region IS Ethiopia THEN state IS
normal
```

Therefore, as mentioned before, the sensor placement inside the hive is very important. To demonstrate that, below Figure 24 represents temperature data in Hive 2 located in Ciburial, Indonesia. As it can be seen, both temperatures are very similar, and in this case, it is hard to tell if the colony is even alive.

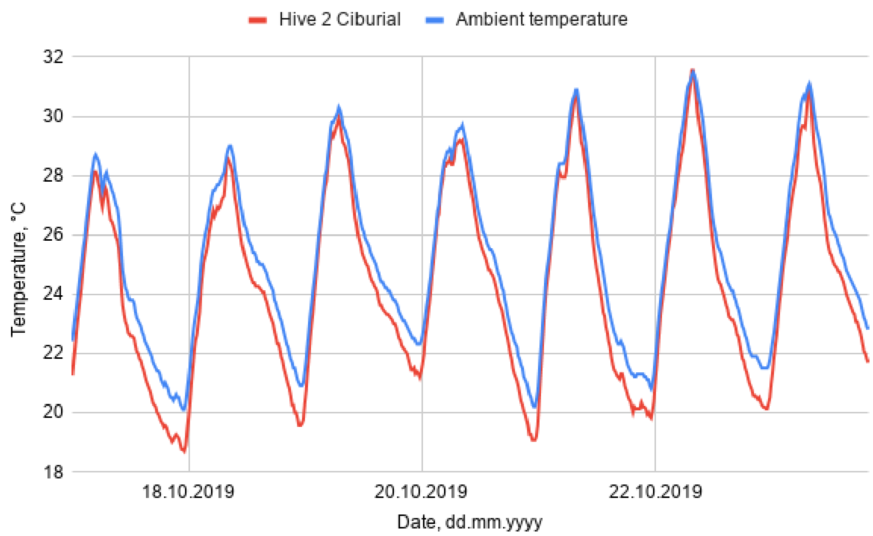

Figure 24. Temperature spike detection during winter season

It is also worth to point out that this analysis was strictly based on temperature dynamics research. Therefore, by adding multiple data sources to the rules, the FIS can be upgraded so it can distinguish states that are more specific.

8. Further development

In the further development of data analysis and interpretation, several improvements shall be made:

- combination of multiple data – temperature and weight. Therefore, weight data needs to be filtered to draw appropriate conclusions. As it was observed, during colony inspection, workers in Ethiopia tend to put equipment, suppers (extra boxes) on top of other hives, hence disturbing the readings (Figure 25):

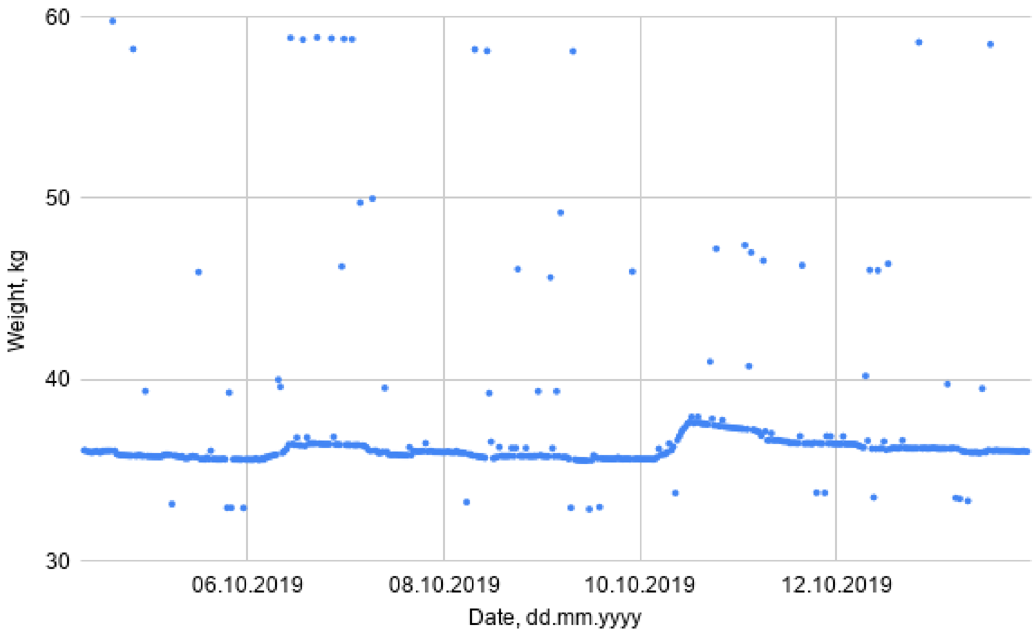

Figure 25. Scattered weight data

- Identification of absconding state – there is a potential to gather data about this phenomenon, since in Indonesia colonies happen to abscond a lot
- FIS adjustments for regions in Ethiopia and Indonesia: there are still challenges the SAMS project has to deal with to ensure a stable monitoring of bee hives to gather reliable and useful data that can be used for final FIS modifications:
    - stable Internet connectivity,
    - device connectivity issues to local network,
    - quality of the hives being monitored,
    - availability of IT experts near the site,
    - location of the hives (time consuming to get there and make adjustments),
    - positioning of the sensors,
    - sensor recording issues (software errors).

Project website: www.sams-project.eu

Project Coordinator contact:

Angela Zur
Deutsche Gesellschaft für Internationale Zusammenarbeit (GIZ) GmbH
An der Alster 62,
20999 Hamburg, Germany
Angela.Zur@giz.de

DISCLAIMER