

Remote monitoring of beehive activity

Uroš Pešović¹, Dušan Marković², Slađana Đurašević¹, Siniša Randić¹

¹*Faculty of Technical Sciences, University of Kragujevac,
Svetog Save 65, 32000 Čačak, Serbia*

²*Faculty of Agronomy, University of Kragujevac,
Cara Dušana 34, 32000 Čačak, Serbia*

Corresponding author: dusan.markovic@kg.ac.rs

Abstract: Colonies of the European honeybee, the most important pollinator insects, are subjected to many negative impacts of modern society, in form of pollution, pests, invading species and colony collapse disorder (CCD). The newly emerging technology of the Internet of Things (IoT), enables remote monitoring using wireless sensors inside beehives. In this paper, we present the model of the system for remote temperature monitoring on a various points inside the beehive colony using IoT devices. Measured data is transmitted to a remote server where received data streams are processed in real-time using Complex Event Processing (CEP) which enables detection of critical events and report it to the beekeeper. These data streams are compared with reference temperature patterns using machine learning algorithms, which give computers the ability to learn to detect events without being explicitly programmed. This system can significantly reduce the beekeeper reaction time and increase chances for a beehive colony overcoming certain types of anomalies with human intervention.

Keywords: Beehive, Internet of Things, machine learning, complex event processing.

Introduction

Beekeeping represents one of the important parts of agriculture, both as a source of honey as well as its importance for fruit pollination. The resolution T6-0579/2008 of the EU parliament noted that 79% of the world's food supply is dependent on honey bees (Aizen and Harder, 2009). As well as many species, honeybees are significantly

affected by negative impacts of modern society in the form of pollution, the introduction of invasive species, climate change, etc.

Beehive colonies require frequent inspections since they could be affected by pests or diseases. In the case of remote beehive locations and the large number of beehives, the inspection could be labor-intensive. Remote sensors can help beekeepers to monitor beehive colony using sensors, without the need of frequent disturbance due to hive opening. The wireless sensor can be used to monitor various beehive parameters, such as temperature and humidity, beehive weight, CO₂ level, etc (Edwards-Murphy *et al.* 2016, Jiang *et al.* 2016). Measured raw data from sensors is collected and transited remotely to the storage server, using the Internet of Things (IoT) concept (Zabasta *et al.* 2019). IoT devices could be accessible through the Internet, anywhere in the world. Except for data storage, cloud server services provide a wide range of tools for data visualization, data analysis, as well implementation of machine learning algorithms which enables extraction meaningful information from measured data to discover or predict some events in beehives (Dineva *et al.* 2018). Coupled with the Complex Event Processing (CEP) incoming streams of beehive data can be processed using the knowledge acquired by machine learning algorithms. This enables that certain events can be identified automatically and beekeepers can be notified to react on them.

This paper represents the possibility to monitor beehive activity using remote temperature measurements in beehive. The presented system can detect patterns in measured data using machine learning techniques in real-time with the CEP server.

Remote temperature monitoring system

The remote monitoring system consists of the number of wireless sensor nodes (one per beehive) that are able to collect and transmit sensor data to the central node using wireless radio. Data set used in this paper is collected during the year 2017 at beehive located at Wurzburg, Germany (HOBOS, 2019). It consists of temperature measurements, measured in thirteen points throughout the beehive. Dataset also includes inflow/outflow of bees at beehive entrance, relative humidity inside beehive and weight of the beehive.

The activity of the honeybee colony can be monitored remotely in real-time by measurements from temperature sensors. This is especially convenient during winter since field inspections could disturb a bee colony and expose it to cold temperatures. During winter months in December and January bee colony in “hibernation”, where worker bees form a cluster in order to maintain the temperature of around 35°C within the center of the cluster. Heat is produced by worker bees at the cluster perimeter, by contractions of their wing muscles. During this period, the bee colony is brood-rearing new workers to maintain the number of bees, due dying of aging “winter bees”. During mid-winter, all temperatures within beehive fluctuate according to the fluctuation of outer air temperature t_{13} , as shown on Figure 1.

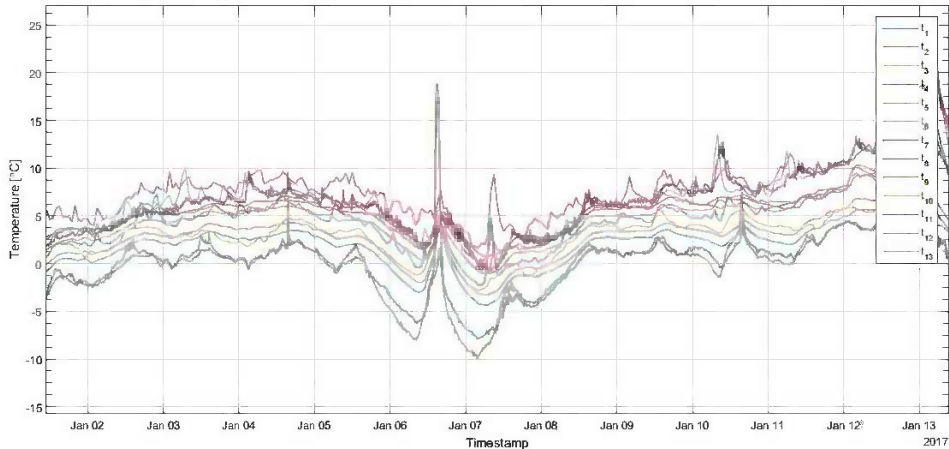


Figure 1. Inactive period in mid-January

During winter due dying of aging “winter bees”, colony reaches the minimum to around ten thousand worker bees. In late-winter, the colony starts brood-rearing a large population of worker bees for early spring pollen collection. This can be seen as the increase in temperatures near the center of the beehive, as shown in figures 2 and 3.

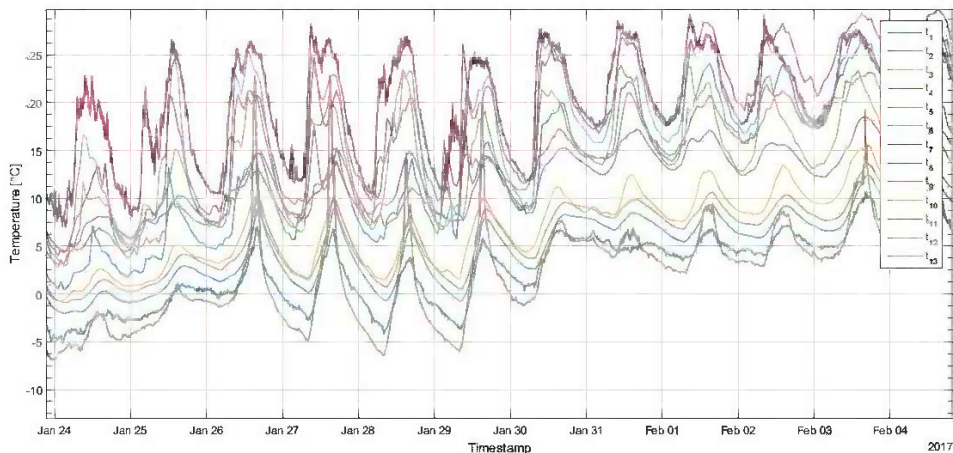


Figure 2. Mid-winter brood rearing in February

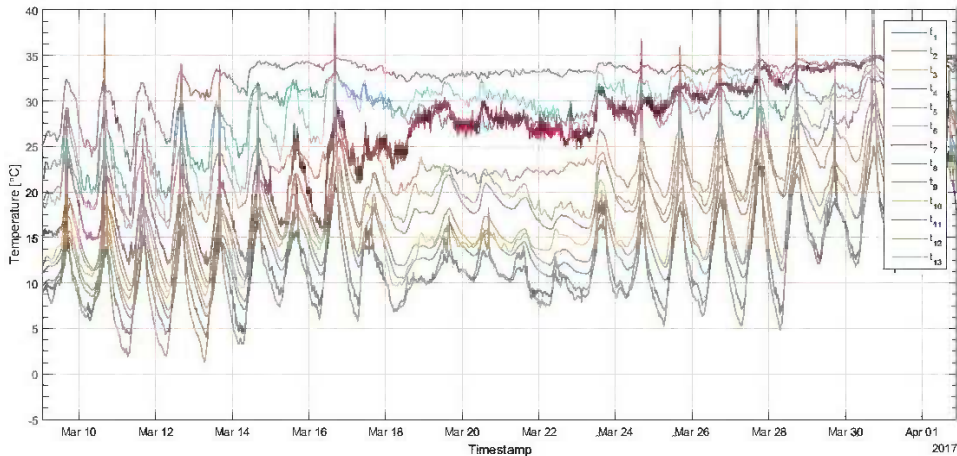


Figure 3. Mid-winter brood rearing in March

When beehive colony reaches its maximum in early spring, temperatures in the center of the beehive start to be regulated around 35°C, no matter of outer temperature, as shown in figure 4.

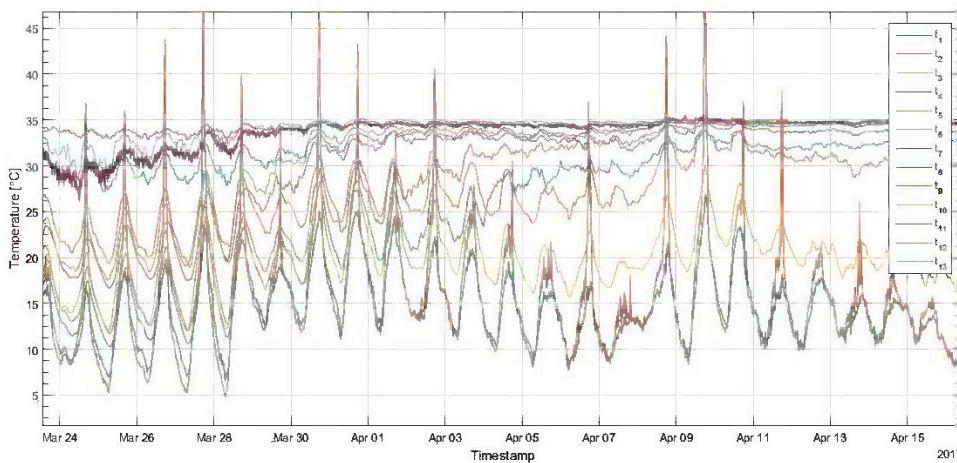


Figure 4. Start of temperature regulation from early spring

Worker bees regulate the temperature in the beehive using water evaporation, throughout the late spring and summer. During this time bees collect nectar and pollen to prepare for the winter hibernation period, which starts in late autumn as shown in figure 5.

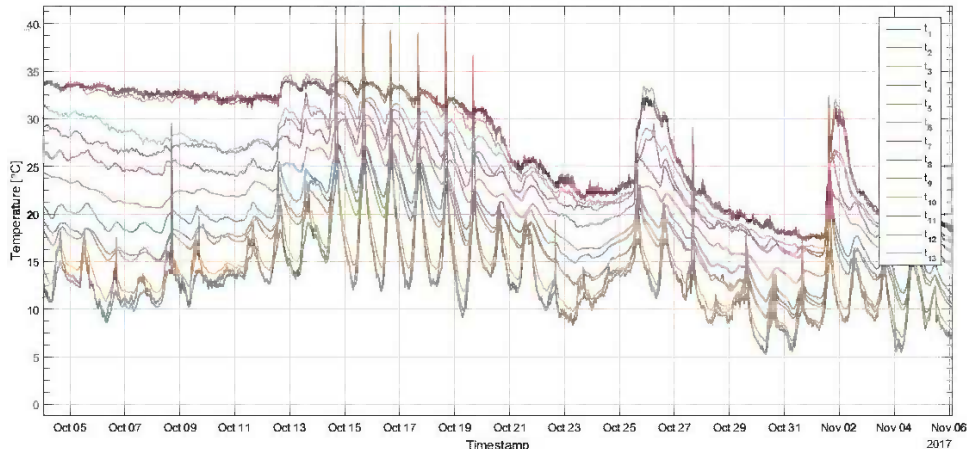


Figure 5. Start of beehive hibernation in late autumn

Based on the temperature measurements from the healthy beehive colony, the model can be build which takes into account outside temperature, the current date and time of the measurements, and expected state of the beehive colony (hibernation, brood-rearing, nectar collection and etc.). When new temperature data is acquired model is comparing this data to expected behavior, and any deviations can be reported as events or anomalies.

Sensor devices were used for detecting events in the real world and they are equipped with the wireless communication modules for transmitting events data towards the server on the Internet. New technology achievements in embedded devices and telecommunication enable emergent of a new concept known as the Internet of Things (IoT). On the base of wireless sensor network IoT found application in many areas of human activity and therefore, more and more data is being acquired on the server. Data in array began to arrive on the server, from multiple sources, requiring activities for their acceptance and storage. Preprocessing of these data could be important to save time and additional recourse utilization if the data were recorded first to the database and then load for subsequent analysis. For the purpose of events processing as they arrived to the server Complex Event Processing (CEP) could be used. Events from individual sensors ware transmitted in time series and could be represented as simple events.

CEP system has the capability to apply predefined rules on simple events from multiple sources and generate new event as a result. This new event would be called a complex event as output from the CEP system and represents an important state or value of data that arrive in streams (Flouris *et al.*, 2017).

A series of events could be processing on CEP systems such as open-source WSO2 CEP server. This CEP server supports event detection patterns, such as filters, time frames, data aggregation and collections of basic events. Also, there is the capability for queries definitions, similar to SQL queries, in order to detect specific events in the input

data stream. Data partitions could be formed in the WSO2 CEP system by grouping data with particular queries. Query execution could be optimized on CEP server within multithreaded, parallel, or distributed environments. WSO2 CEP server is based on Java programming language and it is running on Java Virtual Machine (Perera *et al.*, 2014).

The main component of WSO2 CEP is the Siddhi Java library that is used for accepting data streams, detection of complex conditions, and triggering corresponding actions. WSO2 CEP system could be used efficiently for complex queries with sequence and pattern detection or queries with time windows. This characteristic related to the effective detection of complex events is possible due to its powerful query language. Also, CEP queries can use built-in templates and therefore during the execution they could be dynamically changed. The main part of the CEP system, which forms the basis of the system, is relatively small and can be placed on Edge devices forming the Edge analytic system. This property of the WSO2 CEP system enables the execution of data stream processing on gateway near to the IoT devices (WSO2 CEP, 2019).

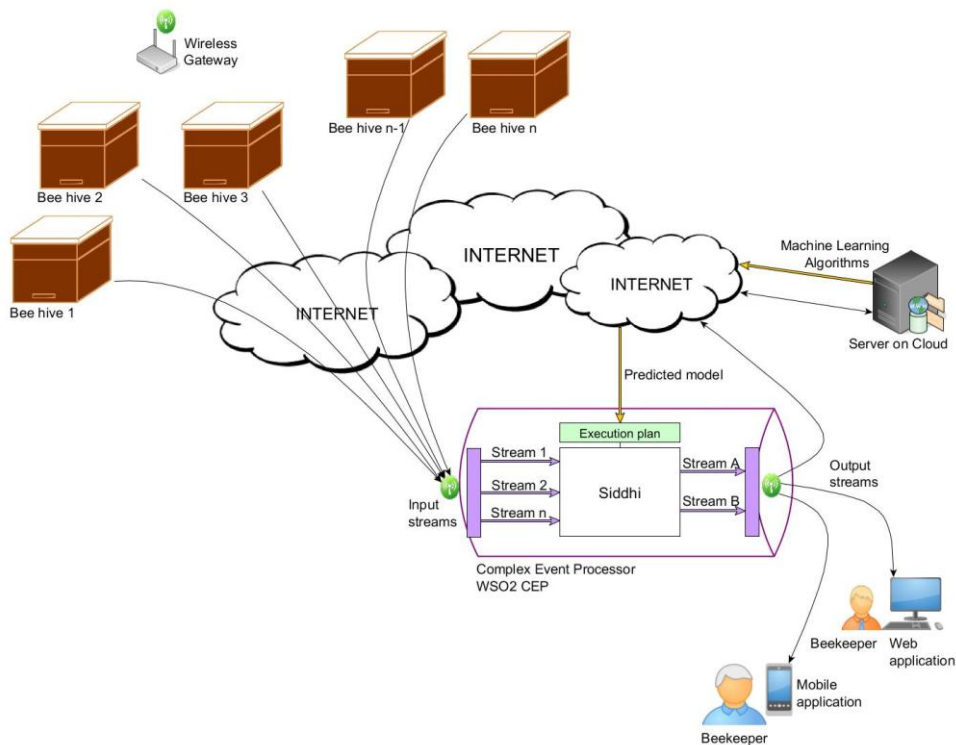


Figure 6. CEP system for detection of information for beekeepers based on predicted model

The model of the CEP system used for detection of new information for beekeepers was shown in Figure 6. Data arrays from beehives were gathering to the server and inputted in the CEP system like event streams. Parameters that characterized conditions in beehives are temperatures from multiple positions inside beehives, flow in and flow out of bees and relative humidity of the air. Measuring results in the past could be used within machine learning algorithms to creating the predicted models. This model could be incorporated to the CEP system so new events that came from beehives could be checked against the expected value and immediately detect the occurrence of anomaly or extreme value of events.

Conclusion

This paper presented the model of the system for remote temperature monitoring of beehive colony using IoT concept, Complex Event Processing enables real-time analytics of received data streams and is able to detect critical events and report it to the beekeeper. This could help in reducing the reaction time of beekeepers and increase chances for a beehive colony to overcome certain types of anomalies with human intervention.

Acknowledgement

The work presented in this paper was funded by grant TR32043 for the period 2011-2019, by the Ministry of Education, Science and Technological Development of the Republic of Serbia.

References

- Aizen M.A., Harder L.D. (2009): The global stock of domesticated honey bees is growing slower than agricultural demand for pollination. *Current Biology*, 19 (11): 915-918.
- Dineva K., Atanasova T. (2018): ICT-Based Beekeeping Using IoT and Machine Learning. In: Vishnevskiy V., Kozyrev D. (eds) *Distributed Computer and Communication Networks. DCCN 2018. Communications in Computer and Information Science*, vol. 919, Springer, Cham: 132-143.
- Edwards-Murphy F., Magno M., Whelan P. M., O'Halloran J., Popovici E. M. (2016): b+WSN: Smart beehive with preliminary decision tree analysis for agriculture and honey bee health monitoring. *Computers and Electronics in Agriculture*, 124: 211-219.
- Flouris I., Giatrakos N., Deligiannakis A., Garofalakis M., Kamp M., Mock M. (2017): Issues in complex event processing: Status and prospects in the Big Data era. *Journal of Systems and Software*, 127: 217-236.
- HOBOS (HoneyBee Online Studies) data, last visited: 11.10.2019., available at https://grafana.biozentrum.uni-wuerzburg.de/dashboards/f/hGca7y7mk/hobos_de.
- Jiang J.-A., Wang C.-H., Chen C.-H., Liao M.-S., Su Y.-L., Chen W.-S., Huang C.-P., Yang E.-C., Chuang C.-L. (2016): A WSN-based automatic monitoring system for the foraging

- behavior of honey bees and environmental factors of beehives. *Computers and Electronics in Agriculture*, 123: 304-318.
- Perera S., Sriskandarajah S., Vivekanandalingam M., Fremantle P., Weerawarana S. (2014): DEBS Grand Challenge: Solving the Grand Challenge Using an Opensource CEP engine. DEBS '14 Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems, India: 288-293.
- WSO2 CEP - WSO2 Complex Event Processor, last visited: 11.10.2019., available at <https://wso2.com/products/complex-event-processor>.
- Zabasta A., Kunicina N., Kondratjevs K., Ribickis L. (2019): IoT Approach Application for Development of Autonomous Beekeeping System. 2019 International Conference in Engineering Applications (ICEA). Sao Miguel, Portugal: 1-6.

DALJINSKI NADZOR AKTIVNOSTI PČELINJEG DRUŠTVA

Uroš Pešović¹, Dušan Marković², Slađana Đurašević¹, Siniša Randić¹

¹Fakultet Tehničkih nauka, Univerzitet u Kragujevcu, Svetog Save 65, 32 000 Čačak, Srbija

²Agronomski fakultet u Čačku, Univerzitet u Kragujevcu, Cara Dušana 34, 32 000 Čačak, Srbija

Rezime

Kolonije evropskih pčela, najznačajnijih insekata oprašivača, izložene su mnogim negativnim uticajima savremenog društva, u obliku zagađenja, štetočina, invazivnih vrsta i poremećaja urušavanja kolonije. Novonastala tehnologija Interneta stvari omogućava daljinsko nadgledanje pomoću bežičnih senzora instaliranim u košnicama. U ovom radu predstavljen je model sistema za daljinsko nadgledanje temperature na raznim tačkama unutar košnice pomoću IoT uređaja. Izmereni podaci se prenose na udaljeni server gde se primljeni tokovi podataka obrađuju u realnom vremenu pomoću složene obrade koja omogućava otkrivanje kritičnih događaja i generiranje alarma koji se šalju pčelaru. Ovi tokovi podataka upoređuju se sa referentnim uzorcima temperature koristeći algoritme mašinskog učenja, koji računarima daju mogućnost da nauče da detektuju događaje bez eksplicitnog programiranja. Ovaj sistem može značajno smanjiti vreme reakcije pčelara i povećati šanse da pčelinje društvo prevaziđe određene vrste anomalija ljudskom intervencijom.

Ključne reči: Pčelinja društva, Internet stvari (IoT), mašinsko učenje, kompleksne obrade događaja.