

UDC: 638.147.7:004.738.5
COBISS.SR-ID: 228212748
Original research paper



Acta Agriculturae Serbica, Vol. XXI, 42(2016) 135-144

Decision support system for temperature monitoring in beehives

Dušan Marković¹, Uroš Pešović², Slađana Đurašević², Siniša Randić²

*¹University of Kragujevac, Faculty of Agronomy, Čačak,
Cara Dušana 34, 32000 Čačak, Serbia*

*²University of Kragujevac, Faculty of Technical Sciences, Čačak,
Svetog Save 65, 32000 Čačak, Serbia*

Abstract: European honeybee colonies are the most important pollinator insects and source of honey and other useful products. Honeybee colonies today face new diseases and pests as well as pollution which threaten their survival and endanger whole food production which relies on honey bee pollination. Internet of Things (IoT) technology enables integration of wireless sensors inside beehives to enable remote monitoring of various beehive parameters from remote location using Internet. Detection of certain critical events in beehive is hard to be explicitly program due to complex dependence between multiple input parameters. Machine learning algorithms give computers the ability to learn to detect these events without being explicitly programmed. Detection of these event from streams of data collected from IoT sensors is possible using Complex Event Processing (CEP) which applies machine induced knowledge do detect and warn beekeepers about certain events in beehive.

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

Introduction

Start of beekeeping dates to dawn of civilization where beehive colonies are used as source of honey, wax and propolis. Today, these natural products are important good for nutrition, medicine and industry. Honeybees are the most important pollinator insects for wide range of cultivated crops. The EU parliament noted in resolution T6-0579/2008, that 79% of the world's food supply is dependent on honey bees (Aizen and Harder, 2009). Honeybee population is affected by various side-effects of modern civilization, such as pollution, climate changes, invading pests and etc. With ever growing human population and food production, it is vital to protect beehive colonies and enable them to maximize their pollination activity as well as honey productivity.

Honeybee colonies require almost daily inspections by beekeepers, since occurrence of beehive diseases and pest outbreaks are very dynamic. Remote monitoring using wireless sensors, placed inside beehives, enables beekeepers clear insight into bee colony health, without need of constant disturbance caused by manual inspection. Various types of parameters can be monitored inside beehive, such as temperature, relative humidity, CO₂ level, weight of beehive, audio and video recordings of colony activity and many others. Wireless communications between sensors enables formation of communication networks, so all beehives can be monitored at once and results can be analyzed and compared.

Remote monitoring of beehive parameters can be based on Internet of Things (IoT) concept. IoT represents the idea of connecting physical objects that contain sensing, networking and computing capabilities, with other objects and services through the Internet. IoT enable possibility that every device with appropriate capability could be accessible through the Internet, anywhere in the world. IoT devices collect raw sensing data from beehives and send them to the server platform for further analysis. Machine learning enables extraction meaningful information from measured data to discover or predict some events in beehives. Complex Event Processing (CEP) enables processing of incoming streams of raw data from beehives using knowledge acquired by machine learning, on which basis action are taken and notification is send to users or react on them.

The aim of this work was to present model of IoT based beehive monitoring system. Such system is able to discover dependences between measured data using machine learning techniques and detecting events in real time with CEP server.

Basis for the system development

WSN devices or IoT devices need to be programmed in such a way to perceive and understand surrounding events in similar way to humans, in order to make correct decisions. Usually such knowledge could not be explicitly programmed because of complex dependence between input and output data, so WSN devices need to rely on intelligent algorithms. These machine learning algorithms give computers the ability to learn without being explicitly programmed. Process of machine learning starts with introduction some data set containing input and output parameters which is used to train a model (data of parameters and events which are subject of knowledge extraction). When model finds dependence between input and output data, it is validated using independent data set, after which it can make predictions on new data. In this way we can create computer program that gives intelligent output to a different inputs which are similar to trained data. Machine learning can be used in prediction and classification problems. Prediction is used to determine output value based on set of input parameters, such as height of beehive cluster in function of set of measured temperatures. It is used to obtain set of regression coefficient which link set of inputs with corresponding outputs of trained data set. Regression coefficients can be obtained iteratively using gradient descent algorithm or using matrix manipulation of normal equation. On the other hand classification is used to determine discrete output of some event, such as is the temperature in beehive low or high enough to cause damage to beehive colony. Classification algorithms rely on logistical regression algorithms where classification problem is presented using logistic (Sigmoid) function (Nilsson, 1998).

Complex Event Processing (CEP) represents emerging technology that could be used for detecting complex conditions in streams of events. Systems that operate using event-base applications have functionality to observe primitive events that were detected in external environments and to interpret them to reveal composite events. Also, systems have ability to send notifications about identified events to the components responsible to react on this detect events value by sending some sort of notification or perform defined action. The process of identifications of composite events is executed by Complex Event Processing Engine. Existing CEP engines have function to timely process generic data using queries that continuously perform calculation of new result on data that arriving to the system (Cugola and Margara, 2012).

One of the open source Complex Event Processing server that is lightweight and easy to use is WSO2 CEP. This server could be used to detect meaningful events in the stream of events, analyzing them and react in the real time. The main part of the WSO2 CEP architecture is Event receivers, Event streams, Event processors and Event publishers. Event receivers have role to receive events

while they coming to the server, then Event streams contain set of attributes with specific types providing structure on which basis the events is processed. Event processing represents the core processing unit of the CEP and where execution plans would be performed and processed event according to define logic. And Event publishers give events to the external systems and store data for further analysis (WSO2 CEP, 2016).

WSO2 CEP as an open source engine is used for real-time analysis of arrival events. It supports patterns for event detection such as filters, windows, joins, aggregation and more advanced features like partitions of events and mapping database values to events. Users define SQL like queries that could be used for detecting particular events from input streams of data. Users also could combine queries and create complex one where a concept of the event partitions may be applied. In that case WSO2 CEP data were grouped within queries forming partitions where CEP server could optimize execution in threads and allow distributed processing. Engine of WSO2 CEP is written in Java and it is running on Java Virtual Machine (Perera *et al.* 2014).

In the book chapter Simmhan and Perera (2016) importance of Stream Processing and Complex Event Processing systems was presented in addressing needs of IoT applications for analyzing gathering data. Scenarios for IoT domains that have big data require specific techniques and analytics so that they could operate in effective manner. In this chapter there was focus on presented IoT domains such as smart cities, smart agriculture, smart health and smart logistics.

Integration of Internet of Things and Complex Event Processing in Healthcare informatics was introduced in article Sheriff *et al.* (2015), where proposed framework has potential to perform analytics across different sub domains of healthcare.

Sensor networks are also source of high volume of data that are transmitting continuously to the server side. Data represents events that form event stream and require adequate processing. Dunkel (2009) and Taylor *et al.* (2011) propose an appliance of CEP in real time analysis of complex event streams where ontology is used as basis for definition of event type structural properties and constraints.

Complex Event Processing is also combined with Machine Learning as based approach for reducing computational time of closed-loop weather monitoring (Chandrathilake *et al.* 2016). CEP engine is used to filter input weather data stream to extract hazardous weather events. These filtered data are then forward to machine learn system. In this case complex weather forecasting models are run only for potentially hazardous events and reduce computation nearly by 75% because using only small set of filtered data.

Model of the beehive detection events system

Winter is particular harsh period for beehive colony, since colony need to survive without any external inputs, relying only onto honey collected prior winter. Worker bees use honey as food source to produce heat using their muscles contractions. They encircle queen bee to form a cluster in order to keep temperature in the center of the cluster between 32-37°C. The temperature of the cluster decreases toward the outside, so bees on the outside when they get so cold that they must rotate to the inside. Honey exploitation leaves bee colony with just enough honey to survive winter, so it important to monitor bee hive activity in order to eventually provide them with supplementary sugar syrup. Inspection of beehives during winter is not recommended because beehive colony can cool too much and die out. Remote temperature monitoring provides insight into beehive activity during winter, so beekeepers can detect reduce activity of the beehive and start providing supplementary food (Edwards-Murphy *et al.* 2016).

Temperature in beehive is measured using several low power DS18B20 digital thermometers. These digital thermometers have measurement range from -55 °C to +125 °C with accuracy of ± 0.5 °C in range from -10 °C to +85 °C. The DS18B20 communicates over a 1-Wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. Each DS18B20 has a unique 64-bit serial code, which allows multiple DS18B20s to function on the same 1-Wire bus. Thus, it is simple to use one microprocessor to control many DS18B20s distributed over a large area. These sensors do not require external components and have and has an alarm function with nonvolatile user-programmable upper and lower trigger points.

Temperature inside beehive is measured on several measuring points, distributed vertically, starting from bottom plate to roof cavity. Based on temperature measurements on different heights, as shown in figure 1, we can determine location of bee cluster.

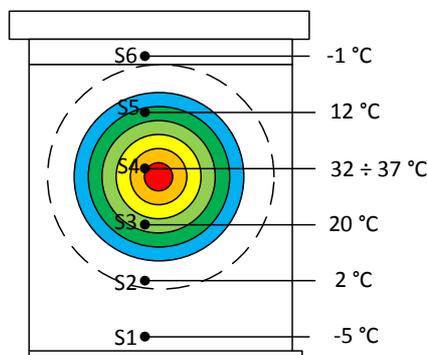


Figure 1. Placement of temperature sensors in beehive

Studies were shown that minimal temperature in beehive should be at least 5°C and temperature in the center of beehive cluster should not be above 36 °C. Measured values from temperature sensors from beehive are sorted and minimum and maximum temperatures are chosen. Figure 2. shows classification set of training data with minimum and maximum temperature boundaries and minimum-maximum boundary (minimum cannot be greater than maximum value). Temperatures which lie outside boundary triangle are extreme for beehive colony and alarm should be sent to beekeeper.

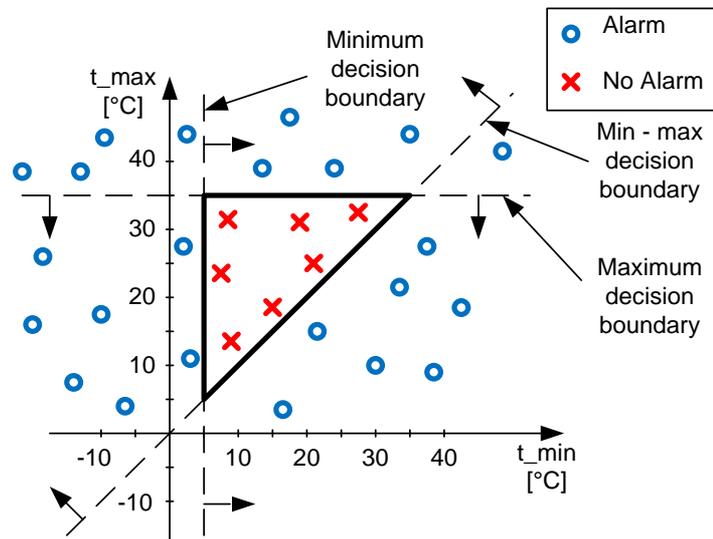


Figure 2. Temperature classification using machine learning

Measured temperature with IoT device, based on ESP8266 module, was send to the remote server. On the server side WSO2 CEP would be running for processing streams of data that are continuously transmitting from devices. Event Flow of the applied processing events was shown in Figure 3. Data was sending with HTTP protocol and Event Receiver was used to accept data and form the Event Stream from them.

CEP Event Flow

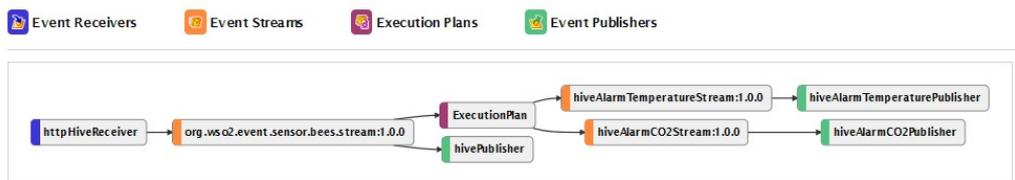


Figure 3. Event flow for processing data from hive sensors

Attribute values from stream are using by defined queries in the Execution plan, which represent the main part of the CEP server responsible to process incoming events in streams and gave meaningful results also in the form of stream that could be used in other part of the system. Queries that could be used for detecting alarming states according to Edwards-Murphy *et al.* (2016) is shown in Example 1.

Example 1. Filtering query for detecting alarm states

```
from hiveStream [min_temperature < 5 and max_temperature > 36 and
min_temperature > max_temperature]
select meta_timestamp, meta_sensorid, meta_sensorName, temperature
insert into hiveAlarmTemperatureStream;
```

Model of the system that uses IoT concept for monitoring beehives using devices based on ESP8266, with WSO2 CEP as Complex Event Processing server and application for Machine Learning is presented in Figure 4.

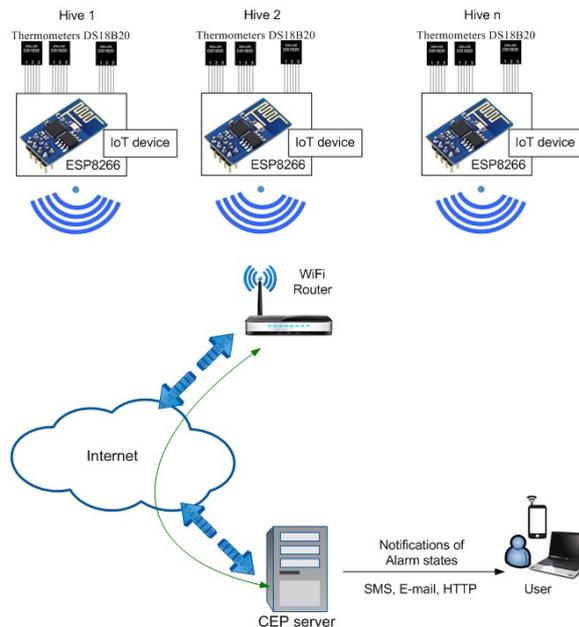


Figure 4. Model of system for Complex Event Processing of IoT data from beehives

After detecting events of the interests by CEP engine notification to the users could be sent by SMS or E-mail. Result's stream in the form of Event Publisher as output from processing queries could be sent, in the appropriate form, to the other components of the system that would start further action.

Conclusion

Model of the system for remote monitoring of beehive parameters based on IoT concept with detection of critical events was presented in this paper. Importance of such a system is in monitoring conditions of honeybee colonies and prevention of bad condition values that could be threat to their survival. The limit value of these parameters could be determined by machine learning algorithms and then in the running process they could be used for detection of events that could cause unfavourable condition. Process of real time analytics using Complex Event Processing could be extend adding more streams of events that could represent stored predefined values or data from monitoring area. In this respect further system development would be toward integration of various sensors with existing IoT devices and their inclusion in more complex calculation for obtaining optimal conditions. The survival of beehive colonies is one of the priorities related to the area of agriculture due food production is associated with honey bee's pollination.

Acknowledgement

The work presented in this paper was funded by grant TR32043 for the period 2011-2016, 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.
- Chandrathilake H.M.C., Hewawitharana H.T.S., Jayawardana R.S., Viduranga A.D.D., Dilum Bandara H.M.N., Marru S., Perera S. (2016): Reducing computational time of closed-loop weather monitoring: A Complex Event Processing and Machine Learning based approach. *2016 Moratuwa Engineering Research Conference (MERCon)*, Moratuwa, 78-83.
- Nilsson J. N. (1998), Introduction to machine learning, *Stanford University, CA, USA*
- Cugola G., Margara A. (2012): Complex event processing with T-REX. *Journal of Systems and Software*, 85(8): 1709–1728.
- Dunkel J. (2009): On complex event processing for sensor networks. *2009 International Symposium on Autonomous Decentralized Systems*, Athens, 1-6.
- Edwards Murphy F., Magno M., Whelan P., Vici E. P. (2015): b+WSN: Smart beehive for agriculture, environmental, and honey bee health monitoring - Preliminary results and analysis. *2015 IEEE Sensors Applications Symposium (SAS)*, 1-6.

- 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.
- 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.
- Sheriff C. I., Naqishbandi T., Geetha A. (2015): Healthcare informatics and analytics framework. *2015 International Conference on Computer Communication and Informatics (ICCCI)*, India, 1-6.
- Simmhan Y., Perera S. (2016): Big Data Analytics Platforms for Real-Time Applications in IoT. *Big Data Analytics: Methods and Applications*, ed. Pyne S., Rao, B.L.S.P., Rao S.B., Springer India, 115-135.
- Taylor K., Leidinger L. (2011): Ontology-Driven Complex Event Processing in Heterogeneous Sensor Networks. *The Semantic Web: Research and Applications: 8th Extended Semantic Web Conference, ESWC 2011, Heraklion, Crete, Greece*, ed. Antoniou G., Grobelnik M., Simperl E., Parsia B., Plexousakis D., De Leenheer P., Pan J., Springer Berlin Heidelberg, 285-299.
- WSO2 CEP, <https://docs.wso2.com/display/CEP420/WSO2+Complex+Event+Processor+Documentation>, last visited: 02.12.2016.

SISTEM PODRŠKE ODLUČIVANJU PRI MERENJU TEMPERATURE U PČELINJIM KOŠNICAMA

Dušan Marković¹, Uroš Pešović², Slađana Đurašević², Siniša Randić²

*¹Univerzitet u Kragujevcu, Agronomski fakultet u Čačku
Cara Dušana 34, 32000 Čačak, Srbija*

*²Univerzitet u Kragujevcu, Fakultet tehničkih nauka, Čačak,
Svetog Save 65, 32000 Čačak, Srbija*

Rezime

Evropska medonosna pčela je najvažniji insekt-oprašivač u svetu kao i izvor meda i niza korisnih proizvoda. Današnje pčelinja društva su izložena nizu invazivnih vrsta i bolesti kao i zagađenju koje ugrožavaju njihov opstanak kao i značajan deo proizvodnje useva koji se oslanjaju na pčele kao oprašivače. Sa napretkom mikroelektronike i komunikacija, tehnologija Interneta Stvari (IoT) omogućava integraciju bežičnih senzora unutar košnica koji omogućavaju praćenje niza parametara pčelinjeg društva sa udaljene lokacije korišćenjem Interneta. Detektovanje određenih kritičnih događaja u pčelinjem društvu vrlo je teško eksplicitno programirati, zbog složene zavisnosti između višestrukih ulaznih parametara. Tehnike mašinskog učenja omogućavaju računarima da uče da detektuju bez potrebe da prethodno budu eksplicitno programirani. Detekcija ovih događaja iz niza sirovih podataka koji se prikupljaju sa niza IoT senzora, moguća je korišćenjem kompleksne obrade događaja (CEP) koji primenjuje stečeno mašinsko znanje kako bi detektovao ove događaje i upozorio pčelare.

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