Smart Agricultural Technology 6 (2023) 100331

Contents lists available at ScienceDirect

Smart Agricultural Technology

journal homepage: www.journals.elsevier.com/smart-agricultural-technology

BeeLive: The IoT platform of Beemon monitoring and alerting system for beehives

Abdelbaset S. Hamza *, Rahman Tashakkori, Bejamen Underwood, William O'Brien, Chris Campell

Department of Computer Science, Appalachian State University, Boone, NC, 28607, USA

ARTICLE INFO

Editor: Spyros Fountas

Keywords:
Beehive
Honey bees
Internet-of-Things (IoT)

ABSTRACT

Monitoring honey beehives is mainly done manually by beekeepers to evaluate the health of their hives and determine their growth and yield. With the emergence of Internet of Things (IoT) devices and tools, there have been some efforts in recent years to automate such monitoring. This can significantly benefit beekeepers as they can obtain critical data and insight into their hives' health and performance more regularly. In this paper, we use IoT devices and the Thingsboard dashboard to track the status of 28 honey beehives installed in the Western region of North Carolina as part of the Appalachian Multi-Apiary Informatics System (AppMAIS) project. In order to acquire data from a beehive, humidity and temperature sensors, as well as a microphone, have been placed inside the hives. In addition, a video camera has been placed at the top of the hive's entrance to obtain video recordings of the bees entering and leaving the hives, and a scale is placed under the hive to report the weight. The data collected from the sensors and peripherals installed in each of the AppMAIS hives are sent to the Thingsboard dashboard for management and visualization. In this paper, we report on our success with Thingsboard IoT tool to monitor honey beehives and take advantage of their capabilities to interact with the devices as needed to adjust operational parameters.

1. Introduction

In recent years, beekeepers worldwide have seen drastic losses of a large number of their hives due to several causes that at some point were collectively referred to as Colony Collapse Disorder (CCD) [1]. In recent years, this loss has been as high as 50% in some regions. The cause of such significant losses is not yet fully understood, but it is believed to be a complex interplay of factors, including pests and pathogens, poor nutrition, pesticide exposure, environmental stressors, lack of genetic diversity, insect predators (i.e., hornets) and poor beekeeping practices. The great loss in honey bee colonies has significant implications for the beekeeping industry and the broader ecosystem, as honey bees are critical in pollinating crops, in general, and almonds, in particular. They pollinate one-third of our entire food supply. Crops from almonds to apples to even broccoli rely on honey bees, and it is estimated that they contribute at least 20 billion dollars to our economy [2]. In addition, a decline in honey bee populations can also harm wild pollinators, leading to a decline in biodiversity [3,4].

Manual inspection, a common practice in conventional beekeeping, has been identified as a source of human error, particularly among novice beekeepers. Additionally, the physical examination of hives can disrupt the internal equilibrium of the colony and increase stress among the bees, resulting in decreased productivity. Furthermore, manual inspections can present logistical challenges, particularly in rural and mountainous regions where hives are dispersed over a vast geographical area.

Utilizing remote Electronic Beehive Monitoring (EBM) systems represents a novel approach to monitoring beehives. These systems involve the installation of sensors within the hive that can collect and transmit data remotely for analysis. The sensors can include but are not limited to temperature measurements, humidity, weight, audio, and visual data. The use of EBM systems has been explored in several studies, and various EBM system designs have been proposed in the literature [5]. These systems can be used for research purposes to determine the overall health of the hive through the data, such as the beehive audio [6], collected inside the hive. It can also be used for commercial purposes, such as Precision Beekeeping (PB), to assist beekeepers in

* Corresponding author.
E-mail address: hamzaas@appstate.edu (A.S. Hamza).

https://doi.org/10.1016/j.atech.2023.100331
Received 17 July 2023; Received in revised form 26 September 2023; Accepted 26 September 2023
Available online 4 October 2023

managing workloads and maximizing the productivity of bees. However, it is worth noting that most current monitoring systems require significant manual setup or frequent visits to retrieve data from the hives. Furthermore, only a limited number of research platforms have the capability of continuous data collection utilizing internet connectivity, and the quality of audio and video recordings captured by previous efforts has been of limited quality.

A previous study by Kulyukin et al. introduced a successful multi-sensor monitoring system called BeePi [5]. This system effectively gathers temperature, audio, and video data, which is then stored on a 25G SD card integrated into the Pi device. Since its initial deployment, the BeePi system has undergone continuous development. Inspired by the BeePi system, subsequent research has explored various techniques for analyzing and quantifying bee movements in video recordings of omni-directional bee traffic [7]. Furthermore, researchers have investigated the application of Digital Particle Image Velocimetry to estimate bee motion [8–12].

The Internet of Things (IoT) is a framework in which data is collected, processed, and used to deliver insights and trends to stakeholders across various applications and business verticals. It involves the integration of internet-connected devices into everyday objects and environments, with the potential to revolutionize the way businesses and individuals interact with the world. For example, IoT devices can be found in smart homes, cities, industrial automation, and healthcare, bringing cost savings, increased efficiency, and improved safety. Similarly, an EBM can be extended to become an IoT-based EBM where data collected from apiaries and hives is used to perform apiary control without interfering with its operation [13,14].

Recently, the application of the IoT in beekeeping has been the subject of research. A number of conceptual solutions have been proposed to track temperature, humidity, mass, and other data related to the health and safety of the system [15–19,13,20–26]. However, most current work in this area focuses on the perception and networking layers of the IoT stack, with less emphasis on upper layers such as middleware development and business layer. As a result, there is a need to establish apiary monitoring and management as a viable business vertical within the IoT domain.

In [27], we presented a detailed overview of the design and implementation of Beemon, a data collection and monitoring system. The system is designed to automatically capture sensor data such as temperature, humidity, and weight and transmit them to an IoT dashboard using MQTT protocol. Additionally, the system captures video and audio recordings at the hives' entrance, which are transmitted to a remote server for analysis and further research. Beemon operates in an outdoor apiary environment and enables near real-time data collection.

The Appalachian State Multipurpose Apiary Informatics Systems (AppMAIS) project is a 1.1 million dollar initiative led by the Department of Computer Science at Appalachian State University and funded by the North Carolina General Assembly Research Opportunities Initiative (ROI). AppMAIS is an interdisciplinary endeavor to investigate the health, development, and genomic diversity of honeybee hives in North Carolina to better understand and effectively prevent CCD. The project comprises 28 hives distributed across various North Carolina's High Country regions. Each of these hives is equipped with the Beemon system, which continuously transmits data to servers for monitoring through our IoT dashboard powered by Thingsboard [28] and for research and data visualization purposes.

This paper presents the IoT platform developed for the AppMAIS project, which enables the monitoring of 28 hives distributed over a vast geographical area in a mountainous region. We will discuss the setup, challenges, and lessons learned during the implementation of the system. Additionally, we will provide details on the multi-level dashboard and rule engine components developed to send automated alerts in case of sensor failure or changes in the beehive's behavior.

The remainder of this paper is organized as follows. In Section 2, we discuss work in the literature related to IoT-based EBMs. We present the Beemon system and the AppMAIS project in Section 3. We dedicate Section 4 to discuss the IoT component of the AppMAIS project, BeeLive. In particular, we discuss the approach used to create the system and demonstrate the operation of the developed IoT system. This section is followed by a discussion of the IoT system in Section 5. Conclusions are discussed in Section 6.

2. Related work

The deployment of IoT solutions for beekeeping has seen limited implementation. Although IoT and electronic remote beehive monitoring are well established, only some projects integrate these domains. For example, in [25], Lyu et al. presented a smart beehive system that monitors temperature, humidity, weight, attitude, and GPS location using the General Packet Radio Service (GPRS) network. The data is then transmitted from the beehive to a monitoring center, where an operator can review the data and determine the status of the hive. In case of abnormalities, the operator can send alerts to the beekeeper. However, this approach relies on the presence of an intermediary at the monitoring center, which can be problematic. To address this issue, developing a dashboard that provides direct access to the data while incorporating an automated warning system is necessary [29]. In [30], Chamaidi et al. present IOHIVE, an IoT-based platform that helps beekeepers keep track of the vitals of the hive with respect to temperature, humidity, and weight. Most recently, Cota et al. presented BHiveSense [31], which is a beehive monitoring system built on IoT and microservices and includes various components such as a low-cost hive sensing prototype, a REST back-end API, a web application, and a mobile application. The authors aim to enhance the integration and sustainability of beekeeping activities by addressing interoperability, scalability, agility, and maintenance issues, ultimately delivering an efficient beehive monitoring system.

Furthermore, a few commercial systems and dashboards are currently available for beehive monitoring. In [32], Dineva and Atanasova reviewed several existing dashboards and evaluated their pros and cons. The systems discussed include Arnia, Beeyard (formerly ApiS), Pollenity (formerly Bee Smart Technology), and HiveTracks. In addition, the authors provided an overview of the general structure and data flow in an IoT-based bee monitoring solution. Other existing solutions include BuzzBox and its OSBeehives Application [33], ApisProtect [34].

The Arnia system, based in England, requires the user to purchase an in-hive sensor, gateway, and bar scale for weight tracking. The sensors include microphones, temperature and humidity, light, accelerometers, and E-compasses. However, each apiary requires its gateway, which requires a data subscription, resulting in potentially high costs. The Arnia dashboard has an attractive layout but offers limited customization options.

Beeyard, based in Portugal, provides the ability to monitor hives through an app or desktop and includes features such as apiary and beehive data monitoring, records management, inventory tracking, task scheduling, and alarms. However, no information regarding their hardware or software costs is provided.

Pollenity, formerly Bee Smart Technology, offers two solutions for monitoring hives: the Beebot and the uHive. The Beebot is a box that connects to a single frame within the hive and measures temperature, humidity, and sound, uploading the data to the cloud for display on their BBoard dashboard. The uHive is a pre-installed sensor hive. A monthly subscription fee is required to use the BBoard dashboard, and a separate Beebot must be purchased for each hive at the cost of $189.

HiveTracks, based in the United States, is not an IoT-based system but provides personalized tasks and task management based on the user's environment, location, and goals.

The biggest challenge with the mentioned IoT solutions is the cost of the services and the inability to modify the dashboards to the consumer needs.

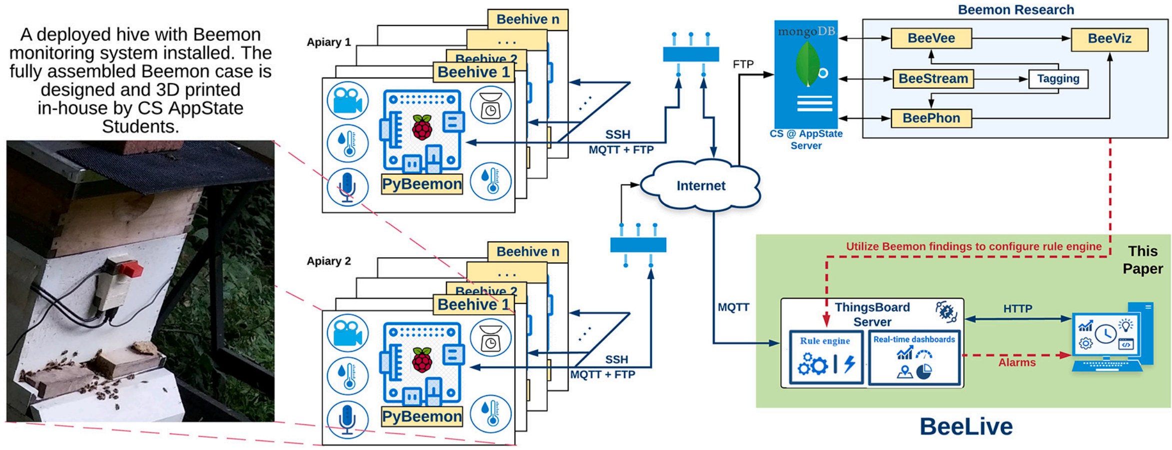

Fig. 1. Overview of the AppMAIS Project and its BeeLive Component.

3. AppMAIS overview

Fig. 1 shows an overview of how data flows in our IoT solution for the AppMAIS project. The data source, in our case, is the beehives. Sensors need to be located at the data source to collect the data for beehives; data on temperatures, humidity inside and outside the hive, the hive's weight, bee counts, and traffic are important for a beekeeper to determine the status of a beehive. A gateway is needed to connect the user to the cloud and the cloud to store the data. Once data has been stored on the cloud, the data can be processed and displayed using a dashboard. The data can also be analyzed and used to feed a rule engine to generate alarms and messages, providing a beekeeper with in-depth insights into the hive's performance.

3.1. Beemon

Beemon is a project that started in the Department of Computer Science at Appalachian State University in 2016 [35–40,27,41–45]. The goal of the Beemon project was to develop a low-cost system that can monitor beehives. The initial system utilized a surveillance camera [36]. However, the most recent version is taking advantage of the computing power and capabilities of the Raspberry Pi4 system to collect various sensor data as well as audio and video recordings.

The project AppMAIS is a project in which the Beemon system has been successfully deployed in 31 hives in the western region of North Carolina across four counties and Belgium. By the time of submission, the Beemon system was utilized by the AppMAIS project for about 18 months for the initial 24 hives and about six months for the additional seven new hives. The system has reliably scaled up and produced data during this period with little interruption.

An example hive can be seen in Fig. 1. Also, this figure shows an array of Beemon systems collecting data and sending them to our server for analysis. Each hive has its microcontroller connected on the outside (see Fig. 2). For this project, the microcontroller used is the Raspberry Pi 4B (RPI). Each RPI has an Adafruit AM2302 sensor connected to record temperature and humidity from inside the hive. Each RPI also has a Raspberry Pi Camera Module V2 connected to record video at the hive's entrance and a UMIK-1 microphone connected to record audio inside the hive. Each of the 28 hives is equipped with a scale designed in our lab and directly transfers its weight reading to the FPGI of the RPI. This data is recorded every five minutes and then uploaded to a server in our department for storage and further analysis.

Our initial challenge revolved around remotely managing the beehives, including making necessary changes and updates. We successfully addressed this challenge by using two key tools: Jenkins for software updates and Wireguard for secure remote access. The details of our solution are discussed in detail in [43].

Fig. 2. Hardware located outside of Beehive.

The data currently sent to Thingsboard is the hive's temperature, humidity, and weight, as well as the file size of the one-minute video that is recorded at the hive's entrance. Most of the data is self-explanatory except for the video file size. The camera that is used in this setup provides entropy encoding. Entropy Encoding is performed on images, or in a video's case, a collection of images, and will compress the image for more efficient-less storage memory-storage. Entropy encoding will allow smaller bits to represent a pixel [46]. In the case of the video recorded, the less activity that happens within the camera frame, the smaller the file size. This gives an idea of how active the bees currently are. Once the data has been collected, it is organized inside a comma-separated value (CSV) file and sent as telemetry data from a Python program on the RPI to Thingsboard.

As pointed out earlier, the hive has several sensors connected to it, collecting various types of data. The sensors are then connected to a gateway, in our case, a Raspberry Pi, that connects the hive to the network. The gateway can support many different types of sensors, whether the sensor is wireless or hard-wired [47]. The biggest job for a gateway is the data processing that has to happen before the data can be delivered to the cloud. In our project, all the collected data is added to a CSV, as mentioned above, and before the data can be delivered to Thingsboard, the data has to be converted from the Python code being used to JSON. One of the main goals for this project is cost reduction, and due to the low cost of RPIs, it was the best choice for a gateway for this project.

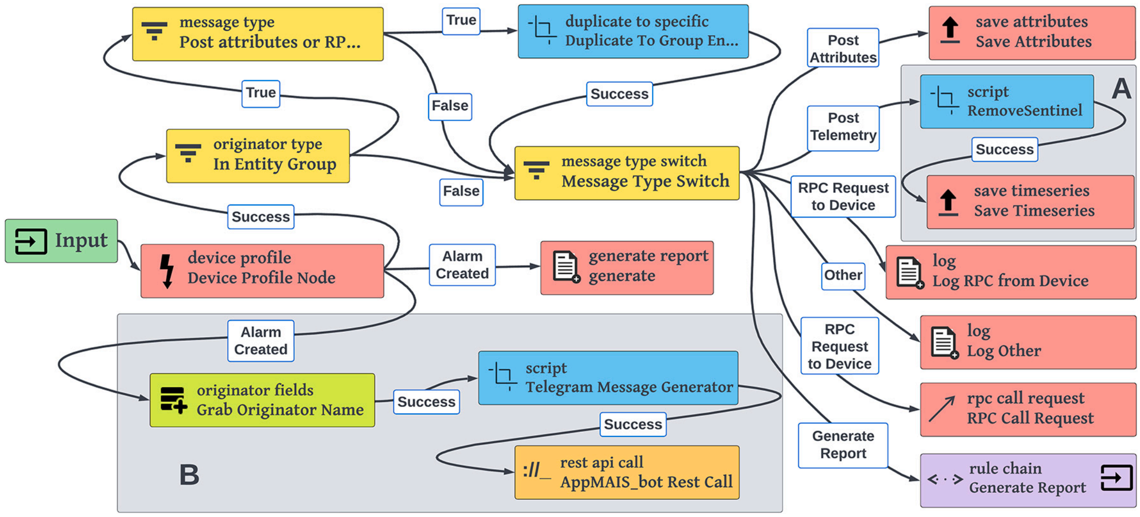

Fig. 3. The root rule chain currently used in BeeLive. Red and blue ovals added for emphasis.

The last step in this IoT life cycle is to send the data from the gateway to the cloud. The primary function of the cloud is to deliver the data to the end consumer. IoT devices have very little storage and low processing capabilities, whereas cloud environments can offer ample storage and processing power. This is why it is crucial to combine IoT with cloud computing [48]. As stated, data is sent to the cloud using the MQTT protocol. MQTT comprises three main components, Publisher, Subscriber, and Broker. The publisher and subscriber are known as the client. The client will always establish the connection to the server, the Broker. The Broker controls the distribution of information by filtering and deciding who gets the messages [49].

4. BeeLive

The primary objective of this paper is to investigate the BeeLive component of AppMAIS, following a comprehensive understanding of AppMAIS. The BeeLive component encompasses the dashboards and the rule engine responsible for the alert generation and early detection through the collected data. Data is gathered from all hives currently utilizing AppMAIS. BeeLive presents it on the dashboards regarding temperature, humidity, weight, and the file size of the latest video recording. In addition, the rule engine includes multiple rule chains that notify users of issues detected. These comprise temperature, humidity, and timeout alarms, which indicate the disconnection of a device. BeeLive operates on the open-source IoT platform known as Thingsboard.

4.1. ThingsBoard

ThingsBoard is a company that specializes in developing software products for the Internet of Things (IoT). Its flagship platform, which operates under the ThingsBoard umbrella, is an open-source project allowing users to create customizable dashboards and widgets tailored to their needs. Various pricing options are available for ThingsBoard, including a free version. Additionally, users can host ThingsBoard on their local server instead of utilizing the company's own server infrastructure. For AppMAIS, we have Thingsboard installed on-premises.

Thingsboard has been used in several use cases, including the healthcare sector [50]. It has also been used in precision agriculture, such as monitoring crops and farming. In [51], Cadavid et al. developed a Monitoring as a Service (MaaS) platform combining IoT capabilities, data analytics, and agricultural experts in Colombia. To achieve this goal, soil sensors were deployed across multiple crops to provide data to the MaaS platform. The platform would issue alerts when certain conditions were met, triggering corrective action or sending notifications to relevant parties. Fortunately, Thingsboard possesses the requisite capabilities to accomplish these tasks.

4.2. Rule engine

In Thingsboard, the Rule Chain is a critical component that is used to process the data received from any device. It is a JavaScript program that executes when Thingsboard receives input from a device, such as telemetry data. For example, a beehive might report its temperature and the time it was recorded. The root rule chain, as depicted in Fig. 3, determines the input source, which can be telemetry, attributes, or other input types.

There is a given default rule chain upon installing ThingsBoard that is built to handle incoming requests. There are two key differences between our rule chain and the default rule chain (see Fig. 3). Firstly, the path highlighted by the oval labeled ‘A’ in Fig. 3 contains a script node called RemoveSentinel. Beemon software uses sentinel values to report errors from sensors, such as using a Not A Number (NaN) value for hive temperature. To avoid displaying these values on the dashboard, the RemoveSentinel node contains a simple JavaScript function that removes all sentinel values from the telemetry before saving the values to the device. Secondly, another major difference is the ‘Alarm Created’ branch from ‘Device Profile Node,’ highlighted by the oval labeled ‘B’ in Fig. 3. If the device telemetry triggers an alarm, a telegram message is created with information about the alarm. The rule chain then queries for the hive name from which the alarm originated and uses another script node to generate the telegram message as a string. Finally, a rest API call is made to a telegram bot to obtain real-time notifications of these critical issues.

The development of new systems using the rule chain presented several challenges. A critical use case was to create a reporting system that would send a telegram message if any hives failed to report any data by 8:30 A.M. each morning. To accomplish this, a rule chain was created that utilized a ‘Generator’ node to create mock telemetry readings every hour and a script node to determine whether the time was 8:30 A.M. If so, the rule chain was duplicated for each device in the AppMAIS group, and the ‘Active’ attribute of each device was checked to determine whether they were online at that time. If a device was found to be offline, a telegram message was sent indicating that the hive did not report data that morning. One problem with this implementation was that if multiple hives were inactive at 8:30 A.M., the rule chain would send a separate message for each hive which could be many messages if there was some widespread problem. Despite efforts to optimize and add new features, such as aggregating messages and addressing non-reporting sensors, these tasks proved difficult to accomplish through

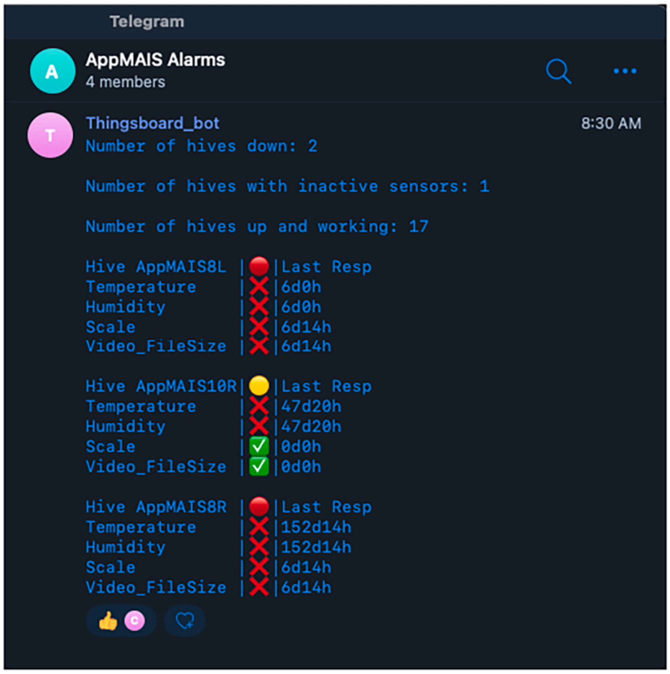

Fig. 4. Telegram morning report sent from the Thingsboard Bot.

the rule chain. As a result, a decision was made to develop an external middleware in Python to address these requirements.

4.3. Python middleware

As discussed earlier, the objective was to enhance the device activity alarm system that notifies when a sensor has not reported for at least an hour. For this purpose, the Thingsboard internal API was employed, which provides useful methods such as retrieving the last telemetry value from its key and the ability to edit device attributes using Server-Side Attributes. However, the Beemon system, which is responsible for collecting data at all the hives, has an active period during daylight, and there is a period when the system goes to sleep. A sleeping Beemon system cannot generate alerts. To detect the Beemon system awakening state in the mornings, we needed an external middleware to identify that state change. To this end, we implemented a python middleware that performs a check on each device by calculating the time difference between the current time and the last non-sentinel telemetry sent by the sensor. If no telemetry is received from a sensor within an hour, a Server-Side Attribute named after the telemetry key is updated, such as "Temperature_Active", which changes from true to false. Additionally, a specific attribute named "ActivityLevel" was created to describe whether or not the device is active. This attribute is set to zero if the device is inactive, one if the device is active but has some inactive sensors, and two if all sensors are active. The information is presented on a dashboard by colored circles, with green representing an activity level of two, yellow for one, and red for zero. Finally, a report of inactive sensors and devices is aggregated and sent as a single message via telegram. The morning report is illustrated in Fig. 4. The middleware has enabled us to detect the state of each sensor in each device and generate separate alerts for them. That has played a significant role in troubleshooting and helped reduce the downtime of the sensors.

4.4. Setting alarms

In Thingsboard, setting and clearing alarms can be achieved by creating alarm rules integral to a device profile. A device profile is assigned to each device and incorporates specific settings, including the types of alarms that can be configured for that device. For instance, one such alarm rule is the "Disconnected" critical level alarm, which is activated whenever the "Active" attribute changes its value to false. Another example includes activating a minor level alarm when a hive's humidity level reaches over 90%, which provides beekeepers with an early warning that the hive may require additional attention. Also, alarms are generated for missing data from any hive sensor, such as a defective scale that only transmits weight data every other recording time. Furthermore, an alarm is triggered when a hive is active, but the "Scale_Active" attribute changes to false after an hour of not receiving scale data. The AppMAIS device profile includes a comprehensive range of alarms that enable beekeepers to stay informed about vital information regarding their hives, such as disconnected hives, inactive sensors, missed readings, and undesirable sensor values.

5. Discussion

Our ultimate objective is to develop a system capable of generating alarms to proactively mitigate future events, including potentially disruptive occurrences such as swarms. To attain this objective, we thoroughly analyze and comprehend the data collected from the hives using visual and audio analysis.

The current dashboard and middleware primarily focus on data display and sensor availability alerts, respectively. However, as we delve deeper into data analysis and better understand the information at hand, our goal is to establish thresholds for different sensors dynamically. This will enable us to generate more precise alarms related to the behaviors exhibited by the bees.

In the creation of a dashboard for beehives, there are three main dashboard states that are utilized to display and gather information. The initial dashboard state, known as the landing page, is depicted in Fig. 5. This serves as the initial point of entry upon accessing the URL. The left half of the screen presents a map that displays the general location of the beehives, with thermometers that change color from orange to green depending on whether a certain threshold of overall hive health has been reached. This provides a quick indication of the temperature and humidity levels in the hive. Clicking on each thermometer generates a display of the temperature and humidity levels above it and a link to a more detailed page on the hives. Furthermore, the map utilizes a clustering effect that groups multiple hives within close proximity to each other, which can be expanded to display individual hives.

The right half of the screen is divided into two panels. The top panel displays active alarms in each hive, providing information on when the alarm was generated, which device it belongs to, the severity of the alarm, and whether it has been acknowledged or cleared. These alarms are generated by implementing a rule chain consisting of three alarm rules: high-temperature detection, high-humidity detection, and inactivity detection. The severity of the alarms can range from minor to critical, with the latter prompting an email and a telegram message to alert the dashboard creator, as illustrated in Fig. 4. The alarm system provides us near-live input on a sensor's status. We have used this to either remotely correct the failure through adjustments in the software tools or have physically gone to the hive to repair or replace the faulty sensors. The bottom panel displays the entity-relationship of all the dashboard states and the devices, with clicking on a specific hive leading to the corresponding dashboard. Additionally, a link has been included to allow users to navigate to the details page.

The following dashboard state accessible to the user is the detailed state, which can be depicted in Fig. 6. This page serves as a concise summary of the most recent data received from each device. The page comprises a topographic map, which offers greater insight into the environmental variations across different hives, in contrast to the street map featured on the landing page. Additionally, the map can be utilized alongside the bar charts situated beneath it to analyze the differences in the data. The bar charts represent recent temperature, humidity, and video file size reports from each hive. At the same time, the numeric

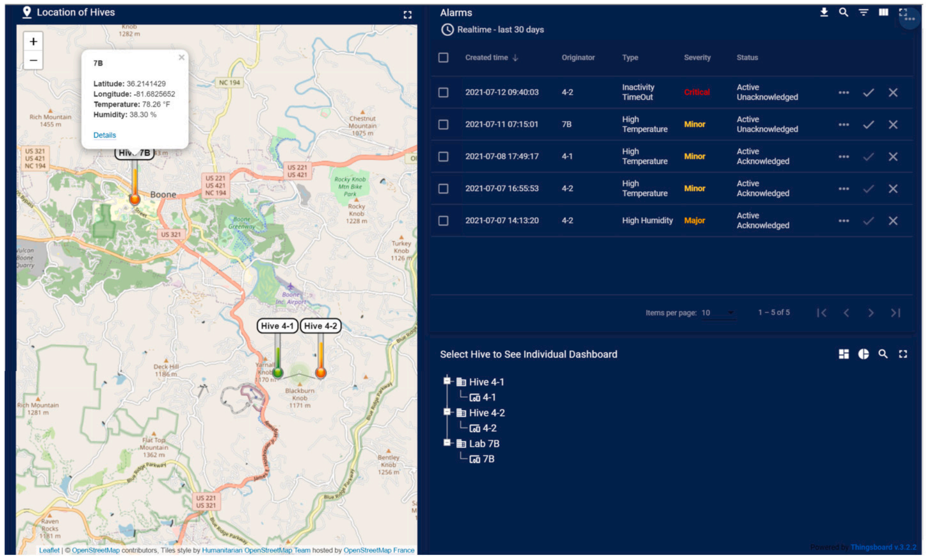

Fig. 5. Landing page for Beemon Dashboard.

Fig. 6. Details page for Beemon Dashboard.

values are displayed in the AppMAIS Hives widget, which exhibits the same information as the landing page, albeit on a larger scale.

The final dashboard state is the detailed page dedicated to the selected hive, presented in Fig. 7. In the top left corner of the page, the values for the four parameters mentioned previously are displayed for the past twelve hours. The first column represents the creation date, while the second and third columns, displaying temperature and humidity, respectively, alter their background colors based on the current readings. The last two columns exhibit the scale and video file size. The hive's current temperature, humidity, and weight are positioned to the right of the sensor values. This information reflects only the most recent data recorded by the sensors. Adjacent to the current weight, a card containing a link to Beestream is located. Beestream allows access to an archive and a current stream of the latest video recorded at each beehive. The remaining section of the dashboard features four graphs demonstrating data for the past month.

In addition to Thingsboard dashboards created as part of the AppMAIS project, we have developed a simple visualization tool [45] to be used internally for our analysis. However, we are currently working on a much more advanced tool for sophisticated visualization and analysis which we have plans to make publicly available sometime in the future.

6. Conclusion

IoT has become increasingly popular for managing applications in different fields in recent years. We have successfully created an IoT solution to manage and monitor the 31 hives of the AppMAIS project. To achieve this, we utilized the Thingsboard IoT platform, which provided flexibility to send the hive data to our server easily. For our project, AppMAIS, we had to develop creative solutions for challenges, such as detecting the failure of a device to wake up and generating specific alarms for remote monitoring. To this end, we developed a middleware that worked with Thingsboard to set alarms and then act upon those alarms locally or remotely. Furthermore, to perform an extensive analysis within Thingsboard, an add-on analytics tool must be purchased separately. As a result, we have been working on developing our own data visualization and analytics tools. As we continue to analyze the data collected through Beemon in our AppMAIS project, we will be able to close the loop and feedback the insights and metrics to set alarm levels leading to a continually improving beehive remote monitoring and management system.

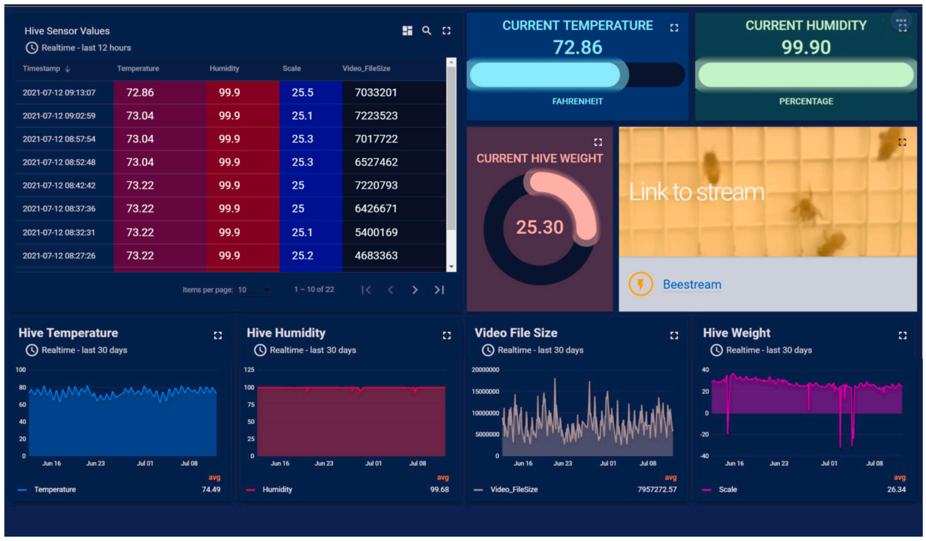

Fig. 7. Dashboard State for each individual state.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

The authors wish to thank the North Carolina General Assembly for funding the AppMAIS project through an NC ROI grant, the Lowe's Distinguished Professor Research Fund, and the Department of Computer Science, Office of Sustainability, Academic Affairs, College of Arts and Sciences at Appalachian State University (106016) for their support. Several graduate and undergraduate students in our department were part of the development and testing of the Beemon and AppMAIS systems. We wish to thank them for their contributions.

References

[1] J. Schwartz, M. Kuenzle, D. Pinsky, Save the bees, URL https://www.greenpeace.org/usa/sustainable-agriculture/save-the-bees/, Jun 2014.
[2] One-third of our food supply relies on a very sick species: honeybees, URL https://www.sierraclub.org/sierra/one-third-our-food-supply-relies-very-sick-species-honeybees, Oct 2019.
[3] Declining bee populations pose threat to global food security and nutrition, online, URL https://www.fao.org/news/story/en/item/1194910/icode/, May 2013.
[4] R.M. Nowierski, Pollinators at a crossroads, online, URL https://www.usda.gov/media/blog/2020/06/24/pollinators-crossroads, Jun. 2020.
[5] V.A. Kulyukin, M. Putnam, S.K. Reka, Digitizing buzzing signals into A440 piano note sequences and estimating forager traffic levels from images in solar-powered, electronic beehive monitoring, in: Proceedings of the International MultiConference of Engineers and Computer Scientists, vol. 1, 2016.
[6] M.Z. Sharif, N. Di, B. Yu, Honeybee (Apis spp.) (Hymenoptera: Apidae) colony monitoring using acoustic signals from the beehive: an assessment by global experts and our feedback, Agriculture 13 (4) (2023), https://doi.org/10.3390/agriculture13040769, https://www.mdpi.com/2077-0472/13/4/769.
[7] V. Kulyukin, S. Mukherjee, On video analysis of omnidirectional bee traffic: counting bee motions with motion detection and image classification, Appl. Sci. 9 (18) (2019), https://doi.org/10.3390/app9183743, https://www.mdpi.com/2076-3417/9/18/3743.
[8] S. Mukherjee, V. Kulyukin, Application of digital particle image velocimetry to insect motion: measurement of incoming, outgoing, and lateral honeybee traffic, Appl. Sci. 10 (6) (2020), https://doi.org/10.3390/app10062042, https://www.mdpi.com/2076-3417/10/6/2042.
[9] V. Kulyukin, S. Mukherjee, A. Minichiello, T. Truscott, BeePIV: a method to measure Apis mellifera traffic with particle image velocimetry in videos, Appl. Sci. 11 (5) (2021) 2276, https://doi.org/10.3390/app11052276.
[10] V. Kulyukin, Audio, image, video, and weather datasets for continuous electronic beehive monitoring, Appl. Sci. 11 (10) (2021) 4632, https://doi.org/10.3390/app11104632.
[11] V. Kulyukin, A. Tkachenko, K. Price, W. Meikle, M. Weiss, Integration of scales and cameras in nondisruptive electronic beehive monitoring: on the within-day relationship of hive weight and traffic in honeybee (Apis mellifera) colonies in Langstroth hives in Tucson, Arizona, USA, Sensors 22 (13) (2022), https://doi.org/10.3390/s22134824, https://www.mdpi.com/1424-8220/22/13/4824.
[12] V.A. Kulyukin, D. Coster, A. Tkachenko, D. Hornberger, A.V. Kulyukin, Ambient electromagnetic radiation as a predictor of honey bee (apis mellifera) traffic in linear and non-linear regression: numerical stability, physical time and energy efficiency, Sensors 23 (5) (2023), https://doi.org/10.3390/s23052584, https://www.mdpi.com/1424-8220/23/5/2584.
[13] A. Zabasta, A. Zhiravetska, N. Kunicina, K. Kondratjevs, Technical implementation of IoT concept for bee colony monitoring, in: 2019 8th Mediterranean Conference on Embedded Computing (MECO), 2019, pp. 1–4.
[14] A. Bourouis, T. Benahmed, K.A. Mokeddem, K. Benahmed, A.S. Lairedj, IoT for smart apiculture: issues and solutions, in: 2022 3rd International Conference on Embedded & Distributed Systems (EDiS), 2022, pp. 61–66.
[15] S.L. Buchmann, S.C. Thoenes, et al., The electronic scale honey bee colony as a management and research tool, Bee Sci. 1 (1) (1990) 40–47.
[16] W. Meikle, N. Holst, G. Mercadier, F. Derouané, R. James, Using balances linked to dataloggers to monitor honey bee colonies, J. Apic. Res. 45 (1) (2006) 39–41.
[17] E. Stalidzans, A. Berzonis, Temperature changes above the upper hive body reveal the annual development periods of honey bee colonies, Comput. Electron. Agric. 90 (2013) 1–6.
[18] B. Kim, C. Jung, Design and implementation of cloud based realtime temperature and humidity monitoring system of honey bee colony, Korean J. Apic. 30 (4) (2016) 263–267.
[19] A. Zacepins, A. Kviesis, A. Pecka, V. Osadcuks, Development of Internet of things concept for precision beekeeping, in: 2017 18th International Carpathian Control Conference (ICCC), 2017, pp. 23–27.
[20] M.A. Md Yunus, Internet of things (IoT) application in meliponiculture, Int. J. Integr. Eng. 9 (4) (Dec. 2017), https://publisher.uthm.edu.my/ojs/index.php/ijie/article/view/2016.
[21] O. Debauche, M.E. Moulat, S. Mahmoudi, S. Boukraa, P. Manneback, F. Lebeau, Web monitoring of bee health for researchers and beekeepers based on the Internet of things, Proc. Comput. Sci. 130 (2018), https://doi.org/10.1016/j.procs.2018.04.103 991–998, http://www.sciencedirect.com/science/article/pii/S1877050918304654, The 9th International Conference on Ambient Systems, Networks and Technologies (ANT 2018) / The 8th International Conference on Sustainable Energy Information Technology (SEIT-2018) / Affiliated Workshops.
[22] S. Kontogiannis, An Internet of things-based low-power integrated beekeeping safety and conditions monitoring system, Inventions 4 (52) (2019) 1–26.
[23] I.Z. Ochoa, S. Gutierrez, F. Rodríguez, Internet of things: low cost monitoring beehive system using wireless sensor network, in: 2019 IEEE International Conference on Engineering Veracruz (ICEV) I, 2019, pp. 1–7.
[24] N.H. Khairul Anuar, M. Amri Md Yunus, M.A. Baharuddin, S. Sahlan, A. Abid, M.M. Ramli, M. Razzi Abu Amin, Z.F. Mohd Lotpi, IoT platform for precision stingless bee farming, in: 2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), 2019, pp. 225–229.

[25] X. Lyu, S. Zhang, Q. Wang, Design of intelligent beehive system based on internet of things technology, in: Proceedings of the 3rd International Conference on Computer Engineering, Information Science & Application Technology (ICCIA 2019), 2019.
[26] P. Catania, M. Vallone, Application of a precision apiculture system to monitor honey daily production, Sensors 20 (7) (2020), https://doi.org/10.3390/s20072012, https://www.mdpi.com/1424-8220/20/7/2012.
[27] R. Tashakkori, A.S. Hamza, M.B. Crawford, Beemon: an IoT-based beehive monitoring system, Comput. Electron. Agric. 190 (2021) 106427, https://doi.org/10.1016/j.compag.2021.106427, https://www.sciencedirect.com/science/article/pii/S0168169921004440.
[28] Thingsboard, open-source IoT platform, URL https://thingsboard.io/.
[29] M.Z. Sharif, N. Di, F. Liu, Monitoring honeybees (Apis spp.) (Hymenoptera: Apidae) in climate-smart agriculture: a review, Appl. Entomol. Zool. 57 (4) (2022) 289–303.
[30] T. Chamaidi, K. Malisova, V. Nomikos, E. Vlachogiannis, C. Alifieris, C. Rigakis, M. Stavrakis, IOHIVE: design requirements for a system that supports interactive journaling for beekeepers during apiary inspections, in: M.M. Soares, E. Rosenzweig, A. Marcus (Eds.), Design, User Experience, and Usability: UX Research, Design, and Assessment, Springer International Publishing, Cham, 2022, pp. 157–172.
[31] D. Cota, J. Martins, H. Mamede, F. Branco, BHiveSense: an integrated information system architecture for sustainable remote monitoring and management of apiaries based on IoT and microservices, J. Open Innov. Technol. Mark. Complex. 9 (3) (2023) 100110, https://doi.org/10.1016/j.joitmc.2023.100110, https://www.sciencedirect.com/science/article/pii/S2199853123002123.
[32] K. Dineva, Computer system using internet of things for monitoring of bee hives, in: 17th International Multidisciplinary Scientific GeoConference SGEM2017, Nano, Bio and Green Technologies for a Sustainable Future, 2017.
[33] link, URL https://www.osbeehives.com/pages/osbeehives-app.
[34] link, URL https://apisprotect.com/.
[35] Department of Computer Science, Appalachian State University, BeeMon: beehive monitoring for new insights, http://cs.appstate.edu/beemon/. (Accessed 14 October 2019).
[36] R. Tashakkori, A. Ghadiri, Image processing for honey bee hive health monitoring, in: SoutheastCon 2015, 2015, pp. 1–7.
[37] D.J. Kale, R. Tashakkori, R.M. Parry, Automated beehive surveillance using computer vision, in: SoutheastCon 2015, 2015, pp. 1–3.
[38] R. Tashakkori, N.P. Hernandez, A. Ghadiri, A.P. Ratzloff, M.B. Crawford, A honeybee hive monitoring system: from surveillance cameras to Raspberry Pis, in: SoutheastCon 2017, 2017, pp. 1–7.
[39] M. Crawford, Automated collection of honey bee hive data using the Raspberry Pi, Master's thesis, Appalachian State University, Boone, NC, USA, 2017.
[40] G. Buchanan, R. Tashakkori, A web-app for analysis of honey bee hive data, in: 2019 SoutheastCon, 2019, pp. 1–6.
[41] W. O'Brien, R. Tashakkori, R.M. Parry, A. Hamza, J. Graber, Estimating the number of drones at the entrance of a honey bee hive using machine learning tools, in: SoutheastCon 2022, 2022, pp. 397–404.
[42] B. Underwood, R. Tashakkori, Detecting anomalies in honey bee hives using their audio recordings, in: SoutheastCon 2022, 2022, pp. 173–177.
[43] C. Campell, J. Graber, R. Tashakkori, W. O'Brien, IoT apiary fleet management with Jenkins, in: 2022 IEEE 13th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), 2022, pp. 0328–0335.
[44] C. Campell, R.M. Parry, R. Tashakkori, Non-invasive spectral-based swarm detection, in: SoutheastCon 2023, 2023, pp. 253–260.
[45] L. Richardson, C. Campell, R. Tashakkori, W. O'Brien, S.E. Davis, AppMAIS simple data visualization app, in: SoutheastCon 2023, 2023, pp. 366–370.
[46] P.E.T. Jorgensen, M.-S. Song, Entropy encoding, Hilbert space, and Karhunen-Loève transforms, J. Math. Phys. 48 (10) (2007) 103503, https://doi.org/10.1063/1.2793569.
[47] A. Glória, F. Cercas, N. Souto, Design and implementation of an IoT gateway to create smart environments, Proc. Comput. Sci. 109 (2017) 568–575, https://doi.org/10.1016/j.procs.2017.05.343.
[48] P. Dhar, P. Gupta, Intelligent parking cloud services based on IoT using MQTT protocol, in: 2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT), 2016, pp. 30–34.
[49] M.B. Yassein, M.Q. Shatnawi, S. Aljwarneh, R. Al-Hatmi, Internet of things: survey and open issues of MQTT protocol, in: 2017 International Conference on Engineering MIS (ICEMIS), 2017, pp. 1–6.
[50] T.M. Kadarina, R. Priambodo, Monitoring heart rate and SpO2 using Thingsboard IoT platform for mother and child preventive healthcare, IOP Conf. Ser., Mater. Sci. Eng. 453 (2018) 012028, https://doi.org/10.1088/1757-899x/453/1/012028.
[51] H. Cadavid, W. Garzón, A. Pérez, G. López, C. Mendivelso, C. Ramírez, Towards a smart farming platform: from IoT-based crop sensing to data analytics, Commun. Comput. Inf. Sci. (2018) 237–251, https://doi.org/10.1007/978-3-319-98998-3_19.