ApIsoT: An IoT Function Aggregation Mechanism for Detecting Varroa Infestation in Apis mellifera Species

Ana Isabel Caicedo Camayo, Martin Alexander Chaves Muñoz and Juan Carlos Corrales

Telematics Engineering Group, Faculty of Electronics Engineering and Telecommunications, Universidad del Cauca, Popayán 190003, Colombia; anitacaicedo@unicauca.edu.co (A.I.C.C.); martinchaves@unicauca.edu.co (M.A.C.M.)

* Correspondence: jcorral@unicauca.edu.co

Abstract: In recent years, the global reduction in populations of the Apis mellifera species has generated a worrying deterioration in the production of essential foods for human consumption. This phenomenon threatens food security, as it reduces the pollination of vital crops, negatively affecting the health and stability of ecosystems. The three main factors generating the loss of the bee population are industrial agriculture, climate changes, and infectious diseases, mainly those of parasitic origin, such as the Varroa destructor mite. This article proposes an IoT system that uses accessible, efficient, low-cost devices for beekeepers in developing countries to monitor hives based on temperature, humidity, CO$_2$, and TVOC. The proposed solution incorporates nine-feature aggregation as a data preprocessing strategy to reduce redundancy and efficiently manage data storage on hardware with limited capabilities, which, combined with a machine learning model, improves mite detection. Finally, an evaluation of the energy consumption of the solution in each of its nodes, an analysis of the data traffic injected into the network, an assessment of the energy consumption of each implemented classification model, and, finally, a validation of the solution with experts is presented.

Keywords: machine learning; Varroa destructor; Apis mellifera; IoT; precision beekeeping; aggregation functions; energy consumption

1. Introduction

The Apis mellifera species is crucial in pollinating various plant species, contributing significantly to agricultural production and food security. According to data from the Food and Agriculture Organization of the United Nations (FAO), approximately 75% of the world’s crops depend heavily on pollination [1]. This activity carried out by bees increases the production of fruits, vegetables, and seeds and improves the quality and diversity of crops. This function is essential for the sustainability of agricultural systems and biodiversity conservation. Furthermore, bees contribute significantly to climate change mitigation, as they promote the regeneration and maintenance of plant ecosystems, helping to absorb carbon dioxide and stabilize the global climate [2].

Western bee colony losses can vary significantly from year to year, but they are becoming worse overall. Annual studies of honey bee colony loss show considerable variation in temporal and spatial rates of colony loss, as well as contributing factors [3,4]. Some studies such as [5] report that the annual colony loss rate of Apis mellifera in Latin America varied between 17% and 48%, depending on the country and the year. The percentage of colony loss of meliponine bees varied between 33% and 46% annually. Varroa destructor mites and the viruses that spread these parasites are now considered the most significant contributors to these losses. However, queen problems, such as a lack of natural food sources, the expansion of urban areas, and excessive use of pesticides, are also factors [6]. This phenomenon threatens global food security by reducing pollination of vital crops. At the same time, declining bee populations negatively impact the health and stability of ecosystems, compromising their ability to provide critical ecosystem services.
The *Varroa destructor* mite, along with the viruses it spreads, is an important catalyst for this decline. This threat generates negative impacts at an individual and collective level, compromising bees’ immunological and nutritional health. It causes physical injuries and promotes the spread of fungi, bacteria, and viruses within the colonies it infests since it acts as a vector of microorganisms [3]. The consequences of *Varroa* infestation significantly impact beekeeping and food security since it has contributed to the decline of bee populations worldwide.

For beekeepers, it is crucial to detect the presence of the mite early and accurately to prevent its spread and protect the health and well-being of bees. Over the years, artisanal techniques have been practiced, such as the *Varroa* test with isopropyl alcohol, considered the most precise and effective, where bees are immersed in a mixture of water and alcohol to count the mites present and establish a percentage of infestation [7]. However, this method can have disadvantages, such as non-representative samples and instant death of bees [8]. Another method is the sanitary floor, which involves replacing the traditional floor of the hives with one that collects fallen mites. However, it has disadvantages, such as damage to the apiaries, and the effectiveness of the test is variable [8]. Finally, the *Varroa* test with powdered sugar involves shaking a colony sample in a jar with sugar and then counting the mites released during the process, which causes stress and death of the bees evaluated [8].

To address this problem, IoT-based monitoring systems and artificial intelligence algorithms are integrated into beekeeping practices [9,10]. This is closely linked to the concept of precision beekeeping, which emphasizes the use of technologies such as advanced sensors, data analysis, and process automation for precise management of bee colonies [11].

This study proposes adopting an integrated solution based on the Internet of Things (IoT) and supervised learning as a critical element in detecting the presence of the *Varroa destructor* mite in hives. To this end, the aggregation of functions and their derived mechanisms are used as the primary strategy to develop a robust and efficient prototype capable of accurately identifying the presence of the mite. This proposal aims not only to guarantee and improve the quality of life of bees but also to support the work carried out by beekeepers and to significantly reduce the restrictions associated with battery, transmission, and data storage. The proposed hardware prototype uses low-cost and easy-to-achieve components, ensuring they are accessible to beekeepers.

The present work involved the following sections. Section 2 presents a systematic review of the available literature on emerging technologies used for hive monitoring and detecting the *Varroa destructor* mite. Section 3 identifies the occurrence of *Varroa* infestation based on discrete variables and captures functional and non-functional requirements; Section 4 shows the prototype’s implementation and hardware and software development. In Section 5, we analyze and discuss the results. Finally, in Section 6, we expose the conclusions and future work.

2. Related Works

Following the adjusted methodology of Petersen [12] and Kitchenham [13], the existing research trends related to the use of IoT against detecting *Varroa* infestation in *Apis mellifera* species are presented.

- Research question: based on the problem of mass mortality of bees [3] and seeking to know the technological proposals from the area of IoT and the aggregation of functions, we pose the following research question:
  What research exists within the IoT area that uses the aggregation of functions to monitor or detect *Varroa* infestation in *Apis mellifera* hives?
- Search strategies: information is collected through scientific databases, such as Scopus and Web of Science. Additionally, we use VOSviewer to analyze the resulting keywords.
- Selection of studies: the search string and the inclusion and exclusion criteria are defined to select the relevant articles (Table 1).
Table 1. Study selection criteria.

<table>
<thead>
<tr>
<th>Search string</th>
<th>Inclusion criteria</th>
<th>Exclusion criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>(“bee” OR “honey bee” OR “apis mellifera”) AND (“IoT” OR “internet of things”) AND (“varroa” OR “varroa infestation”)</td>
<td>Data Aggregation, Multisensor Platform</td>
<td>Artificial Bee Colony Algorithm, Food Products</td>
</tr>
</tbody>
</table>

Through this process, we found 11 related works subjected to keyword analysis to detect possible biases and areas of interest for research.

This article is formulated as a solution to the detection of Varroa infestation in species of Apis mellifera. For this reason, it is essential to analyze the works dedicated to this purpose. In Figure 1, two specific branches are distinguished. The first is dedicated to Varroa infestation detection solutions based on image processing, machine learning, edge, and cloud computing within the framework of precision beekeeping. The second identifies factors that intervene in the detection process, such as climate change, honey quality control, and hive monitoring. Finally, and as the central node, there is the IoT, an emerging technology that allows both gaps to be bridged into a comprehensive solution for the beekeeper.

Figure 1. Keyword mapping.

- Gaps: During the literature review, we found that there are currently no works related to feature aggregation in precision beekeeping. Furthermore, most of the works focus on emerging technologies such as deep learning [14], the cloud, and edge computing, where it is highlighted that the mechanisms proposed by the authors [15–17] are carried out in simulation environments. Although they can provide a general view of a system’s performance, it is necessary to provide a broader view in a hardware environment.

In [18–21], the authors give priority to the implementation of the proposed solution in a natural environment, without regard for precision beekeeping systems, such as battery consumption and processing of large volumes of data. The works do not consider approaches such as analyzing the collected data and processing as a strategy for detecting complex events such as Varroa infestation.

Studies such as [17–21] demonstrate that it is possible to detect anomalies in the hive, including the Varroa destructor mite, using sensors that monitor physical variables. However, the proposed systems offer effectiveness levels greater than 75% and have proven to be minimally invasive techniques for bees. The authors highlight some aspects that have yet to be considered and are addressed in this research, such as incorporating commercial and low-cost sensors for temperature, humidity, CO₂, and TVOC. In addition, a data processing strategy was implemented by aggregating functions and a classification model based on supervised learning to detect the mite.

Some of the sensors used by different authors specifically for detecting Varroa in bee hives are presented in Table 2.
Table 2. Sensors used for different physics variables.

<table>
<thead>
<tr>
<th>Variable Physics</th>
<th>Sensors Used</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO₂</td>
<td>GS4161</td>
<td>[20,22]</td>
</tr>
<tr>
<td></td>
<td>TL6615</td>
<td>[21]</td>
</tr>
<tr>
<td></td>
<td>SGP30, BME680</td>
<td>[19]</td>
</tr>
<tr>
<td>TVOC</td>
<td>SGP30, BME680</td>
<td>[19]</td>
</tr>
<tr>
<td></td>
<td>TGS2600</td>
<td>[20]</td>
</tr>
<tr>
<td>Temperature</td>
<td>MCP9700A, DS18B20</td>
<td>[20,23]</td>
</tr>
<tr>
<td></td>
<td>BME680</td>
<td>[19]</td>
</tr>
<tr>
<td></td>
<td>DHT22</td>
<td>[21,24]</td>
</tr>
<tr>
<td>Humidity</td>
<td>DHT22</td>
<td>[21,24]</td>
</tr>
<tr>
<td></td>
<td>DHT11</td>
<td>[23]</td>
</tr>
<tr>
<td></td>
<td>BME680</td>
<td>[19]</td>
</tr>
<tr>
<td></td>
<td>808H5V5</td>
<td>[20]</td>
</tr>
</tbody>
</table>

3. Materials and Methods

This section focuses on describing the four phases carried out for the research. The first focuses on characterizing the presence of the mite in terms of discrete variables; the second refers to the data preprocessing technique used, and the third presents the classification models based on supervised learning developed to provide a level of alertness—finally, the fourth focuses on integrating the solution into the selected hardware.

3.1. Characterization of the Appearance of Varroa Infestation

Several studies have shown that the behavior of some internal variables of the hive, such as temperature, humidity, and levels of gases such as CO₂ and TVOC, are crucial indicators of bee health and the risk of infections [21,23,25]. Both the Varroa destructor mite and the Apis mellifera species react differently to changes in internal temperature. In this context, the characterization considers the phoretic phase of Varroa, a stage in which the mite is outside the brood cells, and represents a critical moment for evaluating and controlling this threat [8].

Based on interviews with beekeepers in the region, as well as studies such as [25] focused on characterizing the conditions of different variables inside a hive and the presence of Varroa in Colombia, it was possible to identify that for variables such as temperature, the normal ranges inside a hive are between 32 °C and 36 °C, considering a maximum and minimum range between 20 °C and 37 °C, and with relative humidity between 50% and 75%. Meanwhile, for variables such as CO₂ and TVOC, the characterization is performed based on studies such as [19].

3.1.1. Temperature

Some research, such as [26], suggests that temperatures above 38 °C are critical for both bees and Varroa, who are vulnerable and cannot carry out their activities usually. If the temperature remains in that state for prolonged periods, it can cause death. Varroa’s reproduction rate and phoretic phase’s activity phase decrease significantly in this case. Therefore, the risk of contracting Varroa infestation is estimated to be low under these conditions. Between 36 °C and 37.9 °C, bees enter a phase of thermal stress that causes excessive energy consumption and makes them especially vulnerable. Varroa can use this condition to reproduce even if it is not in the best conditions.

Between 33 °C and 35 °C, the ideal environment is established for the reproductive cycle of bees. This temperature also benefits Varroa since the females of this mite can enter the unprotected brood cells and reproduce. However, this temperature does not favor the phoretic phase of Varroa [15].

On the other hand, a temperature below 28 °C makes bees vulnerable due to their inactivity, which reduces their ability to defend themselves against possible threats. In this
scenario, the phoretic phase of the *Varroa* develops to its maximum, allowing easy access to the brood cells. These cells, which maintain a higher internal temperature, facilitate the normal reproductive process of *Varroa* [27].

### 3.1.2. Humidity

Initially, research such as [20] suggests that a humidity level greater than 65% favors the reproduction and spread of the *Varroa destructor* mite, increasing the risk of infection. This extremely humid environment weakens the bees’ immune systems and makes them more susceptible to infections, representing a high health risk. Secondly, a humidity level between 40% and 60% is considered moderate, as it balances bees, allowing them to use their natural defense mechanisms and reducing the risk of infection by *Varroa destructor*. This risk level is classified as medium [19].

Finally, a low humidity level, less than 40%, is considered a dry environment that can dehydrate bees and weaken their immune system, making them susceptible to certain infections. However, *Varroa* mites do not adapt quickly to this environment, so the risk of contracting *Varroa* infestation is low [20].

### 3.1.3. Concentration of Volatile Gases

Recent research [17,19,22] has found that the levels of different types of gases within a hive can be indicators of health, such as CO$_2$ and TVOC. These gases, produced by bees and other biological processes, can regulate the hive’s internal environment and the bees’ well-being. Research has established that TVOC and CO$_2$ levels vary considerably as the level of infestation increases or decreases. For this analysis, it is essential to highlight that gas levels depend on different factors, such as the number of bees in the hive, the quality of the outside air, and the hive’s location. In this case, an average hive was used for beekeeping activities, that is, 40,000 to 60,000 bees. The hive that has not been parasitized with *Varroa* has a TVOC level ranging from 0 ppb to 300 ppb and CO$_2$ from 0 ppm to 600 ppm. A low *Varroa* infestation level has a TVOC level of 300 ppb to 600 ppb and CO$_2$ of 600 ppm to 1000 ppm. An average *Varroa* infestation level has a TVOC level of 600 ppb to 1500 ppb and CO$_2$ of 1000 ppm to 2000 ppm. A high *Varroa* infestation level has a TVOC level more fabulous than 1000 ppb and CO$_2$ greater than 2000 ppm. Finally, a hive receiving chemical treatment against *Varroa* has a TVOC level of 1900 ppb to 3500 ppb and CO$_2$ of 1000 ppm to 1600 ppm.

### 3.1.4. Identified Alerts

After analyzing the presence of the *Varroa* mite using various discrete variables, the information collected is synthesized to establish an alert level, whether very high, high, medium, or low, on the possible presence of the mite in the hive, as shown in the Table 3.

Table 3 establishes the alert level of *Varroa* infestation in a hive in terms of percentages. This considers the result provide by [7] and the information delivered by expert beekeepers in interest. The data provided by the sources are mainly based on *Varroa* tests using different methods and hives of approximately 40,000 to 60,000 bees.
Table 3. Alert-level classification based on discrete variables.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Range</th>
<th>Alert level</th>
<th>% of Infestation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature (°C)</td>
<td>Less than 28</td>
<td>Very high</td>
<td>Greater than 10%</td>
</tr>
<tr>
<td>Humidity (%)</td>
<td>Greater than 75</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TVOC (ppb)</td>
<td>Greater than 1500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO₂ (ppm)</td>
<td>Greater than 2000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Temperature (°C)</td>
<td>28–30</td>
<td>High</td>
<td>Between 5% and 10%</td>
</tr>
<tr>
<td>Humidity (%)</td>
<td>65–75</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TVOC (ppb)</td>
<td>600–1500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO₂ (ppm)</td>
<td>1000–2000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Temperature (°C)</td>
<td>30–33</td>
<td>Medium</td>
<td>Between 2% and 5%</td>
</tr>
<tr>
<td>Humidity (%)</td>
<td>50–65</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TVOC (ppb)</td>
<td>300–600</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO₂ (ppm)</td>
<td>600–1000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Temperature (°C)</td>
<td>34–37</td>
<td>Low</td>
<td>Less than 2%</td>
</tr>
<tr>
<td>Humidity (%)</td>
<td>40–50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TVOC (ppb)</td>
<td>0–300</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO₂ (ppm)</td>
<td>400–600</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.1.5. Requirements Exploration and Capture

A workshop was conducted with 16 beekeepers from the Beekeeping and Agroindustrial Association of Piendamó and Tunía (ASAPIT) to identify the needs and establish the solution’s requirements. The interviews accomplished in this workshop established that the majority knew the Varroa destructor mite and recognized the importance of variables such as temperature and humidity in its development. Although some do not apply preventive treatments, most prioritize hygiene and periodic check-ups. Additionally, beekeepers show interest in a remote system to monitor their hives, preferring weekly notifications (Figure 2).

Figure 2. Apiary visit frequency and notification frequency preference among beekeepers.

Subsequently, considering the needs of beekeepers and the information selected, the functional and non-functional requirements of the system are established.

**Functional requirements:**
1. Capture environmental data such as temperature, humidity, CO₂, and TVOC.
2. Monitor the hive for an extended period.
3. Detect the presence of Varroa infestation from data collected over a week.
4. Determine and classify the Varroa infestation rate in a hive into four levels (very high, high, medium, or low).
5. Notify the beekeeper of said index and provide preventive information through a message.

**Non-functional requirements:**
1. Manage the battery to make the solution work continuously for a long time.
2. Ensure the connection between devices to transmit data.
3. Process data to avoid network saturation.
4. Manage data storage.
5. Adapt the system to different environments to function correctly in various environmental and climatic conditions.

3.2. Function Aggregation

Adding functions in an IoT system refers to applying, combining, and consolidating different functions or features in a system to improve efficiency, usefulness, and data management. This section discusses feature aggregation as a preprocessing strategy for captured data.

3.2.1. Centralized Feature Aggregation Mechanism

We base the proposed IoT solution on a centralized data aggregation mechanism, the main actors of which are the sensor, aggregator, and base station nodes. The focus of the solution is centralized data aggregation (CDA), and this focus marks the pause in its operation. The network diagram is presented below (Figure 3), with the hardware selected.

![Figure 3. Network topology based on the centralized data aggregation model.](image)

Figure 4 shows the system connection diagram for the sensor and aggregator nodes. In red, the connections related to the direct power supply of the source are observed; in orange, the connections that correspond to a voltage input regulated at 5 V; in black, the ground connection; in brown and violet, the data connection for the SGP30 carbon dioxide and volatile organic compounds gas sensor (Sensirion AG, Stäfa, Switzerland); in cyan, the data connection for the Temperature and humidity sensor DHT22 (Adafruit Industries, New York, NY, USA); in gold, the data connection for the Voltage sensor fz0430 (Analog Devices, Inc., Norwood, MA, USA). The aggregator node and the Base station node are made up of the ESP32 development board (Espressif Systems, Shanghai, China) [28].

![Figure 4. System connection diagram.](image)
Two sensor nodes, DHT22 [24], are installed inside the apiary, specifically in the hive’s central frame. They provide temperature and humidity data and SGP30 [29] data at the CO₂ and TVOC levels. These nodes capture this information within the hive and transmit it to the aggregator node.

The aggregator node (AN) performs battery management, supported by the FZ0430 sensor [30]. It also guarantees the system’s storage. It takes the data captured by the sensor nodes, whose primary function is to apply the corresponding aggregation functions and transmit the aggregated data to the base station node (BSN).

For its part, the BSN receives the aggregated data and, through a weighted multi-criteria algorithm, normalizes the information that is subsequently sent to a classification model based on supervised learning, which determines the alert level in the hive; this alert is notified to the beekeeper via a text message in the Telegram application.

The need to use a BSN arises due to the nature of the centralized function aggregation mechanism, designed to generate a scalable solution in case it is necessary to monitor more than one hive, thus facilitating its management.

3.2.2. Battery Management

It is a crucial factor because all nodes in the network run on batteries and, therefore, have energy limitations. Battery life can directly impact system availability and performance. To cover this requirement, we propose the following statements:

- Limit system functionalities: the system takes a certain number of samples (Table 4) depending on the battery’s state before entering an energy-saving state.

<table>
<thead>
<tr>
<th>Battery</th>
<th>Description</th>
<th>Periodicity</th>
<th># of Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greater than 30%</td>
<td>Capture temperature samples</td>
<td>2 h</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>Capture humidity samples</td>
<td>2 h</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>Capture CO₂ samples</td>
<td>12 h</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>Capture TVOC samples</td>
<td>12 h</td>
<td>60</td>
</tr>
<tr>
<td>Less than 30%</td>
<td>Capture temperature samples</td>
<td>4 h</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Capture humidity samples</td>
<td>4 h</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Capture CO₂ samples</td>
<td>12 h</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>Capture TVOC samples</td>
<td>12 h</td>
<td>30</td>
</tr>
</tbody>
</table>

- Hibernation: Using predefined functions of the EP32 board, such as the `light_sleep` method, allows the sensor nodes to enter a hibernation or low-power state until it is time to capture a sample.
- Internet connection: The interconnection between the base station node and the aggregator node occurs only when data need to be sent; otherwise, the devices remain disconnected from the Internet. The solution focuses on minimizing the power consumption of the devices used, especially the internal Wi-Fi modules, reducing the connection only when transmitting or receiving information due to high consumption, according to the ESP32 board datasheet [28].

3.2.3. Transmission Management

It allows the network to reduce traffic overload and optimize battery and bandwidth resources, improving efficiency and minimizing errors and information loss.

- Information reduction: The amount of data captured is reduced, applying average (AVG) as an aggregation function. It consists of taking the data captured in one day and calculating their arithmetic mean, thus reducing the number of samples to a single significant sample per variable.
- Internet connection: The interconnection between the base station node and the aggregator node occurs only when data need to be sent; otherwise, the devices remain disconnected from the Internet.
• Connection with the beekeeper via Telegram: Telegram is chosen as a communication bridge with the beekeeper since it is a free and accessible application. Thanks to its functionality against bot management, it allows data transmission from the hive to the beekeeper.

3.2.4. Data Storage Management

It is an essential process for the system because it optimizes the ESP32 board flash memory and processing resources. It is necessary to guarantee the availability, integrity, and usefulness of the data. In addition to reducing the risk of loss of critical information, it accelerates the identification of relevant data and facilitates adaptation to changing environmental needs.

Figure 5 offers a visual representation of the system’s process of eliminating data redundancy using aggregation functions. Below, we describe the operation in each of the nodes.

- Aggregator node: This node is configured to capture all the data from the sensor nodes, where the data from the entire day are taken and processed at the end of the day through the AVG aggregation functions and COUNT. The processing afterward allows us to obtain the daily arithmetic mean of the four variables inside the hive for subsequent sending to the BSN. Applying these functions allows for freeing up the storage space of the node to prevent its saturation due to lack of memory and loss of information due to possible failures.

- Base station node: This node receives and stores the data corresponding to the arithmetic means of the four variables for each of the five monitoring days. It stores twenty points of data in its memory later converted into four significant variables using the COUNT and AVG functions. These data make up the input data vector for the classification model.

3.2.5. Weighted Multi-Criteria Aggregation Algorithm

It consists of assigning a weight to each variable according to its relevance and calculating a score [31]. The sum of the individual scores of each variable will be the basis for detecting the alert level of the hive [32]. This process is commonly used in various areas of decision-making since it allows for establishing priority classes [33].

For its implementation, initially, the priority level of each variable is defined, with one being the most relevant and four being the least relevant (Equation (1)); temperature is defined as the most pertinent factor (1), followed by humidity (2), CO₂ (3), and TVOC (4). Subsequently, the maximum and minimum values each variable can take are identified, and the range is calculated, which refers to the difference between the maximum and minimum values. The MAX and MIN aggregation functions are used (Equation (2)).

\[
W_j = \frac{1}{J} \sum_{i=1}^{n} \frac{1}{J_i}
\]  

(1)
The priority of each variable was determined considering the results of the systematic review carried out in Section 2 and the capture of requirements applied to ASAPIT beekeepers.

Once the initial data have been calculated, the score or normalized score is identified, assigning a value between 0 and 1 to the value of the captured sample, considering the causal relationship between each variable and the infestation due to *Varroa* infestation [34]. According to the classification of alert levels, Table 3, the score is calculated considering that the temperature variable indicates a low *Varroa* infestation level since it takes higher values (Equation (3)). Likewise, the humidity variables, CO₂ and TVOC, indicate a low level of infestation, which lowers the value taken from the sample (Equation (4)).

\[
x_{nij} = \frac{\text{MAX}_{x_{ij}} - x_{ij}}{\text{MAX}_{x_{ij}} - \text{MIN}_{x_{ij}}} (3)
\]

\[
x_{nij} = \frac{x_{ij} - \text{MIN}_{x_{ij}}}{\text{MAX}_{x_{ij}} - \text{MIN}_{x_{ij}}} (4)
\]

The product between the score and the weight assigned to each variable determines the normalized score for each sample. The basis for the alert level classification is determined by taking the weighted sum of the individual contributions of the four variables. Table 5 and Figure 6 illustrate the algorithm’s application [35].

Using the weighted multi-criteria algorithm allows the collected data to be transformed into normalized data within the range of zero to one. It is less complex for the classification model to identify patterns and alert labels, significantly reducing the computational costs, execution time, and energy consumption [34].

![Figure 6. Flowchart of weighted multi-criteria algorithm.](image)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Weight</th>
<th>Sample</th>
<th>Score</th>
<th>Score × Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature</td>
<td>0.48</td>
<td>22.5</td>
<td>0.8529411765</td>
<td>0.4094117647</td>
</tr>
<tr>
<td>Humidity</td>
<td>0.24</td>
<td>54.8</td>
<td>0.396</td>
<td>0.09504</td>
</tr>
<tr>
<td>CO₂</td>
<td>0.16</td>
<td>600</td>
<td>0.0769230769</td>
<td>0.0123076923</td>
</tr>
<tr>
<td>TVOC</td>
<td>0.12</td>
<td>200</td>
<td>0.08</td>
<td>0.0096</td>
</tr>
</tbody>
</table>
3.3. Classification Models

Using classification models based on supervised learning to detect alerts regarding the health status of hives offers a systematic and automated approach to performing tasks. These models can learn complex patterns from a training dataset, allowing them to make accurate predictions on new data. The ability to process complex data makes machine learning (ML) algorithms a robust and adaptable option for classification in hive monitoring systems, overcoming the limitations of traditional classification methods, such as heuristic rules or expert-based systems.

We divided the models into “white-box” and “black-box” models, representing two contrasting ML approaches (Figure 7). White-box models, such as decision trees, Random Forest, and Gradient Boosting [36], are transparent about how they arrive at their decisions, making them easily interpretable; they provide a clear view of how input features relate to predictions, making the classification process easier to understand. On the other hand, black-box models, such as neural networks, are opaque about their internal workings and do not directly explain their decisions. However, they can capture complex patterns and relationships in data that white box models may miss [37].

![Figure 7. Comparison of classification models.](image)

To date, we have not found a database with actual data that capture the presence of Varroa in hives, considering the discrete variables mentioned above. Therefore, we built a synthetic dataset with 10,000 instances [38] characterizing the hives’ discrete variables following the CRISP-ML(Q) model. The dataset was generated using a Python script designed specifically for this purpose. The script was executed with a series of configurable parameters that allowed the ranges of each class to be adjusted (alert levels considered in Table 3). A descriptive statistical approach was used to examine the class distribution, considering specialized libraries such as Pandas and NumPy for data analysis [39]. As discussed later in this paper, the percentage of the dataset used to train the classification algorithms is an important characteristic, considering the processing and storage capabilities of the hardware used to model deployment. Then, from the dataset built, the C4.5 Decision Tree models were implemented [40] and Random Forest [41], XGBoost (Extreme Gradient Boosting) [36], and neural network [42] algorithms were trained. Figure 5 shows the result of the analysis of the size variation of the training dataset. This analysis shows that white-box models are efficient for the research context because 40% of the data were used for training and 60% for validation. On the other hand, for the black-box model, 20% of the data were assigned for training and 80% for validation.

The original data presented in the study are openly available in Varroa detection with discrete variables on Kaggle at [https://goo.su/39cnh, accessed on 28 April 2024] or [38].

3.4. Hardware Configuration

This section focuses on developing and implementing the solution, considering the essential pillars of IoT. We emphasize the materials and hardware configuration for integrating the classification models.
With the established requirements, the IoT solution proposal is based on accessible, efficient, low-cost devices for beekeepers in developing countries. The functional requirements guided the construction of an IoT system to monitor bee hives, capturing variables such as temperature, humidity, CO₂, and TVOC for approximately three weeks, with weekly notifications on the status of Varroa infestation. A detailed analysis of non-functional hardware and software requirements is needed to achieve these functionalities, including battery consumption and data storage and processing management. Specific sensors, such as the DHT22 for temperature and humidity, the SGP30 for gas concentration, and the FZ0430 for monitoring battery consumption, were selected. The ESP32 board [28] was used for data transmission. Given its compatibility and efficiency in energy consumption, we used the Arduino IDE environment.

3.5. Implementation

Considering the five-layer IoT architecture model for constructing the Internet of Things solutions, the components are identified by layer and represented by a flow diagram (Figure 8). The IoT architecture model provides five defined layers; the perception layer is where the base hardware for the solution is located, such as the aggregator node and the sensor nodes. The network layer transmits information between nodes through Wi-Fi and system–beekeeper communication through the Telegram application. Transversely, there is the battery, which, although part of the perception layer, is in parallel since this component powers the layer to which it belongs and the network layer, mainly the base station node.

The middleware layer is the heart of the solution. It contains the aggregation functions responsible for manipulating and processing the data collected by the components of the perception layer. Within this third layer is the detection mechanism against Varroa infestation, which is the most critical aggregation function for the beekeeper. The application and business layers go hand in hand and are superficial to middleware in that they take the output from the third layer, which is generally an alert-level label, and are responsible for transmitting and documenting it for the end customer. The application layer connects through the network layer with the Telegram application, sending the beekeeper a message to his cell phone with the respective alert level of the hive and the relevant recommendations to act against it (business layer). In this way, the beekeeper can act or make decisions regarding his hive to ensure the health of his colony.

Once the topology, devices, and functional and non-functional requirements of the system have been defined, the scripts for configuring the aggregator and Base Station nodes are developed.

- Hardware configuration of the aggregator node: The aggregator node consists of the configuration of the sensors (SGP30, DHT22) and the ESP32 microcontroller board, as shown in Figure 3. In addition, the FZ0430 sensor has been integrated for sensor management. Following the principles of structured programming, the necessary libraries for the sensors, the Wi-Fi connection, and the transmission protocol (HTTP client/server) are included. In the main loop, the functionalities of the aggregator node are configured, and the battery level is measured before each cycle. With these data, the daily measurement cycle begins, and at the end, the aggregation functions (COUNT and AVG) are applied to send the data to the base station node using HTTP POST. In addition, different exceptions are included to manage possible errors.

- Hardware configuration of the base station node: This node comprises only the ESP32 board, which communicates with the Telegram mobile application through the bot function, as shown in Figure 3. The node includes libraries for connecting to the Internet via Wi-Fi, running the classification model, communicating with the aggregator node, and sending data to the user on Telegram. Wi-Fi network credentials are initially set, and variables are created to store readings and monitoring status. The “sendTelegramMessage” function is configured to send messages via the Telegram API. This function uses the HttpClient class to request HTTP using the POST method.
In the setup of the code, the configuration for the Wi-Fi connection is established, and the connection with the client (aggregator node) is expected; at the same time, the parameters of the classification model are established. In the main loop, the received data are processed, and it is verified that the variable readings are within safe ranges; then, Telegram will send alerts. After five days of monitoring, the averages of the accumulated readings are calculated, and the weighted multicriteria aggregation algorithm is applied, accompanied by the classification model, to determine the alert level. Finally, a detailed report with the alert level and recommendations for the beekeeper are sent to Telegram.

Figure 8. Data flow diagram based on IoT architecture model.

- Integration of classification models in hardware: First, the weighted multicriteria algorithm is applied to integrate the classification model in the hardware (Figure 9), and the generated dataset is processed using the Google Colaboratory development environment. Later, TensorFlow Eloquent, a Python library recognized for simplifying the creation of machine learning models in TensorFlow, was used, following the CRISP-ML(Q) process. In addition, to adapt to microcontrollers, the micromlgen library was used to translate the classification models from Python to C++. The models were implemented in the Arduino IDE environment [43]. An .h file was created for the white-box models and a hexadecimal file for the black-box model. In it, the trained and previously translated model was transcribed using a C++ script. The necessary namespaces were declared in the core sketch (.ino), and it was specified that the trained model should be traversed once the input data vector was received. This ensures continuous improvement, smooth implementation, and efficient execution of the classification model in the hardware environment.
3.6. Description and Approach to Testing

We proposed creating two test scenarios (Table 6) to develop and execute tests.

Table 6. Test scenarios for the aggregator node.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>Cycle with Function Aggregation 24 h cycle with the implementation of aggregation functions and battery optimization.</td>
</tr>
<tr>
<td>S2</td>
<td>Cycle without Function Aggregation 24 h cycle excluding the use of aggregation and battery optimization functions.</td>
</tr>
</tbody>
</table>

Scenario 1 includes all the functionalities implemented in Section 3.2, which refers to the different aggregation and optimization functions for the transmission processes, storage management, and battery management in the aggregator and base station nodes. Finally, it determines an alert level. Scenario 2 excludes the aggregation and optimization functions. Its operation is based on sending and receiving data from the aggregator node to the base station node. The base station node finally runs the classification model and sets the alert level.

Table 7 specifies the metrics used to evaluate the performance of each scenario in each of the tests.

Table 7. Description of evaluation metrics.

<table>
<thead>
<tr>
<th>Test Evaluation Metrics</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Battery consumption in (mAh)</td>
<td>Battery consumption in each scenario for the BSN and AN based on data provided by datasheets and hardware measurements.</td>
</tr>
<tr>
<td>Total number of packages sent</td>
<td>Simulation of packet traffic in a period of 5 days between the AN and BSN for each scenario.</td>
</tr>
<tr>
<td>Memory usage percentage for classification model</td>
<td>Total percentage of flash memory used by the ESP32 board using the Arduino Millis function.</td>
</tr>
<tr>
<td>Run time in seconds and Battery consumption in (mAh)</td>
<td>Run the selected classification model and determine total execution time to estimate the total board consumption when performing this task.</td>
</tr>
</tbody>
</table>

4. Results

This section initially presents the scenarios proposed for the validation tests and also includes results from the implementation of the developed IoT solution, including
an exhaustive validation focused on system optimization and the critical advantages of function aggregation. We established five testing perspectives to evaluate the solution.

4.1. Evaluation of the Energy Consumption of the Solution in Each of Its Nodes

Battery performance tests are performed on each node, considering their specific roles directly affecting battery consumption and duration. For this purpose, information provided by manufacturers through datasheets and direct measurements performed on hardware are used.

Aggregator Node: This test evaluates the performance of the node’s battery with an external power supply or battery. The objective is to analyze the battery capacity concerning operating time, measured in milliampere-hours (mAh). The power consumption of each node component at specific times, both in active and idle mode, is examined. Two scenarios are considered for comparison to understand the energy impact of function aggregation.

Next, total consumption is calculated, considering each component’s duration in seconds (s) in each operating mode for 24 h.

From Table 8, for the active mode in both scenarios, the consumption of the sensor is low since samples are taken for short periods. In Scenario 1, the Wi-Fi module wakes up once a day for 10 seconds to send data, while in Scenario 2, all components, including the Wi-Fi module, are constantly active, quickly depleting the power battery. To the idle mode: although the active mode consumption of the components is higher, the aggregator node remains in active mode for only 0.2% of the total time. It makes the total amount of milliamp-hours (mAh) consumed in an idle mode much higher since this mode operates 9.8% of the time.

Table 8. Total consumption in mAh and total time in seconds of the aggregator node.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Total Active Time (s)</th>
<th>Total Idle Time (s)</th>
<th>Consumption Active (mA)</th>
<th>Consumption Idle</th>
<th>Consumption (mAh)</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>143</td>
<td>86257</td>
<td>3.3455555</td>
<td>224.312553</td>
<td>9.485754618</td>
</tr>
<tr>
<td>S2</td>
<td>798</td>
<td>85602</td>
<td>59.566388</td>
<td>857.6682</td>
<td>38.218108</td>
</tr>
</tbody>
</table>

In Scenario 1, the consumption in idle mode for all sensors, with the low-power light_sleep functionality, is practically zero, on the order of µA. The board’s internal voltage regulator primarily results in battery consumption. In contrast, Scenario Two does not implement low-power features during idle periods, resulting in significant battery consumption and preventing long battery life. Now, an analysis of the performance and duration of commercial batteries that can power the aggregator node is carried out, adapting to the necessary specifications according to the needs of beekeepers. Three battery options are presented. A commercial 9V lithium battery with a capacity of 1200 mAh, with an L7805 voltage regulator to constantly provide 5 volts to the system. As a second option, four AA 1.5 V lithium batteries, with a combined capacity of 3600 mAh, also with an L7805 regulator. Finally, a 5 V power bank with a capacity of 4000 mAh [44].

Figure 10 shows the total operating duration in hours for each battery in each scenario. Battery 1 has the shortest operating duration, powering the node for a single cycle because its mAh capacity is low. In Scenario 1, it lasts approximately 5.25 days, while in Scenario 2, it only reaches 31 h of operation. Battery 2 lasts 15 days in the first scenario and 3.91 days in the second, which limits its ability to complete a monitoring cycle. Battery 3 showed better results in terms of duration. In Scenario 1, the total operation duration was 17.5 days, covering three monitoring cycles, while in Scenario 2, it spanned 4.33 days.
Base Station Node: The node comprises the CPU and the Wi-Fi module of the board, which stores, processes, and sends data. Batteries or the mains can power it, regardless of the hive’s location. Power consumption is calculated and compared in active and idle modes according to the scenarios.

From Table 9, in active mode, it is observed that in Scenario 1, the board is activated once a day to receive data and send a cycle initialization message via Telegram. In Scenario 2, it is activated several times while the Wi-Fi module is in receive mode to receive messages from the base station node. In both cases, data reception is the leading power consumer. Integrating the optimization function “Light_Sleep” in idle mode significantly reduces the board’s power consumption with minimal drain on the internal voltage regulator. In contrast, Scenario 2 lacks power-saving features during idle, resulting in constant power consumption that affects battery life.

### Table 9. Total consumption in mAh and total time in seconds of the base station node.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Total Active Time (s)</th>
<th>Total Idle Time (s)</th>
<th>Consumption Active</th>
<th>Consumption Active</th>
<th>Consumption (mAh)</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>25</td>
<td>86,375</td>
<td>1.35</td>
<td>235.1354</td>
<td>9.7973</td>
</tr>
<tr>
<td>S2</td>
<td>1330</td>
<td>8570</td>
<td>44.125</td>
<td>852.114</td>
<td>35.504</td>
</tr>
</tbody>
</table>

**4.2. Analysis of Data Traffic Injected into the Network**

Data transmission performance tests are carried out to test the premise of reducing traffic injected into the network through function aggregation. We used Wireshark as a tool for network traffic analysis. It allows the transmitted data to be captured and examined in real time, offering a detailed view of the information flow and the protocols used [45].

Considering Table 6, two test scenarios are again analyzed. In this case, the simulation of a five-day cycle is carried out.

Figure 11 presents the total packet traffic between the nodes during each scenario’s data transmission and reception processes. In this test, the tool captured 150 packets over a 5-day cycle in Scenario 1, setting an average of approximately 30 packets daily. In contrast, in Scenario 2, 2887 packets were captured, equivalent to 578 per day.
4.3. Analysis of Hardware Memory Occupancy Concerning the Classification Model Used

This test comprehensively examines how implementing a high-accuracy classification model intended for *Varroa destructor* mite detection impacts the flash memory occupancy of the network BSN. It allows informed decisions regarding model selection and dataset size for respective training.

Figure 12 shows that tree-based models increase their size and memory footprint as the training dataset grows. In contrast, the neural network-based model maintains a stable memory occupancy of approximately 75%, regardless of the training dataset size. To guarantee the correct operation of the BSN CPU, up to 85% of the flash memory can be used, leaving the remaining 15% for executing additional tasks [28].

Figure 13 shows the relationship between flash memory occupancy and accuracy for each implemented model. It is highlighted that the neural network is the model that exhibits the best performance regarding the relationship between memory resource constraints and accuracy. In contrast, tree-based models, such as XGBoost, show a high level of accuracy but consume a considerable amount of flash memory resources, reaching their maximum limit. This situation affects the performance of the BSN during the execution of daily tasks. On the other hand, the C4.5 Tree and Random Forest models are discarded due to their low accuracy in the classification task, even though they require a low amount of flash memory for their implementation.
4.4. Evaluation of the Energy Consumption of Each Implemented Classification Model

This section compares the energy consumption of the four classification models to understand how their execution impacts energy consumption and determine the best solution. For this purpose, a script is proposed using Arduino IDE, which allows each model to execute a certain number of classifications continuously. It offers a vision regarding the time necessary for the card to complete the classification process and thus estimate your energy consumption.

From Table 10 and Figure 14, among the three white-box classification models, Tree C4.5 shows superior performance in terms of battery consumption and lower execution time; this model, being a single tree, can perform classifications faster compared to Random Forest and XGBoost models and employ multiple trees for the classification process. The difference between these two models is that in Random Forest, the prediction is performed by averaging or voting the predictions of each tree individually, which can be more efficient in terms of runtime [46]. In contrast, in XGBoost, the prediction process involves summing the predictions of multiple sequential trees and applying additional regularization processes [47].

Table 10. Classification time of the model’s dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>Average Runtime (ms)</th>
<th>Total Time (h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tree C4.5</td>
<td>123.246</td>
<td>0.34235</td>
</tr>
<tr>
<td>Random Forest</td>
<td>129.825</td>
<td>0.360625</td>
</tr>
<tr>
<td>XGBoost</td>
<td>143.791</td>
<td>0.39941944</td>
</tr>
<tr>
<td>Neural Network</td>
<td>124.014</td>
<td>0.34449166</td>
</tr>
</tbody>
</table>

Figure 13. Flash memory consumption about the accuracy of the implemented classification models.

Figure 14. Total consumption in milliamp-hours for the classification of the dataset of each model.
5. Discussion

By implementing function aggregation on the aggregator node, such as AVG and COUNT for Scenario 1, the captured data are synthesized at the end of the day, temporarily storing the data in flash memory and sending the data only once. In comparison, in Scenario 2, where the data are sent immediately after their capture via Wi-Fi, higher energy consumption is observed, the main differentiating factor in the consumption in active mode in the aggregator node. Furthermore, implementing optimization features such as `light_sleep` on the card reduces battery consumption significantly during idle periods for the two nodes, in contrast to Scenario 2, where their absence makes a notable difference.

Based on the established battery management processes, the total battery lifetime is determined by considering operating conditions when the battery charge is greater than or less than 30%. Battery 1 operates for four days in high and one day in low battery mode, completing only one cycle. Battery 2 lasts eleven days in high and four days in low, equivalent to three complete cycles. On the other hand, battery 3 operates for twelve days in high battery mode and five days in low battery mode, covering three full cycles. During the remaining time, notifications about abnormal ranges or sensor problems and recommendations are sent to beekeepers.

The traffic analysis results show that function aggregation effectively controls the amount of data transmitted, reducing approximately 95% of the total data load. However, considerable packet loss is observed in Scenario 2, which highlights the importance of implementing feature aggregation to improve the reliability and efficiency of network communication. Regarding the size of the training datasets of the models and their implementation, it was decided to select a size of 0.4 for tree-based models, equivalent to 40% of the data for training and 60% for validation. The choice is since the XGBoost model exhibits the highest classification accuracy (94%) and reaches the possible memory limit since it occupies 84% of the flash memory, compared to the other two models (Random Forest and Tree C4.5), whose memory occupations range from 69% to 73%. For the black-box model, a training data distribution of 20% was chosen, and the remaining 80% was reserved for validation. It resulted in a flash memory occupancy of 74%, which resulted in an accuracy of 99%.

Maintaining an appropriate balance between model accuracy and memory occupancy must be considered. The results emphasize the need to find an optimal point where the model accuracy is high while minimizing the consumption of memory resources. This balance ensures the system’s efficient operation in practical environments when facing resource limitations such as those in the BSN.

When comparing the white-box models with the black box, it is observed that the neural network offers fast and efficient battery consumption performance and is on par with the Decision Tree. The results obtained from this comparison reveal the importance of considering both predictive performance and resource consumption when selecting a classification model for practical applications. While the C4.5 Decision Tree emerges as a promising option in terms of efficiency in battery consumption and execution time, the neural network also offers competitive and efficient performance.

Implementing a classification model on the ESP32 [28] microcontroller board represented a significant challenge in practice, mainly because some of the mechanisms used as a basis from other research available in repositories or developer forums needed to meet this research’s requirements satisfactorily. While some approaches differed in nature or scope, others needed more flexibility to adapt to the project’s specific demand. Despite this, the advantages of white-box and black-box models in terms of accuracy, scalability, and interpretability are considered, such as heuristic models or models based on expert knowledge. The research focused on exploring various alternatives designed to address the limitations above and optimize model performance on the board.

After exhaustive research in multiple sources, it was determined that incorporating libraries such as `EloquentTinyML` or `tflm`, developed by [48], surpasses other alternatives, as they are designed for devices with limited resources. These libraries optimize the use of memory and processing. Therefore, this study considers them and is valid as a complete
solution that addresses all the identified limitations. Additionally, they offer a wide range of functionality, integrating multiple machine learning models and enabling sophisticated and versatile solutions in embedded systems.

6. Conclusions and Future Works

This research characterized the climatic conditions affecting the presence of Varroa mites in bee hives, using discrete variables as critical indicators. Significant findings were obtained through a detailed analysis of climatic data, mite infestation records, and expert input. Potential limitations were noted, such as the lack of available information and studies, and caution was made against natural climate variability that may affect data interpretation. Although climatic patterns related to the Varroa mite were identified, it was highlighted that correlation does not imply direct causation. In addition, the possible influence of other factors not considered was mentioned, such as human activity and the use of chemical treatments in hives.

Throughout this research, we encountered limitations when selecting and implementing devices to monitor Varroa mites. To address these limitations, we added multiple additional functions. This allowed for more efficient management of monitoring cycles in the hives, adjusting them according to beekeepers’ needs. Additionally, we conducted a detailed analysis of power consumption to design a solution that minimized battery usage without compromising functionality and accuracy. The integration of these functions facilitated effective management of data capture and transmission times, resulting in maximum battery savings from the card and, simultaneously, a significant reduction in network traffic. Finally, thanks to these efforts, we managed to reduce the battery consumption of the devices by up to 75% compared to implementation without these functions and reduce network traffic by 95%.

Integrating functions as part of our data preprocessing strategy significantly reduced redundancy, reducing the amount of data from 720 to a vector of 4 highly representative aggregated data points. This approach not only optimized storage on limited hardware but also allowed for more efficient management thereof. By combining this hardware with a supervised machine learning model, specifically a neural network, we developed a comprehensive solution for Varroa mite detection. Of all the models tested, the neural network was the most suitable choice, as it allowed us to strike a balance between high accuracy (99%) and low hardware memory consumption (74%). This lays the groundwork for future research in precision beekeeping and automated monitoring of bee health, seeking to overcome the limitations of traditional detection methods, such as Varroa testing, which are laborious and require high human intervention.

Considering the interest of beekeepers in implementing more sensors and devices in the hives and developing additional functionalities based on the collected data, a future approach to designing and creating a more complete and adaptable monitoring and control system is suggested. This system could integrate additional sensors to capture relevant data such as daily bee flow and hive weight and develop advanced algorithms to identify complex patterns related to colony health.

As possible future research work, developing a more robust predictive model for forecasting Varroa mite occurrence in beehives is suggested. It implies considering other climatic constraints, parameters, and variabilities, such as local topography, seasonal variability, and human activity related to beekeepers’ hive management.

In addition, long-term studies are proposed to validate the model’s effectiveness in different regions and climatic conditions, improving the understanding of the relationship between climate and mite infestation.

Author Contributions: A.I.C.C., M.A.C.M. and J.C.C. proposed the concept of this research. A.I.C.C., M.A.C.M. and J.C.C. contributed to the state of art and final paper draft revisions. A.I.C.C., M.A.C.M. and J.C.C. wrote the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by Universidad del Cauca, Popayán, Colombia.

Institutional Review Board Statement: Not applicable.
Data Availability Statement: Data are contained within the article.

Acknowledgments: We would like to thank the Universidad del Cauca and the Beekeeping and Agroindustrial Association (ASAPIT) for participating in the data collection and for providing the interviews and feedback on this work.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations
The following abbreviations are used in this manuscript:

- CO₂: Carbon Dioxide
- TVOC: Total Volatile Organic Compounds
- ASAPIT: Beekeeping and Agroindustrial Association of Piendamó and Tunía
- IoT: Internet of Things
- ML: Machine Learning
- AN: Aggregator Node
- BSN: Station Base Node
- CDA: Centralized Data Aggregation
- AVG: Average
- CPU: Central Processing Unit
- XGBoost: Extreme Gradient Boosting

References


38. Caicedo Camayo, A.I. Varroa Detection with Discrete Variables: Synthetic Dataset for Varroa Detection from Discrete and Normalized Variables; Kaggle: Popayán, Colombia, 2024. Available online: https://www.kaggle.com/datasets/16945694fbf6c04c6e6d51b52a06548ba88c386647c632f9d68c274106112271 (accessed on 4 April 2024).


Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.