UAB
Universitat Autònoma de Barcelona

Dipòsit digital de documents de la UAB

This is the **published version** of the bachelor thesis:

Arpa Robig, Miguel Leonardo. *Multi-Sensor Analysis and Early Detection of Swarming Events in Honey Bee Colonies Using Machine Learning*. Treball de Final de Grau (Universitat Autònoma de Barcelona), 2026 (Grau en Intel·ligència Artificial)

This version is available at https://ddd.uab.cat/record/336306

under the terms of the  license.

Multi-Sensor Analysis and Early Detection of Swarming Events in Honey Bee Colonies Using Machine Learning

Miguel Arpa Robig

June 28, 2026

Abstract

This project investigates early swarming detection and hive management prediction using multi-sensor IoT data from beehives monitored over three seasons (2023–2026) at the UAB campus apiary, covering 23 confirmed swarming events across 8 instrumented hives.

A preliminary behavioural analysis identified two biologically interpretable pre-swarm signatures: a systematic degradation of the weight–temperature Pearson correlation beginning 14 days before departure ($p < 0.001$) and nocturnal temperature increases of $+2$–$+8$ ℃ in the pre-swarm week. Both signals were incorporated as explicit model features.

Three complementary predictive models were developed and evaluated under strict walk-forward temporal cross-validation: a multi-horizon model (XGBoost and LSTM) at 3-, 7- and 14-day horizons; a morning-window classifier with features optimised via Optuna; and a night-enhanced variant adding 13 nocturnal aggregations. XGBoost and both uni- and bidirectional LSTM architectures were compared across all configurations. The best deployed swarm model — a unidirectional LSTM on the multi-horizon daily feature set — achieves G-Mean = 0.866, detecting **13 out of 14** test swarming events 3 days in advance. An exploratory honey super placement model is presented as a secondary application.

Keywords: Swarm detection, beehive monitoring, XGBoost, LSTM, deep learning, walk-forward cross-validation, IoT sensors, precision apiculture

1 INTRODUCTION

Beekeeping plays an important ecological and economic role. Honey bees (*Apis mellifera*) are responsible for pollinating approximately 75% of the world's food crops [1] and their colonies represent a significant productive asset for beekeepers. Two of the most operationally relevant processes in colony management are **swarming** and **honey super placement**, both of which depend on timely beekeeper intervention.

Swarming is the natural reproductive mechanism of honey bee colonies. When a colony grows large enough, the old queen departs with a large number of worker bees (the exact count varies with colony size) to establish a new nest, while the remaining bees raise a new queen. Events occur predominantly in spring (March–June) and follow a recognisable behavioural sequence: the colony raises queen cells, scout bees locate a new nest site and the swarm departs during a brief window of warm, calm weather. A single event can reduce the colony's productive capacity by 30–60%; if the swarm is not recovered, it represents a permanent loss. Traditional monitoring relies on manual hive inspections, which are time-consuming, stressful for the bees and impractical at scale.

Honey super placement refers to the addition or removal of supplementary boxes stacked above the brood chamber to provide extra space for honey storage during the nectar flow season. Currently, beekeepers rely on periodic inspections to make this decision based on visual assessment of colony strength and available space.

The emergence of affordable IoT sensors has enabled continuous, non-invasive monitoring of beehive parameters such as weight, temperature, humidity and acoustic frequency. These streams generate large volumes of time-series data that, when analysed with machine learning (ML) and deep learning (DL) techniques, can provide **automatic early warning of swarming events** days in advance — at horizons of 3, 7 and 14 days — allowing timely intervention, as well as data-driven support for honey super placement decisions.

• Contact E-mail: MiguelLeonardo.Arpa@autonoma.cat
• Supervised by: Vicenç Soler (Departament de Microelectrònica i Sistemes Electrònics)
• Academic Year 2025/26

This project analysed real sensor data collected under the **UABee** initiative [19], an interdisciplinary research and teaching project led by Professor Emeritus Gerardo Caja at the Universitat Autònoma de Barcelona. The UABee experimental apiary, established in May 2019 at the UAB Bellaterra campus, provided three consecutive seasons of continuous hive monitoring (2023–2026) across 18 instrumented hives, yielding 23 confirmed swarming events.

The main objectives of this project are:

- Describe multi-sensor behaviour before, during and after confirmed swarming events and identify biologically interpretable pre-swarm signatures.
- Predict swarming events from continuous IoT sensor data.
- Explore predictive indicators for honey super placement as a secondary hive management application.

This paper is organised as follows. Section 2 reviews related work on IoT-based beehive monitoring and ML-based swarm detection. Section 3 describes the dataset and exploratory analysis. Section 4 presents the behavioural analysis of pre-swarm signatures. Section 5 details the methodology. Section 6 reports experimental results. Section 7 presents the production deployment. Sections 8 and 9 discuss findings and draw conclusions.

2 Background and Related Work

2.1 IoT-Based Beehive Monitoring

The use of electronic sensors in beehives has increased significantly in recent years. Both research and commercial systems typically monitor hive weight, internal temperature [2], humidity and acoustic signals. Weight measurements are particularly useful, as a sudden decrease of 1–5 kg within a few hours is often associated with a swarm departure [2]. Internal temperature is also a key variable: healthy colonies regulate brood temperature within a narrow band of 34–36°C through collective thermoregulation [3] and deviations from this range can indicate changes in colony activity or health.

An additional signal of growing interest is the concentration of CO_2 inside the hive, which rises significantly above ambient outdoor levels due to bee respiration [12]. Preliminary observations suggest that CO_2 may show characteristic variations before swarming events, linked to increased bee activity. Although the evidence remains limited, CO_2 represents a potentially informative additional signal for early detection. It is important to note that the current UABee dataset does not include CO_2 measurements, as this sensor has not yet been deployed in the apiary; its integration is left as future work.

2.2 Colony Behaviour and Key Management Events

Honey bee colonies follow regular daily and seasonal activity patterns that directly motivate sensor-based monitoring. Flight activity peaks in the late morning (approximately 10:00–14:00h) under warm, calm conditions, producing characteristic daily drops in hive weight as foragers depart to collect nectar and pollen. This periodicity makes morning-window aggregations particularly informative for detecting pre-swarm anomalies.

Swarming in the Mediterranean climate occurs predominantly between March and June, driven by the spring population peak and nectar flow. Colony preparation begins one to two weeks before departure: the colony raises queen cells, scout bees start locating new nest sites, and foraging activity progressively decouples from weather conditions as resources are redirected toward swarm preparation. These physiological and behavioural changes produce detectable sensor signatures in the days preceding the event, which motivates the use of 7- and 14-day prediction horizons in this work. Honey super placement is required when the brood chamber nears full capacity: a congested colony lacking storage space is at elevated risk of swarming. Timely addition of supers therefore serves a dual role — expanding honey production capacity and reducing congestion-induced swarming pressure.

2.3 Machine Learning for Swarm Detection

Several studies have applied ML techniques to beehive monitoring. Acoustic-based approaches have been used to detect queen presence and predict swarming [4, 5]: changes in the acoustic spectrum are observable hours before a swarm occurs. More recent work combines multiple sensor signals — temperature, weight, and acoustics — which generally improves detection performance compared to single-variable approaches [6].

A persistent challenge in this domain is **class imbalance** [18]. Under these conditions, standard accuracy is uninformative, as a classifier that never predicts a swarm already achieves high accuracy. The literature therefore recommends metrics that evaluate both classes simultaneously. This work uses G-Mean as the primary model selection criterion, complemented by event-level detection counts to assess operational utility.

2.4 Limitations of Prior Work

Despite promising results, existing approaches share several limitations that motivate this work: (1) very few labelled swarming events, limiting generalisation; (2) many systems focus on a single sensor modality; (3) metabolic signals such as CO_2 are rarely exploited despite their potential relevance; and (4) most approaches detect swarming during or after the event rather than providing advance warning.

3 Dataset and Exploratory Analysis

3.1 Dataset

The dataset used in this project comes from the apiary located at the UAB campus. Sensor data is collected and exported periodically as Excel files by the **BeeHive Monitoring** platform [20], which provides continuous hive telemetry accessible through an online dashboard.

The UABee apiary currently counts 18 instrumented hives. This work focuses on the hives for which confirmed swarming event labels and sufficient continuous sensor data

were available across the monitoring period: 8 hives for the swarm prediction models and 17 out of 18 for the honey super model (only hives with registered super changes are included). Certain periods were excluded from the analysis — in particular, the annual UAB campus festival, during which unusual human activity near the apiary produces atypical colony behaviour not representative of normal dynamics.

Each instrumented hive is equipped with two complementary sensor nodes that log data in interleaved periods at a combined cycle of approximately 10 minutes. Table 1 lists the raw columns exported by the platform.

Raw readings span January 2023 to June 2026. Heart variables provided intermittent data (sensors not always available), while the scales were always online.

Column	Description	Unit
`Hive name`	Hive identifier	—
`Time`	Measurement timestamp	—
`Temperature heart`	Internal temperature	°C
`Humidity heart`	Internal humidity	%
`Frequency`	Acoustic frequency	Hz
`Volume`	Acoustic volume	—
`Temperature scale`	External temperature	°C
`Humidity scale`	External humidity	%
`Weight`	Hive weight	kg

TABLE 1: Raw sensor columns exported by the BeeHive Monitoring platform.

3.2 Confirmed Swarming Events

A total of 23 swarming events were confirmed by the responsible beekeeper across 8 hives over three seasons (Table 2). Following daily feature aggregation and a train/test split at 2026-01-01, the training set contains 3,308 rows (9 events, 0.76% positive) and the test set 944 rows (14 events, 3.92% positive). This severe class imbalance [18] motivates the use of specialised metrics and cost-sensitive learning, described in Section 5.3.

Swarming events show a characteristic intraday pattern: acoustic frequency rises during the morning window (10–14h) and weight drops at the moment of colony departure. This pattern was confirmed by the responsible beekeeper and is corroborated by the exploratory analysis presented in Section 3.3; it directly motivated the morning-window aggregation strategy described in Section 5.1.

Hive	Swarms	Period
Box1	2	Apr 2023, Apr 2026
Box2	1	May 2024
Box3	5	Apr 2025, May 2026
Box4	4	Apr 2025, Apr–May 2026
Box5	2	Apr–May 2026
Box8	3	Apr–May 2026
Box13	4	Apr 2025
Box14	2	Apr 2025, Jun 2026
Total	**23**	2023–2026

TABLE 2: Confirmed swarming events by hive across three monitoring seasons. Train set: 9 events before 2026. Test set: 14 events in 2026.

3.3 Seasonal and Intraday Patterns

Exploratory analysis reveals consistent seasonal weight cycles across hives: rapid growth from February to June (nectar intake and colony expansion), a summer plateau and an autumn decline. Daily amplitude (max − min weight per day) serves as a foraging intensity proxy. Hives exhibit markedly different strategies: Box13 shows the highest daily amplitude (3.67 kg in summer) despite moderate absolute weight, indicating intensive foraging activity, while Box3 accumulates the heaviest reserves (84 kg peak in May–June) with minimal daily variation. These inter-hive differences motivated the use of relative features (deltas, rolling ratios, spike-ratios) rather than absolute sensor values in all ML models.

Figure 1 illustrates these patterns for Box 3, the heaviest hive in the dataset. The intraday profile confirms the standard foraging window (departure ∼8h, return ∼20h), most pronounced in spring, and the monthly distribution shows a sharp weight peak in May (∼85 kg) followed by rapid decline through summer—consistent with honey harvest and reduced nectar availability. These observations align with the colony behaviour described in Section 2, confirming that the sensor data captures the expected seasonal and intraday dynamics as a baseline for anomaly detection.

4 Behavioural Analysis

4.1 Weight–Temperature Correlation Breakdown

Under normal foraging conditions, hive weight shows a positive Pearson correlation with external temperature ($r \approx +0.15$ to $+0.50$): warmer days promote foraging and nectar collection. Inspired by Cecchi et al. [7], who reported daily correlations between CO_2, weight, temperature and humidity inside the hive and given that CO_2 is not available in the current dataset, a 14-day rolling Pearson correlation between daily hive weight and external temperature was computed across all hives with multiple confirmed events.

Figure 2 shows this relationship for Box 13 across three behavioural periods. Under normal conditions the correlation is positive ($r = +0.31$): heavier foraging traffic on warm days adds nectar weight. In the 14 days before a swarm the sign reverses ($r = -0.34$): the colony stops responding to weather cues as it redirects activity toward swarm preparation. After departure the correlation partially recovers ($r = +0.19$), consistent with resumed foraging by the reduced colony. The same sign reversal was observed across all hives with multiple confirmed events, with the largest standardised drops in Box3 ($\Delta z \approx -0.86$) and Box13 ($\Delta z \approx -0.49$).

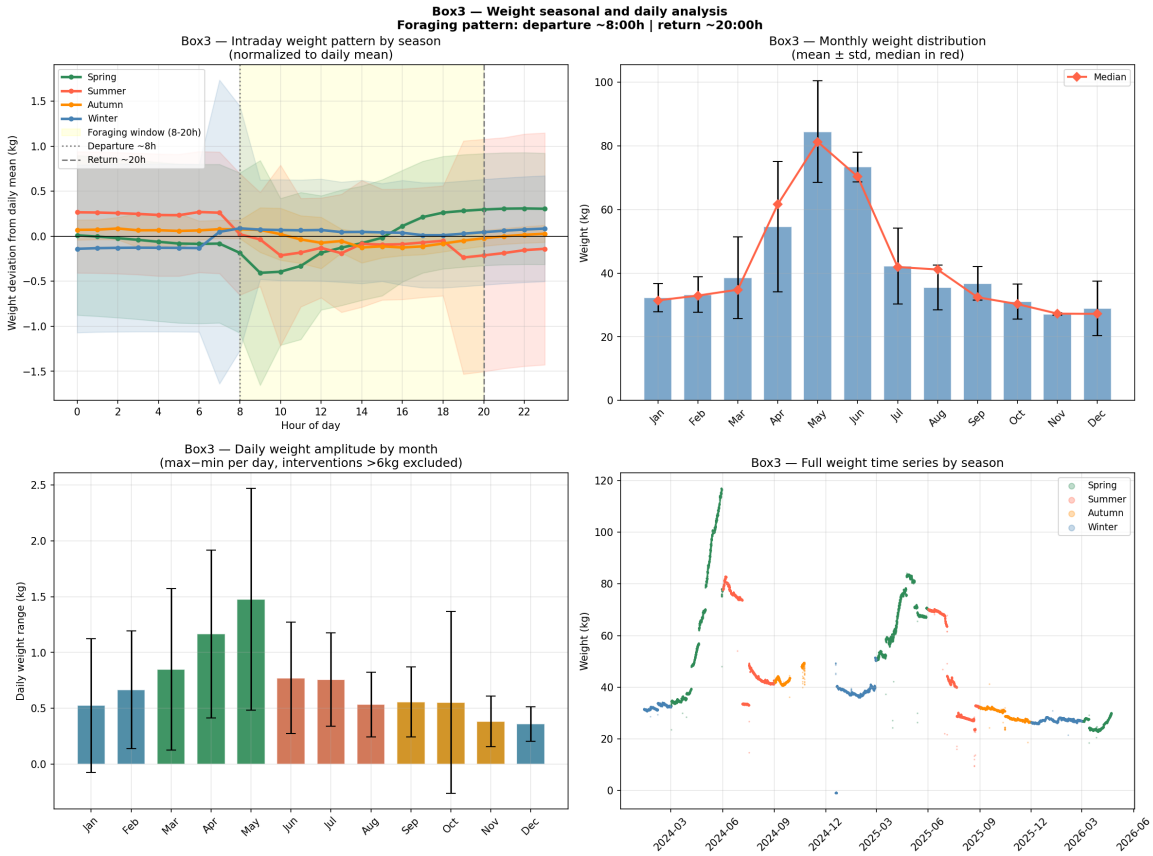


Fig. 1: Box 3 seasonal and daily weight analysis. *Top-left*: intraday weight deviation from daily mean by season; the shaded band marks the foraging window (8–20h). *Top-right*: monthly weight distribution (mean ± std, median in red). *Bottom-left*: daily weight amplitude (max−min) by month, excluding interventions >6 kg. *Bottom-right*: full weight time series coloured by season (2024–2026). Box 3 exemplifies the high-reserve, low-daily-amplitude strategy, in contrast to hives such as Box 13 which show intense daily foraging with moderate total weight.

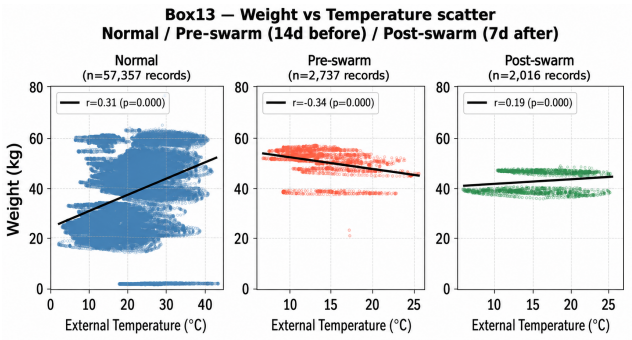


Fig. 2: Weight–temperature scatter for Box 13 across three behavioural periods ($n = 57{,}357$ normal; $n = 2{,}737$ pre-swarm; $n = 2{,}016$ post-swarm). Under normal conditions the correlation is positive ($r = +0.31$); in the 14 days before a swarm the sign reverses ($r = -0.34$); after departure it partially recovers ($r = +0.19$). All coefficients are significant ($p < 0.001$).

This pattern is biologically consistent with the progressive reduction in foraging engagement that precedes swarming [10]: as energy is redirected toward queen-cell development and scout bee activity, the colony decouples from external temperature cues.

Figure 3 shows the full sensor correlation matrices for Box 3 under both periods.

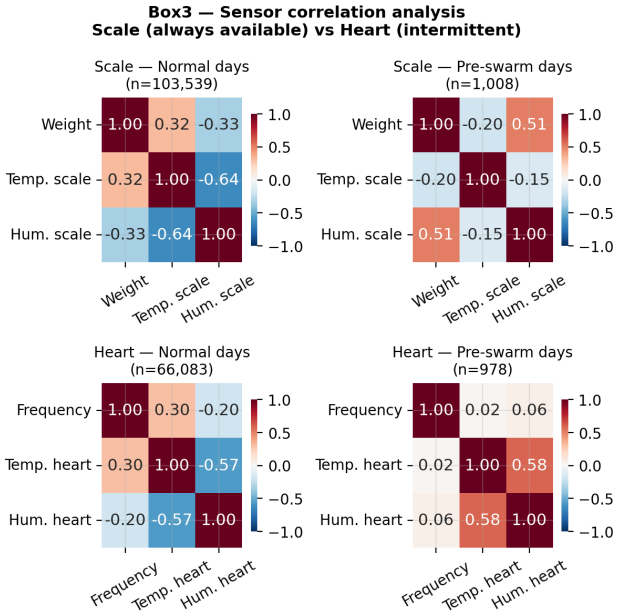


Fig. 3: Sensor correlation matrices for Box 3 under normal (left) and pre-swarm (right) conditions. The weight–temperature correlation reverses sign (from $+0.32$ to -0.20) in the 14-day pre-swarm window, and acoustic frequency decouples from internal temperature (from $+0.30$ to $+0.02$). Both observations motivate the `corr_w_temp` rolling-correlation feature.

4.2 Nocturnal Pre-Swarm Signals

Ferrari et al. [4] report that acoustic frequency rises by approximately +150 Hz and internal temperature increases by +2°C during the nocturnal window (00:00–05:00) in the 7 days preceding swarming. To validate these signatures, we computed nocturnal aggregates (mean frequency and temperature during 00:00–05:00) and compared the 7-day pre-swarm window against a *baseline period* — defined as days 60 to 7 before each confirmed event, representing undisturbed colony behaviour unaffected by swarm preparation — for each confirmed event.

Figure 4 shows the per-event nocturnal changes alongside the Ferrari et al. [4] reference values. Internal temperature increases of +2 to +8°C are consistently observed across all hives and seasons. Nocturnal frequency changes are more variable and none of the observed events reaches the Ferrari et al. [4] +150 Hz threshold: Box4 (+76 Hz) and Box3 (+59 Hz) show the largest increases, while Box2 shows no significant change and Box1 shows a decrease. This suggests that acoustic pre-swarm signatures are hive-dependent and likely influenced by differences in sensor placement or bee population, whereas the nocturnal temperature signal is more universal. These findings motivated the addition of nocturnal features to the model.

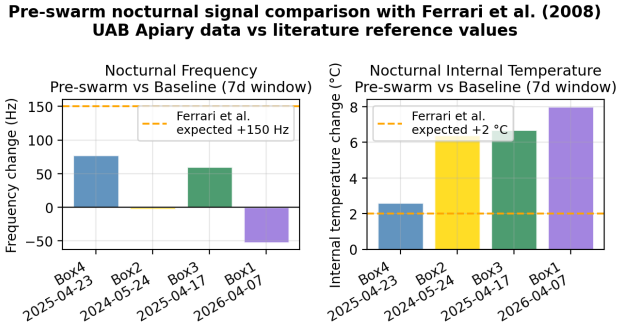


Fig. 4: Nocturnal pre-swarm signal comparison against Ferrari et al. [4] reference values (+150 Hz, +2°C). Internal temperature increases are consistent across all events; frequency changes are hive-dependent.

5 Methodology

5.1 Feature Engineering

Four model variants are developed: three for swarm detection, each built on a progressively richer feature set and a fourth addressing honey super placement prediction. All share the same preprocessing pipeline: the two interleaved sensor streams per hive are merged into a regular 10-minute grid; gaps are forward-filled up to 8 consecutive steps (covering absences up to 80 minutes). This threshold reflects the distinction between a transient sensor dropout — likely a brief communication delay — and a structurally missing period: beyond 2 hours, a gap may indicate a real event such as a sensor removal, battery failure or beekeeper intervention, and filling it would add artificial values across a span large enough to corrupt rolling features computed over hours or days. Gaps exceeding this limit therefore remain as NaN.

To carry out the project, four different configurations have been developed. Table 3 shows these configurations.

C	Task	Features	Horizon
01	Swarm-Daily aggregates	24	3/7/14 d
02	Swarm-10-min + morning win.	31	3 d
03	Swarm-10-min + morn.+night	44	3 d
04	Honey super-Daily aggregates	19–52	7/14 d

TABLE 3: Overview of the four model variants. Feature counts are after NaN-fraction filtering on the training split.

Description of the four configurations:

Daily-aggregated features (Configuration 01). The multi-horizon (3, 7 and 14 days) swarm model operates on one row per hive per day (daily average per variable), constructed from the full daily sensor profile. The 24-feature set covers three domains: weight dynamics (*Weight*, 7- and 14-day moving averages, 7-, 14- and 21-day differences, rolling trend slope and acceleration and 3-day weight drop); acoustic signals (*Frequency* and *Volume* with 7-day moving averages and trend); and a 14-day rolling Pearson correlation between daily hive weight and external temperature (`corr_w_temp`), which captures the correlation breakdown identified in Section 4. Swarm history is encoded via `days_since_last_swarm` and `n_swarms_this_season`; and external and internal temperature (*Temp_scale*, *Temp_heart*) with a 7-day trend, which provide ambient context for the acoustic and weight signals. For prediction windows of 7 and 14 days, training positives are augmented by ±1 day to compensate for the scarcity of labelled events; the 3-day horizon is excluded from augmentation as the signal is too noisy at short range.

Morning-window features (Configuration 02). Building on the daily feature set, this model additionally aggregates 10-minute readings from the peak pre-swarm activity window (10:00–14:00) identified in Section 3.3. The resulting per-day statistics include weight drop, standard deviation and range within the window; frequency mean, maximum and standard deviation; volume mean and maximum; and the frequency spike ratio (`freq_spike_ratio`), defined as the ratio of the morning maximum frequency to its 7-day rolling median — a normalised measure of acoustic bursting relative to the hive's recent baseline. These are combined with the daily-level features and the engineered `weight_std_roll14`, the 14-day rolling standard deviation of daily mean weight. An ablation study over 7 candidate signals confirmed that `weight_std_roll14` is the only feature addition that consistently improves the ML methods used in this project across both different windows; narrowing the morning window from 10–14h to 12–14h was also tested and rejected due to higher false alarm rates. The final feature count is 31.

Night-enhanced features (Configuration 03). This variant extends Configuration 02 by appending 13 nocturnal aggregations extracted from the 00:00–05:00 window: mean and maximum nocturnal frequency with 3- and 7-day moving averages; a nocturnal frequency spike ratio;

mean nocturnal volume; mean overnight weight; overnight weight delta (morning weight minus evening weight) with 3-day smoothing; and the morning-to-night frequency ratio (`night_to_morning_freq`), which captures how much the acoustic profile shifts between the two activity windows. The total feature count grows from 31 to 44. This variant achieves the highest AUC among the 3-day swarm models.

Honey super features (Configuration 04). The honey super (`alza`) prediction task uses daily aggregates from 17 hives, restricted to the active beekeeping season (months 2–6). The feature set (after excluding temporal features) combines four complementary information sources. *Weight dynamics* capture both short- and long-term colony growth through 1-, 7-, 14- and 21-day weight differences, moving averages, rolling variability, trend slope, acceleration and a 7-day positive-day count. *Hive-relative historical features* situate the current weight within the hive's own baseline: a z-score against the historical mean (`weight_vs_historical`), the percentage of the expanding 95th-percentile maximum (`weight_pct_of_max`) and the current consecutive-day weight gain streak. *Apiary-relative features* encode the beekeeper's domain knowledge that honey super readiness is a comparative judgement: z-scores of weight and 7-day growth rate against the daily apiary mean, daily weight and growth percentile ranks and the apiary weight ratio. *Intervention timing* completes the set: `days_since_last_alza`, `days_since_last_ADD` and `n_alzas_this_season`. Three derived binary flags (`above_min_weight`, `overdue_for_alza`, `small_but_growing`) encode threshold-based expert rules. The six temporal features (month, day-of-year and their cyclical encodings) were excluded via ablation — their inclusion caused the model to exploit seasonal calendar patterns rather than learning biologically meaningful signals.

Common conventions. In all swarm models, weight features are supplemented with daily apiary-relative z-scores to reduce sensitivity to inter-hive weight differences. The feature `days_since_last_swarm` is assigned a sentinel value of 999 when no prior event exists for a hive. In Configurations 02 and 03, a companion binary flag `has_prior_swarm` is added to allow the model to distinguish "no history" from "a long time ago."

Target variables. For swarm models, the binary label is defined as $y_d = 1$ if a confirmed swarming event occurs in the interval $[d + 1,\ d + 3]$, i.e., within 3 days ahead. Configuration 01 additionally evaluates 7- and 14-day horizons. The resulting class imbalance is 0.76% positive in the training set and 3.92% in the test set. For the honey super model, $y_d = 1$ if any intervention (addition or removal of a honey super) occurs within $[d+1,\ d+14]$, giving an imbalance of 11.8% positive in the 2026 test set (237 out of 2,012 samples).

5.2 Model Architectures

Two families of binary classifiers are evaluated across all configurations. **Gradient-boosted tree ensembles** (XGBoost [8], LightGBM [21]) combine hundreds of shallow decision trees trained sequentially, each correcting the residual errors of the previous one. They are well-suited to heterogeneous tabular data and require no feature normalisation, provide interpretable feature importance, and natively support class-imbalance weighting. **Recurrent Neural Networks** (LSTM, Bi-LSTM)[9] maintain a hidden state that is updated at each time step, allowing the model to capture dependencies across sequences of consecutive daily vectors; the bidirectional variant processes the sequence in both forward and backward directions before classification. Comparing both families allows us to assess whether the temporal structure of the input sequence adds predictive value beyond what is already captured in the engineered rolling features. For swarm detection (Configurations 01–03) both families are trained and compared; for honey super prediction (Configuration 04) XGBoost is used as the primary model, with LightGBM as a secondary comparison to confirm that results are not architecture-dependent.

XGBoost (swarm models, Configurations 01–03). A gradient-boosted decision tree ensemble [8] is trained on the daily feature vectors. For Configurations 02 and 03 the baseline configuration uses `n_estimators` = 400, `max_depth` = 4, `learning_rate` = 0.05 and `subsample` = `colsample_bytree` = 0.8. Configuration 01 uses a larger ensemble (`n_estimators` = 500) with shallower trees (`max_depth` = 3, `learning_rate` = 0.02) and additional regularisation (`gamma` = 0.1, `reg_alpha` = 0.2) to improve generalisation across three prediction horizons. Class imbalance is addressed by setting `scale_pos_weight` to the negative-to-positive ratio of the training split. Remaining NaN values are imputed with the per-feature median computed on the training split only. For all three swarm Configurations, hyperparameters are further refined with Optuna [11] (50 trials, Tree-structured Parzen Estimator sampler, objective: walk-forward Average Precision across the 2025 and 2026 folds).

LSTM (swarm models, Configurations 01–03). A unidirectional LSTM and a bidirectional LSTM (Bi-LSTM) [9] are compared in each Configuration. For Configurations 02 and 03, both fixed architectures stack 2 recurrent layers with hidden dimension 64, apply dropout with probability 0.3 between layers and feed the last hidden state into a single linear layer with sigmoid activation. The input to each forward pass is a sequence of **14 consecutive daily feature vectors** per hive. Configuration 01 uses sequences of **21 days** to accommodate prediction at longer horizons.

All features are standardised to zero mean and unit variance using a `StandardScaler` fitted exclusively on the training split; residual NaN values are then filled with 0. Training uses the Adam optimiser with $\mathrm{lr} = 10^{-3}$ and weight decay 10^{-4}, and is stopped early when validation loss fails to improve for **15** consecutive epochs. Class imbalance is handled through a weighted `BCEWithLogitsLoss`, where the positive-class weight equals the negative-to-positive ratio of the training split.

A separate Optuna search (20 trials, TPE, objective: validation AUC on the last 20% of training dates by time) explores hidden dimension $\in \{32, 64, 128\}$, number of layers

$\in \{1, 2, 3\}$, dropout rate, learning rate and weight decay, converging for Configuration 01 to hidden = 64, 3 layers and dropout = 0.41 (bidirectional); for Configuration 02 to hidden = 64, 3 layers and dropout = 0.41 (bidirectional); and for Configuration 03 to a wider single-layer architecture (hidden = 128, 1 layer, dropout = 0.35). For Configurations 02 and 03 the Optuna-tuned LSTM underperformed the fixed configuration substantially on the 2026 test set, indicating that the internal validation window was too short to avoid overfitting during search. For Configuration 01 the tuned bidirectional model slightly outperforms the fixed unidirectional configuration; however, given that the internal validation window contains only 25 positive training samples at the 3-day horizon, the search is unstable and the fixed configuration is retained for deployment.

XGBoost (honey super, Configuration 04). The honey super task uses XGBoost with `n_estimators` = 400, `max_depth` = 4, `learning_rate` = 0.03, `subsample` = 0.8, `colsample_bytree` = 0.8, `min_child_weight` = 5 and elastic-net regularisation ($\alpha = 0.1$, $\lambda = 1.0$). The decision threshold is optimised by maximising G-Mean on the walk-forward validation splits (optimal value: 0.435). A secondary comparison with LightGBM on the same feature set yields nearly identical performance, confirming that results are not strongly architecture-dependent at this scale. Optuna (60 trials, TPE, objective: walk-forward AUC) searches over tree depth, learning rate, subsampling rates and regularisation coefficients.

5.3 Evaluation

Walk-forward cross-validation. A strict temporal split is applied throughout to prevent any form of future leakage [13]: the model is trained on all data from years prior to year Y and evaluated on year Y. For the swarm models, two folds are computed: (train 2023–2024, test 2025) and (train 2023–2025, test 2026). The primary results are reported on the 2026 test fold, which contains 14 events and represents the most challenging setting — the most recent season, with the highest positive rate. For the honey super model, where only one complete test season is available, `TimeSeriesSplit` with $n = 4$ folds and a 14-day gap between training and evaluation windows is used instead, ensuring that no label from the prediction horizon leaks into the training set.

Evaluation metrics. Let TP, FP, TN and FN denote true positives, false positives, true negatives and false negatives at a given decision threshold. Four complementary metrics are used throughout.

The **AUC-ROC** summarises discriminative ability across all possible thresholds as the area under the receiver operating characteristic curve:

$$\mathrm{AUC} = \int_0^1 \mathrm{TPR}(t)\, d\,\mathrm{FPR}(t), \tag{1}$$

where $\mathrm{TPR}(t) = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$ is sensitivity and $\mathrm{FPR}(t) = \frac{\mathrm{FP}}{\mathrm{FP+TN}}$ is the false positive rate. An AUC of 1 indicates perfect separation; 0.5 corresponds to chance [14].

The **Average Precision (AP)** is the area under the precision–recall curve, computed as a weighted mean of precisions at each threshold:

$$\mathrm{AP} = \sum_k (R_k - R_{k-1})\, P_k, \tag{2}$$

where P_k and R_k are the precision and recall at the k-th threshold change. Unlike AUC, AP is directly sensitive to performance on the minority class [15, 16] and is therefore the primary optimisation target in all Optuna searches.

The **G-Mean** is the geometric mean of sensitivity and specificity:

$$\text{G-Mean} = \sqrt{\mathrm{TPR} \times \mathrm{TNR}} = \sqrt{\frac{\mathrm{TP}}{\mathrm{TP+FN}} \cdot \frac{\mathrm{TN}}{\mathrm{TN+FP}}}, \tag{3}$$

where $\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{TN+FP}}$ is specificity. G-Mean penalises models that ignore the minority class to maximise overall accuracy [17] and it governs threshold selection.

Event-level detection is an operational metric that counts how many of the 14 test swarming events have at least one positive prediction in the 3 days preceding the event. This is paired with the **false alarm count (FA)** (positive predictions outside any event window). Together these two quantities express model performance in practical terms for the beekeeper: events detected out of total events, and unnecessary inspections per season.

Threshold selection. All models output a continuous risk score in $[0, 1]$. The default threshold of 0.5 is replaced by the value $\hat{\tau}$ that maximises G-Mean on the walk-forward training folds:

$$\hat{\tau} = \arg\max_{\tau \in [0,1]} \sqrt{\mathrm{TPR}(\tau) \times \mathrm{TNR}(\tau)}. \tag{4}$$

This reflects the asymmetric costs of the task: a missed swarm results in a permanent productive loss for the beekeeper, while a false alarm incurs only the cost of an unnecessary inspection.

Model selection criterion. The best model for each configuration is chosen by prioritizing these three metrics: **G-Mean** (balanced sensitivity and specificity at the chosen threshold), **event-level detection rate** (swarm events detected within the pre-event window) and **false alarm count** (positive predictions outside any event window). The AUC and AP are also taken into account as extra information and may include tiebreakers between similar models, but they are not the main selection criterion: both summarize the discrimination capacity at all thresholds, while the geometric mean, detection rate and number of false alarms directly reflect the beekeeper's operational priorities at the chosen decision threshold.

6 Results

6.1 Multi-Horizon – Swarm Prediction

Multi-horizon analysis evaluates 3 architectures × 3 horizons on hives with a history of previous swarming. At the 3-day horizon, LSTM Uni achieves the best balance (AUC = 0.876, G-Mean = 0.866, 13/14 events detected). At

7 days, XGBoost leads in G-Mean (0.802) and event detection (13/14), despite a slightly lower AUC than LSTM (0.855 vs. 0.892), due to its better-calibrated threshold over balanced sensitivity and specificity. At 14 days, LSTM Bi is selected for its markedly lower false alarm count (55 FA vs. 119 for XGBoost) at near-equivalent AUC (0.911 vs. 0.926). Walk-forward validation confirms that the 7d and 14d results are moderately generalisable (walk-forward AUC = 0.802 and 0.821 respectively); the 3d horizon is considered unreliable due to insufficient positive training samples (25 positives).

Full results are shown in Table 4; C01 XGBoost feature importances per horizon are shown in Fig. 5.

TABLE 4: Multi-horizon swarm prediction results on 2026 test set (hives with prior swarm history, seasonal window Feb–Jun). AP = Average Precision, Det = swarm events detected within the prior window / 14 total, FA = False Alarms.

Horizon	Model	AP	G-Mean	Det.	FA
3d	XGBoost	0.116	0.737	12/14	213
	LSTM Uni	**0.144**	**0.866**	**13/14**	**136**
	LSTM Bi	0.110	0.654	7/14	138
7d	XGBoost	**0.335**	**0.802**	**13/14**	182
	LSTM Uni	0.306	0.728	11/14	85
	LSTM Bi	0.303	0.766	12/14	**84**
14d	XGBoost	**0.639**	**0.854**	**14/14**	119
	LSTM Uni	0.561	0.811	13/14	94
	LSTM Bi	0.555	0.815	13/14	**55**

6.2 Morning and Nocturnal Windows – Swarm Prediction

C02 (morning window 10:00–14:00) and C03 (night window 00:00–05:00) compare XGBoost and LSTM using the key feature `weight_std_roll14`. In Table 5, *XGBoost baseline* denotes XGBoost trained on the corresponding feature set *without* `weight_std_roll14`, serving as the ablation reference to isolate the contribution of this feature. In both Configurations, adding this feature improves XGBoost (+0.024–0.045 AUC). LSTM clearly outperforms XGBoost. C03 LSTM Uni achieves the best overall result (AUC=0.882) with only 161 false alarms, compared to 289 for the best in C02 (LSTM Bi, AUC=0.859). Optuna search penalizes LSTM (AUC=0.326/0.645) due to overfitting in the internal validation window, while for XGBoost, Optuna does improve performance (AUC=0.778 C02, 0.772 C03).

XGBoost feature importances for C02 and C03 are shown in Figs. 6 and 7, respectively.

TABLE 5: Swarm prediction comparison (3-day horizon, 2026 test set, 14 events). C02 = morning window (10–14h); C03 = nocturnal window (00–05h).

Cfg	Model	AP	G-Mean	Det.	FA
02	XGBoost baseline	0.112	0.616	11/14	437
	XGBoost	0.122	0.678	10/14	**244**
	LSTM Uni	0.132	0.776	**13/14**	305
	LSTM Bi	**0.141**	**0.807**	**13/14**	289
03	XGBoost baseline	0.134	0.647	10/14	284
	XGBoost	0.138	0.728	10/14	193
	LSTM Uni	**0.153**	**0.823**	12/14	**161**
	LSTM Bi	0.086	0.773	**13/14**	332

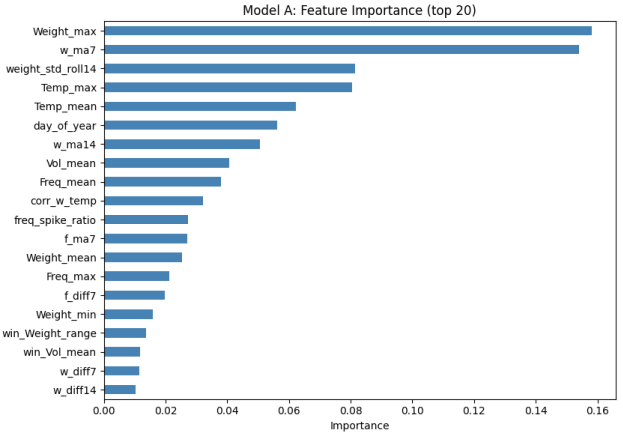


Fig. 6: C02 XGBoost feature importance (top 20). `weight_std_roll14` ranks third, confirming the ablation result from Section 5.1. Morning-window features (`win_Weight_range`, `win_Vol_mean`) appear in the lower ranks, indicating secondary but non-zero contribution.

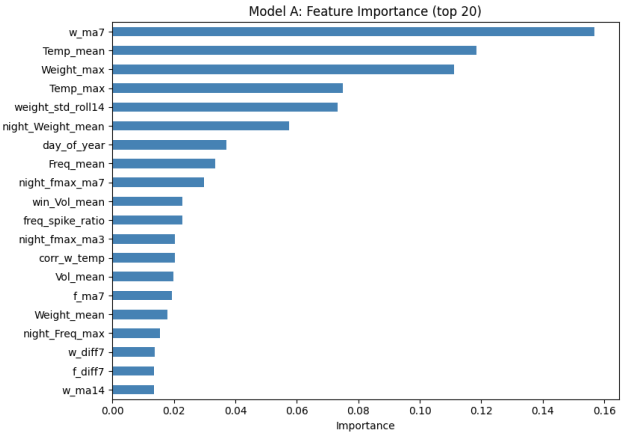


Fig. 7: C03 XGBoost feature importance (top 20). Classical weight and temperature features still dominate (`w_ma7`, `Temp_mean`, `Weight_max`), with `weight_std_roll14` at rank 5. `night_Weight_mean` (rank 6) is the first nocturnal feature to appear, confirming that the 00:00–05:00 aggregations contribute beyond the baseline feature set without displacing the primary weight dynamics.

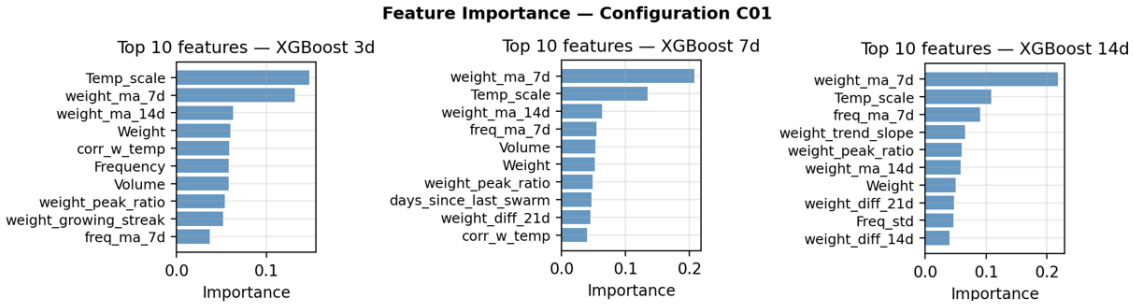


Fig. 5: C01 XGBoost feature importance (top 10) per prediction horizon. At 3 days, external temperature (`Temp_scale`) leads, indicating that ambient conditions modulate short-term pre-swarm weight dynamics. At 7 and 14 days, `weight_ma_7d` becomes dominant, reflecting the shift toward longer-range weight trends as the primary signal. `corr_w_temp` appears in the top 10 at both the 3-day and 7-day horizons.

6.3 Event-Level Detection Analysis

The event evaluation counts as positive if at least one prediction falls within the 3-day window prior to the swarm. The best C02 and C03 models detect 13/14 and 12/14 events, respectively. C03 LSTM Uni fails in 2 of the 14 events but reduces false alarms by 44% compared to the best C02 model (161 vs. 289 FA). The false alarm rate of the XGBoost baseline is very high (>280 FA)—the weight stability feature substantially reduces it in all models.

6.4 Honey Super Placement

Twenty-four configurations (6 feature sets × XGBoost/LightGBM × 7-day and 14-day windows) were evaluated on 20 real-world interventions from 2026. The selection criterion was G-Mean combined with intervention detections. For the 7-day window, LightGBM with full biological features (46 features, no temporal features) achieved the best result (AUC=0.765, G-Mean=0.730, 14/20 detected, CV AUC=0.782±0.041). For the 14-day window, XGBoost with acoustic features (26 features) achieved a good balance between detections and false alarms (AUC=0.701, 15/20 detected, 377 false alarms, CV AUC=0.716±0.052). The 4-fold walk-forward validation confirms moderate robustness (AUC±std $<$ 0.15 in both models).

Full grid search results for the 7- and 14-day windows are shown in Tables 6 and 7, respectively.

TABLE 6: Honey super placement — 7-day window grid search (2026 test, 20 interventions). Numbers in parentheses indicate the number of features.

Model & Feature Set	AP	G-Mean	Det.	FA
L-GBM, bio_no_tmp (46)	**0.15**	**0.730**	14/20	597
XGB, bio_no_tmp (46)	0.11	0.729	14/20	625
L-GBM, hive_plus (28)	0.11	0.686	14/20	724
XGB, hive (19)	0.11	0.657	11/20	**390**

TABLE 7: Honey super placement — 14-day window grid search (2026 test, 20 interventions). Numbers in parentheses indicate the number of features.

Model & Feature Set	AP	G-Mean	Det.	FA
XGBoost, hive (19)	0.235	**0.729**	15/20	565
LightGBM, hive (19)	0.243	0.728	13/20	395
XGBoost, acoustic (26)	**0.260**	0.718	**15/20**	**377**
LightGBM, acoustic (26)	0.260	0.700	13/20	432

Table 8 summarises the best-performing model per configuration across all four tasks.

TABLE 8: Summary of best-performing model per Configuration (2026 test set). Selection: G-Mean + event detection + false alarms. † = deployed to dashboard. * = feature sets detailed in Tables 6–7.

Model	AUC	AP	G-Mean	Det.	FA
C1 (3d), LSTM-U†	0.88	0.14	**0.866**	13/14	136
C1 (7d), XGB	0.86	0.34	**0.802**	13/14	182
C1 (14d), LSTM-B†	0.91	0.56	0.815	13/14	**55**
C2, LSTM-B	0.86	0.14	0.807	13/14	289
C3, LSTM-U	0.88	**0.15**	0.823	12/14	**161**
C4 (7d), LGBM*†	0.77	0.15	0.730	14/20	597
C4 (14d), XGB*†	0.70	0.26	0.718	**15/20**	**377**

7 Production Deployment

To utilize the models developed in this study, the best models have been integrated into a web platform created to support the work of the beekeepers responsible for the UAB apiary. The system combines a FastAPI backend, responsible for data processing and model inference, with a Streamlit control panel that provides an accessible interface for end users.

The platform automatically selects the best-performing model for each prediction task, based on the experimental results obtained in this study. Swarm risk predictions are generated at two horizons: the 3-day alert uses the C01 unidirectional LSTM (G-Mean = 0.866, 13/14 events detected) and the 14-day early warning uses the C01 bidirectional LSTM (55 false alarms per season). Recommendations

for honey super placement use the best-performing models identified in Configuration 04.

To keep the website and models up-to-date, an automated workflow was implemented using n8n [22], an open-source no-code automation platform. Every three days, the platform retrieves the latest sensor readings from the BeeHive Monitoring platform [20], updates the CSV datasets used by the prediction system, recalculates the necessary features and updates the predictions available on the dashboard. This process ensures that the model results are continuously updated without manual intervention.

The dashboard displays the predicted swarming probability and honey super requirements for each monitored hive. Risk levels are shown as percentages, allowing beekeepers to quickly identify colonies that require attention. Other views provide access to sensor trends, historical predictions and model performance information.

Figure 8 illustrates the implementation architecture. The complete system transforms the models developed in this project into an operational tool capable of supporting daily apiary management through automated data collection, predictive analytics and intuitive visualization.

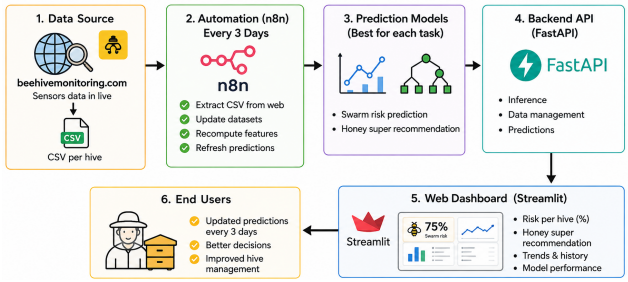


Fig. 8: Architecture of the deployed web system for apiary management, combining a FastAPI inference backend, a Streamlit dashboard and an automated n8n workflow for periodic data ingestion, feature updating and prediction refresh.

However, the models should be considered under development. The limited number of recorded swarming events and interventions related to the placement of honey supers restricts the number of examples available for training, resulting in many false alarms. As new data seasons are incorporated and more real-world events are recorded, the models are expected to progressively improve their generalization, increasing the accuracy of predictions and reducing the FA rate.

8 Discussion

The results show that early swarm detection is possible with multi-season IoT data from a single apiary and that the best-performing model is not the most feature-rich of the configurations evaluated. Several findings warrant closer examination.

The simplicity paradox. As the final results show, the richer feature sets of C02 and C03 have not made their models the best. With the data available so far, the best-performing model remains C01: a unidirectional LSTM with 24 daily features, achieving G-Mean = 0.866 and detecting 13 out of 14 test events with 136 false alarms. The LSTM Uni C03, with 44 features, including 13 nocturnal aggregations, achieves a slightly higher AUC (0.882 vs. 0.876), but detects one fewer event (12/14) and generates 25 more false alarms per season. This suggests that the dominant pre-swarm signal is already present in the daily aggregations (particularly in the `weight_ma_7d`, `Temp_scale` and `corr_w_temp`) and that higher temporal resolution adds noise rather than discriminatory power over a 3-day horizon. The feature importance results (Fig. 7) are consistent with this interpretation: even in C03, nocturnal features do not appear until rank 6, while classical weight and temperature aggregates occupy the top five positions.

LSTM versus XGBoost. The two model types exhibit differences. LSTM consistently outperforms XGBoost in G-Mean over the 3-day window (0.866 vs. 0.737), where the temporal context of a 21-day input sequence captures pre-swarm dynamics that daily snapshots alone cannot resolve. At 7 days, XGBoost recovers, achieving the higher G-Mean and event detection (13/14) at the cost of a higher number of false alarms (182 vs. 84 for LSTM Bi). At 14 days, both architectures converge in detection performance, with LSTM Bi being the preferred option due to its lower number of false alarms (55 vs. 119). The training curves (Fig. 9) reveal a consistent gap between training loss and validation loss across all LSTM configurations, a direct consequence of minority class amplification by weighted `BCEWithLogitsLoss`; early stopping mitigates, but does not eliminate, this effect.

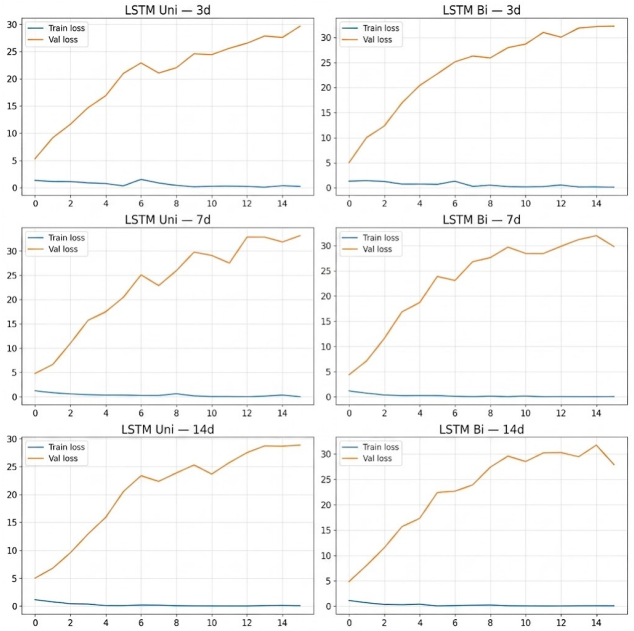


Fig. 9: C01 LSTM training curves (train vs. validation loss) for all six configurations (Uni/Bi × 3d/7d/14d). The large train–validation gap reflects the extreme class imbalance (25 positive training samples at 3 days): the weighted `BCEWithLogitsLoss` amplifies positive-class errors, producing validation losses an order of magnitude above training loss. Early stopping triggers consistently around epoch 12–15.

Validation of pre-swarm behavioural signatures. The studies conducted in Section 4 are reflected in the model results. The rolling weight–temperature correlation (`corr_w_temp`) appears among the top 10 features of C01 XGBoost in both the 3-day (ranked 5th) and 7-day (ranked 10th) time horizons, confirming its importance beyond exploratory data analysis (EDA). The nocturnal temperature signal ($+2$–$+8$°C) also contributes: `night_Weight_mean` is the first nocturnal feature in C03 XGBoost (ranked 6th), reflecting the colony weight gain linked to bee clustering before departure. The acoustic frequency signal, on the other hand, does not reach the threshold of $+150$ Hz reported by Ferrari et al. [4] in any hive, suggesting that this signal depends on the location of the sensor or the size of the population and is not generalized throughout the entire set of colonies at the UAB.

False alarms and operational cost. The implemented 3-day model generates 136 false alarms per season across 8 hives (since the model has only been trained on hives where swarming has been identified, to reduce irrelevant data). This equates to approximately 17 unnecessary inspection activations per hive during the monitoring period from February to June (a narrow window, not year-round, for the same reason as before). For a control panel checked daily by the beekeeper, this rate is manageable. The 14-day model reduces false alarms to 55 (7 per hive per season) at the cost of a longer and less accurate prediction period, as we are assigning more days as swarming days. It is expected that as there are more seasons and therefore more data, these models will improve and become increasingly accurate.

Honey super placement. Honey super placement is inherently more difficult to predict than swarming events, at least for now. The decision to add supers depends on the beekeeper's judgment, the strength of the colony, the local nectar flow and the availability of equipment—all factors that occur simultaneously, none of which can be fully observed using only sensor data. The best 7-day model (LightGBM, G-Mean = 0.730, 14/20 interventions detected) and the 14-day model (XGBoost, 15/20 detected, 377 false alarms) demonstrate that sensor data does contain a predictive signal for this task, but the high number of false alarms limits its practical use for automated recommendations. An additional challenge is that the target variable itself is not entirely objective. Unlike swarming, which is a biological event that either occurs or does not occur, honey super placement is a management decision made by the beekeeper. There is no definitive way to determine whether a super was added at the optimal moment; an intervention may have been performed too early, too late, or exactly when needed. Consequently, the model is partly learning human decision-making rather than a purely biological process, which adds uncertainty that extends beyond what can be captured through sensor measurements alone.

Limitations. This work has several limitations that should be considered when interpreting the results. First, the training set contains only 9 confirmed swarming events, resulting in 25 positive sequences for the 3-day prediction horizon. Although the models achieve promising results, such a small number of positive examples limits the statistical confidence of the evaluation and makes hyperparameter optimization more prone to overfitting.

Second, all data come from a single apiary monitored under relatively consistent environmental and management conditions. As a result, it is still unclear how well the proposed models would generalize to apiaries with different climates, bee populations, nectar availability, sensor setups, or beekeeping practices.

Third, some potentially useful signals were not available in the dataset. In particular, CO_2 concentration has been suggested in previous studies as a promising indicator of pre-swarming activity [12], but no CO_2 sensors were installed during the monitoring period. Incorporating this type of information in future deployments could improve the predictive capability of the models.

Finally, three of the hives evaluated during the 2026 season (Box1, Box5 and Box8) had no previous swarming history available in the training data. Since some of the engineered features rely on historical swarm information, these hives represent a particularly challenging prediction scenario and likely contribute to both missed detections and false alarms. Additional monitoring seasons will be necessary to better assess the performance of the methodology on newly established colonies with limited historical records.

9 CONCLUSIONS

This work demonstrated that early swarming detection from IoT sensor data is feasible with a small but carefully curated multi-season dataset. Behavioural analysis confirmed two biologically interpretable pre-swarm signatures: a systematic reversal of the weight–temperature Pearson correlation in the 14 days before departure and nocturnal internal temperature increases of $+2$–$+8$°C in the preceding week. Both signals were validated across multiple hives and seasons, and incorporated as engineered features that contribute measurably to model performance beyond baseline sensor aggregations.

Four model configurations were developed and evaluated under strict walk-forward temporal cross-validation. The best swarm detection model — a unidirectional LSTM on 24 daily aggregated features (C01) — achieves G-Mean = 0.866 and detects 13 out of 14 test swarming events 3 days in advance, with 136 false alarms per season across 8 hives. Contrary to initial expectations, adding higher temporal resolution (morning and nocturnal windows, C02–C03) did not improve over the simpler daily model: the dominant pre-swarm signal is already captured by daily weight and temperature aggregations, and finer-grained features add noise at the 3-day prediction horizon. LSTM consistently outperforms XGBoost at this horizon, while XGBoost remains competitive at 7 days due to better threshold calibration under the available training data.

The honey super placement task proved substantially harder to model from sensor data alone. LightGBM achieved the best result at the 7-day horizon (G-Mean = 0.730, 14/20 interventions detected) and XGBoost at 14 days (15/20 detected, 377 false alarms), but in both cases the false alarm rates limit operational utility. This reflects the inherently subjective nature of the management

decision, which depends on beekeeper judgement and on factors not observable through hive sensors alone.

The trained models have been integrated into an operational web platform combining a FastAPI inference backend, a Streamlit dashboard and an automated n8n pipeline, providing UABee apiary beekeepers with continuously updated swarm risk and honey super placement predictions.

Several limitations constrain the current work: the training set contains only 9 swarming events, all data originate from a single apiary and CO_2 measurements are unavailable. Future work should incorporate additional monitoring seasons, extend validation to apiaries with different climates and sensor setups, and integrate CO_2 as an additional predictive channel. As the dataset grows, the models are expected to improve generalisation and progressively reduce false alarm rates.

ACKNOWLEDGMENTS

As the author of this work, I would like to thank my supervisor, Vicenç Soler, for his supervision and guidance throughout this project. He facilitated both online and in-person sessions, whether with Gerardo Caja or individually to review data, metrics and models, enabling continuous monitoring and a constant interest in the project's progress.

I would also like to thank the UAB beekeeping team, especially Gerardo Caja, for providing access to hive monitoring data and for answering any questions I had, as well as for their collaboration during the experimental phase of this work.

I would also like to thank the authors of various studies related to bees, sensors and monitoring, whose work I had to access in order to gather information before beginning this study.

Finally, I am grateful for the support and assistance of some classmates who helped me with various questions related to the data.

REFERENCES

[1] Klein, A. M., et al. (2007). Importance of pollinators in changing landscapes for world crops. *Proceedings of the Royal Society B*, 274(1608), 303–313.

[2] Zacepins, A., et al. (2016). Remote detection of the swarming of honey bee colonies by single-point temperature monitoring. *Biosystems Engineering*, 148, 76–80.

[3] Fahrenholz, L., Lamprecht, I., & Schricker, B. (1989). Thermal investigations of a honey bee colony. *Journal of Comparative Physiology B*, 159(5), 551–560.

[4] Ferrari, S., et al. (2008). Monitoring of swarming sounds in bee hives for early detection of the swarming period. *Computers and Electronics in Agriculture*, 64(1), 72–77.

[5] Bromenshenk, J. J., et al. (2009). *Honey bee acoustic recording and analysis system for monitoring hive health*. US Patent 7,549,907.

[6] Terenzi, A., Cecchi, S., & Orcioni, S. (2020). A preliminary study of sound analysis for the estimation of the state of health of a beehive. *Proc. EUSIPCO 2020*.

[7] Cecchi, S., Spinsante, S., Terenzi, A., & Orcioni, S. (2018). A smart sensor-based measurement system for advanced bee hive monitoring. *Sensors*, 18(8), 2726.

[8] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proc. KDD 2016*, 785–794.

[9] Staudemeyer, R. C., & Morris, E. R. (2019). Understanding LSTM – a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586*. https://arxiv.org/abs/1909.09586

[10] Seeley, T. D. (2010). *Honeybee Democracy*. Princeton University Press.

[11] Akiba, T., et al. (2019). Optuna: A next-generation hyperparameter optimization framework. *Proc. ACM KDD 2019*, 2623–2631.

[12] TegwynTwmffat (2023). *AI Honey Bee Swarm Detector / Predictor System* [Project logs]. Hackaday.io. https://hackaday.io/project/190664/logs

[13] Bergmeir, C., & Benítez, J. M. (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences*, 191, 192–213.

[14] Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8), 861–874.

[15] Davis, J., & Goadrich, M. (2006). The relationship between Precision-Recall and ROC curves. *Proceedings of ICML 2006*.

[16] Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, 10(3), e0118432.

[17] Brzeziński, D., & Stefanowski, J. (2018). A theoretical study of classification performance measures in imbalanced domains. *Information Sciences*, 462, 242–261.

[18] He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9), 1263–1284.

[19] Caja, G. et al. *UABee — Projecte d'apicultura de la UAB*. Universitat Autònoma de Barcelona (2019–). https://sites.google.com/view/uabee-esp (accessed June 2026).

[20] BeeHive Monitoring. *Apiary telemetry platform*. https://beehivemonitoring.com (accessed June 2026).

[21] Ke, G., et al. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems (NeurIPS)*, 30.

[22] n8n GmbH. *n8n — Workflow Automation Platform* (open source). https://n8n.io (accessed June 2026).

A APPENDIX

A.1 Feature Definitions

Tables 9–12 enumerate all features used across the four model configurations. Feature names match code variables exactly; rolling windows are fitted on the training split only.

A.2 Data Preprocessing Pipeline

Two sensor nodes per hive log readings in an interleaved pattern at a combined cycle of approximately 10 minutes. The preprocessing path splits into two branches: Configurations 01 and 04 consume a pre-aggregated daily feature CSV (`daily_features_final.csv`) built from the 10-minute raw stream, while Configurations 02 and 03 work directly from raw 10-minute readings to construct morning- and nocturnal-window aggregations.

1. **Time alignment.** Readings from both nodes are merged into a uniform regular grid—10-minute intervals for the daily-feature branch (C01, C04) and 10-minute intervals for the sub-daily branch (C02, C03)—by averaging readings that fall within the same interval.

2. **Gap filling.** Contiguous gaps in the regular grid are forward-filled for up to 8 consecutive steps: 80 minutes on the 10-minute grid. Gaps exceeding this limit are left as NaN; they are presumed to reflect real events (sensor removal, battery failure, or beekeeper intervention) rather than transient communication dropouts.

3. **Outlier removal (C02 and C03 only).** Before window extraction two cleaning rules are applied to the 10-minute data: (a) single-step weight changes $|\Delta W| > 5$ kg are set to NaN, as they indicate sensor faults or beekeeper interventions; (b) weight readings outside the biologically plausible range $[3,\ 120]$ kg are also set to NaN.

4. **Window aggregation.** For C01 and C04, daily statistics (mean, standard deviation, minimum and maximum) are computed per hive per calendar day. For C02 and C03, the morning window (10:00–13:59) is additionally extracted from the 10-minute grid and aggregated. For C03 only, the nocturnal window (00:00–04:59) is extracted directly from the raw 10-minute readings *without forward-fill*, preserving genuine overnight sensor absences rather than imputing across them.

5. **Rolling feature engineering.** Moving averages, cumulative differences, trend slopes and the 14-day Pearson correlation `corr_w_temp` are computed over the complete available timeline for each hive. No parameters are fitted in this step: all computations are deterministic functions of the data.

6. **NaN fraction filtering.** Any feature column with more than 50% missing values in the training split is excluded from the feature set for that run. This threshold removes channels with intermittent sensor coverage (most notably the heart-node sensors on hives where the internal node was deployed later) without over-restricting the feature space for hives with complete data.

7. **Imputation.** Residual NaN values are replaced with the per-feature median computed exclusively on the training split (infinity values are coerced to NaN before this step). No information from the validation or test periods is used.

8. **Scaling and sequence construction (LSTM only).** A `StandardScaler` is fitted on training-split rows and applied to the full dataset. Each LSTM input sample is a tensor of `SEQ_LEN` consecutive daily feature vectors per hive: 21 days for C01, 14 days for C02 and C03.

9. **Training-set augmentation (C01, horizons $\geq$ 7 d).** Each confirmed swarming event in the training split is duplicated with a ± 1-day time shift to compensate for the extreme scarcity of positive labels at longer horizons. Augmentation is not applied to the 3-day horizon, where temporal precision degrades with perturbation.

A.3 Additional Experimental Results

Optuna LSTM hyperparameter search. Table 13 compares the fixed baseline LSTM architecture against the best architecture found by Optuna (20 trials, Tree-structured Parzen Estimator, objective: validation AUC on the most recent 20% of training dates by time) for Configurations 02 and 03. The fixed architecture used in all models is: 2 recurrent layers, hidden dimension 64, dropout 0.30. The Optuna-tuned LSTMs perform dramatically worse on the 2026 test set—AUC = 0.326 for C02 and 0.645 for C03 versus 0.859 and 0.882 for the fixed configurations—confirming that the internal validation window is too short for reliable LSTM architecture search. For C01, the Optuna-tuned bidirectional configuration (hidden = 64, 3 layers, dropout = 0.41) slightly outperforms the fixed unidirectional model but is not deployed, as the search is unstable due to only 25 positive training sequences at the 3-day horizon.

TABLE 13: LSTM Optuna vs. fixed baseline (C02 and C03). Searches: 20 trials, TPE, val. AUC on last 20% of training dates. AUC on 2026 test set.

Cfg	Variant	Hid.	Layers	Drop.	AUC
C02	Fixed Bi	64	2	0.30	0.859
	Optuna Bi	64	3	0.41	0.326
C03	Fixed Uni	64	2	0.30	0.882
	Optuna Uni	128	1	0.35	0.645

In contrast, Optuna improves XGBoost on both configurations (AUC = 0.778 for C02 and 0.772 for C03 vs. the fixed-hyperparameter baselines), consistent with gradient-boosted trees being less sensitive to the small validation window. Despite the higher AUC, the Optuna-tuned XGBoost models are not selected as the best configuration because the fixed LSTM substantially outperforms them in G-Mean and event detection on the 2026 test set.

Feature	Description	Feature	Description
Weight dynamics			
Weight	Daily mean weight (kg)	weight_ma_7d	7-day moving average
weight_ma_14d	14-day moving average	weight_diff_7d	Δweight over 7 days
weight_diff_14d	Δweight over 14 days	weight_diff_21d	Δweight over 21 days
weight_trend_slope	7-day linear slope	weight_acceleration	Rate of change of daily diff.
n_positive_days_7d	Positive-gain days in last 7d	weight_growing_streak	Consecutive days of growth
Swarm history			
weight_peak_ratio	Weight / expanding 95th-pct.	weight_drop_3d	Weight loss over last 3 days
days_since_last_swarm	Days since prior swarm (999=none)	n_swarms_this_season	Swarms in current season
Acoustics			
Frequency	Daily mean frequency (Hz)	freq_ma_7d	7-day moving average
freq_trend_7d	Δfrequency over 7 days	Freq_std	Daily std. dev. of frequency
Volume	Daily mean volume	vol_trend_7d	Δvolume over 7 days
Temperature and correlation			
Temp_scale	External temperature (°C)	temp_trend_7d	Δexternal temp. over 7d
Temp_heart	Internal temperature (°C)	corr_w_temp	14-day Pearson: Weight ↔ Temp

TABLE 9: Notebook 01 feature set (24 features). Daily aggregates; one row per hive per day. LSTM SEQ_LEN = 21 days.

Feature	Description	Feature	Description
Daily weight aggregates (from 10-min data)			
Weight_mean	Daily mean weight (kg)	Weight_std	Daily std. dev. of weight
Weight_min	Daily minimum weight	Weight_max	Daily maximum weight
w_ma7	7-day moving average	w_ma14	14-day moving average
w_diff7	Δweight over 7 days	w_diff14	Δweight over 14 days
weight_std_roll14: 14-day rolling std. dev. of daily mean weight (ablation gain + 0.053 AUC)			
Daily frequency and volume			
Freq_mean	Daily mean frequency (Hz)	Freq_max	Daily max frequency
Freq_std	Daily std. dev. frequency	f_ma7	7-day moving average
f_diff7	Δfrequency over 7 days	Vol_mean	Daily mean volume
Vol_max	Daily max volume		
Temperature, correlation, and swarm history			
Temp_mean	Daily mean temperature	Temp_max	Daily max temperature
corr_w_temp	14-day Pearson: Weight ↔ Temp	days_since_swarm	Days since last swarm (999=none)
has_prior_swarm	Binary: prior swarm exists	day_of_year	Calendar day (1–365)
Morning window 10:00–14:00h			
win_Weight_drop	Weight loss in window	win_Weight_std	Weight std. dev. in window
win_Weight_range	Weight range in window	win_Freq_mean	Mean frequency in window
win_Freq_max	Max frequency in window	win_Freq_std	Freq. std. dev. in window
win_Vol_mean	Mean volume in window	win_Vol_max	Max volume in window
freq_spike_ratio: win_Freq_max / 7-day rolling median of Freq_max (normalised spike)			

TABLE 10: Notebook 02 feature set (31 features). Built from raw 10-min data; 3-day horizon only. LSTM SEQ_LEN = 14 days.

Feature	Description	Feature	Description
Inherited from Notebook 02: all 31 features listed in Table 10			
Nocturnal additions: 00:00–05:00h window (13 new features)			
night_Freq_mean	Mean nocturnal frequency	night_Freq_max	Max nocturnal frequency
night_Freq_std	Nocturnal freq. std. dev.	night_Vol_mean	Mean nocturnal volume
night_Vol_max	Max nocturnal volume	night_Weight_mean	Mean nocturnal weight (kg)
night_f_ma7	7-day MA of nocturnal freq. mean	night_fmax_ma7	7-day MA of nocturnal freq. max
night_fmax_ma3	3-day MA of nocturnal freq. max	night_freq_spike_ratio	Nocturnal freq. max / 7-day rolling median
night_weight_delta: morning (06–09h) minus previous evening (18–21h) weight			
night_to_morning_freq: win_Freq_max / night_Freq_max (morning-to-night acoustic surge ratio)			
night_wdelta_ma3: 3-day moving average of night_weight_delta			

TABLE 11: Notebook 03 feature set (44 features = 31 from C02 + 13 nocturnal additions). Nocturnal aggregations computed without forward-fill. LSTM SEQ_LEN = 14 days.

Morning window boundary ablation. An ablation over the morning window boundary was conducted for C02: the 10:00–13:59 and 12:00–13:59 windows were compared on the walk-forward validation splits. The narrower 12:00–

Feature	Description	Feature	Description
Weight dynamics (12)			
Weight	Daily mean weight (kg)	weight_diff_1d	Δweight over 1 day
weight_diff_7d	Δweight over 7 days	weight_diff_14d	Δweight over 14 days
weight_diff_21d	Δweight over 21 days	weight_ma_7d	7-day moving average
weight_ma_14d	14-day moving average	weight_std_7d	7-day rolling std. dev.
weight_amplitude	Daily range (max−min)	weight_acceleration	Rate of change of daily diff.
weight_trend_slope	7-day linear slope	n_positive_days_7d	Positive-gain days in last 7d
Sensors and correlation (10)			
corr_w_temp	14-day Pearson: Weight ↔ Temp	Temp_scale	External temperature (°C)
Humidity_scale	External humidity (%)	Temp_heart	Internal temperature (°C)
Humidity_heart	Internal humidity (%)	Volume	Daily mean acoustic volume
Freq_std	Daily std. dev. of frequency	freq_ma_7d	7-day moving average of freq.
freq_diff_7d	Δfrequency over 7 days	temp_trend_7d	Δexternal temp. over 7d
Hive identity (1)			
box_id	Categorical hive identifier (encodes hive-specific weight baseline)		
Hive-relative historical (8)			
weight_historical_mean	Hive's per-month historical mean	weight_historical_std	Hive's historical weight std.
weight_vs_historical	Z-score vs hive historical mean	weight_pct_of_max	Weight as % of expanding 95th-pct.
weight_growing_streak	Consecutive days of growth	days_since_last_alza	Days since last super change (999=none)
days_since_last_ADD	Days since last super addition (999=none)	n_alzas_this_season	Super interventions this season
Apiary-relative (5)			
weight_vs_apiary	Z-score vs daily apiary mean	growth_vs_apiary	7d-growth z-score vs apiary
weight_rank_pct	Daily weight percentile in apiary	growth_rank_pct	Daily 7d-growth percentile
weight_apiary_ratio	Weight / daily apiary mean		
Expert threshold flags (3)			
above_min_weight	1 if Weight $\geq$ 20 kg during active season	overdue_for_alza	1 if $>$ 200 days since last super change
small_but_growing	1 if below-median weight but $\Delta_{14\mathrm{d}} > 1$ kg		

TABLE 12: Notebook 04 honey super feature set (45 total; 39 non-temporal). Six temporal encodings excluded via ablation: month, dayofyear, season, sin_dayofyear, cos_dayofyear, days_in_season.

13:59 window produced a higher false alarm count across both XGBoost and LSTM without improving event detection rate, and was therefore rejected. The full 10:00–13:59 window is used in all C02 and C03 models.

Walk-forward cross-validation for C01. Two walk-forward folds are evaluated for the multi-horizon model: (train 2023–2024, test 2025) and (train 2023–2025, test 2026). Primary results are reported on the 2026 fold in Table 4. The average walk-forward AUC across both folds is 0.802 for the 7-day horizon and 0.821 for the 14-day horizon, confirming moderate generalisation across seasons. The 3-day horizon walk-forward estimate is considered unreliable because the training set contains only 25 positive sequences across both folds, making any model selection based on this metric statistically fragile.

A.4 System Architecture Details

The production system consists of three loosely coupled layers: a FastAPI inference backend, a Streamlit monitoring dashboard and an n8n automation workflow. All components run on the same local server and communicate via HTTP.

FastAPI inference backend. The backend is served with Uvicorn at port 8000. At startup, a SwarmPredictor object loads the LSTM model artifacts from api/models/: the PyTorch .pt checkpoint, a StandardScaler object, the ordered feature list and the training-set median fill values. Model inference is offloaded to a four-worker thread pool to avoid blocking the asynchronous event loop. The main endpoints are listed below.

- POST /predict: accepts a raw sensor CSV and a box_id; replicates the notebook feature pipeline and returns the LSTM swarm-risk probability for the next 3 days, labelled HIGH/MEDIUM/LOW using the G-Mean-optimised threshold ($\hat{\tau}$ from Eq. 4).
- POST /weekly-run: orchestrates the full swarm pipeline — downloads an Excel export from beehivemonitoring.com via Playwright browser automation, parses it and calls the predictor for each of the 8 monitored hives. Results are written to latest_results.json; a compact entry is appended to run_history.json (capped at 52 entries, approximately one year of weekly runs).
- POST /weekly-alza-update: invokes update_alza_features.py as a subprocess, which downloads the last 30 days of readings for all 17 honey super hives, merges them with the master CSV and rebuilds daily_features_prod.csv

with all engineered features. A 10-minute server timeout accommodates the full feature recomputation.

- `GET /results` and `GET /history`: return the cached prediction JSON files to the dashboard on demand.

n8n automation workflow. A recurring n8n workflow triggers every 3 days via two sequential HTTP Request nodes: the first calls `POST /weekly-run` (12-minute timeout) and the second calls `POST /weekly-alza-update` (12-minute timeout). If either call returns a non-2xx response, the workflow logs the error. The pipeline requires no manual intervention under normal operation.

Streamlit dashboard. The dashboard is served at port 8501 and organised into six tabs: *Swarm Risk* displays per-hive probability cards and an alert banner for high-risk colonies; *Honey Supers* runs the LightGBM and XGBoost alza models directly against the latest `daily_features_prod.csv` and displays per-hive placement recommendations (add / monitor / wait); *Sensor Data* plots selectable sensor channels over a user-defined time window; *Risk Trends* plots swarm probability over successive weekly runs; *Models* shows a static performance summary; and *History* lists the full run log with a risk-evolution heatmap. Swarm risk thresholds are fixed at HIGH $\geq 50\%$, MEDIUM 20–50%, LOW $< 20\%$. Honey super thresholds are the G-Mean-optimised values stored inside each model bundle (`.pkl` file).

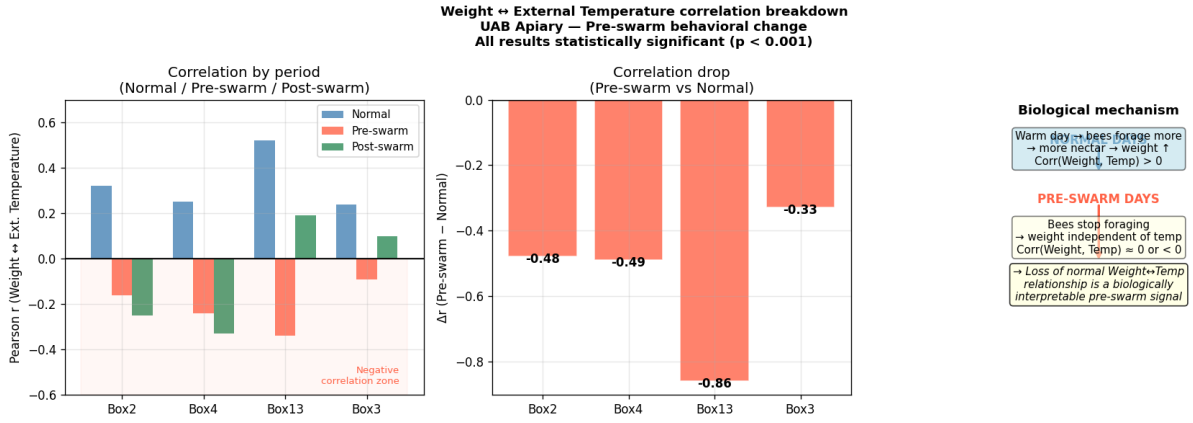


Fig. 10: Weight–temperature Pearson correlation by behavioural period (Normal / Pre-swarm / Post-swarm) for the four swarming hives, computed over daily averages restricted to spring–summer months (March–July). Left: grouped bars coloured by period; right: drop in correlation from Normal to Pre-swarm (Δr). All results are statistically significant ($p < 0.001$). The biological mechanism is annotated: during foraging season bees track ambient temperature, so $r(\text{Weight}, T) > 0$ normally; this coupling collapses pre-swarm as the colony stops foraging, making `corr_w_temp` one of the most informative features in all model configurations.

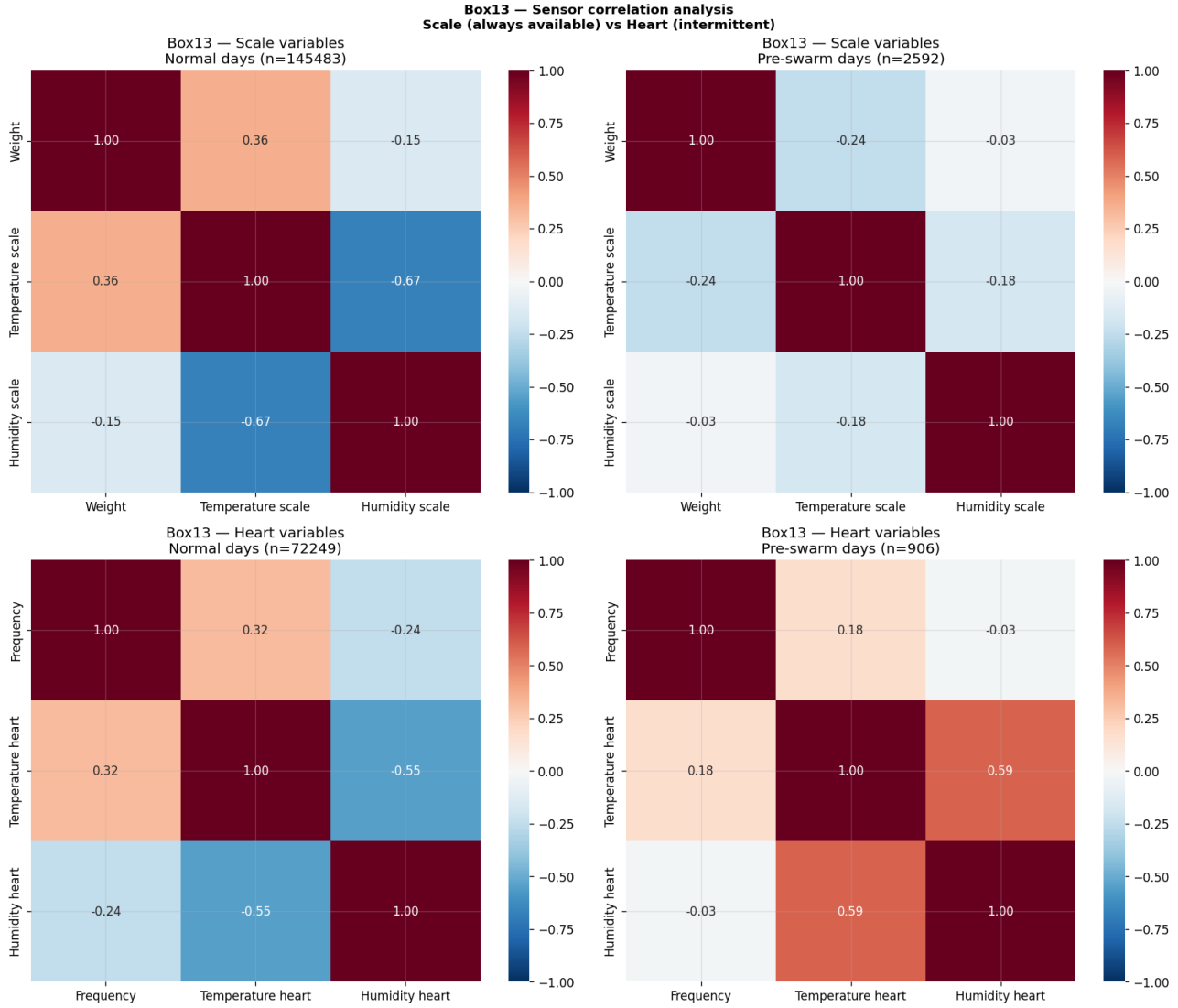


Fig. 11: Box13 sensor correlation matrices for Normal (baseline) and Pre-swarm days, separated by sensor unit: scale (Weight, Temperature scale, Humidity scale) and heart (Frequency, Temperature heart, Humidity heart). During normal operation the scale sensor shows a moderate positive weight–temperature correlation ($r = 0.36$); this collapses to $r = -0.24$ in the pre-swarm window. The heart sensor, intermittently available, shows a similar decoupling of Frequency from Temperature heart ($0.32 \rightarrow 0.18$). Compare with the Box3 analysis in Figure 3.

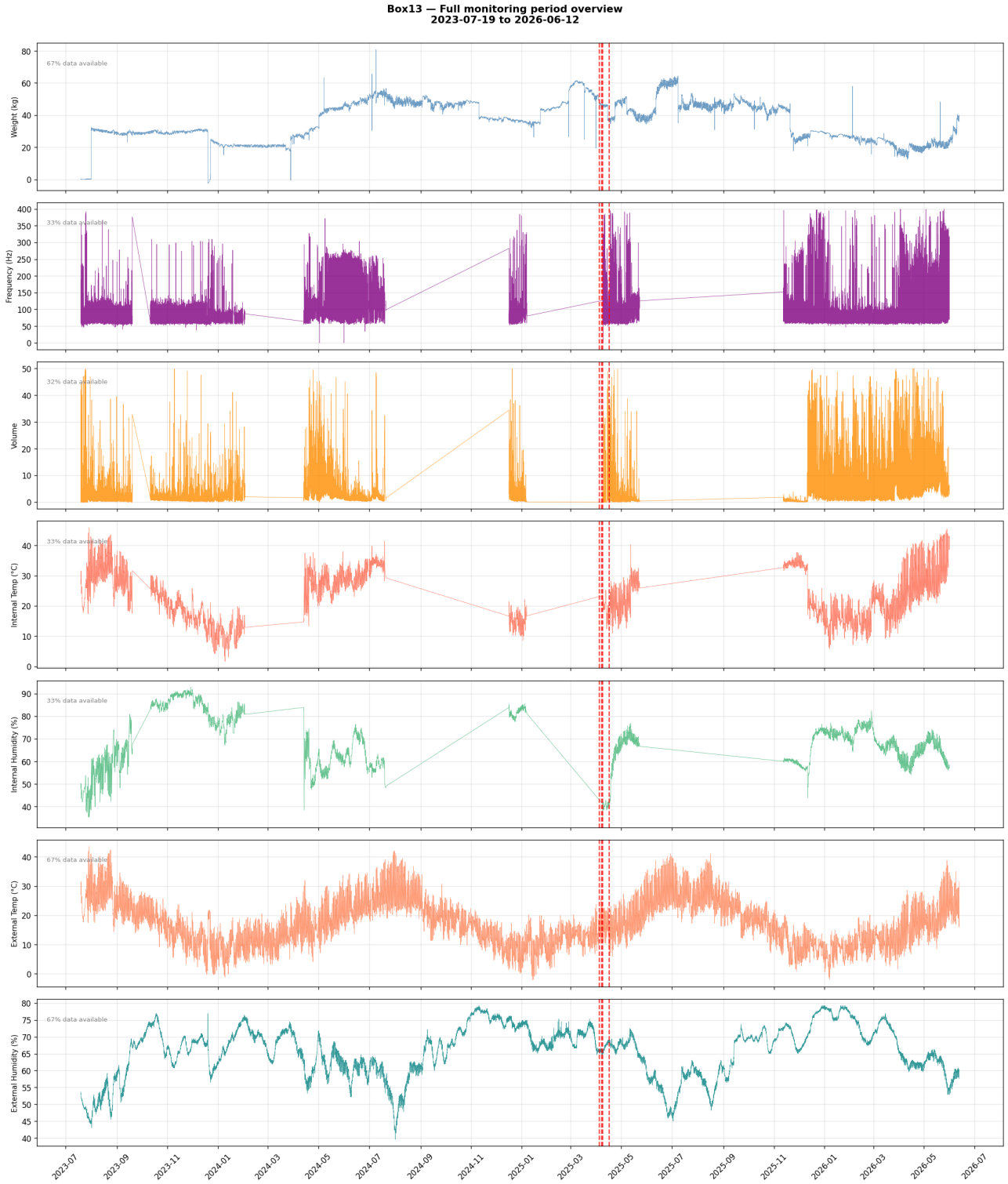


Fig. 12: Box13 full monitoring period (2023-07-19 to 2026-06-12): all seven sensor channels plotted chronologically. The vertical dashed red line marks the swarming events. Note the data gaps (sensor offline) and the clear seasonal pattern in Weight and external temperature.

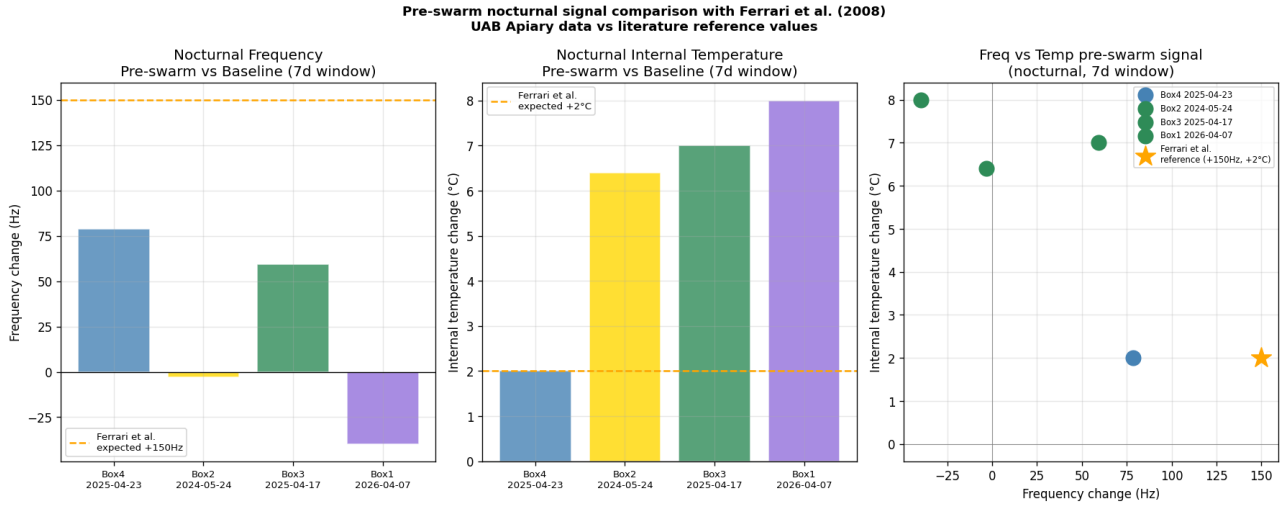


Fig. 13: Extended Ferrari et al. (2008) comparison including the joint frequency–temperature scatter (right panel). Each bubble represents one confirmed swarm event; the orange star marks the Ferrari reference (+150 Hz, +2 °C). Box3 and Box4 reach the temperature threshold; no hive simultaneously meets both criteria, consistent with the high colony-to-colony variability reported in the literature.

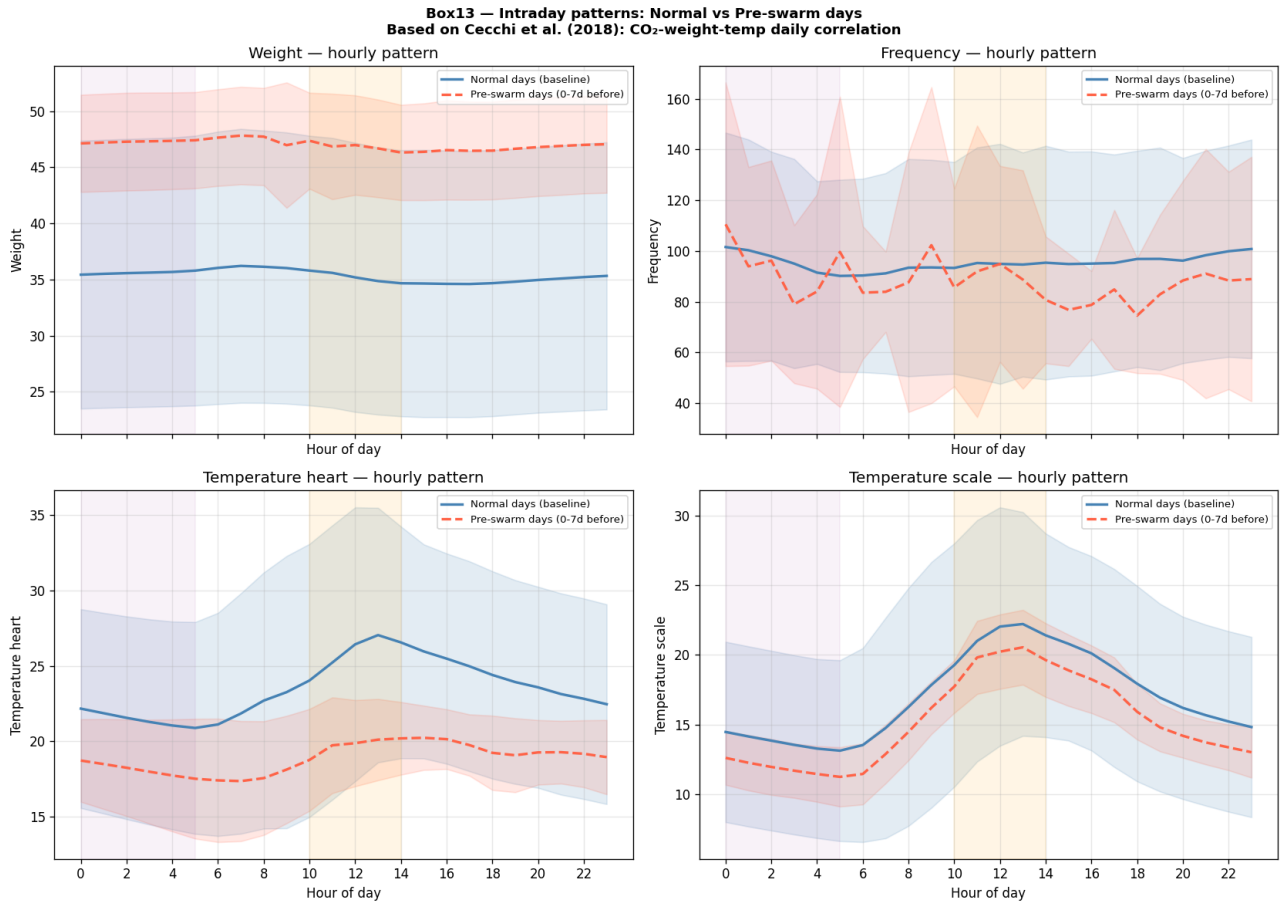


Fig. 14: Box13 mean hourly profiles (Normal vs Pre-swarm days, shaded band = ±1 SD) for Weight, Frequency, Temperature heart, and Temperature scale. Based on Cecchi et al. (2018) CO_2–weight–temperature correlation approach. The morning weight dip during pre-swarm days (bees exiting without foraging return) motivates the morning-window features in C02.

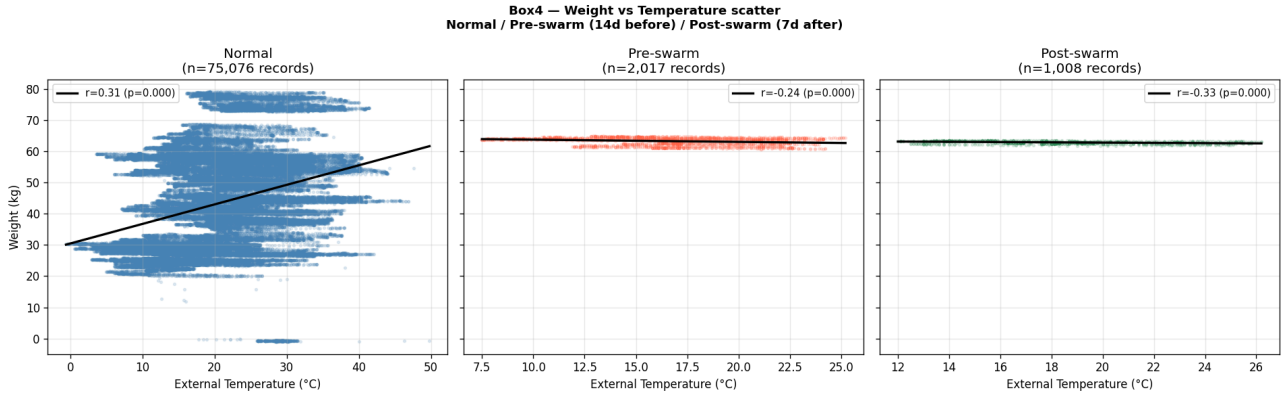


Fig. 15: Box4 weight–external temperature scatter by behavioural period (10-min records, spring–summer only). Normal days show $r = +0.31$ ($p < 0.001$, $n = 75{,}076$); pre-swarm days drop to $r = -0.24$ ($p < 0.001$), confirming the correlation reversal pattern observed in Box13 (Figure 2) and across all hives (Figure 10).

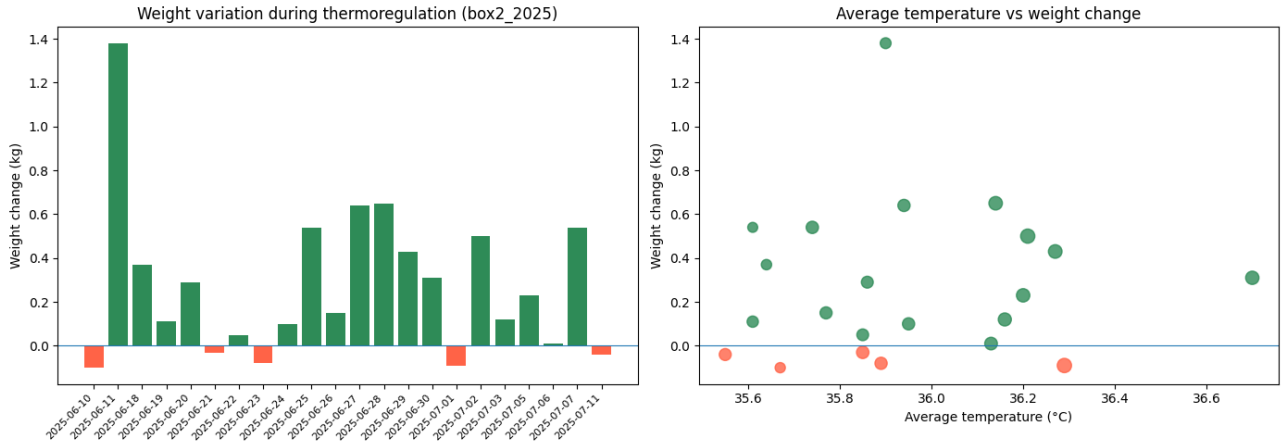


Fig. 16: Box2 (2025) daily weight variation during the thermoregulation period (June–July). Left: bar chart of daily weight change; green bars indicate days with net gain (foraging activity), red bars net loss (swarm preparation or cluster repositioning). Right: scatter of weight change vs. mean nocturnal internal temperature, showing that days with the largest gains cluster around 35.9–36.2 °C, consistent with optimal brood-temperature regulation.

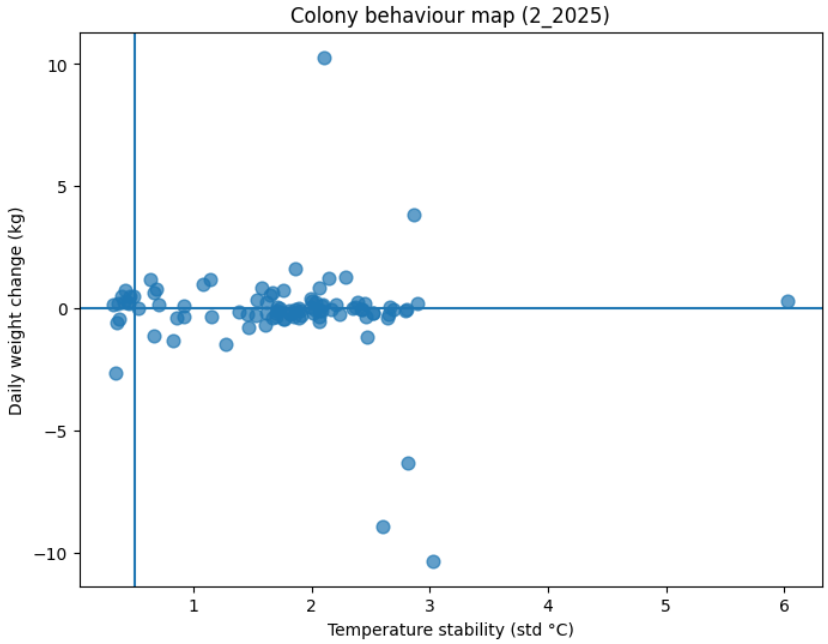


Fig. 17: Colony behaviour map for Box2 (2025): daily weight change vs. temperature stability (std of internal temperature over 24 h). The vertical blue line marks the thermoregulation stability threshold (std $\approx 0.5\,^{\circ}$C); the majority of days fall below it, indicating active brood thermoregulation. The anomalous cluster at temperature stability ≈ 2–$3\,^{\circ}$C with large negative weight changes (down to -10 kg) corresponds to swarming and beekeeper intervention days, where the internal sensor decouples from brood regulation.

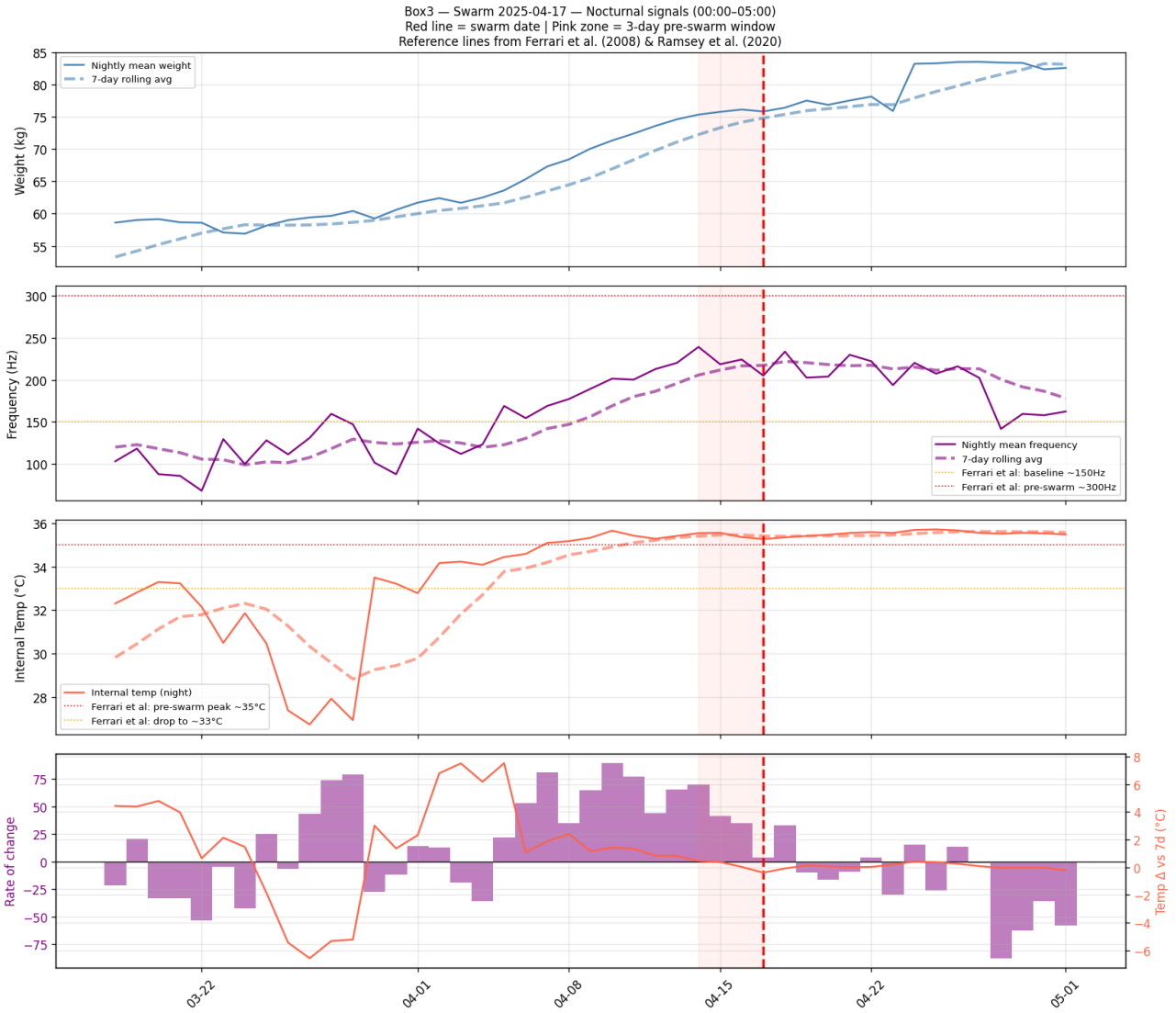


Fig. 18: Box3 (swarm 2025-04-17): nocturnal signals (00:00–05:00) in the 60 days surrounding the event. From top to bottom: nightly mean weight with 7-day rolling average; nocturnal mean frequency with Ferrari et al. [4] baseline (∼150 Hz) and pre-swarm (∼300 Hz) reference lines; internal temperature with Ferrari et al. reference thresholds (+35 °C pre-swarm, drop to +33 °C); and rate of change with temperature delta vs. 7-day average (orange). The pink zone marks the 3-day pre-swarm window; the red dashed line is the swarm date.

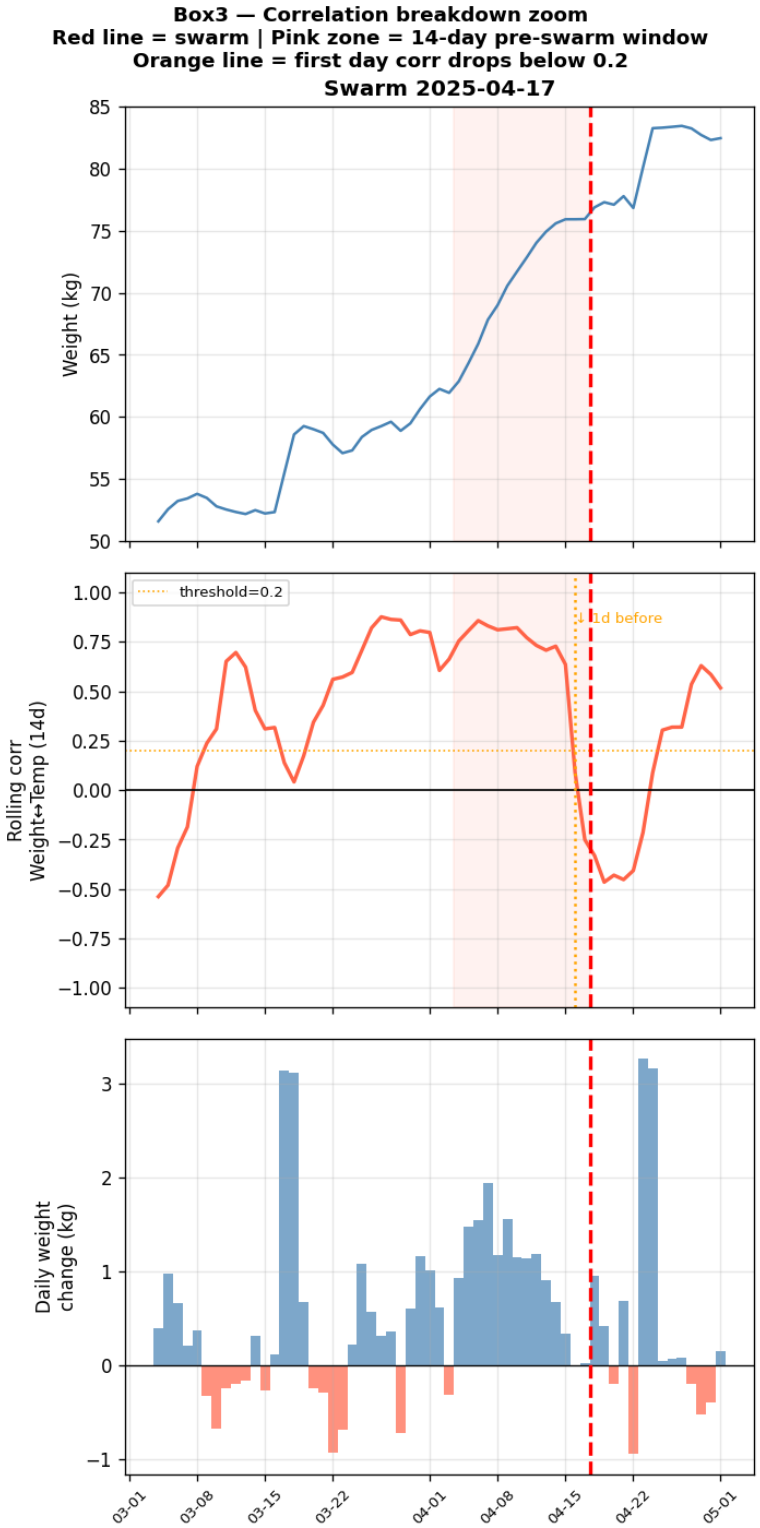


Fig. 19: Box3 (swarm 2025-04-17): rolling 14-day weight–temperature correlation zoom. Top: raw weight (50–85 kg, reflecting the honey accumulation period). Centre: rolling correlation drops below the 0.2 threshold just 1 day before the swarm — the shortest early-warning lead time observed across all events — before recovering post-swarm. Bottom: daily weight change, with the pre-swarm window highlighted.

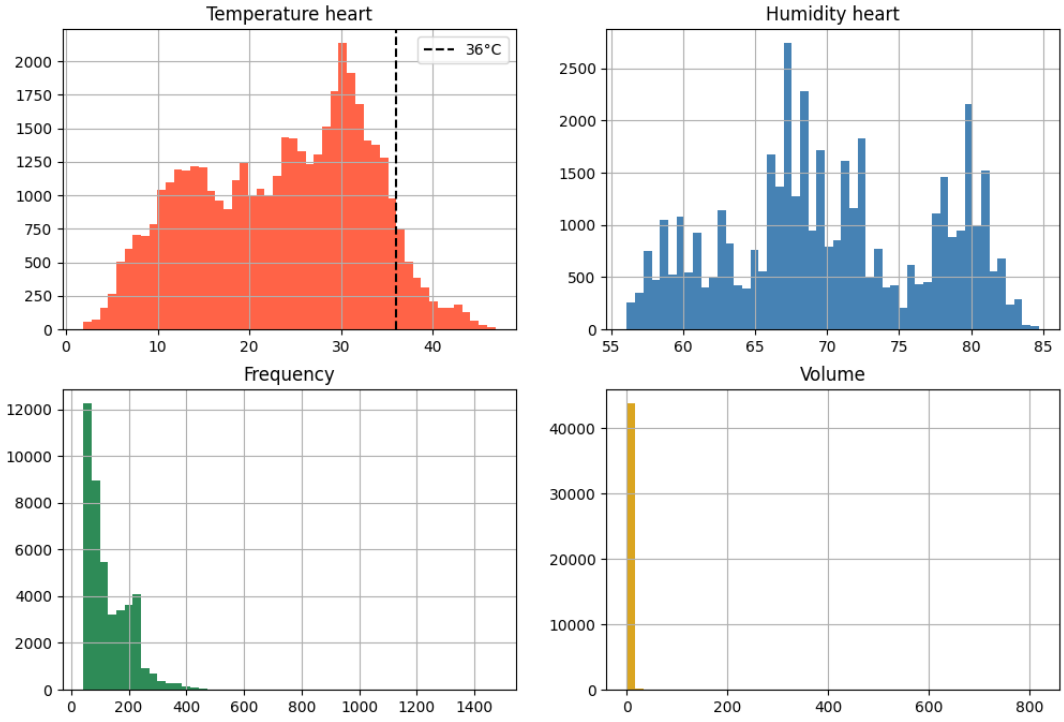


Fig. 20: Distribution of raw heart-node sensor readings across all seasons of hive 4. *Top-left*: internal temperature peaks at 30–35 °C with a reference line at 36 °C (healthy brood temperature); the long left tail reflects winter periods with low colony activity. *Top-right*: internal humidity shows a broad, irregular distribution (55–85%), suggesting high variability between hives and seasons. *Bottom-left*: acoustic frequency is heavily right-skewed, with most readings below 400 Hz and a sparse tail up to 1,500 Hz. *Bottom-right*: volume is near-zero for the vast majority of readings, confirming that this channel carries little discriminatory information and motivating its low feature importance in all model configurations (Tables 9–11).