Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A Sequential Multimodal Variational Autoencoder Framework for State Prediction in Synthetic Multi-Agent Honey Bee Colony Simulations

Piotr Książek

piotr.ksiazek@pwr.edu.pl

Wrocław University of Science and Technology

Jan Przepióra

AGH University of Krakow

Article

Keywords:

Posted Date: November 20th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10831035/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

A Sequential Multimodal Variational Autoencoder Framework for State Prediction in Synthetic Multi-Agent Honey Bee Colony Simulations

Piotr Książek[1,*] **and Jan Przepióra**[2,3]

[1]Wrocław University of Science and Technology, Department of Acoustics, Multimedia and Signal Processing, Wrocław, 50-370, Poland
[2]AGH university of Krakow, Faculty of Mechanical Engineering and Robotics, Cracow, 30-059, Poland
[3]King Abdullah University of Science and Technology (KAUST), Computer, Electrical and Mathematical Science and Engineering Division (CEMSE), Thuwal, 23955-6900, Saudi Arabia
[*]piotr.ksiazek@pwr.edu.pl

ABSTRACT

Automated bee colony monitoring is a maturing field of research in the Industry 4.0 approach to apiculture. Despite the great efficacy of models used for various tasks and some attempts at representation-based solutions for bee state modelling, no concrete framework has been proposed for time series predictions of bee colony states. This work provides a theoretical framework for modelling the behaviour of a simulated bee colony using a multimodal representation learning approach. For this goal, a synthetic bee colony model was devised, simulating the behaviour of individual bee agents at scale. The resulting model was used to create a dataset of simulated readings equivalent to measurements over multiple years in multiple apiaries. The synthetic dataset was then analysed using a MMVAE-LSTM (Multi-Modal Variational Autoencoder - Long-Short Term Memory) model. The multimodal approach was verified in comparison to classical methods of modelling and was found to outperform them significantly. The method has shown great resilience to missing modalities during inference, providing proof the latent space representation contains essential bee colony state information. The presented approach will allow for analysis of future large datasets combining colony weight measurement with the modular multimodal approach allowing for less rigid modality choice when planning large scale bee colony modelling experiments.

Introduction

Beekeeping is an industry and hobby enjoyed by many people around the world. Bees are important to many ecosystems and often serve the role of primary pollinators both in the wild and in agricultural support. The monitoring of bee colonies has been performed manually by humans for millennia. The manual process of verifying the condition of a bee colony is a well-tested method for establishing its current state and predicting future performance. These inspections, however, are best performed with lowest possible frequency, as every inspection provides a stress factor, which affects the bee behaviour and in some circumstances can be detrimental to colony survival[1]. A growing field aiding management of colonies without the need for manual inspections is known as Apiculture 4.0[2].

Within the field, many methods are used for introducing automated monitoring into the apicultural practice including automated colony weight monitoring, which is utilised for verifying the colony yields and colony presence[3], recording of sounds inside and outside the beehive, vision monitoring of the hive entry, and even low-power radar sensors. These sensing technologies are most often connected with computing devices used for data storage and with theoretical capability for on-line data processing. These methods are employed in a variety of tasks, such as detection of queen bee presence and condition[4], early swarming prediction[5], Varroa infection prediction[6], colony population prediction[7], bee specimen recognition[8] and monitoring for changes in colony state with time of day[9] as well as time of year[10].

For more long-term modelling, using contextual data such as presence of food sources, weather, and other colonies in the area, other models were developed, using slow-changing variables for prediction of honey yields and colony prosperity. In this approach, using Geographic Information System (GIS) data, predictions could be made about the potential yields of a colony placed in a given area as a tool for planning the placement of mobile or new apiaries[11]. Machine learning has also been used for similar purposes by Campbell et al. with models informed by environmental sensors showing relatively low absolute error values[12]. This approach, however, does not simulate the roaming of bees and the intra-colony specific behaviours, which is a limitation when considering the day to day changes inside a colony.

In contrast, agent based models (ABM) simulate individual honey bee agents placed in an explicit space to model the predicted colony yields and internal variables. Earlier attempts focused on deeply modelling singular aspects of the bee colony such as HoPoMo and TaskSel models developed by Schmickl et al.[13,14] provided insight into the decision making of the bee colony, but did not simulate all variables accurately. A recent foundational model in this field is the BEEHAVE model, originally developed to model the decline in bee colonies around 2014 by Becher *et al.*[15]. The proposed method implements a classical agent based framework with recruitment mechanics, brood dynamics and roaming simulation to model nectar yields. The model is highly stable, when modifying single parameters and provides insight into the internal workings of the bee colony dynamics. This method has since been used widely in industry and in research, with derivative models such as BEESCOUT[16], which introduces detailed pollination modelling as well as BUMBLE-BEEHAVE[17], which extends the model to bumblebee colonies. More recently, some attempts at creating an open-source model for bee colonies have been made, with the Bee++ model[18], which is an open source framework written in C++ designed for ease of modification.

However, utilising even simple predictive models requires vast amounts of diverse data. There exist many datasets in the field of bee colony monitoring, most focused on the specific niche such as the UrBAN dataset released by Abdollahi et al.[19], which contains sound and environmental data from an urban apiary. Such datasets are most often recorded using IoT devices placed inside or outside beehives in protective cages. Another similar dataset in this field is provided by Kulyukin et al.[3] containing bitmap images of the entry to the bee colony as well as colony mass measurements. Such datasets are mostly centred on providing solutions for a specific research problem and are limited in their modality, mostly containing one or two modalities, without expanding onto others. The expanding capabilities of electronic devices applied to agriculture is leading to an increase in both modality and scale of monitoring. Collection of data for such purposes, however, remains a years-long process which begins with the development of measurement systems and ends in the analysis of very complex data streams with often disparate modalities.

The increasing deployment of Internet of Things systems has enabled continuous monitoring of complex processes through heterogeneous data streams generated by multiple sensors and devices. However, the integration and analysis of such streams remains challenging due to differences in sampling rates, temporal resolutions, data distributions, and measurement characteristics across sensing modalities[20,21]. IoT data are also frequently affected by missing observations, sensor noise, outliers, communication failures, and irregular sampling, which can significantly influence the reliability of subsequent analysis and prediction[22,23]. Furthermore, individual data streams often provide only a partial representation of the underlying system, making joint modelling of heterogeneous modalities essential for capturing interactions between different physical and behavioural phenomena[24]. These challenges become particularly important in long-term monitoring applications, where the objective is not only to reconstruct the current state of the system, but also to identify temporal dependencies and predict its future evolution. Consequently, methods capable of integrating heterogeneous and potentially incomplete sensor streams while learning both their shared structure and temporal dynamics are required for robust IoT-based monitoring and forecasting.

Variational Autoencoders (VAEs) provide a probabilistic framework for learning compact, interpretable latent representations of complex, high-dimensional observations, making them particularly suitable for modelling systems in which multiple heterogeneous data sources describe the same underlying process[25–29]. In multimodal settings, information from different modalities can be encoded into a shared latent space, allowing complementary information to be integrated while accounting for modality-specific characteristics and uncertainty. However, when observations evolve over time, conventional VAEs do not explicitly capture the temporal dependencies that govern the dynamics of the latent representation[30,31]. Integrating Long-Short Term Memory (LSTM) networks into the VAE architecture enables the model to learn temporal relationships from sequential observations and to propagate information from previous time steps when estimating future state of the system[30,32,33]. Such a multimodal recurrent VAE is therefore well suited to complex biological systems, where behavioural, physical, and environmental measurements provide complementary views of the system's evolution.

In our work, we developed an object-based bee colony model utilising abstract 2D space, which was used for the creation of data for modelling a multimodal VAE model for representation of the modelled colonies. The approach was similar to models seen in the literature, with a simplified colony dynamics model taking into account the ambient light level, environmental abundance of nectar as well as variability in bee biology and transmission of information. The simulated multimodal measurements were then used for the development of a representation learning framework, utilising MMVAE-LSTM with Product of Experts (PoE) latent space representation.

Results

Synthetic colony dataset

For this experiment, 32 distinct simulations were performed, each calculated for a maximum length of 60 days of simulated time. Within the 32 calculated simulations, 16 were designated as plentiful nectar yield simulations and 16 were designated as poor nectar yield scenarios.

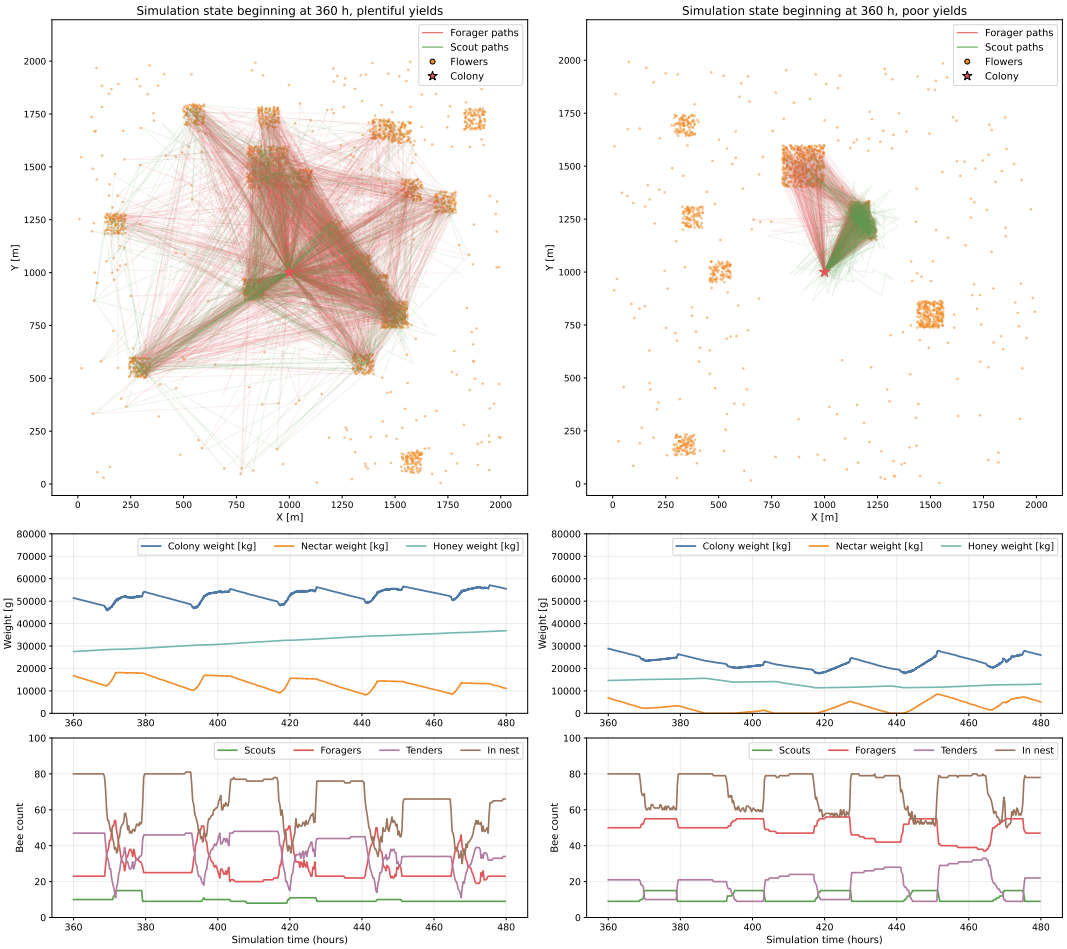


Figure 1. A slice of a selected day of simulated colony activity, with highlighted flight trajectories for workers and drones, simulated measured colony weight and bee role counts.

Figure 1 shows a five days long slice of simulations, starting at the 10-th simulated day. The top subplot shows the positions of flowers within the model environment, with small, dense clusters of flowers grouped in meadows, which simulate more concentrated flower populations and individual flowers spread across the space. The bee flight paths are highlighted in two specific colours, with red paths showing forager flights and green paths showing scout flights. Most paths converge at the hive location, located in the centre of the simulated space. Significant differences can be seen between the plentiful scenario and the poor yield simulation, with the poor yield simulation showing a much lower flower density and the scout activity concentrated on the meadow closest to the central hive. This is caused by the simulation having less nectar-rich flowers, which causes the colony to focus on a known good source of nectar rather than spreading widely to locate all sources in roaming radius.

Within the middle subplot in Figure 1, the honey and nectar amounts can be seen, in addition to colony weight. Note that in the plot, the colony weight is represented by Eq. 1, where m_h is the mass of produced honey, m_n is the mass of gathered nectar and $m_{b,i}$ is the mass of the $i-$th bee. In the poor yields dataset, a decrease in the amount of honey is visible, as the nectar reaches zero during the night. This is in contrast to the plentiful yield scenario, in which the amount of honey increases monotonically.

$$M_c = m_h + m_n + \sum_{i}^{N} m_{b,i} \tag{1}$$

The third subplot in Figure 1 represents the dynamic switching of bee roles, as the colony auto-balances its population to reflect the analogous behaviour of real bee colonies. Within the subplot, a diurnal cycle can be seen, typical of a bee population, with bees returning to the hive during nighttime and leaving quickly in the morning. The daylight level system of the simulation ensures a gradual change in operation from nighttime mode to daytime mode. In the poor yield scenario, the bees show a less

dynamic variation in the number of workers/foragers despite similar similar total sizes of colonies. This is caused by the low amount of nectar - as the colony's role assignment algorithm does not prioritise tender bee presence if the amount of nectar to be turned into honey is not sufficient.

Table 1. Summary of scenarios calculated for the synthetic dataset generation. Parameters are presented with their standard deviation.

Scenario	Collapsed colonies [-]	Simulation length [days]	Honey production [g/week]	Final colony size [agents]
Plentiful	1	56.5 ± 13.9	5801.4 ± 1547.6	139.1 ± 35.9
Poor	14	23.4 ± 17.8	199.8 ± 705.9	24.4 ± 53

Overall, not all of the simulations reached steady state and finished the computation at 60 days. There were examples of scenarios where the colony was unable to locate sufficiently nutritious meadows, which caused a rapid decline in population due to lack of sustenance. Table 1 shows the comparison between scenario types. It can be seen that within the poor nectar availability scenario, nearly all simulations did not conclude, with a mean simulation length of nearly 24 days. In contrast, only a single plentiful simulation was terminated early due to colony collapse. Within the plentiful scenario nearly 6 kg of honey are produced every week on average. This is a much higher amount than colonies typically produce during a week even in high season, however due to the simplifications of the numerical model, it may be taken as a satisfactorily accurate approximation of very high honey production. Note also that the final colony size is presented with the number of agents, with each agent representing 500 bees, meaning the mean final colony counted around 69,5 thousand specimens, which is, similarly to the honey production figure, higher than typically found in nature.

Representation model

The MMVAE-LSTM model training process is seen in Figure 2. Training and validation losses follow a predictable curve, learning the optimal data representation within 26 epochs, with each epoch following showing an increasing difference between training and validation losses. Due to the KL-regularisation used in the loss function, with changing β values, the Kullback Leibler divergence rises within the first epochs and settles in a low point around the 26-th epoch, notably, decreasing slightly as the model reaches overtraining territory. Within the training process, no obvious signs of over- or under-fitting can be seen.

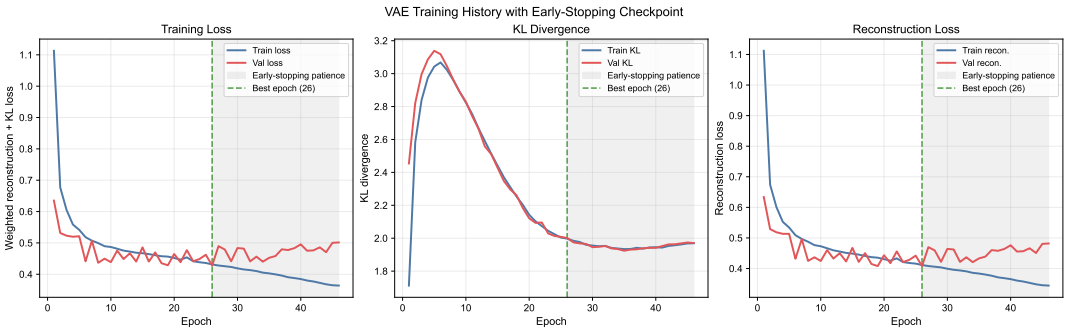


Figure 2. Training and validation losses of the MMVAE-LSTM network with full modality presence.

The reconstructions provided by the complete-modality model for a selected non-steady state scenario with 20 sample horizon can be seen in figure 3. Each panel within the graph corresponds to a modality used within the model. Within the top graph, the reconstruction of the 100 Hz simulated sound spectrum point can be seen, showing good adherence to the overall trends within the test data, but neglecting to model the high-frequency changes in the spectrum itself. This is expected, as high-frequency changes are essentially caused by simulated noise, similarly to the highly variable signal recorded inside the hive within previous experiments and the literature[6,9].

The middle panel of Figure 3 contains the prediction of the hive mass differential, which follows a similar trend as the spectral reconstruction. The model captures the overall trend, including the rapid departure of bees in the morning (downward spike) and the return of bees at night (upward spike). While not fully covering the spikes, the model represents them, showing the representation contains the expression of these changes. Overall, the predictions follow the data closely.

The bottom panel of the figure shows the reconstructed bee counts. Even with this parameter being the most synthetic of the features presented to the model, and highly non-linear, the model is capable of near-perfectly following the data in steady state (on the left of the graph), faltering significantly only in non-stationary state, where the colony switches to intensive search



Figure 3. Predictions for all modalities at maximum horizon of 20 model steps (5,5 hours future prediction).

for resources with large numbers of scouts present during the day, creating essentially a square wave, during which the model struggles to correctly represent the scout counts. This, however, is an extreme case of the model behaviour and, while the differences are stark, they are represented in the predicted reconstruction.

The learnt representation within the latent space of the PoE model shows differentiation in extreme states. As the space is not inherently interpretable, Principal Component Analysis (PCA) was used to show the differences between latent vector values within changing extreme states of the model. Figure 4 shows an example of such representation, with the top and bottom percentiles of the mass differential highlighted as triangles. It can be seen that there is clear separation within this representation between the states, even in such a simplified two-dimensional space. This highlights the non-triviality of the latent representation manifold, and the fact that points connected with the top differential percentile are spread wider than those corresponding to the bottom percentile shows the continuum of the manifold. This is explained by the multiplicity of reasons for high values of differential - sudden influx of nectar, bees returning to hive as compared to the sole reason for lowest values, which is caused by bees leaving the colony in the morning. The ring-shaped representation can be interpreted as the model training a periodic signal, akin to a phase trajectory embedded within the latent values.

The results of the missing modality experiment and the base model evaluation are presented in table 2. The metrics calculated for evaluation are the Mean Absolute Error (MAE) and the Mean Square Error (MSE). While largely interchangeable, the MSE is more sensitive to large residual outliers, making it informative in the context of predictive modelling. The standard deviation of error values over the test signal windows is presented as well, to differentiate the error stability for the methods under test. The VAE model is compared to classical linear regression and persistence modelling with the persistence baseline using the final sample as a prediction (optimistic). The models were evaluated over non-overlapping 20 sample long windows, and the mean errors are derived from the mean residuals within these windows.

The developed MMVAE-LSTM representation model has shown results far outclassing the classical methods over the modelled horizon period. Although the results of estimating the future states of the bee spectra are similar to linear regression, as can be expected from a noisy and slow-changing parameter, both methods are marginally better than the naive persistence model, highlighting the nature of the feature. When it comes to the estimate of the differential mass and the count of bees, the representation-based method shows its performance with results much better than the baseline methods at 0.039 MSE for mass differential prediction and 0.03 MSE for bee count estimation. The bee count estimate is most notable, as the highly non-linear parameter eluded the baseline models completely, providing results on the order of 0.27 to 0.44 MSE. The VAE model is thus

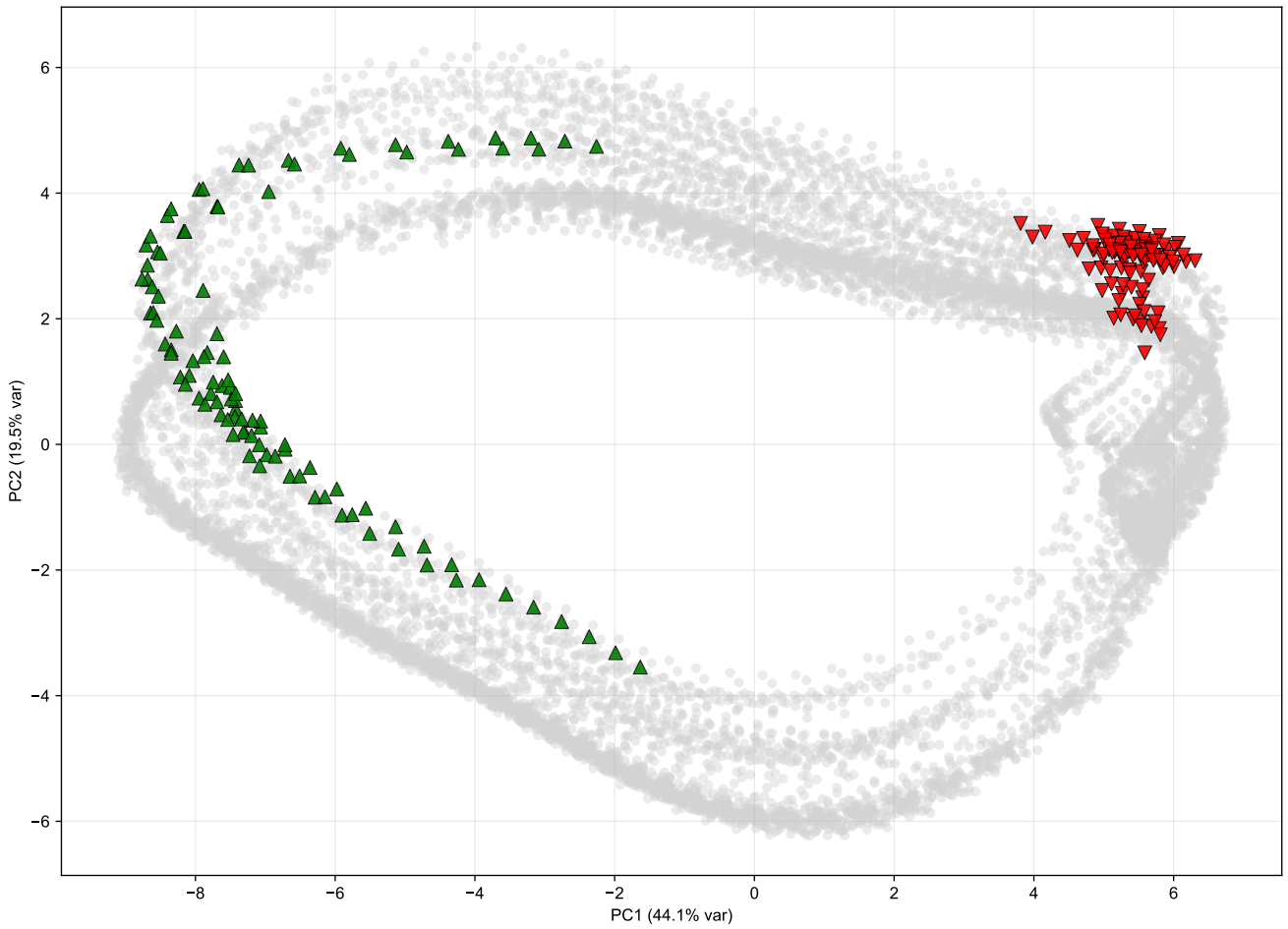


Figure 4. PCA (PC1, PC2) representation of latent space, with green triangles representing recordings corresponding to the 99-th percentile of the mass differential and red triangles representing the 1-st percentile of the mass differential.

capable of storing and predicting the state of the bee colony using its latent 64-neuron representation with both lower errors and error variability than the baseline methods.

The multimodal PoE model was evaluated in terms of its performance under missing modalities. This tested both the modality reconstruction capability as well as overall model performance in the face of missing data. The most important parameter for the model representation is the simulated spectra parameter, the removal of which increases the modelling error of both the remaining modalities significantly from sub-unitary MSE error levels to an error value much greater than 1, especially considering the mass differential feature. While important to the modelling of itself, the context provided by the mass differential expert impacts other features much less, with the highest error seen in the missing modality itself, which, however, is still better than a persistence model and more similar to linear regression. The bee count modality seems to be a middling importance feature, showing a larger error increase when taken out than mass but smaller than spectra. Overall, the results indicate a coherent multimodal representation, where every modality contributes to the overall state of the model, with the dominating feature being the simulated sound spectrum.

Discussion

The presented experiment aimed to develop a theoretical proof of concept multimodal machine learning representation-based model for the prediction of honey bee colony state. To perform this task, an object-based agentic bee colony model was devised with direct simulation of bee agents performing tasks in an abstract space over multiple time steps. The model was utilised to create a multimodal simulated dataset containing the synthesised sound spectra of bees buzzing within the beehive, the mass of the colony itself as well as the bee counters containing information on the caste composition of the hive as well as the number of bees inside. This dataset was then processed using a multimodal MMVAE-LSTM model with a Product of Experts latent

Table 2. Modelling experiment result table with the base experiment as well as the missing modality test.

Model	Missing modality	Modality	MSE	MSE std.	MAE	MAE std.
Persistence	None	Spectra	0.05	0.10	0.13	0.14
		Mass diff	2.62	5.27	1.00	1.27
		Bee counts	0.44	1.02	0.34	0.53
Linear Regression	None	Spectra	0.03	0.07	0.12	0.11
		Mass diff	1.08	3.15	0.63	0.83
		Bee counts	0.27	0.59	0.33	0.37
VAE	None	Spectra	0.03	0.07	0.11	0.10
		Mass diff	0.39	1.13	0.34	0.52
		Bee counts	0.03	0.10	0.11	0.14
VAE	Spectra	Spectra	1.36	1.16	1.05	0.49
		Mass diff	1.02	1.54	0.80	0.61
		Bee counts	0.78	0.83	0.79	0.40
	Mass diff	Spectra	0.05	0.10	0.14	0.13
		Mass diff	1.23	3.52	0.67	0.88
		Bee counts	0.17	0.33	0.27	0.29
	Bee counts	Spectra	0.09	0.20	0.19	0.22
		Mass diff	0.46	1.42	0.38	0.56
		Bee counts	0.54	0.99	0.52	0.48

space formulation and symmetric encoder-decoder pairs. The model learned the representation within 26 training epochs using the Adam optimiser with a KL-Divergence regularisation trick utilised to stabilise the representation. The model was evaluated against classical baseline methods and its behaviour was tested in case of missing modality data.

While other works explored numerical modelling of bee colonies to greater extent and with great accuracy, specifically the BEEHAVE family models[15], the approach to modelling in the case of our experiment worked on different principles and posed different goals. In this work, the emphasis was placed on the high density time-series data of a system highly resembling a bee colony, while not calibrating the model towards any single scenario or dataset, as such multimodal datasets have not been recorded of yet. As such, comparison to existing methods is not strictly possible, with our developed model showing great promise for future modification in an open-source approach as well as the existing model exhibiting highly complex, non-linear behaviour, effectively modelling a bee colony-like system.

Although representation-based models were tested in the literature, notably with the work by Cejrowski et al.[34], where contrastive learning was used in conjunction with Autoencoders with great success, utilising mel-spectrograms along with temperature concatenated to the latent space for conditioning. Our approach differs from the ones seen before in the full implementation of a multimodal VAE model with Product of Experts latent space representation and inherently utilised LSTM encoders and decoders with spectrum additionally processed via CNNs. Our low modelling errors are compared to baselines highlight the efficacy of this approach and show the potential of using such methods on large, multimodal datasets recorded in real bee colonies. The compression of colony parameters into a small latent vector allows for better generalisation of results, allowing downstream models to learn on the most important embeddings instead of sensor noise and measurement spot particularities.

The performance of the developed model when faced with lacking modality data, a common problem in IoT applications in rural, hard to service locations, degrades gracefully, increasing the error, but not failing outright in the case of missing bee count and mass differential data. The overall degradation of results with missing modalities shows that all features are used in the PoE process, but the overall representation allows for a partial feature reconstruction even without the specific feature's leading context. The model's reliance on simulated spectral data stems from its relatively high dimensionality (512) as opposed to the bee counts (4) and hive mass (1). The representation quality is highlighted in the structured latent space PCA representation, which shows a clear separation between top and bottom mass differential values, as well as an overall ring-like structure in 2D PCA, which is indicative of a periodic-like system. In this case correctly, as bees operate on a diurnal cycle.

While the results achieved by the developed model far outclass classical methods, they are not derived from a true, measured dataset. The developed bee colony model allows for simulation of a years-long measurement session in a matter of hours, but is still a much simplified substitute for real measurements. In the context of this work, the model may be expanded to include non-trivial geometry, pollen collection, swarming mechanics, *et cetera*. The model expansion, however may not impact the overall conclusion of the work, where effort may be better spent preparing a large-scale multimodal measurement

session. As such, the simulation used in this work, while limited in scope, serves as an engine for the proof of concept machine learning implementation. The model, being a proof-of-concept, does not implement all the mechanics realated to real-world measurements such as weather events or other external factors impacting the colony and the physical sensor noise, which can significantly impact the performance of application methods[35].

This leads to the natural conclusion regarding future work of this approach, which is the real-world verification of the developed machine-learning method on a real, highly multimodal dataset gathered in a diverse set of locations over at least one apicultural season. This approach may naturally lead to a true representation of a bee colony state, which may power future downstream task applications such as anomaly detection, swarming prediction, queen bee state estimation, and even honey yield modelling and infection prevention. Such applications, however are fully contingent on the creation of large datasets and validation of the models on measurement data.

Methods

To develop the framework for modelling honey bee colony state parameters, a simplified ABM approach was used for modelling of the honey bee colony dynamics, with explicit rules on inter-specimen relationships and colony management. The devised model was then used to create a dataset of simulated measurements containing the results of 16 discrete simulations of colonies placed in environments with plentiful nectar, and 16 simulations of colonies placed in poor yield conditions. The dataset was analysed using a multimodal VAE with LSTM layers to model both the inter-feature interactions and temporal relationships within the colony. The approach was then tested in modality absence scenarios and compared to baseline prediction methods for each modality analysed.

Bee colony model

The devised bee colony model is based on a time-domain agent simulation framework developed using the Python 3.11 programming language. The simulation is object-based, with the main governing object being the simulated world, in which all agents and agent-interactable object instances are stored. The diagram of the simulation is shown in figure 5 along with its main simulation loop.

The simulation logic lives inside the bee and colony objects, with the colony being responsible for managing the number of bees for each role and bees responsible for their behavioural characteristics. The bee objects implement a state machine paradigm, with the possible states varying across specimen types. The three specimen types of bees are forager, tender, and scout. Foragers and scouts are capable of leaving the hive and performing tasks outside, while tenders stay inside the hive and perform actions connected to raising new brood and transforming nectar to honey. Each bee agent simulates a cluster of individuals with configurable size, which for the case of this model is set at 500 bees.

The scout bee can be in an idle state, exploring state, or returning to hive. When in the exploring state, if the scout is unaware of any flowers within its area, it will continue to roam in a Lévy flight pattern, as observed in normal honey bee roaming[36].

If the scout perceives a flower within its vision range, it will move towards it, and when in nectar operation range, it will take a sample, feed if hungry, and continue exploring the area for a set amount of time before returning to the hive and passing the knowledge about the flowers to the colony. If there are no flowers known to contain nectar, but there are historical locations of such flowers, the scout will be drawn in its flight to the nearest known historical meadow location. The flight back to the hive follows the shortest possible path.

The forager bee can be idle, foraging, or returning to hive. When foraging, the bee chooses at random a non-exploited flower from the list of known flowers, with weighting to assign exponentially higher probabilities $P(f_i)$to flowers closer to the colony, as in Equation 2, where f_i is the i-th flower, (x_a, y_a) are the euclidean coordinates of the worker bee and (x_i, y_i) are the coordinates of the flower.

$$P(f_i) = \frac{\frac{1}{\sqrt{(x_a - x_i)^2 + (y_a - y_i)^2 + \varepsilon}}}{\sum_{f_j \in U} \frac{1}{\sqrt{(x_a - x_j)^2 + (y_a - y_j)^2 + \varepsilon}}} \tag{2}$$

The worker then moves towards the chosen flower, gathers nectar, and returns to the colony if its loading capacity is full or if there are no known flowers to exploit. It then deposits the nectar inside the hive and begins the cycle again.

Both forager bees and scout bees utilise the simulated daylight system for their decision making process, with the bees leaving the hive in the morning and returning in the evening at set daylight levels. The daylight model is based on the simulation time, and is represented by a normalised scalar value $L(t)$, modelled with a standard Hanning window according to Equation 3, where t_{sr} is the sunrise time and t_{ss} is the sunset time. The light level is set to 0 for all times during the night.

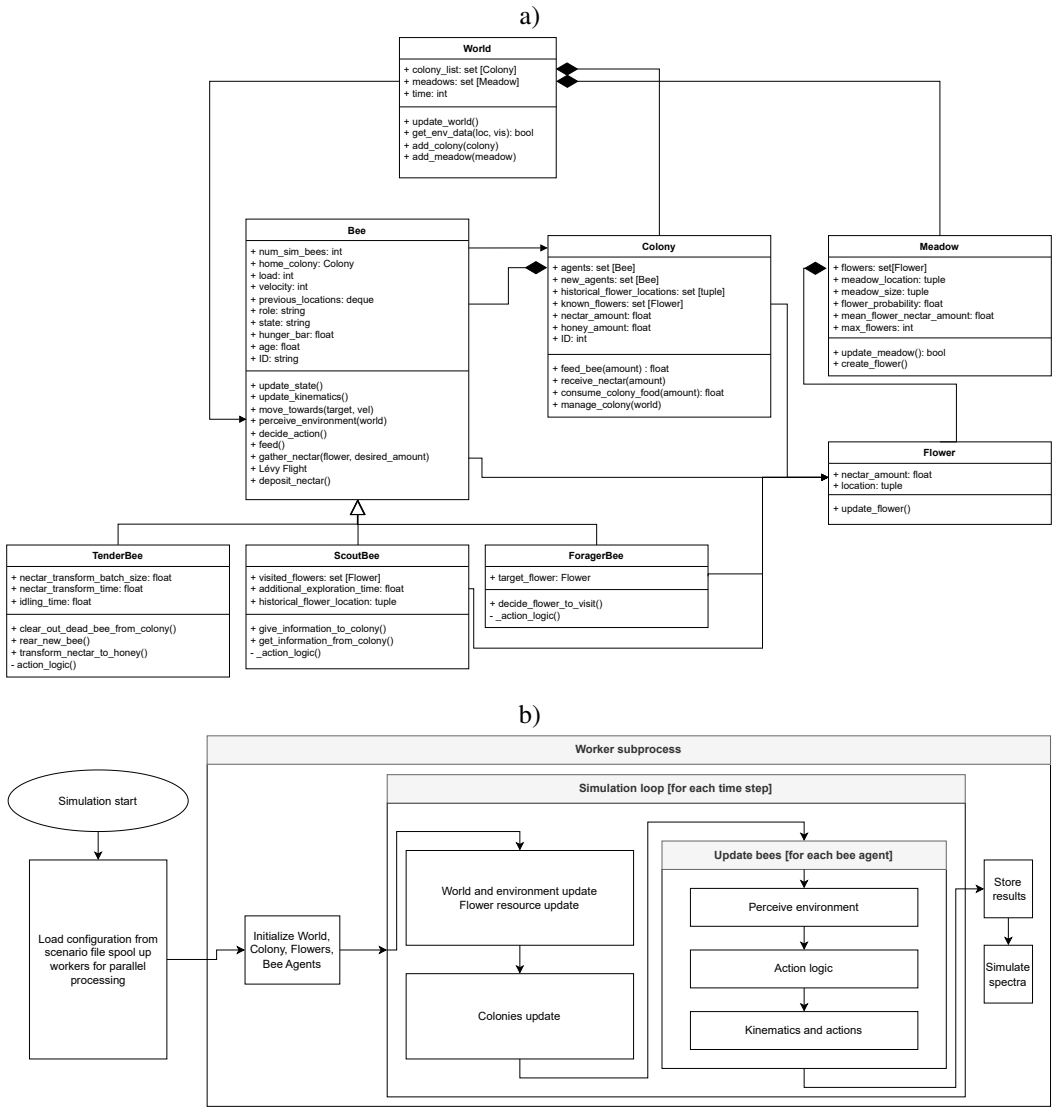


Figure 5. Simulation flow diagrams with a) being the class diagram of objects present in the simulation and b) being the specific simulation loop process used for generating the synthetic dataset.

$$L(t) = \begin{cases} 0.5\left(1 - \cos\left(\frac{2\pi(t - t_{sr})}{t_{ss} - t_{sr} - 1}\right)\right), & t_{sr} \leq t < t_{ss} \\ 0, & t < t_{sr} \vee t \geq t_{ss} \end{cases} \tag{3}$$

The tender bees stay inside the hive and can be idle, clearing dead bees, rearing new bees, or transforming nectar. Each action taken by the tender is connected with a set delay time, to simulate the tender performing the task. When clearing dead bees, the tender agent flags dead bees to be removed from the agents list after the action timer finishes. When rearing new bees, the agent chooses from the list of incoming bees (brood) and feeds them, which accelerates their rearing process. Transforming nectar works by directly subtracting from the colony nectar amount and adding a scaled amount of honey, compensating for the evaporation of water by dividing by a set value.

The colony itself manages the bee population and the knowledge state of the hive. It governs the list of known flowers, current agents, new agents, and the agent composition of the colony. The colony update process first verifies the daylight level to establish the mode of operation, then it filters the exploited flowers from the known flower set. After that, agent population management is performed with bees with positive age (fully reared bees) being put into the standard agents list. Then the dead bees flagged for removal by tenders are removed, and the colony proceeds to role rebalancing. The rebalancing determines uses

target ratios for bee specimen types in the colony composition to rebalance the agent roles, changing the bee object types. This is most visible during the night, where inside a colony with plentiful nectar, foragers are transformed to tenders to accelerate the honey transformation process. Finally, new future agents are placed in the pending agent list with a set negative age, with a maximum of 1 bee per update being placed in the list.

All parameters of the developed model can be modified using the scenario settings file. Many actions performed by bees and their parameters show inherent variability, as such within the simulation parameters are calculated during runtime with a set standard deviation from the scenario-specified mean value. This ensures the numerical divergence of the chaotic systems produced by the simulated bee colonies.

Data preparation

The generated synthetic dataset contained two specific types of scenarios representing different simulated conditions. The first scenario type was the plentiful case, where four main meadow objects are defined, one of them used to simulate randomly growing wildflowers and three others simulating large meadows with high simulated flower density. Besides the four main meadow objects, 15 randomly-placed small meadows are also input into the model. These flower clusters are small in size and contain fewer flowers than the explicitly defined sources.

The poor yield scenario is identical to the test one in every aspect, except for the flower nectar amount, which is reduced by 60% and the random meadow count, which is reduced to 5. Both simulation test scenarios are defined as JSON-formatted configuration files and are available in the model repository.

The prepared model outputs a simulated spectrum measurement, simulated bee type counts, as well as the model hive mass at intervals of 100 model steps, equivalent to 1000 seconds, due to the model time step. The sound spectrum simulation is performed by additive synthesis with settings based on the actions taken by each bee specimen. The sound model can be written as a sum of background Gaussian noise and corresponding sinusoidal sounds of possible actions a within the set AC of all the bee agents as seen in Equation 4.

$$S(t) = s_n + \sum_{a \in AC} s_a(t) \tag{4}$$

The action sound is described in detail in in Equation 5, where summing over the set of all bees performing the action a, $A_{a,k}$ is the set amplitude randomised with a uniform distribution in a set range $A_{a,k} \sim U(A_{min}, A_{max})$. similarly, $\gamma_{1,a,k}, \gamma_{2,a,k}$ are randomisation factors for the frequency of the performed action, taken from a uniform distribution $U(\gamma_{min}, \gamma_{max})$. Finally, f_a is the set base frequency corresponding to the action being performed by the specific bee and w_k is the attenuation factor based on the relative location of the bee and the simulated microphone.

$$s_a(t) = \sum_{k=1}^{K_a} A_{a,k} w_k \left(\sin(2\pi\gamma_{1,a,k} f_a t) + 0.5 \sin(4\pi\gamma_{2,a,k} f_a t) \right) \tag{5}$$

The attenuation factor is calculated according to Equation 6, where $\mathbf{r}_k$ is the location vector of the k-th bee agent and $\mathbf{r}_{mic}$ is the location vector of the microphone.

$$w_k = \frac{1}{||\mathbf{r}_k - \mathbf{r}_{mic}||^2} \tag{6}$$

The fundamental frequency for each action performed by bee agents was set to simulate the approximate spectrum of sounds recorded within a bee colony as seen in previous experiments and literature[10]. The frequencies used for actions performed include the following.

- Transforming nectar: 90 Hz,
- idle: 120 Hz,
- rearing new bees: 140 Hz,
- exploring: 220 Hz,
- returning: 220 Hz,
- foraging: 220 Hz,

- clearing dead bees: 250 Hz.

The sound is synthesised with a sampling frequency of 4 kHz and subsequently the linear spectrum estimate is calculated using the Welch method with an FFT length of 1024 samples providing a 512 value long feature vector.

The simulation results, containing sampled simulated data, required transformation for use in neural networks training. For this purpose, the hive weight, simulated spectrum, and bee count data from each simulation is loaded. The simulation time step is used as a common key to align the independent observations, which produces a relationship between model step and calculation step, which in the case of the model is 100 to 1, meaning one model step is 100 simulation steps. Then the mass numerical differential is calculated and the metadata is prepared per-simulation to preserve the distinction of different runs within the ML dataset. Finally, the feature arrays are normalised independently using Z-score standardisation and the array is saved for training with a total single vector length of 517 values, where 512 is the spectrum, 4 is the bee count data and 1 value corresponds to the hive mass differential.

MMVAE-LSTM modelling

The proposed Multimodal Variational Encoder–LSTM framework is designed to learn a compact probabilistic representation of the temporal state of the beehive from heterogeneous, temporally aligned sensor modalities and use the representation for simultaneous multi-modal forecasting. The architecture follows a encoder-latent representation-decoder structure presented in Figure 6. The input data are first separated into modality-specific streams, including spectral measurements, beehive mass, bee count, and any potentially additional sensing modalities - the system can be easily expanded to harbour modalities not yet accounted for. In addition to the measurement data, each modality is associated with a binary modality flag that indicates whether the corresponding modality is enabled for training. When the flag is set to false, the selected modality is excluded from the encoding and subsequent multimodal fusion, allowing the model to be trained with different combinations of available modalities without requiring changes to the overall architecture. Each data stream is processed by a dedicated encoder designed according to the characteristics of the corresponding data. The spectral modality is first transformed using a one-dimensional convolutional neural network, which extracts local features from the spectral measurements, and is subsequently processed by an LSTM to capture their temporal evolution. The beehive-mass and bee-count streams are independently processed using dedicated LSTM encoders. Additional modality data can be processed via a modality specific encoder-dependent on the data characteristic. This design allows each modality to retain its individual temporal and structural characteristics before multimodal fusion. The architecture therefore separates modality-specific feature extraction from cross-modal representation learning.

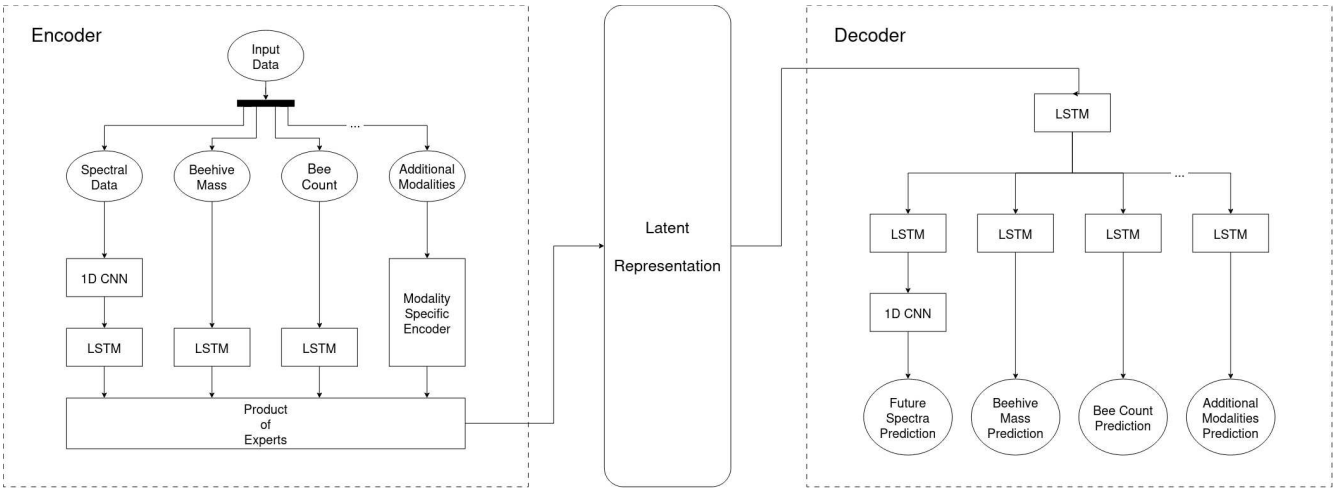


Figure 6. Multi-Modal Variational Autoencoder structure diagram

The outputs of the modality-specific encoders are combined using a Product of Experts (PoE) formulation. In contrast to direct feature concatenation, PoE represents each modality as an expert contributing information to a shared probabilistic latent space[37]. For each modality m, the corresponding encoder produces the parameters of a Gaussian latent distribution,

$$q_m(\mathbf{z} \mid \mathbf{x}_m) = \mathcal{N}\left(\mu_m, \mathrm{diag}(\sigma_m^2)\right). \tag{7}$$

where μ_m and σ_m^2 denote the modality-specific latent mean and variance, respectively. The multimodal representation is then obtained by multiplying the modality-specific expert distributions together with the unit Gaussian,

$$q(\mathbf{z} \mid \mathbf{x}_1, \ldots, \mathbf{x}_M) \propto p(\mathbf{z}) \prod_{m=1}^{M} q_m(\mathbf{z} \mid \mathbf{x}_m), \tag{8}$$

where

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{9}$$

For Gaussian experts, the PoE can be expressed in terms of precision $\tau_m = \sigma_m^{-2}$. The precision of the resulting joint distribution is therefore

$$\tau = \mathbf{I} + \sum_{m=1}^{M} \sigma_m^{-2}. \tag{10}$$

and its variance is given by

$$\sigma^2 = \left(\mathbf{I} + \sum_{m=1}^{M} \sigma_m^{-2} \right)^{-1}. \tag{11}$$

The corresponding mean is the precision-weighted combination of the modality-specific means,

$$\mu = \sigma^2 \left(\sum_{m=1}^{M} \sigma_m^{-2} \mu_m \right). \tag{12}$$

Thus, modalities with lower uncertainty and consequently higher precision have a greater influence on the resulting latent representation. The resulting distribution,

$$q(\mathbf{z} \mid \mathbf{x}_1, ..., \mathbf{x}_M) = \mathcal{N}\left(\mu, \mathrm{diag}(\sigma^2)\right). \tag{13}$$

provides the common probabilistic latent representation. The resulting multimodal latent representation provides a compact description of the temporal state of the beehive. During VAE training, this representation is treated probabilistically and sampled using the reparametrization mechanism, allowing the model to jointly learn the latent representation and its uncertainty. The latent representation is then provided to the decoder, which is responsible for reconstructing and forecasting the individual modalities.

The decoder first applies a shared LSTM to the latent representation, establishing a common temporal representation of the future state. The resulting representation is then distributed to modality-specific LSTM decoder branches, with each branch dedicated to predicting one output stream, such as future spectra, beehive mass, bee count, or any additional modalities. In the spectra branch, the LSTM is followed by a CNN model. Thus, while the encoder performs modality-specific temporal encoding followed by probabilistic multimodal fusion, the decoder performs shared temporal decoding followed by modality-specific prediction. This arrangement enables the model to exploit relationships between the different sensing streams while preserving their individual output characteristics.

Loss function and learning process

The proposed model is trained end-to-end by minimising a composite variational loss that combines multi-modal reconstruction error with Kullback-Leiber (KL) divergence regularisation term. For an input context sequence, the modality-specific encoders and Product-of-Experts mechanism produce the parameters (μ) and ($\log \sigma^2$) of the shared latent distribution, from which the latent variable ($\mathbf{z}$) is obtained. The decoder subsequently generates a sequence of future observations, and the reconstruction loss measures the discrepancy between the predicted and target trajectories using mean squared error (MSE). The reconstruction component is calculated separately for each modality and can incorporate modality-specific weighting, while a horizon-dependent weighting scheme assigns progressively greater importance to later prediction steps. In addition, temporal-difference losses are applied to penalise discrepancies between the predicted and observed changes between consecutive time steps, thereby encouraging the model to reproduce the temporal dynamics of the underlying system.

The complete reconstruction objective can therefore be expressed as

$$\mathscr{L}_{\text{rec}} = \mathscr{L}_{\text{spectra}} + \lambda_m \mathscr{L}_{\text{mass}} + \lambda_b \mathscr{L}_{\text{bee}} + \lambda_{\Delta,s} \mathscr{L}_{\Delta,\text{spectra}} + \lambda_{\Delta,m} \mathscr{L}_{\Delta,\text{mass}} + \lambda_{\Delta,b} \mathscr{L}_{\Delta,\text{bee}}. \tag{14}$$

The variational component is defined as the Kullback–Leibler (KL) divergence between the learned latent posterior and the unit Gaussian prior,

$$\mathscr{L}_{\text{KL}} = D_{\text{KL}} \left[q(\mathbf{z} \mid \mathbf{x}) \,\|\, \mathscr{N}(\mathbf{0}, \mathbf{I}) \right], \tag{15}$$

resulting in the total objective

$$\mathscr{L} = \mathscr{L}_{\text{rec}} + \beta_{\text{eff}} \mathscr{L}_{\text{KL}}. \tag{16}$$

During training, β_{eff} is gradually increased from zero to the prescribed value during an initial KL warm-up period, reducing the risk of premature latent-space collapse. The model parameters are optimised using the Adam algorithm through backpropagation, with gradients optionally clipped to improve training stability. For each batch, the input context is encoded, the latent variable is sampled, future trajectories are generated by the decoder, and the resulting loss is propagated through the network to update the encoder, PoE-related parameters, and decoder simultaneously. The training procedure evaluates the model on a separate validation set after each epoch, retains the checkpoint with the lowest validation loss, and terminates early when the validation performance fails to improve for a predefined number of epochs.

Model validation and missing modality experiment

Model validation was performed on a held-out part of the prepared dataset with simulation run separation to exclude the possibility of data leakage. The validation was performed at a maximum time horizon of the MMVAE-LSTM model, which constituted 20 model steps, equal to 20,000 seconds in model time.

The model forecasting evaluation utilised the best model checkpoint inferred during training, which in the case of this experiment was the early stopped epoch model weights. Then the model performance was evaluated per-modality using the representation model as well as two baselines. The first baseline model was a persistence-based approach, which naïvely repeats the final provided vector value as the predicted model value. The second baseline was linear regression, which provided predicted samples within the window auto-regressively generating single samples based on the previous step context.

The resulting predictions were captured in sliding sequence windows across the entire test set for all predicted modalities. Then, the predictions were compared to true values corresponding to their model points by using the mean absolute error (MAE) and mean square error (MSE). The mean error values were averaged over the entire prediction horizon and the standard deviation of both errors was calculated.

The missing modality experiment was calculated without baselines, using purely the VAE model. All possible combinations of missing modalities were verified, however, only a single modality omission was analysed further for experiment clarity. The missing modalities were wholly disabled during inference via passing a drop flag to the model, triggering their corresponding branch of the encoder within the VAE to not perform any calculations, while the decoder output was calculated as normal.

Acknowledgements

The authors would like to acknowledge Prof. Adam Roman, Prof. Bogusław Szlachetko, and Dr. Urszula Libal for their continued guidance in scientific endeavours.

Funding disclosure

This research was supported by funding from the King Abdullah University of Science and Technology (KAUST).

Author contributions statement

P.K. conceived the experiment framework and the bee colony simulation, P.K. and J.P. conducted the machine learning experiments, analysed the results, and were responsible for preparation of the manuscript. All authors reviewed the final manuscript and accept its contents.

Additional information

The authors declare no competing interests.

All code related to the project is available in the Zenodo archive and Github repository[38]. The generated simulation dataset is preserved for reproducibility in a Zenodo archive[39].

References

1. Étienne Minaud *et al.* How stressors disrupt honey bee biological traits and overwintering mechanisms. *Heliyon* **10**, e34390, DOI: https://doi.org/10.1016/j.heliyon.2024.e34390 (2024).
2. Verbeke, W. *et al.* European beekeepers' interest in digital monitoring technology adoption for improved beehive management. *Comput. Electron. Agric.* **227**, 109556, DOI: https://doi.org/10.1016/j.compag.2024.109556 (2024).
3. Kulyukin, V. A., Coster, D., Kulyukin, A. V., Meikle, W. & Weiss, M. Discrete time series forecasting of hive weight, in-hive temperature, and hive entrance traffic in non-invasive monitoring of managed honey bee colonies: Part i. *Sensors (Basel, Switzerland)* **24**, DOI: 10.3390/s24196433 (2024).
4. Kanelis, D. *et al.* Decoding the behavior of a queenless colony using sound signals. *Biology* **12**, DOI: 10.3390/biology12111392 (2023).
5. Ferrari, S., Silva, M., Guarino, M. & Berckmans, D. Monitoring of swarming sounds in bee hives for early detection of the swarming period. *Comput. Electron. Agric.* **64**, 72–77, DOI: https://doi.org/10.1016/j.compag.2008.05.010 (2008). Smart Sensors in precision livestock farming.
6. Abdollahi, M. *et al.* On the prediction of varroa mite infestations in honey bee colonies via acoustic monitoring. *IEEE Sensors J.* **26**, 12157–12167, DOI: 10.1109/JSEN.2026.3666127 (2026).
7. Abdollahi, M. *et al.* Audio modulation spectral features for improved honeybee colony population prediction. *IEEE Sensors J.* **25**, 44378–44391, DOI: 10.1109/JSEN.2025.3625178 (2025).
8. Libal, U. & Biernacki, P. Non-intrusive system for honeybee recognition based on audio signals and maximum likelihood classification by autoencoder. *Sensors* **24**, DOI: 10.3390/s24165389 (2024).
9. Książek, P., Libal, U. & Król-Nowak, A. Spectral components of honey bee sound signals recorded inside and outside the beehive: An explainable machine learning approach to diurnal pattern recognition. *Sensors* **25**, DOI: 10.3390/s25144424 (2025).
10. Książek, P., Szlachetko, B. & Roman, A. Honey bee lifecycle activity prediction using non-invasive vibration monitoring. *Appl. Sci.* **16**, DOI: 10.3390/app16010188 (2026).
11. Xavier Janssens, Étienne Bruneau & Philippe Lebrun. Prévision des potentialités de production de miel à l'échelle d'un rucher au moyen d'un système d'information géographique. *Apidologie* **37**, 351–365, DOI: 10.1051/apido:2006006 (2006).
12. Campbell, T., Dixon, K. W., Dods, K., Fearns, P. & Handcock, R. Machine learning regression model for predicting honey harvests. *Agriculture* **10**, DOI: 10.3390/agriculture10040118 (2020).
13. Schmickl, T. & Crailsheim, K. Hopomo: A model of honeybee intracolonial population dynamics and resource management. *Ecol. Model.* **204**, 219–245, DOI: https://doi.org/10.1016/j.ecolmodel.2007.01.001 (2007).
14. Schmickl, T. & Crailsheim, K. Taskselsim: a model of the self-organization of the division of labour in honeybees. *Math. Comput. Model. Dyn. Syst.* **14**, 101–125, DOI: 10.1080/13873950701846662 (2008). https://doi.org/10.1080/13873950701846662.
15. Becher, M. A. *et al.* Beehave: a systems model of honeybee colony dynamics and foraging to explore multifactorial causes of colony failure. *J. Appl. Ecol.* **51**, 470–482, DOI: https://doi.org/10.1111/1365-2664.12222 (2014). https://besjournals.onlinelibrary.wiley.com/doi/pdf/10.1111/1365-2664.12222.
16. Becher, M. *et al.* Beescout: A model of bee scouting behaviour and a software tool for characterizing nectar/pollen landscapes for beehave. *Ecol. Model.* **340**, 126–133, DOI: https://doi.org/10.1016/j.ecolmodel.2016.09.013 (2016).
17. Becher, M. *et al.* Bumble-beehave: A systems model for exploring multifactorial causes of bumblebee decline at individual, colony, population and community level. *J. Appl. Ecol.* **55**, DOI: 10.1111/1365-2664.13165 (2017).
18. Betti, M., LeClair, J., Wahl, L. M. & Zamir, M. Bee++: An object-oriented, agent-based simulator for honey bee colonies. *Insects* **8**, DOI: 10.3390/insects8010031 (2017).
19. Abdollahi, M. *et al.* Urban: Urban beehive acoustics and phenotyping dataset. *Sci. Data* **12**, 536, DOI: 10.1038/s41597-025-04869-1 (2025).
20. Yasumoto, K., Yamaguchi, H. & Shigeno, H. Survey of real-time processing technologies of iot data streams. *J. Inf. Process.* **24**, 195–202 (2016).
21. Anagnostopoulos, C. *et al.* Multimodal federated learning in aiot systems: Existing solutions, applications, and challenges. *IEEE Access* **12**, 180864–180902 (2024).

22. Goknil, A. *et al.* A systematic review of data quality in cps and iot for industry 4.0. *ACM Comput. Surv.* **55**, 1–38 (2023).

23. Belay, M. A., Blakseth, S. S., Rasheed, A. & Salvo Rossi, P. Unsupervised anomaly detection for iot-based multivariate time series: Existing solutions, performance analysis and future directions. *Sensors* **23**, 2844 (2023).

24. Liu, A., Jiang, W., Huang, S. & Feng, Z. Multi-modal integrated sensing and communication in internet of things with large language models. *IEEE Internet Things Mag.* **8**, 103–111 (2025).

25. Kingma, D. P. & Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

26. Narsina, A. G., Begani, D. & Bhattacharjee, V. Exploring latent spaces for generating challenging datasets using variational autoencoders. In *2025 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)*, vol. 3, 1–5, DOI: 10.1109/IATMSI64286.2025.10985261 (2025).

27. Baião, A. R. *et al.* A technical review of multi-omics data integration methods: from classical statistical to deep generative approaches. *Briefings bioinformatics* **26**, bbaf355 (2025).

28. Lee, M. & Pavlovic, V. Private-shared disentangled multimodal vae for learning of latent representations. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 1692–1700 (IEEE, 2021).

29. Ballard, J. L., Wang, Z., Li, W., Shen, L. & Long, Q. Deep learning-based approaches for multi-omics data integration and analysis. *BioData Min.* **17**, 38 (2024).

30. Girin, L. *et al.* Dynamical variational autoencoders: A comprehensive review. *Foundations Trends Mach. Learn.* **15**, 1–175 (2022).

31. Chung, J. *et al.* A recurrent latent variable model for sequential data. *Adv. neural information processing systems* **28** (2015).

32. Lin, S. *et al.* Anomaly detection for time series using vae-lstm hybrid model. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4322–4326 (Ieee, 2020).

33. Ibrahim, M. R. *et al.* Variational-lstm autoencoder to forecast the spread of coronavirus across the globe. *PloS one* **16**, e0246120 (2021).

34. Cejrowski, T. & Szymański, J. Detection of anomalies in bee colony using transitioning state and contrastive autoencoders. *Comput. Electron. Agric.* **200**, 107207, DOI: https://doi.org/10.1016/j.compag.2022.107207 (2022).

35. Abdollahi, M., Henry, E., Giovenazzo, P. & Falk, T. H. The importance of context awareness in acoustics-based automated beehive monitoring. *Appl. Sci.* **13**, DOI: 10.3390/app13010195 (2023).

36. Woodgate, J. L., Makinson, J. C., Lim, K. S., Reynolds, A. M. & Chittka, L. Life-long radar tracking of bumblebees. *PLOS ONE* **11**, e0160333, DOI: 10.1371/journal.pone.0160333 (2016).

37. Hinton, G. E. Training products of experts by minimizing contrastive divergence. *Neural computation* **14**, 1771–1800 (2002).

38. Książek, P. & Przepióra, J. L0ranos/bee_sim_modelling_2026: v1.0.0, DOI: 10.5281/zenodo.22111657 (2026).

39. Książek, P. & Przepióra, J. Synth_bee, DOI: 10.5281/zenodo.22111489 (2026).