Anomaly Detection in Beehives: An Algorithm Comparison

Padraig Davidson(✉), Michael Steininger, Florian Lautenschlager, Anna Krause,
and Andreas Hotho

Institute of Computer Science, Chair of Computer Science X, University of Würzburg,
Am Hubland, Würzburg, Germany
{davidson,steininger,lautenschlager,anna.krause,
hotho}@informatik.uni-wuerzburg.de

Abstract. Sensor-equipped beehives allow monitoring the living conditions of bees. Machine learning models can use the data of such hives to learn behavioral patterns and find anomalous events. One type of event that is of particular interest to apiarists for economical reasons is bee swarming. Other events of interest are behavioral anomalies from illness and technical anomalies, e.g. sensor failure. Beekeepers can be supported by suitable machine learning models which can detect these events.

In this paper we compare multiple machine learning models for anomaly detection and evaluate them for their applicability in the context of beehives. Namely we employed Deep Recurrent Autoencoder, Elliptic Envelope, Isolation Forest, Local Outlier Factor and One-Class SVM. Through evaluation with real world datasets of different hives and with different sensor setups we find that the autoencoder is the best multi-purpose anomaly detector in comparison.

Keywords: Precision beekeeping · Anomaly detection · Deep learning · Autoencoder · Swarming

1 Introduction

Supporting beekeepers in their care decisions is the goal of precision apiculture. To this end, sensors are used which collect data on 1) apiary-level (i.e. meteorological parameters), 2) colony-level (i.e. beehive temperature), or 3) individual bee-related level (i.e. bee counter) [24]. For colony-level data, environmental sensors are installed in beehives in order to monitor and quantify the beehive's state continuously. Sensor values we expect most of the times are defined as normal regions of observations, while values differing considerably from this norm are called anomalies. Defining norm and anomaly is always contingent on the context of the analysis. We differentiate between behavioral anomalies, sensor anomalies and external interference. The first anomaly type is characterized by irregular behavior of the bees, the second type occurs when there are irregular measurements due to the sensors, and the last type represents anomalies induced by any external force.

An important behavioral anomaly for beekeepers is swarming, which describes a queen leaving her hive accompanied by worker bees in order to establish a new colony.

A. Ahrens et al. (Eds.): SENSORNETS 2020/2021, CCIS 1674, pp. 1–20, 2022.
https://doi.org/10.1007/978-3-031-17718-7_1

First, there is the *prime swarm* where the current queen leaves the hive with a large number of worker bees. This can be followed by multiple *after swarms* with fewer workers departing. These events can even lead to the complete depletion of a colony [23]. Beekeepers want to prevent swarming as it reduces honey production. Additionally, swarming requires immediate action to recollect the new colonies. Due to the highly stochastic nature of this reproduction process the prediction of these events is difficult.

Anomalies which are not directly related to bees can also occur. On the one hand, there are sensor anomalies which are caused by defective sensors. These require repair in order to restore a beehive's complete functionality. On the other hand, there can be anomalies due to external interference. This usually occurs through physical interaction of the beekeeper with the hive, e.g. when the hive is opened to yield honey.

For large datasets of beehive data it is infeasible to find anomalies manually. Therefore, we apply automatic anomaly detection methods. A number of machine learning algorithms have shown to provide this functionality in other domains. It is therefore interesting to assess how these methods perform in the context of beekeeping.

In this work, we evaluate multiple common anomaly detection models, namely Deep Recurrent Autoencoders, Elliptic Envelope, Isolation Forests, Local Outlier Factor and One-Class SVMs, for their applicability with beehive data. We evaluate these models on three datasets for this work: Two short term datasets, one from [25] and the other from we4bee (https://we4bee.org/), and one long term dataset from the HOBOS (https://hobos.de/) project containing four years of data. These datasets contain labelled swarming events (e.g. observed events by the apiarist) and other anomalies without labels (e.g. hidden or unobserved). The models are trained to find anomalies based on temperature readings of a beehive in an univariate setting (with one temperature sensor) and in a multivariate setting (with three temperature sensors). We use the labelled swarms to assess anomaly detection performance of our models quantitatively. Our results suggest that recurrent autoencoders provide consistently good results across the datasets for both, the univariate and the multivariate setting, compared to the other models. Elliptic Envelope's performance is inconsistent, since it showed by far the best performance when trained on one beehive but also the worst when trained on another beehive. This implies, prediction quality is strongly dependent on the training data. The other models have also shown to provide relatively good performance. Furthermore, we present other types of anomalies found through automatic anomaly detection, namely through the recurrent autoencoder, for which no labels exist and discuss the usage of anomaly detection for non-swarming anomalies.

Our contribution is twofold: First, we compare typical anomaly detection machine learning models for swarm detection in both a univariate and a multivariate sensor setting. Second, we present other types of anomalies found by the recurrent autoencoder in the beehive datasets and discuss anomaly detection for these anomalies.

In this work we present an extension of our work in [6]. This includes a broader spectrum of anomaly detectors, not solely the recurrent autoencoder. Furthermore we added a quantitative analysis of the swarm prediction quality of all detectors. The analysis was done in a univariate sensor setting, as well as a multivariate setting.

This work is structured as follows: Related research is presented in Sect. 2. Section 3 describes the datasets used in this work. The different anomaly detection models of our comparison are presented in Sect. 4. A description of our experiments can be found in Sect. 5 while their results are shown in Sect. 6. We discuss our results in Sect. 7 before concluding the work in Sect. 8.

2 Related Work

There are a number of works which encompass monitoring and detection of swarms in beehives.

Ferrari et al. [8] analyzed humidity, temperature and sound in beehives to understand how these variables change before and during swarming. To this end, they used data from three beehives where nine swarming events had occurred. The authors identified that a change in temperature and a shift in sound frequency might be useful indicators for swarming.

Kridi et al. [11] determined pre-swarming behavior through clustering temperature data. If measurements cannot be assigned to clusters of typical beehive temperature patterns for several hours, the authors consider this an anomaly.

Zacepins et al. [25] proposed an rule-based algorithm for swarming detection using data from a single temperature sensor. Their algorithm (from here on denoted as RBA) detects a swarming event if the temperature is above 35.5 °C for between two and twenty minutes. Events with shorter or longer temperature anomalies are not considered to be swarms.

Zhu et al. [27] link a linear rise in temperature to pre-swarming behavior. They recommend placing a temperature sensor between the bottom of the first frame and the beehive's wall, as this is the most suited location for measuring this increase in temperature.

While some of these works propose swarm detection methods, none of them evaluated a larger set of common machine learning approaches for anomaly detection. Popular models include One-Class SVMs [21], Local Outlier Factor (LOF) [3], Elliptic Envelope [19], Isolation Forests [13] and neural networks [20]. As for neural networks, recurrent autoencoders performed particularly well on sequential data across many anomaly detection settings [4,9,15,22]. Therefore, we evaluate these algorithms to identify which is the most promising for this task.

3 Datasets

We obtained datasets from three sources for our studies: HOBOS, we4bee, and a subset of Zacepins et al. [25] dataset. We selected two HOBOS beehives in Würzburg and Bad Schwartau and one we4bee hive in Markt Indersdorf for our experiments. Zacepins et al. data was collected in Jelgava. From here on, we refer to all datasets by the location of the beehive.

3.1 Würzburg and Bad Schwartau

HOBOS collected evironmental data from five sensor equipped beehives (species *apis mellifera*; beehive type: zander beehive) in Bournemouth, Münchsmünster, Gut Dietlhofen, Bad Schwartau and Würzburg. We selected the hives in Bad Schwartau as there are three verified swarming events. In contrast, data for the Würzburg beehive is completely unverified. We use this beehive to assess cross-beehive applicability of our models. HOBOS beehives come with different sensor configurations. Figure 1 shows the maximum sensor configuration: 13 temperature sensors, named T_1 to T_{11} mounted between the honeycombs and T_{12} and T_{13} mounted on the back and front of the hive respectively, plus weight, humidity, and carbon dioxide (CO_2) sensors. The beehives in Bad Schwartau and Würzburg are both missing some of the temperature sensors: Bad Schwartau is not equipped with T_2, T_3, T_9, T_{10} and T_{12} and Würzburg is not equipped with T_2 and T_3. HOBOS collected data from May 2016 to September 2019. During this time, sensor readings were collected once per minute for every sensor. As the typical swarming period for honey bees is May to September [7], we limit data for our preliminary study to the swarming period. HOBOS granted us access to their complete dataset.

Analysis. The Pearson correlation coefficients between different sensors, e.g. inter-sensor correlations, are visualized in Fig. 8 in the appendix. Within the normal data portion of the dataset, these correlations are strong, especially between adjacent sensors. Correlations are even higher for the sensors T_4–T_{10} placed in the center of the beehive, and go beyond directly adjacent sensors, i.e. sensors T_4 and T_{10} still correlate positively. The sensors placed at the outer margins of the apiary tend to correlate with the sensors placed outside, as well as their opposite counterpart.

During days containing anomalies, correlations are not that strong, except for neighboring sensors. This implies, that certain sensors are more sensible for swarm detection, as also stated in [27].

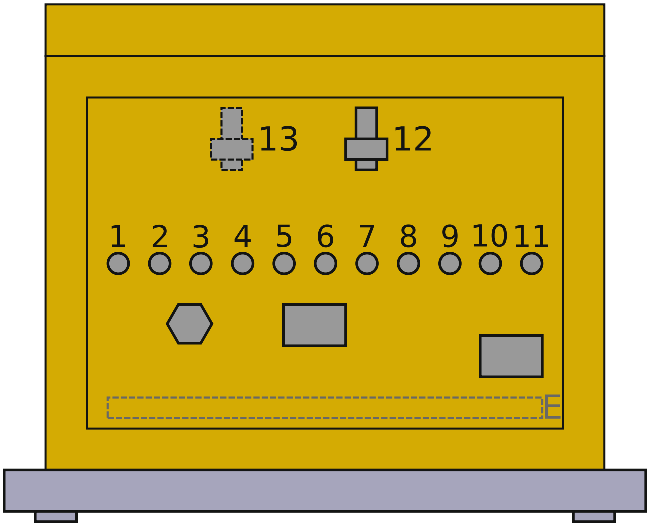


Fig. 1. Back of a HOBOS beehive. Temperature sensors T_1–T_{11} are mounted between honeycombs, temperature sensors T_{12} and T_{13} are mounted on the back and the front of the hive, respectively. E denotes the hive's entrance on the front of the beehive [6].

3.2 Jelgava

Zacepins et al. monitored ten colonies (*apis mellifera mellifera*; norwegian-type hive bodies) with a single temperature sensor placed above the hive. The observation ran from May to August in 2015 and recorded one measurement per minute. The authors recorded nine swarming events during their observation period and granted us access to the nine days in the dataset that contain these events.

3.3 Markt Indersdorf

we4bee started rolling out 100 smart top bar hives to schools and interested individuals all over Germany in 2019. In the same year, first bee colonies have been introduced into the hives. One successful hive of this first project year is the hive in Markt Indersdorf (*apis mellifera*; top bar hive). Figure 2 shows the cutaway view of a we4bee hive: it includes four temperature sensors on the inside of the hive and one on the outside. Three temperature sensors inside the hive distributed along the length of the hive, the fourth is located at the back. The inner temperature sensors are referred to as T$_l$, T$_m$, and T$_r$ for the sensors in the hive body; the sensor at the back is named T$_i$. The outside sensor is called T$_o$. we4bee hives also report other environmental quantities: air pressure, weight, fine dust, humidity, rain and wind. For Markt Indersdorf we obtained data from June (when the colony was introduced to the hive) to September 2019. All sensors except fine dust reported one measurement per second; fine dust was recorded once every three minutes.

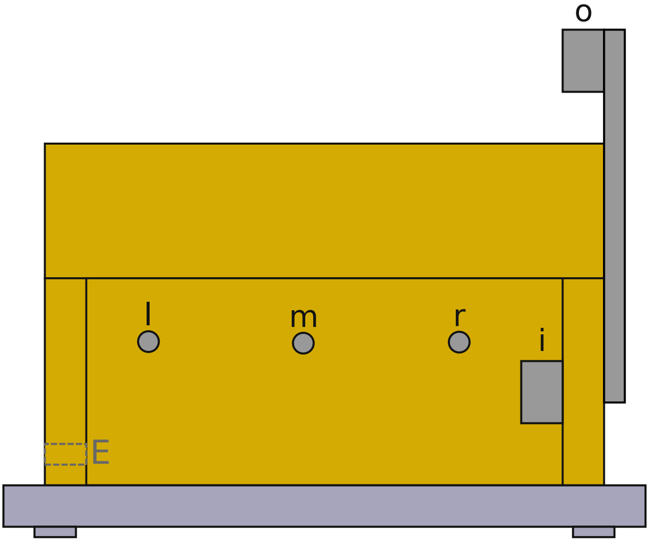


Fig. 2. Cutaway view of a we4bee beehive. T$_l$, T$_m$, T$_r$, and T$_i$ are mounted on the inside, laterally to the honeycombs. T$_o$ is placed outside at the pylon. E denotes the entrance on the front of the beehive [6].

4 Methods

4.1 (Recurrent) Autoencoder

An autoencoder (AE) consists of two neural networks, an encoder ϕ and a decoder ψ. The encoder maps the input space $\mathcal{X}$ into the feature space ($\phi : \mathcal{X} \rightarrow \mathcal{F}$). In contrast, the decoder remaps the feature space into the input space ($\psi : \mathcal{F} \rightarrow \mathcal{X}$). The task of the encoder-decoder pair is to adapt both mapping steps in a way, that decoding an encoded sequence closely resembles the input itself: $\bar{x} = \psi(\phi(x)) \sim x$. When training the AE with normal data, this kind of data is encoded very well within the feature space, whereas anomalous data cannot be reconstructed properly, incurring a high difference in prediction and input.

This difference is quantified by a loss function $\mathcal{L}$, which is often the l_2 norm [26] or the MSE (mean squared error) [22]. The optimization task can be stated as follows:

$$\phi, \psi = \text{arg min}_{\phi,\psi} \mathcal{L}(x, \psi(\phi(x))).$$

An anomaly is any input with a resulting loss that is greater than α, which is the anomaly threshold: $\mathcal{L}(x, \bar{x}) \geq \alpha$. This hyperparameter can either be set manually or determined with a labelled anomaly set in a second training step. Optimally, α is set high enough to detect all anomalies, but not too low to be overly sensible within predictions [16].

4.2 Local Outlier Factor

The local outlier factor [3] estimates the degree or probability of an instance being an anomaly, rather than performing a binary classification. The algorithm is based on the idea of density based clustering, which generally requires two parameters: a minimum number of objects k and a volume value. Together, these parameters define a local density. Regions with densities higher than the density threshold form clusters and are separated by regions with densities below the density threshold.

As an extension of this idea, the local outlier factor algorithm only relies on one parameter, the minimum number of neighbors k. Densities are calculated by using the k-heighborhood. The local reachability density is the average reachability distance of a point to its k neighborhood. The reachability distance of two points is the maximum of the k-distance or the distance between the two points: $rd_k(A, B) := \max\{k - \text{distance}(B), d(A, B)\}$ [3]. Finally, the local outlier factor of a point is calculated as the average local reachability density of the k neighborhood divided by the local reachability of a given point.

A local outlier factor of $\leq$1 indicates an (cluster) inlier, whereas values $>$1 indicate outliers.

4.3 Isolation Forest

While most anomaly detectors build internal profiles of normal data and report anomalies that do not fit in these profiles, the isolation forest is based on the idea of isolating anomalies [13, 14]. An isolation forest consists of several isolation trees, or iTrees. An iTree is a binary search tree, which consists of external nodes (e.g. nodes without children) and internal nodes (e.g. nodes with exactly two children). Internal nodes split the data by a random attribute q and a corresponding split value p. If a data point fullfills the "test" function $q < p$ the path to the first child is followed, otherwise the second path is pursued. This structure is repeated until all data points x in the dataset are isolated in an external node.

Since anomalies are isolated more easily, the path length $h(x)$ for an anomalous point is shorter than for normal data. An anomaly score can be computed as $s(x,m) = \mathrm{pow}(2, -E(h(x))/c(m)$, where $E(h(x))$ is the average $h(x)$ from all iTrees and $c(m)$ represents the average path length given the sample size m. The anomaly score indicates an anomaly if it is close to 1 and normal data for values close to or below 0.5.

4.4 Elliptic Envelope

Elliptic envelope is an anomaly detection algorithm that is based on the minimum covariance determinant (MCD) estimator and assumes the data to be sampled from an elliptically symmetric unimodal distribution. MCD is a highly robust estimator of multivariate location and scatter [18].

The method subsamples the data $\mathbf{X}$ in $\mathbf{H}_1$ and computes an estimate of the location $\mathbf{T}_1$ and the covariance of each sample $\mathbf{S}_1$. A new subsample $\mathbf{H}_2$ is built with the $h = (n+p+1)/2$ samples with the lowest robust distance [19], where n is the number of samples in $\mathbf{H}_1$ and p the number of features. This subsampling process is repeated until the determinant of the covariance converges within a given tolerance. Elliptic envelope finally flags every sample as outlier that has a robust distance above a cutoff value $\sqrt{\chi^2_{p,0.975}}$ [19].

4.5 One-Class SVM

The one-class support vector machine (SVM) [12], is an extension of the standard support vector machine [5] for unsupervised outlier detection. The SVM algorithm is normally applied to supervised two class classification problems. Input is classified by finding a hyperplane with maximum distance to the closest instances of the two classes. Data points of the same class, in our setting normal data or anomalous data, are grouped on the same side of this plane. To account for datapoints still being non-seperable, a penalty parameter is introduced. The One-Class SVM reuses the SVM algorithm by setting all class labels to the same class. This means, the separating hyperplane is an envelope around the normal data, with a maximum distance towards all anomalies.

5 Experimental Setup

All models described in Sect. 4 are evaluated on the datasets outlined in Sect. 3.

Data Splitting. We used the HOBOS hives for training and validation purposes. That is, we trained on Bad Schwartau and Würzburg in independent settings using the reported and found [6] swarming events. Explicitly, we built two setups: one with training the models on the normal behavior of Bad Schwartau, using its anomalous behavior as a validation set for the parameter search, and one with the training step consisting of the normal behavior of Würzburg, while validating on its anomalous behavior set. The datasets from Jelgava and Markt Indersdorf, as well as the test set from the untrained hive were only used for evaluation. All models were provided with the same splits of input data to ensure comparability.

As customary in novelty detection (i.e. AE, Isolation Forest), the training data shouldn't be polluted by outliers. For that, we visually examined the datastream and marked each day as normal behavior, or as an outlier, e.g. an anomaly. We defined normal behavior as any temperature sensor trace remaining nearly constant at 34.5 °C as the core temperature [8,25]. Any larger deviation form this norm temperature were considered as anomalous days. Figure 3 shows sensor data to be expected from a normal day. Training and validation parts consist exclusively of normal data, while test and holdout sets combine normal and anomalous data. Test sets are any portions of the dataset with anomalous data, that don't originate from the beehive used for training. The holdout set, which contains the anomalous behavior from the training hive, is used for the parameter search of the estimators.

An exemplary view of the data splitting procedure for the Bad Schwartau hive can be seen in Fig. 4. Keep in mind, that the test and holdout sets also contain slices of normal data and are not necessarily only windows with anomalies.

Input Data. In any univariate sensor setting, we use central temperature sensors. For the hives in Würzburg and Bad Schwartau, those are $T_6 – T_8$, from which we evaluate data on T_6 and T_8. In Markt Indersdorf this is T_m, which we additionally downsampled to one minute resolution to be consistent with the other datasets. For Jelgava the single temperature sensor at the top is used. In the multivariate sensor setting, we used senors T_6, T_7, T_8 for the hives Bad Schwartau and Würzburg, whereas we used T_l, T_m, T_r in Markt Indersdorf.

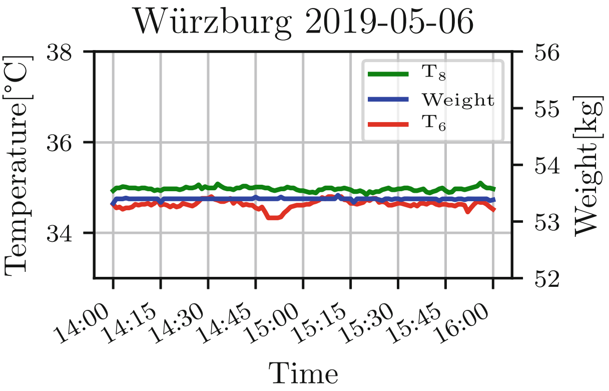


Fig. 3. Normal behavior of all three sensors [6].

Any model is given a 60 min window of sensor data, which corresponds to 60 consecutive input values of temperature data per sensor. According to [8,25,27] swarming events last from 20 min to 60 min in duration.

In the multivariate sensor setting, the AE is provided with sensor data of 60×3, whereas the other models are given $60 \cdot 3$ values, i.e. concatenating the three sensors.

Input data for the AE was normalized via standard scaling (e.g. their z-score). The non-autoencoder models were provided with the raw and unscaled sensor values, since scaling impaired their predictions.

Model Training. The optimal parameter settings for the models were found employing a random search [2]. A table with all parameters that were optimized is given in the appendix (see Table 4). Parameters were optimized using the normal data of a beehive, while using the anomalous behavior of that hive as the validation set for this search. The F_1 score of predicting swarms was used as the metric to be maximized.

For the models Local Outlier Factor, Elliptic Envelope, Isolation Forest and One-Class SVM we relied on the implementations in [17]. In the same setting we searched for the remaining parameter α for the pre-trained AE, which we could not do in [6] due to missing labels. Within this (second) parameter setting step for the swarm detection, we used a windowing technique, shifting the window by 15 min forward in time to extract the next window.

Pre-training of the AE was done in a preliminary random search (see appendix), finding the best hyperparameters for the reconstruction task itself. The *Adam* optimizer [10] was used with the default parameters ($lr = 10^{-3}$) and the mean squared error (MSE) as the loss function. Early stopping with five epoch patience was employed to prevent overfitting. For pre-training, we used all suitable measurement windows by shifting the window one time step further.

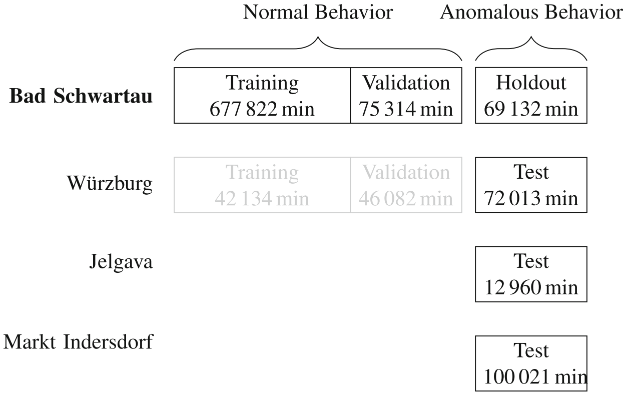


Fig. 4. The data splits used for Bad Schwartau. The autoencoder is trained on Bad Schwartau's 'Training'. The hyperparameters and α are tuned using its 'Validation' and 'Holdout', respectively. The model is then tested on all 'Test'. For Würzburg, the splits are set accordingly using its 'Training', 'Validation', and 'Test' as 'Holdout'. We provide the recording time for all splits [6].

Predictions. RBA [25] is utilized on all anomalous behavior subsets to predict swarming events. We additionally used it on the normal behavior portions of the dataset to ensure no swarming events in the training steps of both training hives.

Predictions with all other models were made by using the best model configuration found in the random search, while predicting events within the tests sets, e.g. the anomalous sets. For instance, a model was trained on Bad Schwartau, using its hold-out set for the grid search, while predicting the anomaly sets of Würzburg, Jelgava and Markt Indersdorf.

Evaluation. To evaluate the various classifiers, we used standard classification metrics. *True positives* (TP) contain all time series correctly classified as swarms, whereas *true negatives* (TN) represent all time series correctly labelled as non-swarms. The other two units quantify miss-classifications. *false positives* (FP) are any non-swarms classified as swarms, and *false negative* (FN) any swarms categorized as non-swarms.

With these quantifiers, we can calculate performance measures of the classifiers:

$$P := \frac{TP}{TP + FP} \quad R := \frac{TP}{TP + FN} \quad F_1 := \frac{2 \cdot P \cdot R}{P + R}$$

where P represents the precision, R the recall and F_1 the F_1-measure.

6 Results

6.1 Univariate

Table 1 lists the classification metrics for swarming events in the univariate sensor setting for temperature sensor T_8 on the hives Würzburg and Bad Schwartau respectively. The left hand side shows classification metrics using Bad Schwartau as the training hive, the right hand side shows this for Würzburg. The best results in the category precision and F_1 are highlighted in bold, except for RBA. For full disclosure, Markt Indersdorf is listed in this table, too, but since there are no true positives for swarms, the metrics are degrade and therefore it is not taken into account for calculations.

Discussion. As already mentioned in Sect. 5, we only optimized the parameters regarding the F_1 score for predicting swarming events. This has direct implications on the displayed metrics of Tables 1 and 2, since any true anomaly that is not a swarm is reported as a false positive. As we only have labels for swarming events, these tables are meant to show the differences in predictions when automatically optimizing models with very sparse (Würzburg: 8, Bad Schwartau: 24 swarming windows) events and no specialization.

When comparing Table 1 predictions from both different training hives, there are a few remarkable differences: Overall, the classification results are better when training on Bad Schwartau (F_1 : $[.09, .12]$) in contrast to training on Würzburg (F_1 : $[.03, .12]$). The main reason for that is the very high inclination of the classifiers towards never predicting a swarming event in Würzburg. Some anomaly detectors even report no true positive for swarming events (Elliptic Envelope, One-Class SVM). Even the metrics on the Jelgava test set decline significantly (Bad Schwartau: F_1 : $[.48, .69]$, Würzburg:

Table 1. Overview of classification metrics and results. Results are only calculated by the true positives of swarms! The estimators are trained on Bad Schwartau on the left hand side and Würzburg on the right hand side (separated by vertical double lines), both with sensor T_8. Precision (P), Recall (R), and F_1 are reported, and set to 0 for no correct classification and F_1 set as NA. Corresponding true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN) values are also reported. The overall metrics are calculated from the weighted scores from each hive.

Classifier	Hive	P[%]	R[%]	F_1[%]	TP	FP	TN	FN	Hive	P[%]	R[%]	F_1[%]	TP	FP	TN	FN
Local Outlier Factor	Jel.	0.32	1.00	0.48	36	78	723	0	Jel.	0.07	1.00	0.13	36	484	317	0
	Wü.	0.01	0.63	0.02	5	485	4268	3	B. S.	0.01	1.00	0.01	24	3925	716	0
	All	0.06	0.68	0.09	41	563	4991	3	All	0.02	1.00	0.03	60	4409	1033	0
	M. I.	0.00	0.00	NA	0	1532	5185	0	M. I.	0.00	0.00	NA	0	303	6414	0
Elliptic Envelope	Jel.	0.5	0.97	0.66	35	35	766	1	Jel.	0.17	1.00	0.29	36	174	627	0
	Wü.	0.00	0.00	NA	0	51	4702	8	B. S.	0.01	0.88	0.01	21	3772	869	3
	All	0.07	0.15	0.10	35	86	5468	9	All	0.03	0.89	0.05	57	3946	1496	3
	M. I.	0.00	0.00	NA	0	58	6659	0	M. I.	0.00	0.00	NA	0	17	6700	0
Isolation Forest	Jel.	0.33	0.75	0.45	27	56	745	9	Jel.	0.21	0.89	0.34	32	121	680	4
	Wü.	0.00	0.50	0.00	4	2443	2310	4	B. S.	0.00	0.50	0.00	4	2443	2310	4
	All	0.05	0.54	0.07	31	2499	3055	13	All	0.03	0.56	0.05	36	2564	2990	8
	M. I.	0.00	0.00	NA	0	5226	1491	0	M. I.	0.00	0.00	NA	0	4056	2661	0
One-Class SVM	Jel.	0.59	0.83	0.69	30	21	780	6	Jel.	0.30	0.89	0.45	32	75	726	4
	Wü.	0.00	0.00	NA	0	298	4455	8	B. S.	0.01	0.88	0.01	21	2255	2386	3
	All	**0.09**	0.12	0.10	30	319	5235	14	All	0.05	0.88	0.08	53	2330	3112	7
	M. I.	0.00	0.00	NA	0	1363	5354	0	M. I.	0.0	0.0	NA	0	204	6513	0
AE	Jel.	0.57	1.0	0.73	37	28	772	0	Jel.	0.50	1.00	0.67	36	36	765	0
	Wü.	0.01	0.50	0.02	4	506	4247	4	B. S.	0.01	0.88	0.02	21	2329	2312	3
	All	**0.09**	0.57	**0.12**	40	535	5019	4	All	**0.08**	0.89	**0.12**	57	2365	3077	3
	M. I.	0.00	0.00	NA	0	251	6466	0	M. I.	0.00	0.00	NA	0	1934	4783	0
RBA	Jel.	1.00	0.50	0.67	18	0	801	18	Jel.	1.00	0.50	0.67	18	0	801	18
	Wü.	0.07	0.25	0.11	2	27	4726	6	B. S.	0.57	0.33	0.42	8	6	4635	16
	All	0.21	0.29	0.19	20	27	5527	24	All	0.64	0.36	0.46	26	6	5436	34
	M. I.	0.00	0.00	NA	0	4	6713	0	M. I.	0.00	0.00	NA	0	4	6713	0

$F_1 : [.13, .45]$) for all detectors except the AE (Bad Schwartau: $F_1 : .73$, Würzburg: $F_1 : .69$).

In both cases, the AE is the best swarm detector within the machine learning algorithms (highlighted in bold). It also seems to be more robust regarding the origin of data, since the F_1 score (0.12) is the same for both training scenarios.

RBA is the best swarm detector regarding the metrics. It does however miss more swarming events (4 vs. 24), some due to the windowing technique used, since in relies on the base temperature 30 min pre-swarming. The major contributing factor for the better metrics performance is the very low false positive rate. This is to be expected, since it is only built for swarm detection and isn't drawn away towards other anomalies and thus inherently has a lower false positive rate. For example, any temperature deviation below 34.5 °C is completely ignored, but may in fact be an anomaly.

6.2 Multivariate

Table 2 lists the classification metrics in the multivariate sensor setting for temperature sensors T_6, T_7, T_8 for the hives Bad Schwartau and Würzburg. Calculation of the metrics is done in the same manner as in the univariate setting. The table lists only the multivariate datasets. The classification metrics of Würzburg are reported when training on Bad Schwartau and vice versa. Furthermore the lines with Markt Indersdorf$_X$ show the reports when training on hive X and predicting Markt Indersdorf.

Table 2. Overview of classification metrics and results in the multivariate setting. Results are only calculated by the true positives of swarms! The estimators are trained on Bad Schwartau predicting Würzburg (and vice versa) and sensor T_6, T_7, T_8. Precision (P), Recall (R), and F_1 are reported, and set to 0 for no correct classification and F_1 set as NA. Corresponding true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN) values are also reported. The overall metrics are calculated from the sum of the number of classifications.

Classifier	Beehive	P[%]	R[%]	F_1[%]	TP	FP	TN	FN
Local Outlier Factor	Würzburg	0	0	NA	0	55	4698	8
	Bad Schwartau	0.005	0.875	0.010	21	4086	222	3
	Overall	**0.005**	0.656	0.010	21	4141	4920	11
	Markt Indersdorf$_W$	0	0	NA	0	5132	1585	0
	Markt Indersdorf$_S$	0	0	NA	0	1055	6552	0
Elliptic Envelope	Würzburg	0	0	NA	0	177	4576	8
	Bad Schwartau	0.006	0.875	0.011	21	4063	578	3
	Overall	**0.005**	0.656	0.010	21	4240	5154	11
	Markt Indersdorf$_W$	0	0	NA	0	3955	3762	0
	Markt Indersdorf$_S$	0	0	NA	0	2842	3875	0
Isolation Forest	Würzburg	0.001	0.500	0.002	4	3206	1547	4
	Bad Schwartau	0.005	0.875	0.011	21	3841	800	3
	Overall	0.004	0.781	0.007	25	7047	2347	7
	Markt Indersdorf$_W$	0	0	NA	0	6694	23	0
	Markt Indersdorf$_S$	0	0	NA	0	6387	330	0
One-Class SVM	Würzburg	0.001	0.500	0.002	4	3206	1547	4
	Bad Schwartau	0.005	0.875	0.011	21	3841	800	3
	Overall	0.004	0.781	0.007	25	7047	2347	7
	Markt Indersdorf$_W$	0	0	NA	0	6694	23	0
	Markt Indersdorf$_S$	0	0	NA	0	6387	330	0
AE	Würzburg	0.001	0.125	0.002	1	1292	3461	7
	Bad Schwartau	0.007	0.875	0.015	21	2841	1800	3
	Overall	**0.005**	0.688	**0.011**	22	4133	5261	10
	Markt Indersdorf$_W$	0	0	NA	0	6683	43	0
	Markt Indersdorf$_S$	0	0	NA	0	5534	1183	0

Discussion. In the multivariate setting, the AE is also the best option for detecting anomalies. Still Table 2 shows, that all metrics drop in contrast to the univariate, single temperature sensor setting. The reason for that is the much higher false positive rate, which means, that more non-swarms are confused with swarms. This means the additional measurements introduce more noise as would be necessary for predicting swarms. As shown in Fig. 8, adjacent sensors correlate strongly within the normal data, thus they bear no additional information during training, but weaker so within the anomaly set. Only including new sources of information (like the scale) would help in the multivariate sensor setting, as shown in Figs. 6a and 6b.

6.3 Methodology

In Sect. 3 we described our empirically founded, but manual approach of splitting data into anomalous behavior and normal sensor data. However, this data splitting method is ambiguous and highly susceptible to missing days in the corresponding dataset, i.e. missing anomalies and therefore mislabeling specific days. A general, rule-based approach of splitting anomalous and normal data, i.e. all windows with sensor values drifting for more than two standard deviations, doesn't work, since it removes most swarming events from the test set. A clearer split of training and testing data can only be ensured by very thorough labeling of the sensor values, which has to be done on different sensors independently.

In this work we evaluated predictions in an automated manner by using a random search for the best parameter settings (Table 4) using only labelled information of swarming events. In previous work [6] we selected the parameter α for the AE for detecting anomalies manually. This is a first step towards the automation of the anomaly detectors, but still has the problem of only being optimized for one anomaly class and still results in false positives for swarming events, but true positives for other anomalies.

Summarizing the results, the AE is the best all purpose swarm detector within the machine learning algorithms. It is out-performed to RBA for swarming detection, but it is also capable of predicting other anomalies without the knowledge of special rules.

7 Analysis

In this section we analyze the found anomalies and will outline different types of anomalies reported by the AE. We used this model exemplary to show interesting observations from the predictions, not only focussing on swarms, but also the aforementioned false positives, as well as true positives for other anomalies, hidden from the above discussion.

Swarming Events. All swarming events predicted with temperature sensors T_6 and T_m by the AE and RBA can be found in Table 3. Events observed by apiarists on site are marked with $\star$. This table lists all swarm like events detected by RBA, as well as additionally missed swarming events. In other words, we used RBA to verify the results of the AE and vice versa, as described in [6]. Figure 5 shows a sensor data plot for a prototypical swarm for the hive in Bad Schwartau. Swarming events can be found within the table as *location (S)*, whereas other anomalies are denoted with *location (O)*. A more detailed view of the findings regarding swarming events is given in [6].

Table 3. Detected Anomalies. The first column shows the shortened name of the used test (anomaly) set (B. S. is Bad Schwartau, Wü. is Würzburg, Jel. is Jelgava, M. I. is Markt Indersdorf). (S) signifies that the set contains swarms while (O) stands for other anomalies. The next column displays the date of the event, and—where suitable—a reference to figures in the text. The last two columns indicate whether RBA or our method (AE) detected the anomaly. Predictions on HOBOS-hives are based on sensor T_6, on T_m for we4bee. We used the Bad Schwartau trained model to predict the swarms in any other beehive, except for Bad Schwartau itself [6].

Dataset	Timestamp	Detected	
		RBA	AE
B. S. (S)	2016-05-11 11:05[5]	✓	✓
	2016-05-22 07:30	✓	✓
	2017-06-06 15:02	✓	✓
	2019-05-13 09:30*	✓	✓
	2019-05-21 09:15*	✓	✓
	2019-05-25 12:00*	✓	✓
B. S. (O)	2016-08-03 17:24	✓	✓
Wü. (S)	2019-05-01 09:15[6a]		✓
	2019-05-10 11:15[6b]	✓	✓
Wü. (O)	2019-04-17 16:22[6c]	✓	✓
Jel. (S)	2015-05-06 18:02*	✓	✓
	2016-06-02 13:48*	✓	✓
	2016-05-30 10:03*	✓	✓
	2016-06-16 15:50*	✓	✓
	2016-06-01 13:20*	✓	✓
	2016-06-03 09:11*	✓	✓
	2016-06-13 03:30	✓	✓
	2016-06-16 10:52*	✓	✓
	2016-06-13 13:32*	✓	✓
M. I. (O)	2019-07-26 08:10	✓	✓
	2019-08-31 17:08[7b]	✓	

Other Anomalies. Figure 6 depicts anomalies easily confused with swarms in at least one sensor. Figure 7 on the other hand, show anomalies categorized as external interference. They all display the same sensors, two temperature sensors (HOBOS: T_6, T_8; we4bee: T_r, T_m) and the weight on the scale. An exemplary plot of a training sample can be seen in Fig. 3. Sensors in Fig. 5 show the expected behavior for a swarming event, as already stated in Sect. 5.

Detecting swarms only in traces of temperature data, also has its drawbacks, as Fig. 6c shows, since the values of the weight sensors tend to describe normal behavior, whereas the temperature sensors follow the expected inverted parabola.

Similar implications can be seen in Fig. 6b, as a slice of the window actually contains a swarm, shown by all three sensors, whereas a later slice only indicates a swarm temperature-wise.

Figure 6a shows a swarming event, which RBA only detects in T_8, but not T_6, since it is not covered by the defined rules for swarms. The AE on the other hand is capable of detecting this swarm in both temperature sensors.

Figure 7a depicts the sensor traces of an opened apiary, which becomes obvious in the fast and strong drop in weight, and with varying delay in time, in the temperature sensors. This is due to the influx of ambient air, cooling the temperature within the beehive. As soon as the hive is closed the expected values, the same as before opening, are reported again.

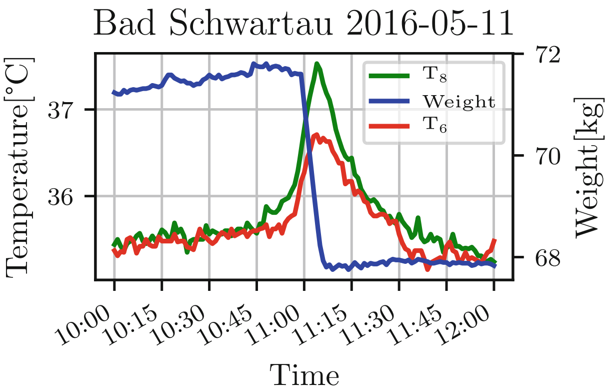
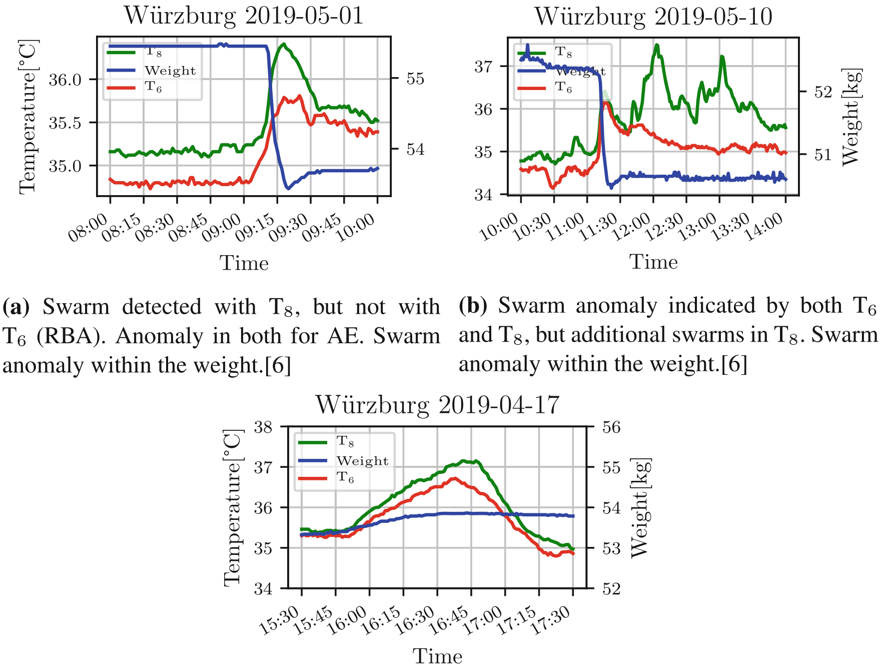


Fig. 5. (Prototypical) Swarm as indicated by T_6 and T_8, detected by RBA and AE [6].



(a) Swarm detected with T_8, but not with T_6 (RBA). Anomaly in both for AE. Swarm anomaly within the weight.[6]

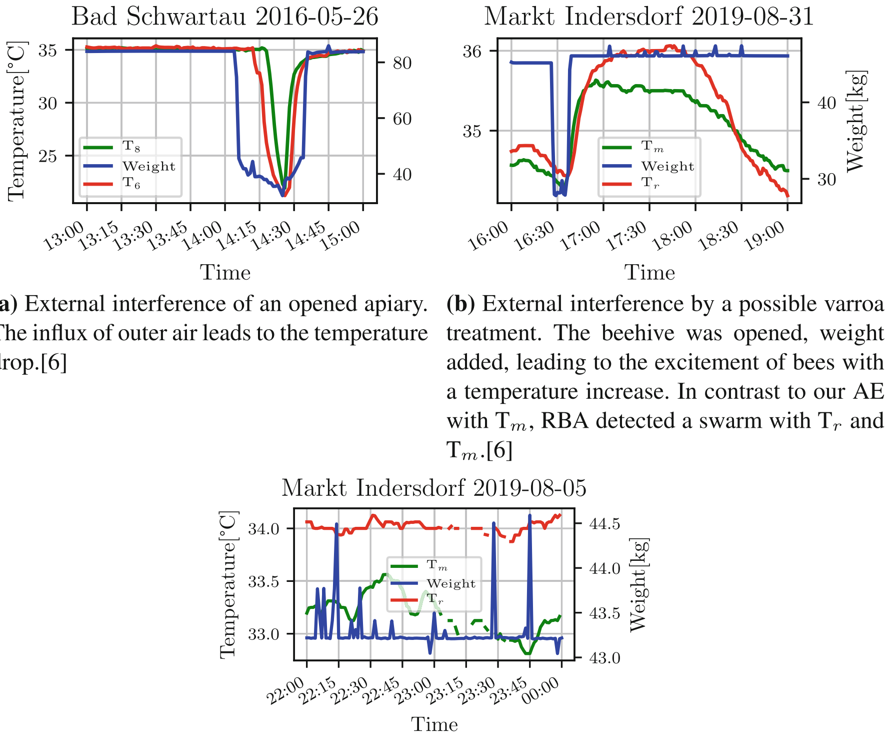
(b) Swarm anomaly indicated by both T_6 and T_8, but additional swarms in T_8. Swarm anomaly within the weight.[6]

(c) Swarm-like anomaly in sensors T_6 and T_8, but not within the measured weight.[6]

Fig. 6. Special cases of swarming events. (a) shows a swarm only detected with one temperature sensor, but not the other (RBA). (b) shows a swarming event followed by subsequent swam-like temperature curves in T_8. (c) shows a swarm-like anomaly in the temperature sensors, but not in the scale.

The beehive must sometimes be opened for treatment purposes. An example of a varroa treatment with a substance (i.e. formic acid) is displayed in Fig. 7b. The resulting additional weight after closing the hive in visible in the weight sensor. RBA confuses this as a swarm in both temperature sensors, whereas the AE only reports a swarm for T_r. T_m only fluctuates within one standard deviation of training data, which can be captured by the feature space of the AE.

Aforementioned anomalies are only a subset of reported anomalies, since the AE detects a lot more. Some of them are not as easily classified, but normally are temperature values far lower than 30 °C. Even sensor anomalies are detected by AE, as can be seen in Fig. 7c.



(a) External interference of an opened apiary. The influx of outer air leads to the temperature drop.[6]

(b) External interference by a possible varroa treatment. The beehive was opened, weight added, leading to the excitement of bees with a temperature increase. In contrast to our AE with T_m, RBA detected a swarm with T_r and T_m.[6]

(c) Sensor anomaly with missing values in T_r and T_m, but not in the measured weights.[6]

Fig. 7. External interference anomalies. (a) shows an opened hive with no modifications, whereas (b) is opened for treatment with a substance added. (c) shows missing sensor values.

8 Conclusion/Future Work

In this work we evaluated the use of machine learning models for anomaly detection in beehives. We compared the models Elliptic Envelope, Isolation Forests, Local Outlier Factor, One-Class SVMs, and recurrent autoencoders quantitatively for swarm detection. The results show that the AE is the best multi-purpose anomaly detector in comparison. It is able to detect swarms with high accuracy even by only optimizing the decision threshold with very sparse swarm instances. Within the multivariate temperature sensor setting we found, that combining three sensors incurs more noise than information, and still needs further experiments and evaluation. Especially combining different sensor types, i.e. temperature and weight, seems to be more promising. Multiple aspects of anomaly detection in beehives require more work in the future:

Evaluation of Deep Generative Models. Other types of deep neural networks will have to be explored in future work. For example, generative models like variational autoencoders or generative adversarial networks show particular promise, since they have two advantages: A) anomalies may exist within the training set, and B) they allow for probability-based classification instead of relying solely on the reconstruction error [1].

Dataset Generation. Machine learning models require data to correctly learn their task. The amount of beehive data available is limited, especially when considering data with labeled anomalies like swarming. To this end, we hope to improve data availability in the project we4bee, where sensor-equipped apiaries are distributed mostly across Germany, allowing us to collect a large dataset of beehive data. Any events or anomalies can be marked by apiarists participating in we4bee, providing us with more valuable labeled data. Predictive alert-systems can be implemented to warn beekeepers in case of anomalies. The beekeepers may provide feedback for the warnings, which allows further improvements in prediction quality.

Winter Period. During winter, bees enter a passive state where their behavior changes significantly in comparison to summer time [24]. To learn normal behavior of bees for their active summer time, we excluded data from October through March for all datasets (cf. Sect. 3). Detecting anomalies during winter can also be of interest but this remains future work.

Acknowledgements. This research was conducted in the we4bee project sponsored by the Audi Environmental Foundation.

Appendix

Sensor Correlations

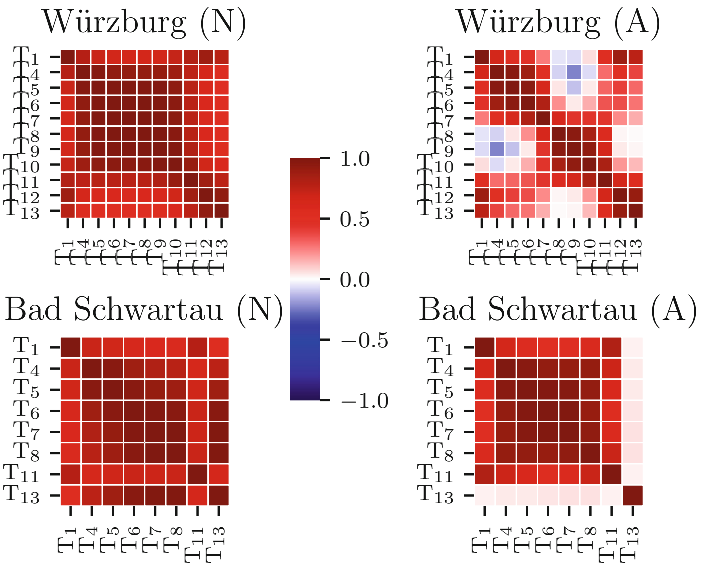


Fig. 8. Sensor correlations. All figures display the Pearson correlation between temperature sensors within a given beehive. (N) stands for the dataset containing normal behavior and (A) for the dataset with anomalous behavior [6].

Hyperparameters

Table 4. Optimized parameters and their ranges for the anomaly detectors within the random search. $\mathcal{U}_x$ describes an uniform distribution with $[0, x)$, whereas $\mathcal{I}_{a,b}$ represents a random integer distribution with $[a, b]$. $\mathcal{LU}_{a,b}$ is a log uniform distribution with parameters a, b.

Classifier	Hyperparameter	Range
Local Outlier Factor	*n neighbors*	$\mathcal{I}_{1,100}$
	algorithm	ball tree, kd tree
	leaf size	$\mathcal{I}_{1,150}$
	contamination	$\mathcal{U}_{0.5}$
	metric	chebyshev, cityblock, euclidean, infinity, l1, l2, manhattan, minkowski
Elliptic Envelope	*assume centered*	True, False
	support fraction	$\mathcal{U}_1$
	contamination	$\mathcal{U}_{0.5}$
Isolation Forest	*n estimators*	$\mathcal{I}_{10,100}$
	max samples	auto
	contamination	$\mathcal{U}_{0.5}$
	max features	$\mathcal{U}_1$
	bootstrap	True, False
One- Class SVM	*kernel*	linear, poly(degree=3), rbf(coef0=0), sigmoid
	shrinking	True, False
	γ	$\mathcal{LU}_{0.0001,1}$
	ν	$\mathcal{LU}_{0.0001,1}$
AE	*hidden size*	$\mathcal{I}_{2,64}$
	layers	$\mathcal{I}_{1,4}$

References

1. An, J., Cho, S.: Variational autoencoder based anomaly detection using reconstruction probability. Special Lecture IE **2**(1) (2015)
2. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. J. Mach. Learn. Res. **13**, 281–305 (2012)
3. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: LOF: identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, pp. 93–104 (2000)
4. Chalapathy, R., Chawla, S.: Deep learning for anomaly detection: a survey. CoRR abs/1901.03407 (2019). http://arxiv.org/abs/1901.03407
5. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995). https://doi.org/10.1007/BF00994018
6. Davidson, P., Steininger, M., Lautenschlager, F., Kobs, K., Krause, A., Hotho, A.: Anomaly detection in beehives using deep recurrent autoencoders. In: Proceedings of the 9th International Conference on Sensor Networks (SENSORNETS 2020), pp. 142–149. No. 9, SCITEPRESS - Science and Technology Publications, LDA (2020)
7. Fell, R., et al.: The seasonal cycle of swarming in honeybees. J. Apic. Res. **16**(4), 170–173 (1977)

8. 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**(1), 72–77 (2008)
9. Filonov, P., Lavrentyev, A., Vorontsov, A.: Multivariate industrial time series with cyber-attack simulation: fault detection using an LSTM-based predictive data model. In: NIPS Time Series Workshop 2016 (2016)
10. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
11. Kridi, D.S., Carvalho, C.G.N.d., Gomes, D.G.: A predictive algorithm for mitigate swarming bees through proactive monitoring via wireless sensor networks. In: Proceedings of the 11th ACM Symposium on PE-WASUN, pp. 41–47. ACM (2014)
12. Li, K.L., Huang, H.K., Tian, S.F., Xu, W.: Improving one-class SVM for anomaly detection. In: Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 03EX693), vol. 5, pp. 3077–3081. IEEE (2003)
13. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: 2008 Eighth IEEE International Conference on Data Mining, pp. 413–422. IEEE (2008)
14. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation-based anomaly detection. ACM Trans. Knowl. Discov. Data **6**(1), 1–39 (2012)
15. Malhotra, P., et al.: Multi-sensor prognostics using an unsupervised health index based on LSTM encoder-decoder. In: 1st SIGKDD Workshop on ML for PHM, August 2016
16. Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., Shroff, G.: LSTM-based encoder-decoder for multi-sensor anomaly detection. arXiv preprint arXiv:1607.00148 (2016)
17. Pedregosa, F., et al.: Scikit-learn: machine learning in python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
18. Rousseeuw, P.J.: Least median of squares regression. J. Am. Stat. Assoc. **79**(388), 871–880 (1984)
19. Rousseeuw, P.J., Driessen, K.V.: A fast algorithm for the minimum covariance determinant estimator. Technometrics **41**(3), 212–223 (1999)
20. Ryan, J., Lin, M.J., Miikkulainen, R.: Intrusion detection with neural networks. In: Advances in Neural Information Processing Systems, pp. 943–949 (1998)
21. Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. Neural Comput. **13**(7), 1443–1471 (2001)
22. Shipmon, D.T., Gurevitch, J.M., Piselli, P.M., Edwards, S.T.: Time series anomaly detection; detection of anomalous drops with limited features and sparse examples in noisy highly periodic data. arXiv preprint arXiv:1708.03665 (2017)
23. Winston, M.: Swarming, afterswarming, and reproductive rate of unmanaged honeybee colonies (apis mellifera). Insectes Soc. **27**(4), 391–398 (1980)
24. Zacepins, A., Brusbardis, V., Meitalovs, J., Stalidzans, E.: Challenges in the development of precision beekeeping. Biosys. Eng. **130**, 60–71 (2015)
25. Zacepins, A., Kviesis, A., Stalidzans, E., Liepniece, M., Meitalovs, J.: Remote detection of the swarming of honey bee colonies by single-point temperature monitoring. Biosys. Eng. **148**, 76–80 (2016)
26. Zhou, C., Paffenroth, R.C.: Anomaly detection with robust deep autoencoders. In: Proceedings of the 23rd ACM SIGKDD, pp. 665–674. ACM (2017)
27. Zhu, X., Wen, X., Zhou, S., Xu, X., Zhou, L., Zhou, B.: The temperature increase at one position in the colony can predict honey bee swarming (apis cerana). J. Apic. Res. **58**(4), 489–491 (2019)