Towards sustainable dairy management - a machine learning enhanced method for estrus detection

Kevin Fauvel
Univ Rennes, Inria, CNRS, IRISA
kevin.fauvel@inria.fr

Véronique Masson
Univ Rennes, Inria, CNRS, IRISA
veronique.masson@irisa.fr

Élisa Fromont
Univ Rennes, Inria, CNRS, IRISA
elisa.fromont@irisa.fr

Philippe Faverdin
PEGASE, INRA, AGROCAMPUS OUEST
philippe.faverdin@inra.fr

Alexandre Termier
Univ Rennes, Inria, CNRS, IRISA
alexandre.termier@irisa.fr

ABSTRACT

Our research tackles the challenge of milk production resource use efficiency in dairy farms with machine learning methods. Reproduction is a key factor for dairy farm performance since cows milk production begin with the birth of a calf. Therefore, detecting estrus, the only period when the cow is susceptible to pregnancy, is crucial for farm efficiency. Our goal is to enhance estrus detection (performance, interpretability), especially on the currently undetected silent estrus (35% of total estrus), and allow farmers to rely on automatic estrus detection solutions based on affordable data (activity, temperature). In this paper, we first propose a novel approach with real-world data analysis to address both behavioral and silent estrus detection through machine learning methods. Second, we present LCE, a local cascade based algorithm that significantly outperforms a typical commercial solution for estrus detection, driven by its ability to detect silent estrus. Then, our study reveals the pivotal role of activity sensors deployment in estrus detection. Finally, we propose an approach relying on global and local (behavioral versus silent) algorithm interpretability (SHAP) to reduce the mistrust in estrus detection solutions.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; • **Applied computing** → **Agriculture**.

KEYWORDS

Sustainable Dairy Management, Machine Learning, Classification, Interpretability

KDD'19, August 4-8, 2019, Anchorage, Alaska - USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

As underlined in the report *Creating A Sustainable Food Future* [29], ruminant livestock (cattle, sheep, and goats), used for dairy and meat production, occupy two-thirds of global agricultural land and contribute roughly half of agriculture's production-related emissions. Increased efficiency of resource use in farms is one of the most important steps toward meeting both food production and environmental goals. As a response, precision livestock farming (PLF) is a promising way to improve farm performance [31]. PLF is the use of continuous information to optimize an individualized animal management.

Nowadays, data (e.g. temperature, activity, body weight, milk production) is collected in dairy farms through different types of sensors to support farmers' decision making in various aspects of management (e.g. reproduction, diseases, feeding, environment). Machine learning methods can help to exploit the value of this ever-growing volume of data.

Reproduction is a key factor for dairy farm performance. It directly impacts milk production as cows start to produce milk after giving birth to a calf; and milk productivity declines after the first 3 months. The most prevalent reason for cow culling, the act of slaughtering a cow, is reproduction issue (e.g. long interval between 2 calves) [3]. So, it is crucial to detect estrus, the only period when the cow is susceptible to pregnancy, to timely inseminate cows and therefore increase farm efficiency.

Traditionally, estrus detection relies on visual observation of animal behaviors. Activity usually increases markedly in cows during estrus [16] unless the cow is experiencing a silent estrus (estrus without obvious behavioral signs - 35% of total estrus). In practice, less than 50% of estruses are detected visually [26] due to two main reasons: first silent estrus cannot be detected visually and second, sexual behaviors are mostly expressed at night. Different methods have been developed to aid visual detection. The reference method is estrus estimation using automated progesterone analysis in milk [10]. However, the cost of this solution prohibits its extensive implementation.

As a result, affordable activity and body temperature sensor data are considered having potential for automatic estrus detection [28]. Some solutions based on activity data

are available. However, their adoption rate remain moderate [31]. These commercial detection solutions face two major shortcomings. First, solutions based on activity cover only behavioral estruses (estruses associated with obvious behavioral signs - 65% of total estrus). Second, false alerts and lack of explanation behind detections generate solutions mistrust from farmers. Therefore, aside from an enhanced performance, key justifications for estrus alerts are also needed to expand automatic detection solution adoption.

1.1 Our Contributions

Our research tackles the challenge of milk production resource use efficiency in dairy farms with machine learning methods. We aim to enhance estrus detection, especially on the currently undetected silent estrus, and allow farmers to rely on automatic estrus detection solutions based on affordable data (e.g. activity, temperature). With our real world data analysis and an exhaustive estrus labeling (behavioral, silent) approach, this study will:

- Present LCE, a local cascade based algorithm for estrus detection;
- Show that LCE significantly outperforms a commercial reference in estrus detection;
- Evaluate the relevance of deploying a combination of 2 affordable sensors (activity and temperature);
- Identify the key drivers supporting estrus alerts at global and local level (behavioral versus silent) based on algorithm interpretability and propose an approach to reduce solution mistrust.

2 RELATED WORK

Multivariate time series (MTS) collected from activity and temperature sensors are labeled as either estrus or anestrus, the period of sexual inactivity between two periods of estrus. Estrus detection can be formulated as a binary classification problem. In this section, we first discuss the classifiers suited to our study. Then, we examine literature on classifier interpretability. Finally, we present existing work on estrus detection through machine learning methods.

2.1 Classification

Among state-of-the-art binary classifiers for numerical data and in the context of our problem, we can exclude the use of classifiers dedicated to MTS. MTS classifiers do not fit our needs for different reasons. First, current literature in animal science does not provide information to make assumption about a particular physiological model. Second, intervals [5] and shapelets [18] classification algorithms are excluded due to the short time windows we consider. Data from sensors are 24hr aggregated (the relevant period for estrus evaluation); and according to animal scientists, data on day of estrus and the day before estrus could be sufficient for estrus detection (time window size of two). Finally, dictionary representation approaches do not allow us to exploit temporal interactions between variables due to the aggregated representation of series on time [4]. Our dataset has the same frequency among variables, we manage the time aspect by setting the different timestamps as column variables.

Accordingly, we explore state-of-the-art classifiers in the following classes: k-nearest neighbors, regularized logistic regressions, support vector machines, neural networks and ensemble methods.

Firstly, we consider elastic net [32], the logistic regression combining L1 and L2 regularization methods, which constitutes the reference in regularized logistic regression.

Then, given the lower number of features than the number of samples in our dataset, we test a support vector machine with a radial basis function kernel.

Among categories of neural networks (multilayer perceptron - MLP, convolutional neural network - CNN and recurrent neural network - RNN), we consider small MLPs. Deep MLPs, without convolutional layers, are difficult to train due to the large number of parameters and the vanishing gradient problem [24]. Moreover, our dataset size (18,000 samples) and the inexistence of a pretrained CNN network on a comparable problem do not allow us to use CNNs. Then, RNNs are not suited to the short time windows we consider.

Lastly, the explicit (bagging and boosting) and implicit (negative correlation learning and mixture of experts) approaches exhibit respective strengths and limitations therefore a hybrid ensemble method is encouraged [21]. The strengths and limitations of explicit and implicit approaches concern their ability to generalize beyond the training dataset. Generalization performance depends on the balance found between an algorithm which is not capturing the underlying structure of the training dataset (underfitting - high bias) and an algorithm which is learning too closely the training dataset (overfitting - high variance). This challenge is called the bias-variance tradeoff. Negative Correlation Learning (NCL) attempts to train individual classifiers in an ensemble and combines them in the same learning process. On the entire training set, individual classifiers are trained simultaneously and interactively through the correlation penalty terms of their error functions to adjust the bias-variance tradeoff. The disadvantage is that all individual classifiers are concerned with the whole ensemble error. Mixture of Experts (ME) is an ensemble method based on the divide and conquer principle in which the problem space is divided between few experts (e.g. classifiers), supervised by a dynamic weighted average scheme (gating network). It allows each expert to learn a part of the training data with its corresponding individual error. However, there is no control over the bias-variance tradeoff. Combinations of NCL and ME implicit approaches exist [1, 13]. These methods integrate an error function correlation penalty term to encourage different classifiers (NCL), through a divide and conquer approach (ME), to learn using different parts of the training data. However, implicit approaches combinations do not benefit from the improved generalization ability of explicitly creating different training sets by probabilistically changing the distribution of the original training data (bagging, boosting). A method combining the explicit boosting approach with implicit ME divide and conquer approach exists [14]. Nonetheless, the

low bias distribution change of boosting does not ensure a bias-variance tradeoff.

Therefore, given the lower performance of small MLPs compared to ensemble methods in average (confirmed by our experiments), we propose a new hybrid ensemble method. It combines an explicit bagging-boosting approach to handle the bias-variance tradeoff and an implicit ME divide and conquer approach to learn different parts of the training data.

As previously mentioned, we cannot separate classifiers detection performance from interpretability. This will be explored in the next section.

2.2 Interpretability

There is no mathematical definition of interpretability. A definition proposed by [22] states that the higher the interpretability of a machine learning algorithm, the easier it is for someone to comprehend why certain decisions or predictions have been made.

Our problem requires insights into the type of estrus (behavioral versus silent), which suggests local explanations. Morevover, we need a method able to work for the different classifiers identified (model-agnostic). State-of-the-art methods meeting these requirements (local, model-agnostic) are Local Interpretable Model-agnostic Explanations (LIME) [27] and SHapley Additive exPlanations (SHAP) [20]. SHAP values come with the black box local estimation advantages of LIME, but also with theoretical guarantees. Therefore, we use SHAP in order to interpret the output of our machine learning algorithm. This technique is inspired by game theory, which is used to determine how much each player in a collaborative game has contributed to its success. In our study, SHAP values measure how much an activity or temperature variable impacts estrus predictions. A higher absolute SHAP value of a variable compared to other variables means that this variable has a higher predictive or discriminative power in detection algorithm. SHAP values are calculated by the average marginal contribution of a feature value towards the prediction over all possible coalitions. SHAP interaction values, an extension of SHAP values based on Shapley interaction index [15], capture pairwise interaction effects. In addition, SHAP values are available at local level. We analyze it to compare the impact of variables on algorithm predictions in behavioral and silent estrus.

2.3 Automatic Estrus Detection

There are a couple of studies about the application of machine learning methods on estrus detection [12, 19, 23]. None of them uses the currently recognized method for behavioral and silent estrus identification as labels (progesterone profiles), so their estrus labeling methods are not exhaustive. Moreover, two studies use different variables (milk volume, milking order, days since last estrus) rather than the affordable activity or temperature measurements. Finally, none of them gives insights on algorithm predictions based on its interpretability.

[23] bases the study on time series data of milk volume and milking order, using visual detection as the ground truth. Two learning schemes were tested - FOIL and C4.5. Algorithms detected 69% of estruses identified by visual method and a large number of false positives occurred (74%).

[19] learns a MLP on time series data of activity and the number of days since last estrus, using successful insemination as the ground truth. The model showed a sensitivity, a specificity and an error rate of 77.5, 99.6 and 9.1% on 373 estrus.

And lastly, [12] bases the study on time series data of activity, using visual detection as the ground truth (65.6% of all estruses). Three machine learning techniques were tested - random forest, linear discriminant and MLP. Algorithms showed 91%-100% accuracy on a limited dataset of 18 cows.

3 LCE: LOCAL CASCADE ENSEMBLE

As mentioned previously, we propose a new hybrid ensemble method which combines an explicit bagging-boosting approach to handle the bias-variance tradeoff and an implicit ME divide and conquer approach to individualize classifier error on different parts of the training data. We have decided to start from an existing combined implicit (NCL and ME) stacking-based approach (cascade generalization [30]): local cascade [17]. The bagging/boosting potential of local cascade decision tree divide and conquer method motivates our choice. In this section, we first introduce local cascade, the initial implicit stacking-based approach. Next, we explain LCE, our augmented (explicit and implicit) version of local cascade, and then compare LCE performance to local cascade. Figure 1 illustrates the presentation of local cascade and LCE.

3.1 Local Cascade

First of all, cascade generalization uses a set of classifiers sequentially and at each step adds new attributes to the original dataset [30]. The new attributes are derived from the class probabilities given by a base classifier (e.g. $H_0(D)$, $H_1(D_{01})$ in Figure 1). The bias-variance tradeoff is obtained by negative correlation learning: at each stage of the sequence, classifiers with different behaviors are selected. It is recommended in cascade generalization to begin with a low variance algorithm to draw stable decision surfaces (H_0 in Figure 1) and then use a low bias algorithm to fit more complex ones (H_1 in Figure 1). Local cascade [17] applies cascade generalization locally following a divide and conquer strategy based on mixture of experts principle. The objective of this approach is to capture new relations that cannot be discovered globally. The local cascade divide and conquer method is a decision tree. When growing the tree, new attributes (class probabilities from a classifier - base classifier) are computed at each decision node and propagated down the tree. In order to be applied as a predictor, local cascade stores, in each node, the model generated by the base classifier.

3.2 LCE: Local Cascade Ensemble

Our contribution intervenes in our explicit manner of handling the bias-variance tradeoff whereas local cascade approach is implicit, alternating between base classifiers behaviors (bias, variance) at each level of the tree.

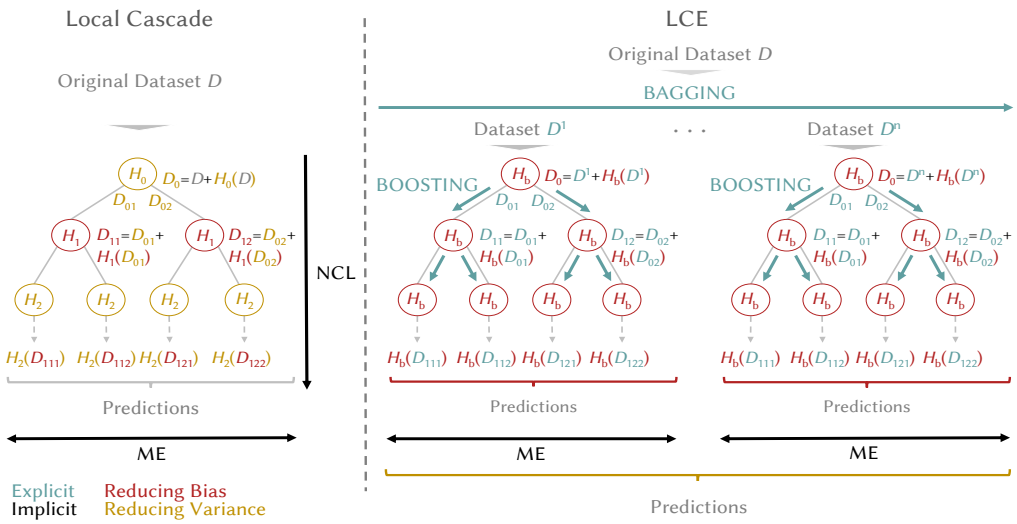

Figure 1: Local Cascade versus LCE

LCE reduces bias across decision tree through the use of boosting as base classifier (H_b in Figure 1). Boosting base classifier iterative data distribution change (reweighting) decreases the bias at each tree level. In addition, boosting is propagated down the tree by adding class probabilities of the base classifier to the training dataset (new attributes). Class probabilities contain information about the ability of the base classifier to correctly classify a sample. At the next tree level, class probabilities added to the dataset are exploited by the base classifier as a weighting scheme to focus more on previously misclassified samples.

Then, the overfit generated by the decision tree divide and conquer bias reduction approach is mitigated by the use of bagging. Bagging provides variance reduction by creating multiple decision trees from different subsamples of the original dataset (random sampling with replacement, see $D^1 \dots D^n$ in Figure 1). Trees are aggregated with a simple majority vote.

LCE new hybrid ensemble method enables to balance the bias-variance tradeoff without the need for an interactive learning between individual classifiers (NCL), while benefiting from the improved generalization ability of explicitly creating different training sets (bagging, boosting). Furthermore, LCE divide and conquer method ensures that classifiers learn on different parts of training data without the need for a supervision scheme (gating network).

We present LCE pseudocode in Algorithm 1. A function (LCE_Tree) builds a tree and the second one (LCE) the forest of trees through bagging.

There are 2 stopping criteria during a tree building phase: when a node has an unique class or when the tree reaches the maximum depth. We set the range of tree depth from 0 to 3 in LCE instead of 0 to 5 in local cascade. This hyperparameter is used to control overfitting. Our choice of low bias boosting base classifiers justifies the maximum depth adjustment to 3. In this study, the set of low bias base classifiers is limited

Algorithm 1 LCE: Local Cascade Ensemble

Require: A dataset D, a set of classifiers H, maximum depth of a tree max_depth, number of trees n_trees

function LCE(D, H, n_trees, max_depth)
  $F \leftarrow \emptyset$
  for each i in $[1, n\_trees]$ **do**
    $S \leftarrow$ A bootstrap sample from D
    $t \leftarrow$ LCE_Tree(S, H, max_depth, 0)
    $F \leftarrow F \cup t$
  return F
function LCE_Tree(D, H, max_depth, $depth$)
  if max_depth or uniform class **then**
    return leaf
  else
    $D' \leftarrow$ Concatenate$(D, H_{depth}(D))$
    Split D' on attribute maximizing Gini criterion
    $depth \leftarrow depth + 1$
    for $D'^{(j)} \in \mathcal{P}(D')$ **do**
      $Tree_j =$ LC_Tree$(D'^{(j)}, H, max_depth, depth)$
    return tree containing a decision node, storing classifier $H_{depth+1}(D)$ and descendant subtrees $Tree_j$

to the state-of-the-art boosting algorithm (extreme gradient boosting - XGB [9]).

In addition, we removed two rules implemented in local cascade to reduce variance: the maximum base classifier error rate and the minimum class representation in a node. The first rule requires the stopping of propagation down the tree to prevent overfitting if the base classifier, in a node, had an error rate below a certain threshold (0.5). Our approach suggests a variance reduction through bagging, and not during a tree construction; so we did not keep this rule. In order to restrict the attention to well populated classes, the second rule requires considering a class in a node if the number of examples belonging to this class is greater than N times (3)

the number of attributes. We did not keep the second rule for the same reason.

3.3 Performance Comparison: Local Cascade versus LCE

Our comparison aims to underline the superior performance of LCE compared to a local cascade on our real-world dataset, induced by their different approaches of handling bias-variance tradeoff (explicit versus implicit approach). LCE is implemented according to the description given in the previous section. Local cascade implementation corresponds to the description of the original paper and as recommended, we use naive bayes for low variance base classifier. In order to be comparable, the low bias base classifier is XGB. Depth is set to 1 for LCE and the local cascade. Results are presented in Table 1.

Table 1: F1-score with 95% confidence interval of LCE versus local cascade (LC) on our dataset

Trees	**1**	**5**	**10**	**30**	**50**	**70**	**90**
LCE	68.1 ±3.2	69.2 ±2.6	68.9 ±2.8	69.1 ±2.4	69.1 ±2.5	68.9 ±2.4	68.9 ±2.5
LC	53.2 ± 2.8						

As expected, results show a higher variability across folds of LCE compared to the local cascade when the number of tree is set to 1 due to its low bias orientation (standard error of 1.6% versus 1.4% on F1 score, performance calculation detailed in section Experiments). However, LCE on 1 tree exhibits a higher detection performance than local cascade (F1 score: 68.1% versus 53.2%).

Additionally, through bagging, we observe LCE variability reduction to a lower level than local cascade as well as an increase of detection performance (F1 score 95% confidence interval: 68.1 ± 3.2 with 1 tree versus 68.9 ± 2.4 with 70 trees versus 53.2 ± 2.8 with local cascade).

Therefore, this comparison affirms the superiority of our explicit bias-variance tradeoff approach compared to the implicit NCL approach of local cascade on our dataset. The intrinsic different behavior of LCE and local cascade is confirmed in the results and discussions section.

4 EXPERIMENTS

In this section, we present the composition of our real-world dataset, the preprocessing performed and the structure of the 5 folds used for cross-validation.

4.1 Dataset

Our dataset is offline. From 2014 to 2017, an experiment was conducted at the INRA Méjusseaume dairy farm (4806' N, 147' W, Brittany, France). This experiment enrolled 125 Holstein cows housed in free stalls representing 153 lactations.

Each cow was equipped with a collar-mounted activity meter (HeatPhone - Medria Technologies, Châteaubourg, France) and a temperature sensor in first stomach (Thermobolus - Medria Technologies, Châteaubourg, France). Based on its good performance compared to other solutions [8] and its international market presence, we hold that Medria estrus detection system is a reasonable basis of comparison. In the following sections, Medria is called the commercial solution (CS). The dataset consists of visual estrus alerts, Medria estrus alerts and Medria numeric variables with a 5-minute frequency (*rumination*, *ingestion*, *rest*, *standing up*, *overactivity*, *other activity*, *temperature*, and *temperature corrected*). *Temperature corrected* takes into account the cooling effect of water ingestion by the cows. Concerning the visual estrus alerts, visual observation was conducted by farm staffs. Staff also checked the commercial solution alerts before inputting their visual records, thus these visual estrus alerts are shown as Visual&CS in the study. The preprocessing applied on the data collected is a 24hr aggregation (activity: sum, temperature: mean) which corresponds to the relevant window for both estrus detection and, from an alert standpoint, farmers' needs. We assume that the treatment operated by Medria on raw data to generate variables is stable during our experiment.

Our novel approach addresses both estrus categories detection (behavioral and silent). Therefore, we labeled estrus by measuring the progesterone concentration in whole milk, the current reference for an exhaustive estrus identification. This time-effective and non-invasive method for the cow induces commonly accepted errors (progesterone measurements, profiles analysis [2]). We mark an estrus as behavioral estrus when either a visual detection or a Medria alert occurred. An estrus is considered silent when neither visual detection or a Medria alert occurred. Our dataset is composed of 671 estruses with 37% of silent estrus which is aligned with the rate of 35% observed in literature [25].

Days preceding estrus are a valuable source of information for estrus detection, we set it as a hyperparameter. Every value in the range from 1 to 21 days, the length of a regular ovarian cycle, are tested. Past days of variables are added as feature columns.

4.1.1 Feature Selection. We perform feature selection in this study because of the sensitivity of the method chosen to interpret the detection algorithm (SHAP) to high correlations among features. We conduct a subset selection on pairs of collinear features based on the Pearson correlation coefficient (threshold 0.8). One pair of features is above the threshold (0.9: *temperature corrected*, *temperature*). Since *temperature* is affected by the cooling effet of water ingestion, the variable *temperature corrected* is selected. From this point onwards, *temperature corrected* is named temperature. After this feature selection, no Pearson pairwise correlation in the case of the 21 past days dataset is above the threshold.

4.1.2 Dataset Structure. We make a 5-fold cross validation. Dataset split is presented in Table 2.

The split has kept the same number of days in estrus in each fold (1,144 days). We made this choice to avoid overfitting on a particular animal. We discuss the impact of

Table 2: Dataset Split

	Fold1	Fold2	Fold3	Fold4	Fold5	All
Estrus	126	136	118	141	153	671
Silent %	33	40	24	40	46	37

a split keeping the same number of animals per fold in the detection performance section. Moreover, we do not observe any structural imbalance on silent estrus percentage across the folds.

4.2 Experimental Setting

We present in this section algorithms and methods used in our experiments.

4.2.1 Algorithms. We tested our hybrid ensemble LCE (explicit - implicit approaches) versus the initial implicit approach (local cascade) and the state-of-the-art algorithm for each explicit approach (bagging: random forest, boosting: extreme gradient boosting). K-nearest neighbors, elastic net, support vector machines and small MLPs are also tested.

- k-nearest neighbors - KNN: we use the implementation neighbors.KNeighborsClassifier in the scikit-learn package for Python[1]
- Elastic net - EN: we use the implementation linear model SGDClassifier in the scikit-learn package for Python[1]
- Support Vector Machine - SVM: we use the implementation svm.SVC in the scikit-learn package for Python[1]
- Random Forest - RF: we use the implementation ensemble.RandomForestClassifier in the scikit-learn package for Python[1]
- Extreme Gradient Boosting - XGB: we use the implementation in the xgboost package for Python[2]
- Local cascade - LC: algorithm has been reimplemented in Python 2.7 based on the description of the paper since no public version available.
- LCE: algorithm implemented in Python 2.7
- Multilayer Perceptron - MLP: we use the implementation available in the package Keras for Python[3] and limit the neural network architecture to 3 layers

4.2.2 Optimization of Hyperparameters. Hyperparameters of classifiers are set by hyperopt, a sequential model-based optimization using a tree of Parzen estimators search algorithm [7]. Hyperopt chooses the next hyperparameters decision from the previous choices and a tree-based optimization algorithm. Tree of Parzen estimators meet or exceed grid search and random search performance for hyperparameters setting [6]. We use the implementation available in the Python package hyperopt[4] and hyperas wrapper for keras. Optimization is undertaken to maximize F1-score. The choice of this metric is driven by 2 reasons. First, we do not make assumption about the dairy management style; farmers can favor a higher estrus detection rate (higher recall) or fewer false alerts (higher precision) according to their needs. Second, we face a class imbalance (33% of estrus days) which renders irrelevant the accuracy metric.

4.2.3 Classification Performance. Our experiments use progesterone profiles as ground truth for exhaustive estrus identification. The levels of progesterone allow us to identify a time window of 3 days for estrus with a duration of less than 24 hours, in the standard scheme. Adopting a conservative approach, we decided to aggregate by the maximum of our daily predictions on estrus/anestrus period to calculate the classification performance. In addition, we observe that for high thresholds (threshold > 0.95), classifiers performances are unstable with a significant decrease in estrus detection rate (recall below 70%). In addition, for low thresholds (threshold < 0.1), classifiers are equivalent to a random classifier. So, we decided to adopt a F1-score calculation based on the average of F1-score on threshold range 0.1-0.95. This calculation does not modify the classifier selection results or the comparison result with the commercial solution. Nonetheless, it corresponds to the plausible range of calibration for dairy management and shows a detection performance closer to real conditions.

4.2.4 Algorithm Selection. Based on a 5-fold cross-validation 60/20/20 train/validation/test split, the best classifier is selected based on the highest F1-score on validation sets.

4.2.5 Statistical Test. As recommended by [11], we have used a 5×2 cross validation t-test for statistical significance of machine learning algorithms on one dataset.

4.2.6 Interpretability. As mentioned in the related work section, we use the SHAP implementation available in the Python package shap[5].

5 RESULTS AND DISCUSSIONS

This section is structured into two parts: performance and interpretability. The detection performance part compares LCE to other detection methods (classifiers, commercial solution) and evaluates the relevance of deploying 2 sensors. Then, we identify the key drivers (variables impact, temporal interactions) behind the estrus detection alerts at global and local level (behavioral versus silent) based on algorithm interpretability (SHAP) and propose an approach to reduce the solution mistrust.

5.1 Detection Performance

Classification results on test sets are presented in Figure 2.

The best classifier on validation sets is LCE with the following hyperparameters: 3 past days, depth equals to 1 and 70 trees. We do not observe an overfit of LCE, the performance observed on test sets (F1-score: 68.9) is stable compared to the one of the validation sets (F1-score: 68.1).

Furthermore, the performance of LCE responds to the objective of an increase in performance in both estrus detection rate and fewer false alerts compared to the commercial

[1] https://scikit-learn.org/stable/
[2] https://xgboost.readthedocs.io/en/latest/python/
[3] https://keras.io/
[4] https://github.com/hyperopt/hyperopt
[5] https://github.com/slundberg/shap

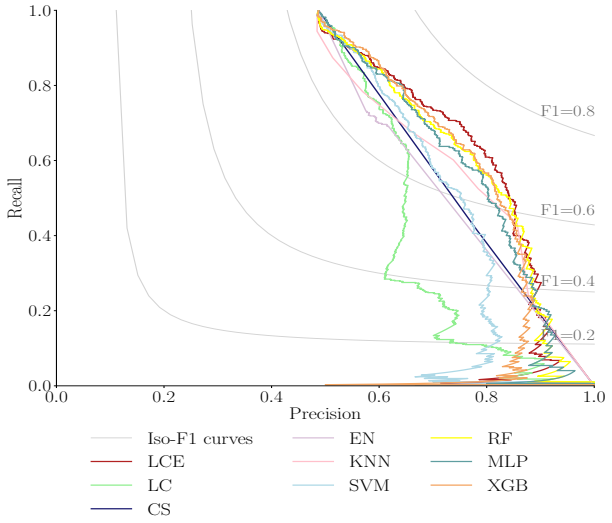

Figure 2: Precision recall curves on test sets of the classifiers versus the commercial solution

solution (CS). At the same precision, LCE recall is constantly higher than commercial solution recall. At a precision of 78%, the precision rate of the commercial solution in this study, our algorithm detects 22% more estrus.

5.1.1 Comparative Analysis. We compare the error rate correlation of LCE to those of other detection methods. This comparison allows us to:

- gain insights into the shortcomings of the commercial solution and LCE detections
- identify limitations of our approach for deployment

A low correlation indicates that classifiers err in different regions of the instance space. Table 3 presents Pearson correlations of LCE prediction errors with other detection methods (classifiers and commercial solution) on test sets. In order to be comparable, we have set the threshold of each classifier with the same precision as the commercial solution (78%).

Table 3: Pearson pairwise correlations of LCE prediction errors with other detection methods on test sets

KNN	EN	SVM	MLP	RF	XGB	LC	CS
0.61	0.19	0.57	0.69	0.73	0.8	0.41	0.37

First, the commercial solution shows an intrinsic different behavior from that of LCE (correlation: 0.37). This low correlation is mainly explained by the null performance of the commercial solution on silent estrus detection across the herd. On 67% of the cows, composed of a slighlty higher proportion of silent estrus compared to average (40% versus 37%), predictions correlation of the commercial solution with LCE is 0.21 ± 0.03.

Next, the low correlation between LCE and local cascade (0.41) confirms the value added by the explicit bias-variance tradeoff of the LCE approach. This low correlation is explained by the low recall (11%) of the local cascade for a precision of 78%. The stable decision surface drawn by naive bayes at the root of the local cascade decision tree substantially limits the range of performance of the algorithm on our dataset (recall drops with a precision higher than 66%). We observe this performance drop for precision above 66% in Figure 2.

Finally, the classifier with the closest behavior to LCE is XGB (0.8). However, the correlation difference remains substantial and is explained by some divergence among few cows. The divergence, an error rate correlation below 0.6, concerns 12% of the cows comprising a proportion of silent estrus aligned with average (35%). Therefore, our bias-variance approach enhances XGB performance on standard cases (cows with 35% of silent estrus). Nevertheless, we observe a poor performance of LCE on 11% of the cows exhibiting a high proportion of silent estrus (F1 score $< 55\%$, silent estrus proportion: 54%). Silent estrus are not equally distributed among cows. In our dataset, 16% of the cows represent 40% of the silent estrus. LCE performance per cow is exposed to the animal estrus type proportion. It is confirmed by the LCE performance drop when assessed on the activity and temperature dataset generated by a stratified 5-fold on animals (66.3 ± 3.4). LCE performance per cow variability according to the animal estrus type proportion is a limitation of our solution for deployment; meanwhile it is also a driver for detection improvement. We suggest further investigation to incorporate additional animal individual features.

5.1.2 One or Two Sensors? In order to answer this question, we compare the detection performance on test sets of LCE on the temperature, the activity and both variables. We also compare LCE detection results to the commercial solution and visual method.

First, the results confirm the potential of data science techniques for automatic estrus detection versus visual detection as concluded by [12]. We observe that LCE for both behavioral and silent estrus detection, trained on activity and temperature data, manifests significantly better performance (F1-score and lower variability) than Visual&CS (68.9 ± 2.4 versus 60.4 ± 4.6, $P < 0.05$). Our Visual&CS performance is aligned with the state-of-the-art [26]; the detection rate is slightly below 50% (47%).

Second, we observe a better performance (higher F1-score and lower variability) with our algorithm trained on activity and temperature than activity or temperature alone (68.9 ± 2.4 versus 67.0 ± 3.0 versus 55.9 ± 2.3). The performance difference is only significant when compared to the algorithm trained using the temperature. We infer that, in the conditions of our experiment, only activity sensor should be deployed: the performance is not significantly lower than that trained with two sensors (activity and temperature).

Nonetheless, temperature information cannot be excluded. We observe a markedly lower variability of the algorithm based on temperature across folds which allows the algorithm based on activity and temperature to reduce its variability. It

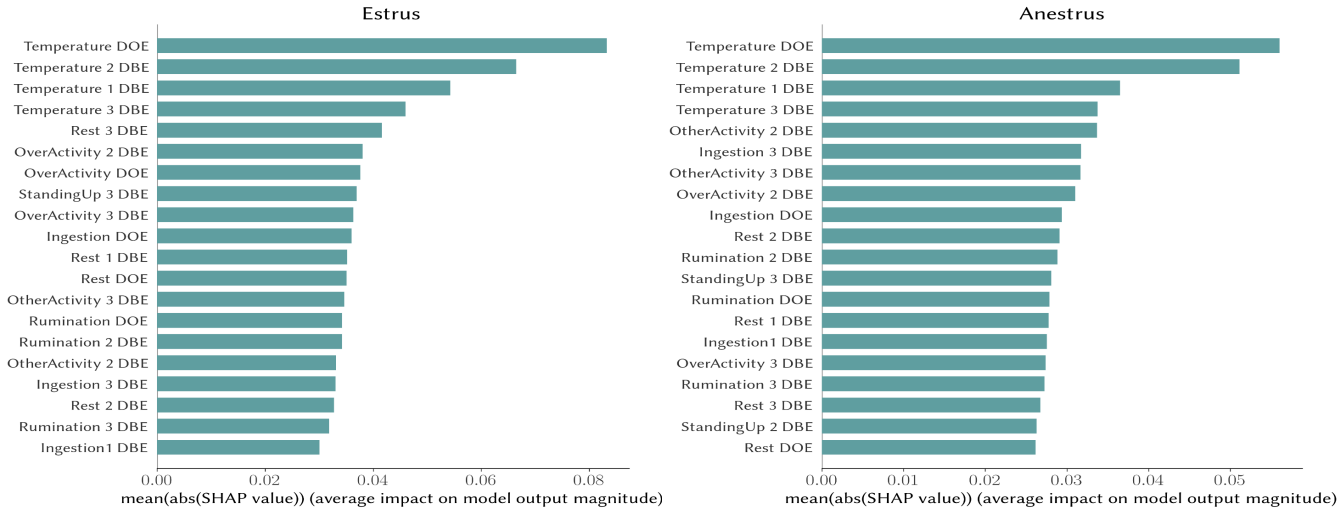

Figure 3: Average impact of the attributes on algorithm predictions for estrus and anestrus. Abbreviations: DOE - Day Of Estrus; DBE - Day Before Estrus

means that the algorithm based on temperature is consistent on different data. It implies a possible higher discriminative and generalizing power. We propose to further study the potential of temperature data for estrus detection with a broader data heterogeneity (cows breed, environment). The next step would consist of a partnership with an automatic detection solution provider to have access to a more diverse dataset.

5.2 Interpretability of our Solution

In this section, we firstly present the relative impact of variables in LCE predictions and their temporal interactions. Then, we propose an approach to give insights on estrus detection to the farmers based on these elements.

Figure 3 shows the average impact of each variable on algorithm predictions for estrus and anestrus by decreasing order.

These results confirm the discriminative power of the temperature and its potential for improving estrus detection performance. The variable with the strongest impact to algorithm predictions is the temperature on the day of estrus for both estrus and anestrus classes.

Next, we observe that the ranking of all activity variables are different with a significant rank change between estrus and anestrus. Therefore, the relative impact of each activity variable in LCE predictions differs between estrus and anestrus. Overactivity on the day of estrus, a typical characteristic of most estrus (65%), appears as the third most impactful variable after temperature estrus and does not appear on the top 20 of variables for anestrus.

By taking the same impact ranking approach locally for behavioral versus silent estrus, we also observe a significant change on the ranking of activity variables (75% of rank change). Rumination 2 days before estrus is a key variable in silent estrus detection. It is the third most impactful activity variable for silent estrus and appears at the 19th position for behavioral estrus.

Finally, temporal relations among variables differ between behavioral and silent estrus. SHAP interaction values reveal that algorithm predictions are more impacted by activity variables further to the day of estrus for silent estrus than behavioral estrus. For example, the variable of highest interaction with rumination on the day of estrus is the rest 3 days before estrus for silent estrus versus the rest 2 days before estrus for behavioral estrus. This observation holds true for over activity, standing up and ingestion (two third of activity variables).

Therefore, in order to support LCE estrus alerts and ease solution adoption, we propose an approach based on LCE interpretability (activity sensor only). First, communicate to the farmer the relatedness of the estrus detection to historical cases through a confidence indicator and the amplitude of differences in the 3 most impactful activity variables (rest 3 days before estrus, over activity 2 days before estrus and over activity on the day of estrus). The confidence indicator corresponds to the weighted average of absolute SHAP values differences by the ranking of impact variables for estrus from our reference presented above. Second, in case of estrus, inform the farmer about the type of estrus (behavioral/silent) with a confidence level and which temporal interactions are satisfied. The information about the type of estrus aims to reassure farmers when they are not able to verify the estrus alert by visual behavioral signs, therefore reduce potential mistrust. Confidence level is calculated like the previous one but using ranking of variables impact of silent estrus as a reference. In addition, temporal interactions are communicated in decreasing order of variable impact.

6 CONCLUSION

Our study confirms the significant performance improvement of LCE on estrus detection compared to commercial solutions, a result driven by silent estrus detection. It also proves the pivotal role of activity sensors deployment in these

detections. The interpretability of LCE offered by SHAP, disclosing information about the relatedness of the predictions to historical cases and the possibility of visually verifying the estrus (behavioral versus silent), promises mistrust reduction from farmers. Concerning the deployment of our solution, the homogeneity (cows breed, environment) of our dataset is a limitation. The next step would consist of a partnership with an automatic detection solution provider to have access to a heterogeneous dataset.

ACKNOWLEDGMENTS

We thank all technical staff of INRA Méjusseaume dairy farm who helped managing and monitoring this long-term experimentation. We also thank Medria for its collaboration by providing activity and temperature sensor data. This work was supported by the French National Research Agency under the Investments for the Future Program (ANR-16-CONV-0004), French national project Deffilait (ANR-15-CE20-0014) and APIS-GENE.

REFERENCES

[1] E. Abbasi, M. Shiri, and M. Ghatee. 2016. Root-quatric mixture of experts for complex classification problems. *Expert Systems with Applications* 53 (2016), 192–203. https://doi.org/10.1016/j.eswa.2016.01.040

[2] I. Adriaens, W. Saeys, T. Huybrechts, C. Lamberigts, L. Franois, K. Geerinckx, J. Leroy, B. De Ketelaere, and B. Aernouts. 2018. A novel system for on-farm fertility monitoring based on milk progesterone. *Journal of Dairy Science* 101, 9 (2018), 8369–82. https://doi.org/10.3168/jds.2017-13827

[3] S. Bascom and A. Young. 1998. A Summary of the Reasons Why Farmers Cull Cows. *Journal of dairy science* 81, 8 (1998), 2299–305. https://doi.org/10.3168/jds.S0022-0302(98)75810-2

[4] M. Baydogan and G. Runger. 2014. Learning a symbolic representation for multivariate time series classification. *Data Mining and Knowledge Discovery* 29, 2 (2014), 400–22. https://doi.org/10.1007/s10618-014-0349-y

[5] M. Baydogan, G. Runger, and E. Tuv. 2013. A Bag-of-Features Framework to Classify Time Series. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 11 (2013), 2796–2802. https://doi.org/10.1109/TPAMI.2013.72

[6] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. 2011. Algorithms for Hyper-Parameter Optimization. In *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 2546–2554.

[7] J. Bergstra, D. Yamins, and D. Cox. 2013. Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. In *Proceedings of the 30th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Sanjoy Dasgupta and David McAllester (Eds.), Vol. 28. PMLR, Atlanta, Georgia, USA, 115–123.

[8] A. Chanvallon, S. Coyral-Castel, J. Gatien, J. Lamy, D. Ribaud, C. Allain, P. Clment, and P. Salvetti. 2014. Comparison of three devices for the automated detection ofestrus in dairy cows. *Theriogenology* 82, 5 (2014), 734–41. https://doi.org/10.1016/j.theriogenology.2014.06.010

[9] T. Chen and C. Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 785–794. https://doi.org/10.1145/2939672.2939785

[10] E. Cutullic, L. Delaby, Y. Gallard, and C. Disenhaus. 2011. Dairy cows' reproductive response to feeding level differs according to the reproductive stage and the breed. *Animal* 5, 5 (2011), 731–40. https://doi.org/10.1017/S1751731110002235

[11] T.G. Dietterich. 1998. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation* 10, 7 (1998), 1895–1923. https://doi.org/10.1162/089976698300017197

[12] K.A. Dolecheck, W.J. Silvia, G.Jr Heersche, Y.M. Chang, D.L. Ray, A.E. Stone, B.A. Wadsworth, and J.M. Bewley. 2015. Behavioral and physiological changes around estrus events identified using multiple automated monitoring technologies. *Journal of Dairy Science* 98, 12 (2015), 8723–31.

[13] R. Ebrahimpour, S. Arani , and S. Masoudnia. 2013. Improving combination method of NCL experts using gating network. *Neural Computing and Applications* 22, 1 (2013), 95–101. https://doi.org/10.1007/s00521-011-0746-8

[14] R. Ebrahimpour, N. Sadeghnejad, S. Arani, and N. Mohammadi. 2013. Boost-wise pre-loaded mixture of experts for classification tasks. *Neural Computing and Applications* 22, 1 (2013), 365–77. https://doi.org/10.1007/s00521-012-0909-2

[15] K. Fujimoto, I. Kojadinovic, and J.L. Marichal. 2006. Axiomatic characterizations of probabilistic and cardinal-probabilistic interaction indices. *Games and Economic Behavior* 55, 1 (2006), 72–99. https://doi.org/10.1016/j.geb.2005.03.002

[16] C. Gaillard, H. Barbu, M.T. Srensen, J. Sehested, H. Callesen, and M. Vestergaard. 2016. Milk yield and estrous behavior during eight consecutive estruses in Holstein cows fed standardized or high energy diets and grouped according to live weight changes in early lactation. *Journal of Dairy Science* 99, 4 (2016), 3134–43.

[17] J. Gama and P. Brazdil. 2000. Cascade Generalization. *Machine Learning* 41, 3 (2000), 315–43. https://doi.org/10.1023/A:1007652114878

[18] I. Karlsson, P. Papapetrou, and H. Boström. 2016. Generalized random shapelet forests. *Data Mining and Knowledge Discovery* 30, 5 (2016), 1053–85. https://doi.org/10.1007/s10618-016-0473-y

[19] J. Krieter. 2005. Oestrus detection in dairy cows using control charts and neural networks. In *Proceedings of 56th Annual Meeting of the European Association for Animal Production. Commission on Cattle Production*. 1–11.

[20] S. Lundberg and S. Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 4765–4774.

[21] S. Masoudnia and R. Ebrahimpour. 2014. Mixture of experts: a literature survey. *Artificial Intelligence Review* 42, 2 (2014), 275–93. https://doi.org/10.1007/s10462-012-9338-y

[22] T. Miller. 2017. Explanation in Artificial Intelligence: Insights from the Social Sciences. *CoRR* abs/1706.07269 (2017).

[23] R.S Mitchell, R.A. Sherlock, and L.A. Smith. 1996. An investigation into the use of machine learning for determining oestrus in cows. *Computers and Electronics in Agriculture* 15, 3 (1996), 195–213. https://doi.org/10.1016/0168-1699(96)00016-6

[24] M.A. Nielsen. 2015. *Neural Networks and Deep Learning*. Determination Press.

[25] M.A. Palmer, G. Olmos, L.A. Boyle, and J.F. Mee. 2010. Estrus detection and estrus characteristics in housed and pastured Holstein-Friesian cows. *Theriogenology* 74, 2 (2010), 255–64. https://doi.org/10.1016/j.theriogenology.2010.02.009

[26] O.A. Peralta, R.E. Pearson, and R.L. Nebel. 2005. Comparison of three estrus detection systems during summer in a large commercial dairy herd. *Animal Reproduction Science* 87, 1 (2005), 59–72. https://doi.org/10.1016/j.anireprosci.2004.10.003

[27] M.T. Ribeiro, S. Singh, and C. Guestrin. 2016. Why Should I Trust You?: Explaining the Predictions of Any Classifier. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1135–44. https://doi.org/10.1145/2939672.2939778

[28] M. Saint-Dizier and S. Chastant-Maillard. 2012. Towards an automated detection of oestrus in dairy cattle. *Reproduction in Domestic Animals* 47, 6 (2012), 1056–61.

[29] T. Searchinger, R. Waite, C. Hanson, J. Ranganathan, P. Dumas, and E. Matthews. 2018. *Creating a Sustainable Food Future*. World Resources Institute.

[30] M. Sesmero, A. Ledezma, and A. Sanchis. 2015. Generating ensembles of heterogeneous classifiers using Stacked Generalization. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 5, 1 (2015), 21–34. https://doi.org/10.1002/widm.1143

[31] W. Steeneveld and H. Hogeveen. 2015. Characterization of Dutch dairy farms using sensor systems for cow management. *Journal of Dairy Science* 98, 1 (2015), 709–17. https://doi.org/10.3168/jds.2014-8595

[32] H. Zou and T. Hastie. 2005. Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 67, 2 (2005), 301–320.