Computers and Electronics in Agriculture 191 (2021) 106566

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Leveraging sequential information from multivariate behavioral sensor data to predict the moment of calving in dairy cattle using deep learning

Arno Liseune [a,*], Dirk Van den Poel [a], Peter R. Hut [c], Frank J.C.M. van Eerdenburg [c], Miel Hostens [b,c]

[a] Faculty of Economics and Business Administration, Ghent University, Tweekerkenstraat 2, B-9000 Ghent, Belgium
[b] Faculty of Bioscience Engineering, Ghent University, Coupure Links 653, B-9000 Ghent, Belgium
[c] Faculty of Veterinary Medicine, Utrecht University, Yalelaan 7, 3584CL Utrecht, Netherlands

ARTICLE INFO

Keywords:
Calving management
Sensors
Sequential Models
Animal Monitoring
Deep Learning

ABSTRACT

Calving is one of the most critical moments during the life of a cow and their calves. Timely supervision is therefore crucial for animal welfare as well as the farm economics. In this study, we propose a framework to predict calving within 24 h, 12 h, 6 h, 3 h and 1 h of dairy cows using sequential sensor data. In particular, data were extracted from 2363 cows coming from 8 commercial farms between August 2016 and November 2020. Two sensors attached to the neck and leg of each cow measured rumination, eating, lying, standup, walking and inactive behavior on a minute basis. A novel methodology was used to impute the missing values in the sensor sequences by leveraging the observed values of all the behavioral activities recorded by the sensors. A deep learning model was then used to predict the moment of calving on an hourly basis using the imputed sensor sequences. Results show that 65% of the calvings within 24 h can be detected with a precision of 77%, while 57% of calvings occurring within 3 h can be identified with a precision equal to 49%. Moreover, we find that using the missing value imputations significantly improves the predictive performance for observations containing up to 60% of missing values. The framework proposed in this study can be used by farmers to optimize their calving management and hence improve animal monitoring.

1. Introduction

Calving is one of the most critical moments of both the cow's and the calf's life (Barrier et al., 2013; Mee, 2013). Dystocia, i.e. difficulties or abnormalities encountered during calving, can severely affect health and welfare of dairy cattle (Barrier and Haskell, 2011). In particular, dams that experience dystocia can be at increased risk of injury as well as contracting uterine diseases such as metritis and endometritis (Rutten et al., 2017). Moreover, it is reported that dystocia is one of the most painful conditions for dairy cows (Laven et al., 2009). Dystocial calves on the other hand can experience many physiological problems such as prolonged hypoxia and significant acidosis (Lombard et al., 2007) as well as physiological stress and internal injuries (Berglund et al., 2003). This in turn can reduce the calf's long-term survival or even result in stillbirth (Lombard et al., 2007). In fact, 7% of all the calves born in the United States die directly within 48 h and 50% of the stillbirths can be directly attributed to dystocia (Meyer et al., 2000). Difficulties with calving can therefore negatively affect animal welfare as well as farm economics (Mee, 2004). Specifically, dystocia can be very costly to dairy farmers as it is associated with a lower fertility, milk production and survival rate of the dam (Tenhagen et al., 2007). Additionally, the need for veterinary assistance contributes to the economic cost of dystocia. In particular, the total cost associated with a difficult calving has been estimated at €500 (McGuirk et al., 2007). The financial losses related to stillbirth even average $938 per case (Mahnani et al., 2018). Reducing difficulties with calving is therefore crucial to the dairy producer.

Several risk factors causing dystocia include parity, calf weight, sex, body size and pelvic diameters of the dam as well as seasonal effect and environmental stress (Tenhagen et al., 2007). Yet, farm management such as breeding decisions and human supervision can strongly influence calving difficulties as well (Rutten et al., 2017; Van Pelt and de Jong, 2011). More specifically, it has been shown that providing timely human intervention reduces the risk of dystocia, the pain experienced during labor and the reproductive decline of the dam (Borchers et al.,

* Corresponding author.
E-mail addresses: Arno.Liseune@ugent.be (A. Liseune), Dirk.VandenPoel@ugent.be (D.V. den Poel), p.r.hut@uu.nl (P.R. Hut), f.j.c.m.vaneerdenburg@uu.nl (F.J.C.M. van Eerdenburg), Miel.Hostens@ugent.be (M. Hostens).

https://doi.org/10.1016/j.compag.2021.106566
Received 2 July 2021; Received in revised form 9 November 2021; Accepted 11 November 2021
Available online 19 November 2021

Table 1
The sensor activities and their corresponding features recorded on an hourly and daily basis.

Activity	1 h features	24 h features
Walking	minutes	
		minutes
Standing	minutes	
		minutes
		number of bouts
Eating	minutes	
		minutes
		number of bouts
		avg bout minutes
		avg inter bout minutes
Rumination	minutes	
		minutes
		number of bouts
		avg bout minutes
		avg inter bout minutes
Lying	minutes	
		minutes
		number of bouts
		avg bout minutes
Inactivity	minutes	
		minutes
		number of bouts
		avg bout minutes
		avg inter bout minutes
Leg activity	number of steps	
		number of steps

2017). Individual animal monitoring, however, becomes increasingly more difficult as the number of cattle per farm globally increases over time (Raussi, 2003). In fact, even with intensive monitoring, it remains difficult to forecast the moment of calving correctly (Lange et al., 2017). One way to organize human supervision more efficiently is by employing models that are able to accurately predict the moment of parturition. Such models can automatically alert farmers of an imminent calving and hence facilitate timely calving supervision (Ouellet et al., 2016). Physical and behavioral changes may provide clues to detect when cows are about to calve (Huzzey et al., 2005). More specifically, it has been shown that behaviors such as eating, rumination and grooming decrease, while restlessness and lying bouts increase during the period around calving (Miedema et al., 2011; Jensen, 2012; Schirmann et al., 2013). Visually assessing these behavioral changes, however, is subjective, time consuming and prone to human error (Ouellet et al., 2016). Several frameworks were, therefore, presented to predict the onset of calving by automatically processing these changes in behavioral patterns. Ouellet et al., 2016, for example, constructed three different models to predict the moment of calving based on four calving indicators, i.e., vaginal temperature, rumination time, lying time and lying bouts. More specifically, three logistic regression models were built that predicted the start of parturition within 24 h, 12 h and 6 h based on the optimal combination of the four aforementioned indicators. Similarly, Fadul et al., 2017 trained a logistic regression model with a stepwise selection procedure to predict the onset of calving within the next 3 h based on rumination time and chews, lying bouts, boluses as well as other activities not related to ruminating, feed intake or drinking. Whereas the two previously mentioned studies removed missing data, all observations with missing values were assigned to the training set in the study presented by Zehner et al., 2019. A Naive bayes model was then trained and evaluated on a validation set, which exclusively consisted of observations with complete information. Rutten et al., 2017 on the other hand, presented a methodology to impute the missing values by a weighted average of sensor data recorded during the previous three days at the same time period. A logistic regression model was then trained on the imputed data to generate the calving predictions. Borchers et al., 2017 applied more complicated machine learning techniques such as random forests, linear discriminant analysis and neural networks to predict the start of calving. The same dataset was used in a subsequent study conducted by Keceli et al., 2020 who applied a Bidirectional Long Short-Term Memory (Bi-LSTM) to process the data sequentially.

Yet, in most of the previously mentioned studies, the proposed frameworks disentangle the temporal information in the sensor sequences. As result, these models are not able to leverage the sequential patterns in the behavioral changes, which can negatively affect model performance. Additionally, the previously presented frameworks are difficult to generalize and may not be suitable for practical applications. In particular, in most studies, observations with missing values are removed (Fadul et al., 2017; Ouellet et al., 2016; Borchers et al., 2017; Zehner et al., 2019; Keceli et al., 2020). As a result, these models won't be able to generate reliable calving predictions when missing values are present in the observed sensor sequences. In one study, however, missing values were imputed by the moving averages of sensor recordings observed at previous time steps (Rutten et al., 2017). Yet, in case of large periods with missing data, this approach will also not be able to impute the missing values in a reliable way. Furthermore, all of the previously mentioned studies were conducted on datasets with a limited number of recorded calvings, mostly coming from one herd. Hence, the reported performance scores of these models were obtained on very limited test observations and are thus difficult to generalize towards calving events not observed in the data.

In order to fill this gap in literature, we present a framework that is generalizable and suitable for practical implementations. More specifically, this study was conducted on a large dataset containing sensor data coming from 2363 animals from 8 different herds (Hut et al., 2021). Additionally, we propose a novel methodology to infer missing values by leveraging the values recorded by all the sensors. Finally, we present a model that accurately predicts the moment of calving by sequentially processing the multivariate sensor sequences. There are several reasons why we believe that the proposed framework can be valuable for calving management. First, human supervision for calving can be organized better as farmers are automatically alerted when a cow goes into labor. This way, stock personnel does not need to permanently supervise their cattle. Second, cow welfare can be drastically increased as timely supervision significantly reduces the negative consequences of dystocia (Borchers et al., 2017; Schuenemann et al., 2011; Schuenemann et al., 2013; Szenci et al., 2012). Finally, we propose a model that generates reliable predictions, irrespective of the data quality of the recorded sensor data. This is a valuable tool as missing values and outliers frequently occur in sensor sequences due to faulty data transmission or malfunction of the sensors.

2. Materials and Methods

2.1. Data

For this study, data was collected from 2363 cows coming from 8 commercial dairy farms with freestall barns in the Netherlands between August 2016 and November 2020. No external personnel was employed by the farms. From the 8 farms, 6 farms were Holstein Friesian, 1 were Fleckvieh and 1 farm were crossbreeding Holstein Friesian, Fleckvieh and Scandinavian Red. From the moment the Nedap infrasture (Nedap, Groenlo, The Netherlands) was completely implemented at a farm, each cow was equipped with the Nedap Smarttag Leg and Nedap Smarttag Neck sensor for the entire period of this study. The sensors were attached to the front legs and the neck of the cow with the former recording the number of steps, standing time, walking time and lying time and the latter recording the eating time, rumination time and inactive time, i.e., time not spend eating and ruminating. Sensor data was recorded every minute. Hourly as well as daily measurements were obtained by summing all the values of each activity recorded during each hour and day respectively. For the data aggregated on a daily basis, the data supplier provided some additional features, e.g., the number of bouts, the average bout length as well as the average length between different

Table 2
Overview of data used in this study.

	Hourly Prediction	Daily Prediction
Calving events	572	3902
Farms	8	8
Parity 1	110	782
Parity 2	148	927
Parity 3+	314	2193
Recording interval	1 h	24 h
Sliding window size	24 h	14d
Number of sequences	8275	31216
Features	7	19
Training size	4896	18728
Validation size	1721	6240
Test size	1658	6248

bouts for several activities. Additionally, the parity, i.e. the number of different times a dam has had an offspring, and the season of calving (summer, spring, autumn, winter) were provided for each calving event. Table 1 shows the raw sensor recordings as well as the derived features on an hourly and daily basis obtained from the data provider.

The moment of calving was manually recorded by the farmer. In total, the day of calving was registered for 3902 different calvings. For 572 of these calvings, the exact timestamp was registered by the farmer at the moment the farmer visually observed the parturition. In total, 159 calvings were registered in the morning (from 6am to 12 pm), 170 in the afternoon (from 12 pm to 6 pm), 178 in the evening (from 6 pm to 12am) and 65 at night (from 12am to 6am). For each calving event, the daily features observed during the 21 days before calving were extracted. In order to extract the features and labels, a sliding window of 14 days was shifted over the sequences by one day. This resulted in 8 observations for each calving event, with one observation containing the sensor sequences observed the day before calving, and 7 observations with sensor sequences observed 2 or more days before calving. For calving events with an exact time stamp, the hourly sensor values were extracted from the day before calving until the moment of parturition. A sliding window of 24 h was then shifted over the sequences by one hour. In total, 31216 sequences of daily data X^d and 8275 sequences of hourly data X^h were extracted. For the sensor data aggregated on a daily basis, each observation X_i^d contained 14 recordings $X_{i_t}^d$, with each recording comprising 19 sensor values. For the sensor data recorded on an hourly basis, every observation X_i^h consisted of 24 recordings $X_{i_t}^h$ of 7 sensor values. Outliers were removed by the median absolute deviation method (Leys et al., 2013). This method consists of removing observations according to the absolute difference between the observation and the median value. Hence, it is more robust for extreme outliers. In total, 4783 outlying sensor recordings were replaced by missing values, which comprises 0.3% of the data. After the removal of the outliers, sensor values were normalized between 0 and 1. Finally, a training, validation and test set was constructed by randomly sampling 60%, 20% and 20% of the observations respectively. Table 2 gives an overview of the data used in this study.

2.2. Deep Learning Models

Mutlilayer Perceptron Models (MLP) are a type of neural networks and consist of an input layer, one ore more hidden layers and an output layer. In each hidden layer, every neuron is a linear combination of all the neurons from the previous layer, followed by a non-linear activation function. In particular, if h_j represents the outputs of layer j, then the output of layer $j+1$ can be calculated as follows:

$$h_{j+1} = f(h_j \cdot W_{j+1} + b_{j+1})$$

with W_{j+1} and b_{j+1} being the weights matrix and biases corresponding to layer $j+1$, and f being a non-linear activation function, commonly a ReLU function. The activation function of the final layer is generally a sigmoid, softmax or identity function, depending on whether the label is binary, multiclass or continuous respectively. In general, MLPs are suitable for any supervised learning task. In practice, however, MLPs are rarely used when temporal or spatial dependencies exist among the features of the input data. Long Short-Term Memory Models (LSTM) on the other hand, have been specifically designed to process time-series data as they have recurrent connections between the different inputs (Hochreiter and Schmidhuber, 1997). In particular, information from each time step t is fed to an LSTM unit, which is composed out of four units: a memory cell c_t, an input gate i_t with the corresponding weight matrices W_R^i, W_I^i and b^i, an output gate o_t with the corresponding weight matrices W_R^o, W_I^o and b^o and a forget gate f_t with the corresponding weight matrices W_R^f, W_I^f and b^f, as shown by the following equations:

$$\begin{aligned}
i_t &= \sigma\left(W_R^i h_{t-1} + W_I^i x_t + b^i\right)\\
f_t &= \sigma\left(W_R^f h_{t-1} + W_I^f x_t + b^f\right)\\
c_t &= c_{t-1} \odot f_t + i_t \odot tanh\left(W_R^c h_{t-1} + W_I^c x_t + b^c\right)\\
o_t &= \left(W_R^o h_{t-1} + W_I^o x_t + b^o\right)\\
h_t &= o_t \odot tanh(c_t)
\end{aligned}$$

where x_t represents the observed features at the current time step, h_{t-1} represents the output of the previous time step, c_{t-1} represents the cell state of the previous time step and σ and *tanh* represent the sigmoid and hyperbolic tangent function respectively. The memory cell c_t stores information extracted from the previous time steps and the gates determine the information flow between the cells. The output at the last time step represents a compact summary of the entire observed sequence. Sometimes, however, one LSTM layer is insufficient to compress all the observed data into one single feature vector. In such cases, more informative vectors can be obtained by stacking multiple LSTM layers on top of each other. In contrast to LSTMs, Convolutional Neural Networks (CNN) were originally developed for computer vision applications (LeCun et al., 1998; Krizhevsky et al., 2012; Szegedy et al., 2015). Lately, however, they have also shown great performance on time series data as they can extract time-dependent features in parallel (Zhao et al., 2017). In general, a CNN exists of multiple convolutional blocks, with each block typically comprising a linear transformation and a non-linear activation stage for feature extraction. In particular, for a time series with K features and T time steps, a filter of size KxS with $S < T$ is slided over the sequential data along the time dimension. Each time the filter is shifted one position, the filter weights are multiplied with the elements of the data that are covered by the filter at that point. Subsequently, a non-linear activation function, such as ReLU, is applied to the sum of the outputs of the multiplication and results in a new time series of the features extracted by that filter. In order to downsample the output and to make the model invariant to small translations in the input, a pooling stage or strided convolution is used at some of the layers to summarize the presence of the feature in every specific time window. By applying multiple convolutional blocks and flattening the output of the last layer, a vector is obtained representing all the features extracted from the input data. By altering the number of filters of the filter size, or by adding convolutional blocks, the model can learn more complex patterns. Finally, hybrid approaches are now also used to leverage the unique capabilities of different models. C-LSTM models that combine CNNs with LSTMs for example, have been successfully used to process time-series data (Alhussein et al., 2020; Pak et al., 2018). In these architectures, a CNN first extracts a set of time-dependent features from the input data, while an LSTM then sequentially processes these features and encodes them into a one-dimensional feature vector. The motivation to use this kind of architecture is that the CNN is able to extract meaningful features in parallel from the multivariate timeseries, while the LSTM can extract temporal patterns from long-term sequences.

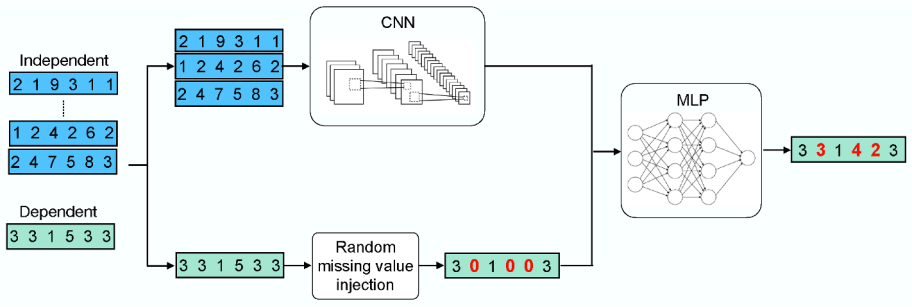


Fig. 1. Schematic overview of the missing value imputation model.

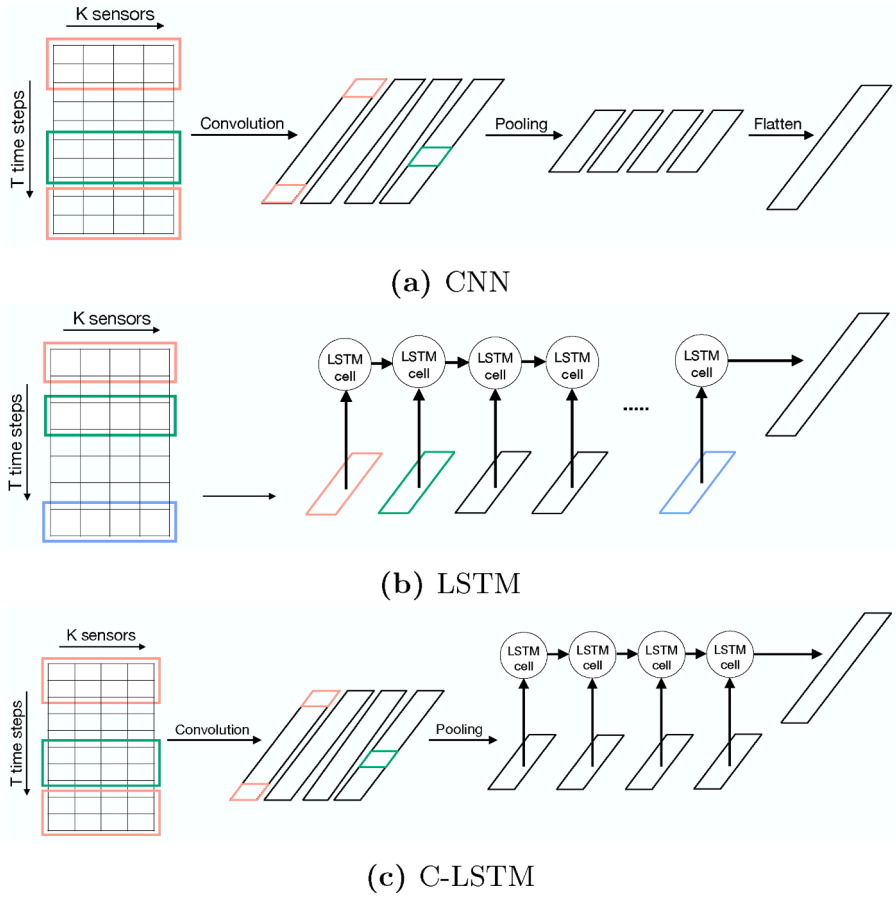


Fig. 2. Deep learning architectures used in this study.

2.3. Missing Value Imputation

A major concern regarding the data quality of sensor data is the frequent occurrence of missing values. Missing gaps in sensor sequences can occur due to several reasons such as malfunction of the sensors and faulty transmission of data. One way of dealing with missing values present in the data sequences is by imputation by the mean, whereby the missing values of a certain feature are replaced by the feature mean. For time series data, however, this often results in unrealistic realizations of sequences, as the imputed value does not take into account the values observed before or after the missing value (Liseune et al., 2020). In contrast, linear and spline interpolation impute missing values by interpolating between known data points. While the linear interpolant equals a straight line between two known points, the spline interpolant is a piecewice polynomial fitted to a small subset of known values. However, in case of a multivariate time series, correlation may exist among the different sequential features, which can not be leveraged by linear or spline interpolation. Hence, in addition to the three previously mentioned imputation methods, a model was also built to impute the missing values for each of the behavioral sequences (e.g. eating) based on the values observed in that behavioral sequence as well as the values recorded in the other behavioral sequences, hereinafter referred to as the dependent and independent sequential features. More specifically, a CNN was used to obtain a one-dimensional vector from the independent sequential features. In order to leverage the values that were observed in the dependent sequential feature, the sequence was used as input as well. During the training stage, observed values of the dependent sequential feature were randomly set to missing to obtain a set of missing and true values. This vector was then concatenated with the CNNs output and was subsequently fed to an MLP which predicted the entire dependent sequential feature. Finally, the mean squared error loss between the values of the dependent sequential feature that were set to

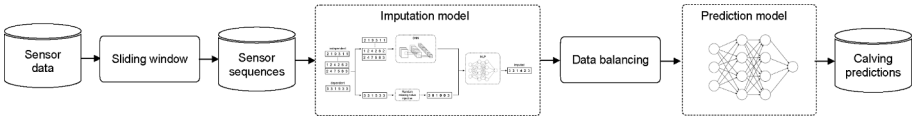


Fig. 3. Overview of methodology.

Table 3
Performance in terms of the AP of the models on test set for the different prediction windows.

Prediction window	Non-Imputed					Imputed				
	LR	RF	LSTM	CNN	C-LSTM	LR	RF	LSTM	CNN	C-LSTM
24 h	0.32	0.65	0.72	0.75	0.73	0.52	0.76	0.79	0.77	0.78
12 h	0.89	0.89	0.89	0.90	0.88	0.86	0.89	0.90	0.90	0.89
6 h	0.65	0.68	0.64	0.66	0.66	0.62	0.68	0.64	0.65	0.68
3 h	0.41	0.46	0.41	0.47	0.44	0.41	0.46	0.44	0.49	0.44
1 h	0.18	0.21	0.21	0.24	0.19	0.19	0.21	0.23	0.24	0.29

Table 4
Performance of the best performing models in terms of Sensitivity, Specificity and Precision for different thresholds.

Prediction Window	Threshold	Sensitivity	Precision	Specificity
24 h	0.8	0.65	0.77	0.97
	0.5	0.79	0.53	0.90
	0.3	0.87	0.40	0.81
	0.1	0.93	0.28	0.67
12 h	0.8	0.57	0.89	0.79
	0.5	0.89	0.81	0.39
	0.3	0.98	0.77	0.15
	0.1	1.0	0.75	0.01
6 h	0.8	0.43	0.66	0.85
	0.5	0.77	0.58	0.63
	0.3	0.91	0.52	0.43
	0.1	1.00	0.42	0.09
3 h	0.8	0.12	0.67	0.99
	0.5	0.57	0.49	0.85
	0.3	0.80	0.37	0.65
	0.1	0.95	0.26	0.32
1 h	0.8	0.30	0.31	0.95
	0.5	0.66	0.16	0.75
	0.3	0.88	0.13	0.55
	0.1	0.99	0.09	0.21

Table 5
Performance of the C-LSTM model for the different imputation techniques in terms of the AP.

Prediction Window	Mean Imputation	Linear Interpolation	Spline Interpolation	Model Imputation
24 h	0.16	0.27	0.36	0.78
12 h	0.81	0.82	0.86	0.89
6 h	0.56	0.59	0.50	0.68
3 h	0.31	0.29	0.27	0.44
1 h	0.09	0.11	0.10	0.29

missing and the corresponding predictions was calculated and back-propagated through the entire network. For each feature of the daily and hourly data, a missing imputation model was trained and was used to impute all the missing values. An example of how one particular sequential feature is imputed by a missing value imputation model is given in Fig. 1.

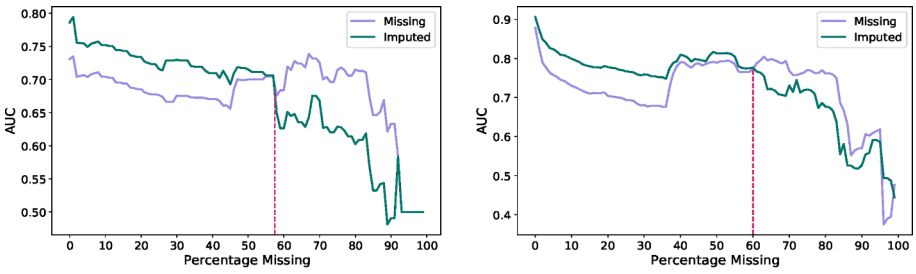
2.4. Predictive Models

In order to predict the moment of calving, two machine learning models and three deep learning models were trained on the sensor data. For predicting the moment of parturition within 24 h, the sensor data aggregated on a daily basis X^d was used as input. The hourly sensor data X^h was used to predict calving within 12 h, 6 h, 3 h and 1 h. For the machine learning models, the data was flattened to obtain non-sequential observations. In particular, for the daily data X^d, each observation X_i^d was flattened by concatenating each of the 14 recordings $X_{i_t}^d$ of 19 sensor values into a one-dimensional vector: $X_{i_{1_1}}^d, X_{i_{1_2}}^d, \ldots, X_{i_{14_{19}}}^d$. Likewise, the hourly data was flattened by concatenating each observation's recording into a one-dimensional vector: $X_{i_{1_1}}^h, X_{i_{1_2}}^h, \ldots, X_{i_{24_7}}^h$. For every prediction window, each model was trained on the imputed data as well as the raw data with the missing values. Like most of the previous studies, a logistic regression model was trained on the flattened daily and hourly sensor data as this model is not able to sequentially process the input features. For the daily and hourly predictions, the logistic regression model can be expressed as follows:

$$y_i^d = \frac{1}{1+\exp^{(-\beta_0+\beta_1 * X_{i_{1_1}}^d+\beta_2 * X_{i_{1_2}}^d+\ldots+\beta_{266} * X_{i_{14_{19}}}^d)}}$$

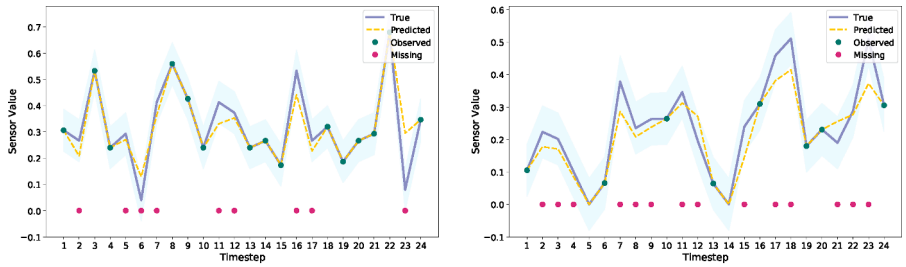
$$y_i^h = \frac{1}{1+\exp^{(-\beta_0+\beta_1 * X_{i_{1_1}}^h+\beta_2 * X_{i_{1_2}}^h+\ldots+\beta_{168} * X_{i_{24_7}}^h)}}$$

with y_i^d being the predicted probability of calving the next day for observation i of the daily data and y_i^h being the predicted probability of calving the next 1 h, 3 h, 6 h or 12 h for observation i of the hourly data. Additionally, a random forest model was trained on the flattened data as this model does not assume a linear decision boundary, unlike the logistic regression model. In contrast to the machine learning models, three deep learning models that are able to sequentially process the sensor data were used to predict calving. A CNN model was implemented by applying multiple convolutional layers on the time series data. In each layer, several filters were shifted along the time dimension to extract different sets of time-dependent features. Pooling stages were used to downsample the feature space. The output of the last layer was flattened to obtain a vector comprising all the extracted features. For the LSTM model, the sensor values observed at each time step were processed sequentially by one or two LSTM layers. The output of the last LSTM cell was used as a compact summary of all the observed sensor sequences. Finally, the C-LSTM model comprised a CNN and LSTM unit, with the CNN extracting several time-dependent feature vectors by applying multiple convolutional blocks, and the LSTM processing these features sequentially and obtaining a compact feature representation. The feature representations obtained by the LSTM, CNN and C-LSTM models were passed to an MLP with a sigmoid activation function in the final layer to predict the probability of calving. An overview of the three deep learning models applied in this study are shown by Fig. 2



(a) AUC C-LSTM 1h (b) AUC C-LSTM 24h

Fig. 4. AUC of the C-LSTM models trained on the data with missing values as well as imputations on subsets of test observations with increasing amounts of missing values. Purple solid line = AUC of model trained on data with missing values, Green solid line = AUC of model trained on data with imputed missing values.



(a) Inactivity (b) Leg Activity

Fig. 5. Visualization of the imputation model's predictions of missing sensor values for two random sensor sequences of the test set. Blue solid line = true sensor sequence, Pink dots = sensor values randomly set to missing, Green dots = observed sensor values, Orange dashed line = predicted sensor values.

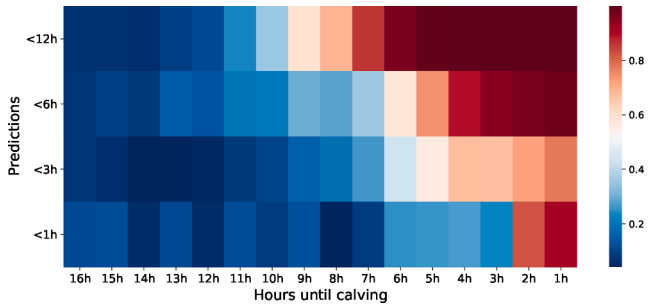


Fig. 6. The predicted probabilities of the 12 h, 6 h, 3 h and 1 h calving models for different time periods until calving.

2.5. Model Training

All the deep learning models were trained by using the back-propagation algorithm (Rumelhart et al., 1986). In this algorithm, the gradient with respect to loss is calculated and propagated through the network by using the chain rule. The Adam gradient-based optimization algorithm was then used to update the weights (Kingma and Ba, 2014). For the missing value imputation model, the mean squared error between the values of the dependent sensor that were randomly set to missing and the corresponding predictions was calculated and back-propagated through the entire network. The negative log-likelihood between the predicted probabilities and the true calving observations was used to train the prediction models. All the models were trained on the training set by using the early stopping procedure in which model training continues as long as the performance on the validation set improves in order to avoid overfitting. The hyperparameters of all the prediction models were tuned using a random search. More specifically, for each training cycle of a model, a hyperparameter setting was determined by randomly sampling values from the model's predefined hyperparameter space. After a predefined number of training cycles, the optimal hyperparameter setting was determined by obtaining the model with the highest validation performance. For the logistic regression model, the regularization method and strength as well as the number of training iterations were optimized. The number and depth of trees as well as the number of samples required to split an internal node and to be at a leaf node were set as hyperparameters for the random forest model. For the deep learning models, the number of layers, the number of neurons in each layer, the activation function, the dropout rate as well as the inclusion of batch normalization were all considered as tunable parameters. Finally, for every prediction model, the inclusion of the static data features, i.e. the parity and season of calving, as well as the balancing scheme was considered as a hyperparameter as well. In particular, the data could be upsampled or downsampled, the loss function could be weighted with respect to the class proportions or no adjustment could be made to the data. An overview of all the hyperparameters that were assessed for each of the different models is provided in A. In Fig. 3, a schematic overview of the methodology for each prediction model is given.

2.6. Model Evaluation

The performance of the prediction models was evaluated by five metrics that are widely accepted as appropriate evaluation metrics for binary classification algorithms, namely the AUC (Area Under ROC Curve), the Sensitivity (Se), the Precision or Positive Predicted Value (PPV), the Specificity (Sp) and the Average Precision (AP). The Sensitivity equals the proportion of correctly classified positive examples. The Precision measures how much positive examples were retrieved from the positive predictions. The Specificity calculates the proportion of negative examples that were identified by the model. In contrast to the aforementioned evaluation metrics, the AUC and AP are not dependent on a specific threshold, i.e. the cutoff point above which a predicted probability is considered as a positive prediction and a negative prediction otherwise. Hence, these metrics allow to compare how well models are ordering the predictions, without considering any specific

decision threshold. The AUC can be interpreted as the probability that a random positive observation gets a higher score than a random negative observation (Schetgen et al., 2021). An AUC score of 0.5 represents a model that does not perform any better than random, while an AUC score of 1 is obtained by a perfect model. In case of imbalanced data, however, it has been shown that the AP is more informative than the AUC when evaluating binary classification models (Saito and Rehmsmeier, 2015). The AP is the area under the precision recall curve (PRC) and indicates how well the model can correctly identify all the positive examples without predicting too much negative examples as positive. A random classifier has an AP equal to the proportion of positive examples while a perfect model has an AP equal to 1.

2.7. Model Selection

Each missing value imputation model comprised 5 convolutional layers, with 32, 64, 64, 128 and 128 filters respectively. In each layer, a filter size of 3 and a ReLU activation function was applied. The output of the second and fourth layer was downsampled by applying a stride of 2. The output of the last layer was flattened and passed to an MLP with one hidden layer of size 100 and a ReLU activation function. Each imputation model was evaluated in terms of the RMSE on the validation set every 5000 training iterations with a batch size of 32. Every time the validation RMSE decreased, the model's weights were saved. Training was terminated when the performance did not improve for 5 consecutive times. For each of the predictive models, 50 random hyperparameter configurations were assessed. After convergence on the training set, the AP of the machine learning models on the validation set was calculated. Every deep learning model was evaluated on the validation set in terms of the AP after 1 training epoch. Model weights were saved when the AP on the validation set increased. Training was terminated when the validation AP did not increase for 5 consecutive times. For each predictive model, the parameter configuration that rendered the highest validation performance was retrained on the combination of the training and validation set and was evaluated on the test set. The learning rate applied for the Adam optimization algorithm was 0.001 for both the missing value imputation as well as the deep learning predictive models.

2.8. Programming Tools

All data processing and analyses were done in Python 3.9 (Python Software Foundation, https://www.python.org/) with the add-on packages Pandas (pandas development team, 2020) and NumPy (Harris et al., 2020) for data preprocessing, scikit-learn (Pedregosa et al., 2011) for machine learning modeling and model evaluation, TensorFlow (Abadi et al., 2015) and Keras (Chollet, 2015) for deep learning modeling and Matplotlib (Hunter, 2007) and seaborn (Waskom, 2021) for data visualization.

3. Results

3.1. Model Performance

The performance of the models in terms of the AP on the test set for each prediction window is presented in Table 3. For the daily predictions, the deep learning models clearly outperformed the machine learning models on the data with missing values. While the logistic regression and random forest model achieved an AP of 0.32 and 0.65, the LSTM, CNN and C-LSTM obtained AP scores of 0.72, 0.75 and 0.73 respectively. The highest scores, however, were obtained on the imputed data. While the performance of the CNN and C-LSTM increased by 3% and 7% respectively, the AP of the LSTM increased by 12.5% to 0.79, resulting in the best performance on the daily predictions. Likewise, the performance of the LSTM and C-LSTM improved considerably when trained on the imputed data for the smallest prediction window. More specifically, the LSTM's performance increased by 0.02 when trained on the imputed data. The C-LSTM on the contrary, improved its performance from 0.19 to 0.29, which resulted in the highest score for the 1 h prediction interval. The added value of the imputations with respect to the predictive performance is also visible for the other prediction intervals. The CNN trained on the imputed data obtained the highest performance scores for the 3 h prediction window with an AP equal to 0.49, thereby outscoring the best performing model trained on the missing data with 0.02. For the 6 h prediction interval, the best performance was obtained by the C-LSTM and random forest model trained on the imputed data as well as the random forest model trained on the data with missing values.

In contrast to the AP, the Se, Sp and PPV are dependent on the chosen threshold. For each prediction interval, the values of these metrics are therefore shown for different thresholds in Table 4 for the best performing model, i.e., the C-LSTM model trained on the imputed data for the 6 h and 1 h prediction interval, the CNN model trained on the imputed data for the 12 h and 3 h prediction interval and the LSTM model trained on the imputed data for the 24 h prediction interval. As expected, the Se increases for lower thresholds, as more observations are classified as positive. Yet, as more observations are predicted as being positive, the number of false positives will increase as well, hence resulting in lower levels of Precision. For a threshold equal to 0.8, the daily prediction model is able to detect 65% of calvings that will occur in 24 h with approximately 77% of the positive predictions being correct. When the threshold is lowered to 0.3, almost 90% of all calving events are identified but with a lower PPV being equal to 0.4. For the same threshold, the model predicting a calving event within 1 h is comparable to the model predicting the moment of calving within 24 h in terms of the Se. The PPV of the 1 h model, however, is 0.13 and, therefore, considerably lower than that of the 24 h model.

Furthermore, Table 5 shows the performance of the C-LSTM model for the different imputation strategies in terms of the AP. Regarding the traditional imputation methods, the spline interpolation renders the highest performance for the 24 h and 12 h prediction interval, with an AP equal to 0.36 and 0.86 respectively. Imputations made by linear interpolation on the contrary, achieve the highest results for the 6 h and 1 h interval, with AP scores equal to 0.59 and 0.11 respectively. Yet, for every prediction window, using the imputations inferred by the deep learning model clearly results in better performance with respect to predicting the moment of calving than using the imputations made by the more traditional imputation methods. In particular, for the 1 h interval, the C-LSTM model trained on the model imputations outperforms the C-LSTM model trained on the imputations made by linear interpolations by 0.18. For the 24 h interval, the C-LSTM model leveraging the model imputations even outperforms the best performing model using a traditional imputation method by 0.42. Additionally, the performance of the models trained on the data with missing values as well as the missing values imputed by the imputation model is visualized in more detail in Fig. 4. More specifically, the AUC of the C-LSTM model trained on the missing and imputed data for the smallest and largest prediction interval are compared for different subsets of test observations comprising a minimum percentage of missing values. As expected, the AUC of the models decreased when more missing values were present in the observations. For observations with at least 20% of the sensor values missing, the AUC of both models for the 1 h prediction interval decreased by 0.05 compared to the AUC obtained on observations with no missing values. For the 24 h prediction interval, the AUC of the model trained with missing data decreased by 0.17 while the AUC of the model trained on the data with imputations only decreased by 0.13. Yet, while the performance of all the models steadily decreased for increasing amounts of missing values, the models trained on the imputed data clearly outperformed the models trained on data with missing values for observations with a tolerable number of missing values. In particular, for sensor sequences with at least 30% of the values missing, the model leveraging the imputations scored an AUC of 0.73 and 0.76 on the 1 h and 24 h prediction interval respectively. In contrast, the models that

didn't have access to the imputations obtained an AUC score of 0.68 for the same subset on both prediction intervals. However, when approximately 60% or more of the sensor values were missing, the AUC of the models trained on the imputed data started to decrease rapidly, resulting in higher performance scores obtained by the models trained on the missing data. This could be explained by the fact that for these observations, the imputations only rely on a small subset of recorded values, hence resulting in less qualitative estimations. Yet, for more reasonable amounts of missing data, imputations are far more precise and therefore the resulting calving predictions as well.

An example of how the imputation model infers the missing values of two different sensor sequences recorded on an hourly basis is visualized in Fig. 5. For the sensor sequence measuring Inactivity, 10 of the 24 values were randomly set to missing. By observing the remaining Inactivity values as well as the sequences representing the 6 other behavioral activities, the imputation model is capable of accurately approximating the true sensor values. For time step 6 for example, the model correctly infers a strong decrease in Inactivity, before increasing back to a local maximum. For time step 11 and 23, the model also correctly identifies the true direction of the sensor activity, yet slightly underestimates the true increase and decline of the sensor values. For the sequence representing Leg Activity behavior, 14 values were randomly set to missing. Again, the imputation model is able to correctly infer the direction of the sensor values for most of the time steps. From time step 2 to 5, the model rightly predicts a slight increase followed by strong decrease. Likewise, the model is able to detect an increase in sensor values for time steps 7, 11, 18 and 23. For time step 21 however, the model assumes an increase in Leg Activity behavior while a decrease in sensor activity was truly observed.

Finally, an example of how the hourly calving models change their predicted probabilities according to the time until calving is shown by Fig. 6. For one animal, the probabilities are generated by the models by observing the sequence of sliding windows of sensor data before calving. As the moment of calving approaches, the predicted probabilities of the 4 models increase. For the model trained to predict calving within 12 h, the probabilities become considerably larger than 0.5 when calving starts in 9 h. The 6 h model on the other hand, only starts generating probabilities larger than 0.5 when the moment of parturition is in 6 h or less. The predictions made by the 3 h model start to increase rapidly when calving approaches within 4 h, while the 1 h model only predicts probabilities larger than 0.5 when calving starts within 2 h.

4. Discussion

The results depicted in Table 3 clearly indicate that the deep learning models, which are able to leverage the sequential patterns in the sensor data, perform better than the more traditional machine learning models, which used the flattened sensor data as input. Except for the 6 h prediction interval, the highest AP was always obtained by one of the deep learning algorithms, irrespective of the data preprocessing method. This indicates that the temporal patterns in the sequences of sensor data contain valuable clues regarding the moment of calving. Traditional machine learning models are not able to leverage sequential information as they do not process the time series in a sequential fashion.

Furthermore, it is also clear from Table 3 that the models trained on the data imputed by the imputation model predict calving more accurately than when the data with missing values was used. For every prediction window, the best performing model trained on the data imputed by the deep learning model performed as well or better than the best performing model trained on the missing data. For predicting calving within 24 h and 1 h, the missing value imputations had the largest impact, with an increase of 0.04 and 0.05 in terms of AP respectively. Additionally, Table 5 shows how the predictions with respect to the moment of calving were considerably more accurate for every prediction window by using the imputations made by the deep learning model than by using the imputations made by the more traditional imputation methods. Moreover, the results from Tables 3 and 5 indicate that imputation by the mean, linear interpolation or spline interpolation even harm performance, as the C-LSTM model trained on the non-imputed data obtains higher AP scores for every prediction window. This can be attributed to the fact that entire gaps of missing values are more present in the sequences than single missing data points, which in turn may be the result of sensors not transmitting data for a certain period, rather than a single moment. For such large gaps of missing data, the imputations made by the more traditional imputation methods will likely be unrealistic. In particular, imputations generated by the mean and linear interpolation will lie on a horizontal and linear line respectively, while the imputations generated by spline interpolation will lie on a parabolic line. The deep learning imputation model, however, is able to leverage all the information available in the data, including the observed values of other features, and is able to generate more complex patterns for the gaps of missing data. The added value of the imputations was also visualized by Fig. 4. In particular, it was demonstrated that for observations with reasonable amounts of missing values, the models trained on the imputed data perform consistently better than the models trained on the missing data. For test observations with 1% to approximately 60% of missing values, the 1 h as well as the 24 h predictions were more accurate when the missing values were imputed. However, for more missing values, the accuracy of the imputations starts to decrease rapidly and hence results in even worse performance than using the raw data as input. These results suggest that as long as no more than half of the data is missing, using intelligent imputation methods can considerably increase the predictive performance to predict the moment of calving.

In order to investigate the added value of using sequential deep learning models for imputation as well as prediction in further detail, the results from this study are compared with the results obtained by similar studies. In the study presented by Rutten et al., 2017, a logistic regression model that used the relative differences in sensor values to predict calving within 1 h obtained a Se of 0.21 with a PPV of 0.05. For approximately the same level of Precision, the Naive Bayes model presented by Zehner et al., 2019 obtained a much higher Se of 0.82. In this study, the C-LSTM trained on the imputed data was able to detect more positive calving events at a higher precision. More specifically, for a threshold of 0.3, 88% of the true calving events were detected with a PPV of 0.13, while for a threshold of 0.1, the model was able to detect 99% of positive cases with a PPV of 0.09. The logistic regression model was also used by Rutten et al., 2017 to predict the start of calving within 3 h. For this prediction interval, they reported a Se and a PPV of 0.42 and 0.09 respectively. In this study, a Se of 0.95 with a PPV equal to 0.26 was achieved by the CNN trained on the imputed data, given a threshold of 0.1. A logistic regression model using the relative changes in sensor values was also proposed by Fadul et al., 2017 to predict calving for a 3 h interval. For multiparous cows, they reported a Se of 0.85 in correspondence to a Sp of 0.74. In this research, a similar Se of 0.8 was obtained for a slightly lower level of Sp equal to 0.65, given a threshold of 0.3. Yet, the results presented by Fadul et al., 2017 were obtained on the same 9 observations which were used to fit the model parameters and therefore could be biased. For predicting the start of calving within 6 and 12 h, a logistic regression model was also used by Rutten et al., 2017 and Ouellet et al., 2016. For the 6 h predictions interval, Ouellet et al., 2016 reported a Se of 0.71 and a PPV of 0.17, while for the 12 h interval, a Se and PPV of 0.7 and 0.3 were obtained. These results, however, should be interpreted with caution as they were also obtained on the same 33 calving events used to train the model. A better comparison can therefore be made with the 6 h and 12 h models presented by Rutten et al., 2017, as they did use a separate test set to evaluate the predictive performance. For the 6 h prediction interval, the model proposed by Rutten et al., 2017 obtained a Se and PPV of 0.49 and 0.11 respectively, while for a window of 12 h, a Se of 0.51 and a PPV of 0.13 was reported. For a threshold of 0.8, the 6 h model proposed in this study achieved a similar level of Se of 0.43, but at a much higher PPV, i.e. 0.66. Likewise,

the CNN model trained on the imputed data was much more accurate in predicting the start calving within 12 h. In particular, 89% of the positive cases could be detected at a predictive accuracy of 81% for a threshold of 0.5. Finally, the model proposed in this study that predicted calving within 24 h obtained a Se of 0.65 with a PPV of 0.77 given a threshold of 0.8. The model proposed by Rutten et al., 2017 obtained lower values for both the Se as well as PPV, namely a Se of 0.36 and a PPV of 0.6. Borchers et al., 2017 on the other hand, presented an MLP that was able to detect every single positive calving event with a PPV of 0.4. Better performance scores were even reported by Keceli et al., 2020 who used an LSTM architecture on the same dataset. In particular, they reported a Se and PPV of 1.0. However, while the results in this study are obtained on a test set comprising 115 calvings coming from 8 different herds, the results reported by Borchers et al., 2017 and Keceli et al., 2020 were obtained on only 10 calving events coming from the same herd. Additionally, while in this research the results are obtained on test observations containing missing values, observations with missing values were removed from the analysis conducted by the two aforementioned studies. This is also true for the frameworks proposed by Fadul et al., 2017 and Ouellet et al., 2016. In practice, however, sensor sequences often contain missing values. The prediction errors reported by these studies will therefore be underestimates of the true errors obtained on new observations containing missing values. Finally, the models presented by Borchers et al., 2017 and Keceli et al., 2020 are categorical classification algorithms that predict the number of days until calving during the two weeks preceding calving. In order to predict the number of days until calving, a fixed window of 14 days of observed data was used as feature by Keceli et al., 2020. As a result, the model can correctly predict the number of days until calving by solely counting the number of available features, regardless of the values of these features. For unseen calving events, however, the days until calving are unknown and therefore also the number of features. As a result, it is much more difficult to generalize these results towards other calving events than the results obtained by the present study.

5. Conclusion

Dystocia is a major problem for the dairy cattle industry as it significantly affects the animal welfare as well as the farm economics. Accurately predicting the moment of calving is, therefore, a valuable tool for dairy farmers as it allows them to provide timely supervision. In this study, we propose a framework to predict the moment of calving by using sensor data measuring behavioral activities such as eating, ruminating, walking and lying. The present study shows that leveraging the sequential patterns from the sensor data increases the performance of calving prediction models. More specifically, we show how deep learning models are able to accurately infer missing values by using all the behavioral activities observed by the sensors. In addition to increasing the overall predictive performance, using the missing value imputations also significantly improves the performance on observations containing up to 60% of missing values. Additionally, we show how using sequential deep learning algorithms are better able to predict the moment of parturition than more traditional machine learning algorithms, which are not able to exploit the sequential patterns hidden in the sensor data. In particular, the presented models could detect 65% of the calvings within 24 h with a precision of 77%, while 57% of calvings occurring within 3 h could be identified with a precision equal to 49%. Hence, the framework proposed in this study can be used to enhance calving predictions, and therefore facilitate timely supervision as well as improve animal welfare.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research was supported by the Dutch research program SenseOfSensors, which is a collaboration between Utrecht University (Utrecht, the Netherlands), Wageningen University & Research (Wageningen, the Netherlands), Nedap Livestock Management (Groenlo, the Netherlands), Vetvice (Bergen op Zoom, the Netherlands) and Elsbeth Stassen (Adaptation Physiology Group, Wageningen University, De Elst 1, 6708 WD Wageningen).

Appendix A. Hyperparameters

Table A.6.

Table A.6
Hyperparameters of the models.

Predictive Model	Hyperparameter	Settings
Logistic Regression	Number of iterations	1000, 2000, …, 5000
	Regularization method	None, L1, L2, Elastic Net
	Regularization strength	0.001, 0.01, 0.1, 0.2, 0.5, 1, 10, 100
	Balancing method	None, Downsample, Upsample, Weighted
Random Forest	Number of trees	100, 200, …, 1000
	Maximum depth of a tree	10, 20, …, 100
	Minimum number of samples per split	2, 5, 10
	Minimum number of samples per leave	1, 2, 4
	Maximum features per split	sqrt(number of features)
	Use static features	True, False
	Balancing method	None, Downsample, Upsample, Weighted
LSTM	Number of LSTM layers	1, 2
	Size of hidden state	50, 100, 200
	Activation function	ReLU, Leaky ReLU
	Dropout Rate	0.0, 0.1, …, 0.5
	Use batch normalization	True, False
	Number of MLP layers	0, 1, 2
	Size of MLP layers	50, 100
	Use static features	True, False
	Balancing method	None, Downsample, Upsample, Weighted
CNN	Number of CNN layers	2, 4, 6, 8
	Number of filters	16, 32, 64, 128
	Size of filter	3
	Downsample layer	None, Stride, MaxPool
	Stride or MaxPool size	2
	Activation function	ReLU, Leaky ReLU
	Dropout Rate	0.0, 0.1, …, 0.5
	Use batch normalization	True, False
	Number of MLP layers	0, 1, 2
	Size of MLP layers	50, 100
	Use static features	True, False
	Balancing method	None, Downsample, Upsample, Weighted
C-LSTM	Number of CNN layers	1, 2
	Number of filters	16, 32, 64
	Size of filter	3
	Downsample layer	None, Stride, MaxPool
	Stride or MaxPool size	2
	Number of LSTM layers	1, 2
	Size of hidden state	50, 100
	Activation function	ReLU, Leaky ReLU
	Dropout Rate	0.0, 0.1, …, 0.5
	Use batch normalization	True, False
	Number of MLP layers	0, 1, 2
	Size of MLP layers	50, 100
	Use static features	True, False
	Balancing method	None, Downsample, Upsample, Weighted

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., & Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. https://www.tensorflow.org/ software available from tensorflow.org.

Alhussein, M., Aurangzeb, K., Haider, S.I., 2020. Hybrid CNN-LSTM model for short-term individual household load forecasting. IEEE Access 8, 180544–180557. https://doi.org/10.1109/ACCESS.2020.3028281.

Barrier, A., Haskell, M., 2011. Calving difficulty in dairy cows has a longer effect on saleable milk yield than on estimated milk production. J. Dairy Sci. 94, 1804–1812. https://doi.org/10.3168/jds.2010-3641.

Barrier, A., Haskell, M., Birch, S., Bagnall, A., Bell, D., Dickinson, J., Macrae, A., Dwyer, C., 2013. The impact of dystocia on dairy calf health, welfare, performance and survival. Vet. J. 195, 86–90. https://doi.org/10.1016/j.tvjl.2012.07.031.

Berglund, B., Steinbock, L., Elvander, M., 2003. Causes of stillbirth and time of death in Swedish Holstein calves examined post mortem. Acta Vet. Scand. 44, 111. https://doi.org/10.1186/1751-0147-44-111.

Borchers, M., Chang, Y., Proudfoot, K., Wadsworth, B., Stone, A., Bewley, J., 2017. Machine-learning-based calving prediction from activity, lying, and ruminating behaviors in dairy cattle. J. Dairy Sci. 100, 5664–5674. https://doi.org/10.3168/jds.2016-11526.

Chollet, F. et al. (2015). Keras. https://github.com/fchollet/keras.

Fadul, M., Bogdahn, C., Alsaaod, M., Hüsler, J., Starke, A., Steiner, A., Hirsbrunner, G., 2017. Prediction of calving time in dairy cattle. Animal Reproduction Science 187, 37–46. https://doi.org/10.1016/j.anireprosci.2017.10.003.

Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M.H., Brett, M., Haldane, A., del Río, J.F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., Oliphant, T.E., 2020. Array programming with NumPy. Nature 585, 357–362. https://doi.org/10.1038/s41586-020-2649-2.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural computation 9, 1735–1780.

Hunter, J.D., 2007. Matplotlib: A 2d graphics environment. Computing in Science & Engineering 9, 90–95. https://doi.org/10.1109/MCSE.2007.55.

Hut, P., Hostens, M., Beijaard, M., van Eerdenburg, F., Hulsen, J., Hooijer, G., Stassen, E., Nielen, M., 2021. Associations between body condition score, locomotion score, and sensor-based time budgets of dairy cattle during the dry period and early lactation. J. Dairy Sci. 104, 4746–4763. https://doi.org/10.3168/jds.2020-19200.

Huzzey, J., von Keyserlingk, M., Weary, D., 2005. Changes in feeding, drinking, and standing behavior of dairy cows during the transition period. J. Dairy Sci. 88, 2454–2461. https://doi.org/10.3168/jds.S0022-0302(05)72923-4.

Jensen, M.B., 2012. Behaviour around the time of calving in dairy cows. Applied Animal Behaviour Science 139, 195–202. https://doi.org/10.1016/j.applanim.2012.04.002.

Keceli, A.S., Catal, C., Kaya, A., Tekinerdogan, B., 2020. Development of a recurrent neural networks-based calving prediction model using activity and behavioral data. Computers and Electronics in Agriculture 170, 105285. https://doi.org/10.1016/j.compag.2020.105285.

Kingma, D., Ba, J., 2014. Adam: A method for stochastic optimization. ArXiv e-prints.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems 25, 1097–1105.

Lange, K., Fischer-Tenhagen, C., Heuwieser, W., 2017. Predicting stage 2 of calving in Holstein-Friesian heifers. J. Dairy Sci. 100, 4847–4856. https://doi.org/10.3168/jds.2016-12024.

Laven, R., Huxley, J., Whay, H., Stafford, K., 2009. Results of a survey of attitudes of dairy veterinarians in New Zealand regarding painful procedures and conditions in cattle. New Zealand Veterinary Journal 57, 215–220. https://doi.org/10.1080/00480169.2009.36904.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proc. IEEE 86, 2278–2324.

Leys, C., Ley, C., Klein, O., Bernard, P., Licata, L., 2013. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. J. Exp. Soc. Psychol. 49, 764–766. https://doi.org/10.1016/j.jesp.2013.03.013.

Liseune, A., Salamone, M., Van den Poel, D., Van Ranst, B., Hostens, M., 2020. Leveraging latent representations for milk yield prediction and interpolation using deep learning. Computers and Electronics in Agriculture 175, 105600.

Lombard, J., Garry, F., Tomlinson, S., Garber, L., 2007. Impacts of dystocia on health and survival of dairy calves. J. Dairy Sci. 90, 1751–1760. https://doi.org/10.3168/jds.2006-295.

Mahnani, A., Sadeghi-Sefidmazgi, A., Keshavarzi, H., 2018. Performance and financial consequences of stillbirth in holstein dairy cattle. Animal 12, 617–623. https://doi.org/10.1017/S1751731117002026 https://www.sciencedirect.com/science/article/pii/S1751731117002026.

McGuirk, B.J., Forsyth, R., Dobson, H., 2007. Economic cost of difficult calvings in the United Kingdom dairy herd. Veterinary Record 161, 685–687. https://doi.org/10.1136/vr.161.20.685.

Mee, J.F., 2004. Managing the dairy cow at calving time. Veterinary Clinics of North America: Food Animal Practice 20, 521–546. https://doi.org/10.1016/j.cvfa.2004.06.001.

Mee, J.F., 2013. Why do so many calves die on modern dairy farms and what can we do about calf welfare in the future? Animals 3, 1036–1057. https://doi.org/10.3390/ani3041036 https://www.mdpi.com/2076-2615/3/4/1036.

Meyer, C., Berger, P., Koehler, K., 2000. Interactions among factors affecting stillbirths in Holstein cattle in the United states. J. Dairy Sci. 83, 2657–2663. https://doi.org/10.3168/jds.S0022-0302(00)75159-9.

Miedema, H.M., Cockram, M.S., Dwyer, C.M., Macrae, A.I., 2011. Behavioural predictors of the start of normal and dystocic calving in dairy cows and heifers. Applied Animal Behaviour Science 132, 14–19. https://doi.org/10.1016/j.applanim.2011.03.003.

Ouellet, V., Vasseur, E., Heuwieser, W., Burfeind, O., Maldague, X., Charbonneau, É., 2016. Evaluation of calving indicators measured by automated monitoring devices to predict the onset of calving in Holstein dairy cows. J. Dairy Sci. 99, 1539–1548. https://doi.org/10.3168/jds.2015-10057.

Pak, U., Kim, C., Ryu, U., Sok, K., Pak, S., 2018. A hybrid model based on convolutional neural networks and long short-term memory for ozone concentration prediction. Air Quality, Atmosphere & Health 11, 883–895.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830.

Raussi, S. (2003). Human-cattle interactions in group housing. Applied Animal Behaviour Science, 80, 245–262. doi: 10.1016/S0168-1591(02)00213-7. Behavior and welfare of cattle housed in large groups.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back-propagating errors. Nature 323, 533–536.

Rutten, C., Kamphuis, C., Hogeveen, H., Huijps, K., Nielen, M., Steeneveld, W., 2017. Sensor data on cow activity, rumination, and ear temperature improve prediction of the start of calving in dairy cows. Computers and Electronics in Agriculture 132, 108–118. https://doi.org/10.1016/j.compag.2016.11.009.

Saito, T., Rehmsmeier, M., 2015. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PloS one 10, e0118432.

Schetgen, L., Bogaert, M., Van den Poel, D., 2021. Predicting donation behavior: Acquisition modeling in the nonprofit sector using Facebook data. Decis. Support Syst. 141, 113446. https://doi.org/10.1016/j.dss.2020.113446.

Schirmann, K., Chapinal, N., Weary, D., Vickers, L., von Keyserlingk, M., 2013. Short communication: Rumination and feeding behavior before and after calving in dairy cows. J. Dairy Sci. 96, 7088–7092. https://doi.org/10.3168/jds.2013-7023.

Schuenemann, G., Bas, S., Gordon, E., Workman, J., 2013. Dairy calving management: Description and assessment of a training program for dairy personnel. J. Dairy Sci. 96, 2671–2680. https://doi.org/10.3168/jds.2012-5976.

Schuenemann, G., Nieto, I., Bas, S., Galvão, K., Workman, J., 2011. Assessment of calving progress and reference times for obstetric intervention during dystocia in Holstein dairy cows. J. Dairy Sci. 94, 5494–5501. https://doi.org/10.3168/jds.2011-4436.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–9.

Szenci, O., Nagy, K., Takács, L., Mádl, I., Bajcsy, Á.C., et al., 2012. Farm personnel management as a risk factor for stillbirth in hungarian Holstein-Friesian dairy farms. Magyar Állatorvosok Lapja 134, 387–393.

pandas development team, T. (2020). pandas-dev/pandas: Pandas. URL https://doi.org/10.5281/zenodo.3509134. doi:10.5281/zenodo.3509134.

Tenhagen, B.-A., Helmbold, A., Heuwieser, W., 2007. Effect of various degrees of dystocia in dairy cattle on calf viability, milk production, fertility and culling. J. Vet. Med. Ser. A 54, 98–102. https://doi.org/10.1111/j.1439-0442.2007.00850.x.

Van Pelt, M., de Jong, G., 2011. Genetic evaluation for direct and maternal livability in The Netherlands. Interbull Bulletin.

Waskom, M.L., 2021. seaborn: statistical data visualization. Journal of Open Source Software 6, 3021. https://doi.org/10.21105/joss.03021.

Zehner, N., Niederhauser, J.J., Schick, M., Umstatter, C., 2019. Development and validation of a predictive model for calving time based on sensor measurements of ingestive behavior in dairy cows. Computers and Electronics in Agriculture 161, 62–71. https://doi.org/10.1016/j.compag.2018.08.037. BigData and DSS in Agriculture.

Zhao, B., Lu, H., Chen, S., Liu, J., Wu, D., 2017. Convolutional neural networks for time series classification. Journal of Systems Engineering and Electronics 28, 162–169.