3D acceleration for heat detection in dairy cows

Sebastián R. Vanrell[1,2], José O. Chelotti[1], Julio R. Galli[3], Hugo L. Rufiner[1,2], and Diego H. Milone[1,2]

[1] sinc(i) – Centro de Investigación en Señales, Sistemas e Inteligencia Computacional
Facultad de Ingeniería y Ciencias Hídricas – Universidad Nacional del Litoral
[2] CONICET – Consejo Nacional de Investigaciones Científicas y Técnicas
[3] Facultad de Ciencias Agrarias – Universidad Nacional de Rosario

Abstract. Accurate and reliable detection of heat in dairy cows is essential for a controlled reproduction and therefore, for maintaining milk production. Classical approaches like visual identification are no longer viable on large dairy herds. Several automated techniques of detection have been proposed, but expected results are only achieved by expensive or invasive methods, because practical methods are not reliable. We present a method that aims to be both practical and accurate. It is based on simple attributes extracted from 3D acceleration data and well known classifiers: multilayer perceptrons, support vector machines and decision trees. Results show promising detection ratios, above 90% in several configurations of the detection system. Best results are achieved with multilayer perceptrons. This information could be readily incorporated to the automated system in a dairy farm and help to improve its efficiency.

Keywords: estrus recognition, dairy cattle, binary classification, multilayer perceptron, support vector machine, decision tree, accelerometer

1   Introduction

Dairy cows must calve on intervals of 12-14 months to maximize average daily milk production on their lifetime. A deficient detection of heat (estrus) is one of the main factors that negatively affects calving intervals and reproductive management. Timely insemination is crucial for reproduction and heat detection is the key to the successful use of artificial insemination [1]. The most common cause of a poor detection of heat is failure to watch cows for long enough periods. Cows should be watched at least three times a day, for a period of 20-30 minutes. Thus, as monitoring technology progress, several systems for automated heat detection has been proposed [2]. However, current methods show satisfactory results only for expensive measurements, while simpler methods present high error rates and none of them is completely automatic.

Cows go into heat with an average cycle of 21 days (normal cycle may vary from 18 to 24 days). The average duration of standing heat is 18 hours, and periods between 12 and 24 hours are considered normal. Hot and cold environmental temperatures tend to reduce the length of estrous periods, and increase

the difficulty in detecting heat [1]. Successful artificial insemination relies on correct heat identification because it is only possible in a short-time window after heat [3]. Missing a cycle will delay the next calving unnecessarily, causing economical and sustainability complications.

When a cow is in heat exhibits behavioral patterns which are distinctly different from the rest of the estrous cycle. For example, some signs of heat are: increased nervousness and/or restlessness which result in greater physical activity, cows that mount each other, and group formation of sexually active cows [1]. Except for [3], no works on automatic recognition of heat behaviors has been found in the literature. In similar problems, identification of behaviors or activities from sensors has been tried on humans [4, 5] and animals [6–10]. For example, authors in [4] attempted to identify everyday human activities with wearable sensors. In [7] is presented an automatic system that decodes ingestive sounds of cattle and gives information of grazing behavior. In [10] a method for behavior recognition of cows using a 3D accelerometer and support vector machine is proposed. Behaviors analyzed —such as standing, feeding, walking— were distinguishable by direct observation of a non expert. However, visual detection of heat behavior is possible but require an expert. Some activities last for minutes and other just for a few seconds ill conditioning the detection problem. Classification shows weak results because some behaviors were often confused.

In the past two decades several methods have been proposed for automatic heat detection [2]. Some of them have excellent accuracy but they involve measurements that are expensive, impractical or invasive. For example, body or milk temperature were considered, but they are not useful for practical applications, since both are highly influenced by external factors [11]. Milk progesterone achieved good rates of detection but at the expense of high costs and extra labor. Voluntary activity of cows were considered as an important indicator of heat, but additional observations and the use of a cow calendar are suggested as necessary conditions for sufficiently high detection. In recent years, technological improvements have enabled the development of devices that combine activity sensing with other measurements, such as lying/standing conditions. For example, in [3] a change detection algorithm was proposed by combining information from step count and lying time.

Acceleration signals have been found as a feasible source for automated identification of behavior patterns in animals and humans [9, 8, 12], however, no attempts to automatically detect heat from acceleration data have been reported. In this work is presented a strategy that has shown promising results and that has several advantages: data is collected in a simple and not invasive way, by attaching the device to the collar of the animal; recording device and data processor are not expensive; data processing is fast, results can be obtained seconds after that data is downloaded from the device; expert can concentrate only on those cows that show altered behaviors, a useful attribute in large herds. Acceleration records are filtered and segmented. After that, simple statistical attributes are extracted from each segment. Extracted features from a segment are shown to a classifier, which decides if the record belongs to a normal cow or a cow in

heat. Tested classifiers are multilayer perceptrons, support vector machines and decision trees.

The rest of this article is organized as follows: Section 2 introduces the analyzed signals and gives a detailed description of the detection method, including feature extraction and classification. Section 3 describes data collection and datasets generation. Results are presented and discussed in Section 4 and finally the conclusions are given in Section 5.

2 Detection method

Analyzed signals consist of three-dimensional acceleration varying through time. A fragment of a typical acceleration record is shown in Fig. 1. The three upper plots correspond to the acceleration on north (a_n), east (a_e), and up (a_u) directions. The bottom plot represents the magnitude of the total acceleration vector $|\boldsymbol{a}| = \sqrt{a_\mathrm{n}^2 + a_\mathrm{e}^2 + a_\mathrm{u}^2}$. Accelerations on vertical axes are given in m/s². Time on horizontal axes is given in s. Visual differences are not evident when signals recorded on normal cows and cows in heat are compared. Thus, a threshold, or some other rule, can not be defined.

A pattern recognition approach where the proposed system learns from labeled examples is applied. In a simplified scheme, the recognition system can be seen as a pipeline with two main stages: feature extraction (Section 2.1) and classification (Section 2.2). The feature extraction stage involves the conversion of the raw signal into a vector of attributes. In the classification stage, these feature vectors are given as an input to the classifier, which estimates the class that each vector belongs to. In the following subsections both stages will be explained in detail.

2.1 Basic feature extraction

The problem of interest is the detection of discriminative changes in the dynamic patterns contained in the records. A record could be extremely long and the translation into only one feature vector would not be the best choice. Furthermore, records of different duration are the typical situation. A common strategy in the field of signal processing involves the division of the original records into time fixed-length segments and then the translation of each segment into a feature vector. The flow of this processing strategy can be seen in Fig. 2. Each feature vector $\boldsymbol{x}_\mathrm{i}$ represents a segment labeled as i and is integrated by attributes extracted from a_n, a_e, a_u and $|\boldsymbol{a}|$. Therefore, from each record a collection of feature vectors is generated. Five attributes are extracted from the acceleration signal in each direction, which finally are organized to form a feature vector $\boldsymbol{x}_\mathrm{i}$ of twenty elements.

The shadowed area in Fig. 2 corresponds to the feature extraction detailed for a_n in Fig. 3. The same is applicable to a_u, a_e and $|\boldsymbol{a}|$. First, a high pass filter is applied to the a_n segment, thus, baseline components of the signal are separated from those related to movements. The cutoff frequency will be defined
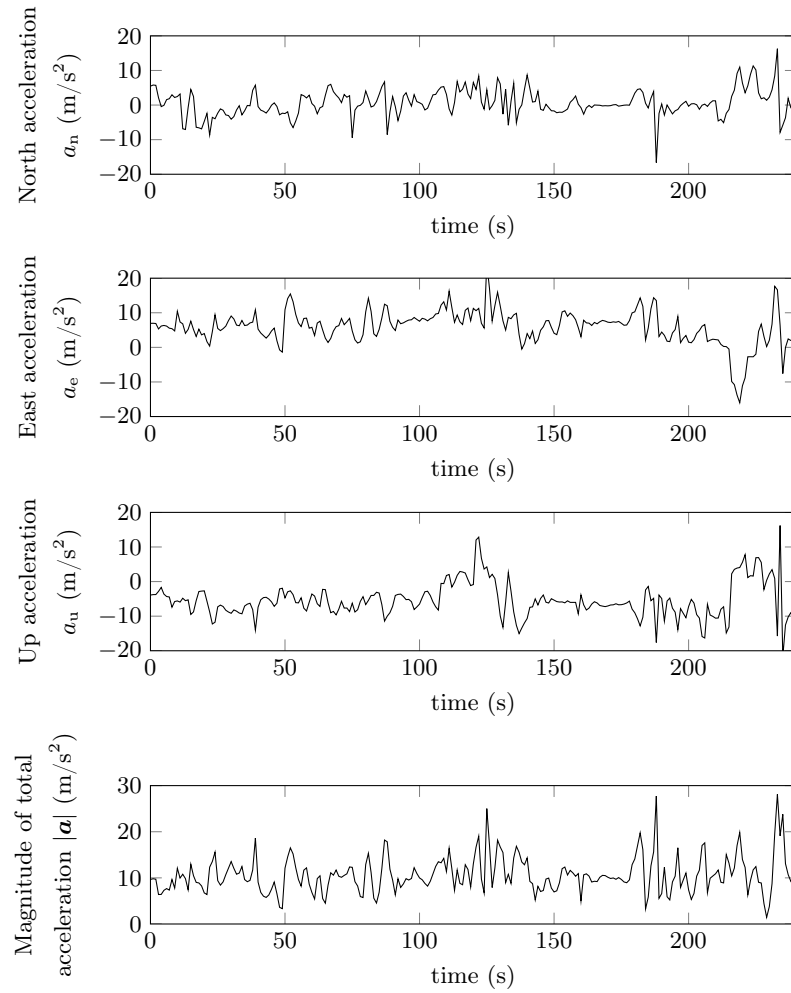
Fig. 1. Fragment of a typical record. From top to bottom, a_n, a_e and a_u plots correspond to the acceleration on north, east and up directions. The magnitude of the total acceleration vector $|\boldsymbol{a}| = \sqrt{a_n^2 + a_e^2 + a_u^2}$ is shown in the last plot.
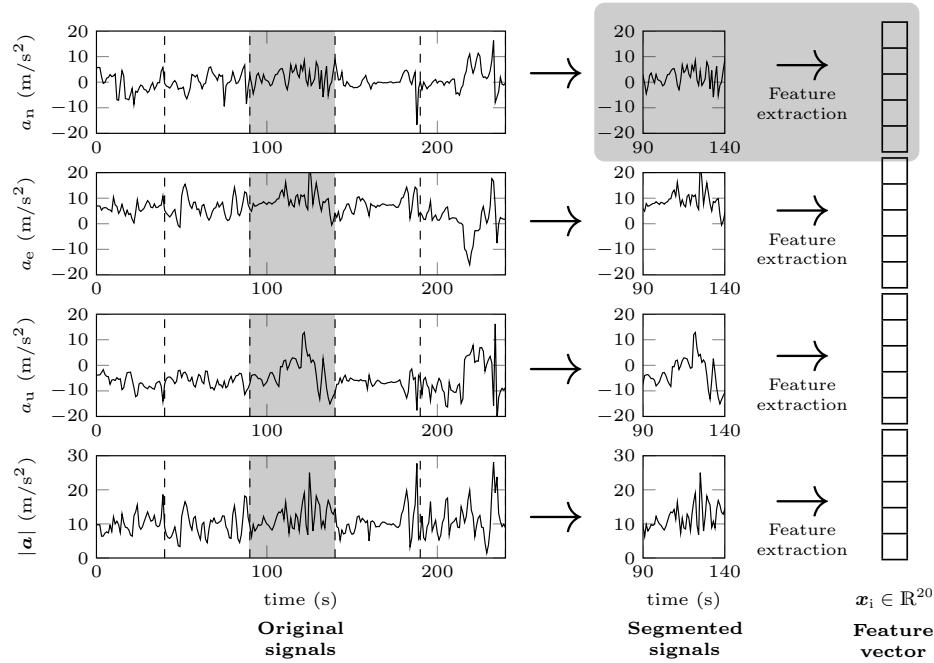
Fig. 2. Extraction of feature vector from a segment of the original signal.

later. Then, from the filtered signal five features are extracted: energy, maximum, minimum, amplitude (max − min) and standard deviation. These characteristics are stored as five float numbers that are arranged in a feature vector. Feature extraction processing is developed using Matlab.

In a following stage, feature selection is achieved using methods implemented in [13]. Correlation-based Feature Subset Selection (CFS) is selected for evaluation in combination with an exhaustive search. CFS evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them. Subsets of features that are highly correlated with the target class, while having low intercorrelation, are preferred. Restricted by the available data, only a preliminary feature selection is done.

2.2 Classification

A proper design of a classifier involves the separation of data into training and testing sets. Each feature vector in the training set contains one target value (the class label) and several attributes (the extracted features). Based on the training data, the goal of the classifier is to predicts the target values of the test feature vectors given only the attributes [14].

After successfully representations are constructed from segmented acceleration data, the problem of automatic heat recognition follows naturally. It may be
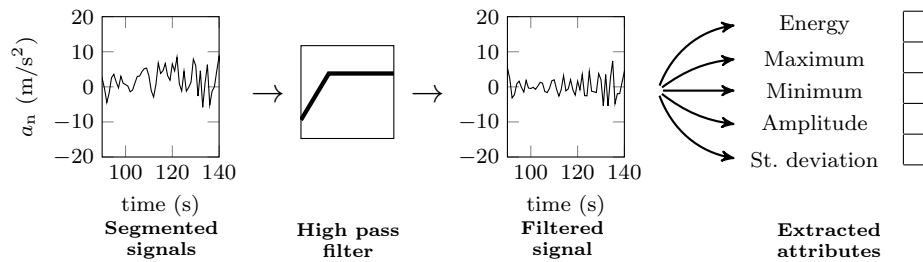
Fig. 3. Features extraction from a segment of raw signal.

simply considered as a binary classification problem where each feature vector-label pair consist of 20 attributes and a label that indicates *heat* or *normal*. In this work, three conventional static classifiers are tested: feed forward neural networks, support vector machines, and decision trees.

The assumption of this approach is that each segment contain enough information to represent one of the two target classes. That is, the discriminatory information is conserved after segmentation. Moreover, segments are considered independent. Thus, feature vectors can be presented to classifiers in any order. All classifiers are implemented using [13], except for the support vector machines, that are implemented with [15].

Multilayer Perceptron (MLP) is a conventional feed-forward artificial neural network design that can deal with non-linearly separable data. Consists of several layers of nodes in a directed graph, with each layer fully connected to the next one. Each neuron uses a hyperbolic tangent as the activation function. For further details see [16].

In this work, the number of inputs is set to 20 and the number of outputs to 2. One hidden layer is used. Attributes are normalized and labels binarized. Learning rate is fixed to 0.3, momentum to 0.2 and epochs to 1000. The remaining options are leaved in their default values. In preliminary experiments, significant changes were not observed using different number of neurons in the hidden layer. Thus, 5 neurons are used as it provided the best results.

Support vector machines (SVM) is a useful machine learning technique for binary classification problems. Conceptually, input feature vectors are non-linearly mapped to a very high-dimension feature space. The mapping to a higher dimension is done by a kernel function K. In this feature space a linear decision surface is built. That surface is a hyperplane and is located to achieve the maximal margin of separation between the two classes (see [17] or [18] for detailed explanations). In this work, a radial basis function $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{-\gamma|\boldsymbol{x}_i - \boldsymbol{x}_j|^2}$ is chosen with a soft margin penalty for misclassifications. Penalty coefficient C and parameter γ are optimized over the independent dataset (sets will be described in Section 3). Remaining parameters of the classifier are kept on their default values.

Decision Trees (DT) are characterized by the fact that an unknown feature vector is classified into a class using one or several decision functions in a sequential manner. In general, a decision tree consists of a root node, a number of interior nodes, and a number of terminal nodes. The root node and interior nodes, collectively referred as non-terminal nodes, are linked into decision stages; the terminal nodes represent final classifications. Associated with the root node is the entire set of classes into which a feature vector may be classified. Each node consists of a set of classes to be discriminated, the set of features to be used, and the decision rule for performing the classification. For further details see [19, 20]. In this work, a decision tree is built using the C4.5 algorithm. Changes in pruning confidence did not reflected significant differences. Thus, default settings are used.

3 Data collection and datasets

Three-dimensional acceleration were recorded using an inertial measurement unit (IMU) placed inside a box securely attached to the collar of the animal (Fig. 4). The IMU combines triaxial gyroscope, accelerometer, and compass sensors in conjunction with advanced processing to give reliable measurements. Normal signals were recorded during grazing on a cow that was not in heat. Records corresponding to heat were recorded on a cow that exhibited signs of heat. The data collected when the animal was tied or influenced by the experiment itself were excluded from the analysis. Sampling rate was fixed to 1 Hz for all records. The mean record duration was 5 minutes.
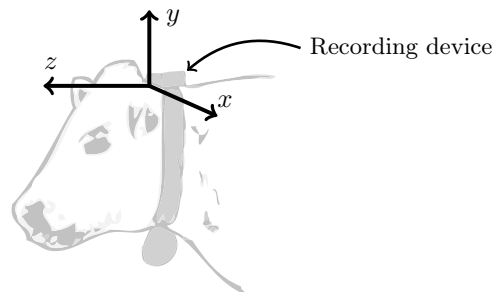

Fig. 4. Schematic ilustration of recording device attached to the top of the collar. Direction of positive x, y and z acceleration axes are shown.

Gravity and north directions were registered as versors at the same sampling rate in conjunction with accelerations. East versor was obtained by cross product between gravity and north versors. Up direction was obtained simply as the opposite to gravity. After that, acceleration on x, y and z axes were projected over north, east and up directions.

In each record, signals were segmented in blocks of 20, 40, 60, 80 and 100 seconds. The segments, or blocks, were generated by a fixed-length time window, without overlap. The window was moved across the original signals from the end to the beginning, and any remaining samples were discarded. As it was detailed in Section 2.1, feature vectors were created by filtering and extracting features from one segment at a time. The first five datasets listed in Table 1 correspond to the extraction of features without filtering. The remaining ten datasets were obtained by applying a high pass filter with a cutoff frequencies of 0.15 and 0.25 Hz. Those frequencies were selected in preliminary experiments as they seem to give a good balance between energy conservation and low frequencies attenuation.

The resulting number of feature vectors extracted from raw signals are listed in Table 1. Each feature vector belongs to *heat* or *normal* class and is represented by the 20 attributes detailed in Section 2.1.

Table 1. Description of datasets generated from raw signals. In the columns: size of segments, number of feature vectors, and cutoff frequency of the filter.

Dataset name	Segments size (seconds)	High pass frequency (Hz)	Number of feature vectors
D_1	20	0.00	49
D_2	40	0.00	24
D_3	60	0.00	15
D_4	80	0.00	11
D_5	100	0.00	9
D_6	20	0.15	49
D_7	40	0.15	24
D_8	60	0.15	15
D_9	80	0.15	11
D_{10}	100	0.15	9
D_{11}	20	0.25	49
D_{12}	40	0.25	24
D_{13}	60	0.25	15
D_{14}	80	0.25	11
D_{15}	100	0.25	9

To perform feature and model selection, a subset of 12 feature vectors was extracted from D_1. Selected feature vectors were removed from D_1 to non invalidate the cross validation of the next section. Therefore, two new sets were defined, D_{1s} (for selection) and D_{1c} (for classification), where $D_{1s} \cup D_{1c} = D_1$ and $D_{1s} \cap D_{1c} = \emptyset$. Analogous division was performed over D_6 and D_{11} sets. Parameter selection for SVM was carried over the union set $D_{1s} \cup D_{6s} \cup D_{11s}$.

4 Results and discussion

Because the number of feature vectors in each dataset were small, *leave-one-out* cross-validation was applied. In leave-one-out cross-validation the number of folds was fixed to the number of feature vectors in the dataset. In each iteration, one feature vector was used for testing and the remaining feature vectors were used for training the classifier. The test results were collected and averaged over all folds [21]. Performance of classifiers was measured by counting the proportion of correctly predicted feature vectors in the unseen test dataset. That is, the classifier accuracy.

As it was introduced at the end of Section 2.1, a feature selection was carried on. The selected attributes were: total maximum ($\max_t$) in D_{1s}, up energy (ene_u) in D_{6s} and east amplitude (amp_e) in D_{11s}. During classification we compared the performance of using all attributes and only the 3 selected. Datasets that contain all features will be named as D_{Af}, and datasets that contain only the 3 selected as D_{3f}.

4.1 Detection results

Classification accuracy for the proposed classifiers is shown in Fig. 5. Top plot corresponds to MLP, the middle to SVM and the bottom one to DT. Labels on horizontal axis represent the conditions under each dataset was created. For example, a dataset created by segments of 40 seconds and a filtering frequency of 0.15 Hz is shown as the seventh pair of columns, from left to right. White bars are the result of using all the features and grey bars correspond to accuracy classification using only the selected attributes ($\max_t$, ene_u and amp_e).

Results for MLP (Fig. 5(a)) using all features are analyzed first. Accuracy achieved over datasets created without filtering (D_1-D_5) improves as the length of segments gets reduced. The best result is obtained over the dataset D_2, that corresponds to segments of 40 seconds length. Classification accuracy on datasets created by filtering at 0.15 Hz (D_6-D_{10}) improves as the length of segments is increased. Thus, short segments are preferred for non-filtered signals, and large segments for filtered signals. The reason behind these results is related to the attributes that contribute more to the final decision. Recalling feature selection, total maximum (a local attribute) was selected for non-filtered signals (D_{1s}), and up energy (a global measure) was selected for filtered signals at 0.15 Hz (D_{6s}). Local attributes are attenuated during filtering process and that explain the apparently opposed results. Finally, accuracy achieved on datasets created by 0.25 Hz filtering frequency (D_{11} to D_{15}) are not influenced by the length of the segments. Because all discriminatory information, that improve results over datasets D_1 to D_{10}, is no longer available after the 0.25 Hz filtering, in D_{11} to D_{15}. Better results are expected for MLP using selected attributes ($\max_t$, ene_u and amp_e), because least and relevant features tend to generalize better than using all features. An improvement can be seen for the datasets that correspond to 0.15 Hz filtering and short segments (≤ 60 s).

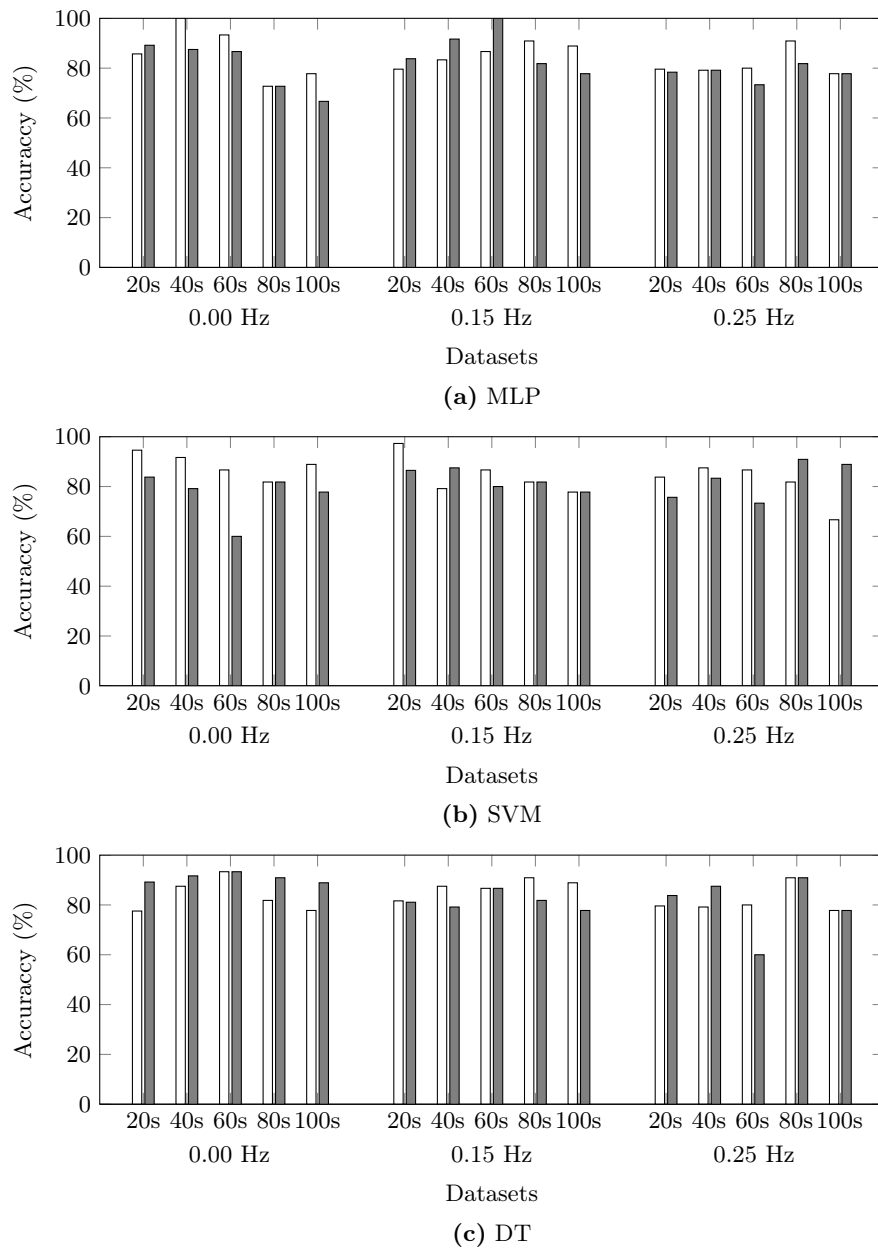**(a)** MLP

(b) SVM

(c) DT

Fig. 5. Classification accuracy for (a) MLP, (b) SVM and (c) DT. Labels on horizontal axis correspond to the conditions under each dataset was created (size of segments and filtering frequency). White bars were obtained using all the features and gray bars using selected attributes (max_t, ene_u and amp_e).

Obtained results using all features and SVM (Fig. 5(b)) were better for short segments, under any filtering condition. This tendency could be influenced by the difficulty to generalize when just a few feature vectors are in the training dataset (the number of feature vectors decreases as length of segments increases). In general, classification using only the selected features show poorer or similar results. The SVM classifier is more immune to irrelevant features, thus, eliminating them do not change the results. Instead, if an important feature is dropped out the performance of the classifier will be reduced. Best result for SVM corresponds to 20 seconds segments, 0.15 Hz filter cutoff and all features.

Classification accuracy for DT (Fig. 5(c)) show results that are reasonably good in almost all datasets, above 77% and reaching a 93% in the best situation. Over datasets created without filtering, a peak is observed on 60 seconds segments and all features. Using only selected features the results improve in the five datasets, and a unique feature is used to build the trees ($\max_t$). The improvements seen on trees built after feature selection are explained by the reduction of the number of possible architectures, a fact that simplify the search. Trees created for datasets corresponding to 0.15 and 0.25 Hz give results that are similar or poorer than previously analyzed.

The analysis of the architecture of trees shows that, for datasets in D_{Af} corresponding to 20 seconds segments, trees look like the one in Fig. 6(a). It is a tree of 7 nodes, 3 non-terminal and 4 terminal (2 for heat and 2 for normal). Threshold values for decisions are given in non-terminal nodes. The remaining trees created (over D_{Af} and D_{3f}) consist of the simplest tree, 1 root node and 2 terminal nodes (1 for each class). An example of such trees is shown in Fig. 6(b). In these simplest trees, the selected feature was total maximum (16 of 27 times). Without feature selection (over D_{Af}) the selected feature was north energy (5 of 15 times). DT architecture is interesting as it gives some clues about the classification problem. The classification with only one feature shows that the classes can be distinguish in a simple way. However, it is not always the same feature in the root node. Unexpectedly, north features get that position 10 of 15 times (over D_{Af}). Results of classification with trees reveal that minimum and standard deviation are not important attributes to discriminate between classes. An unexpected situation was the absence of features that came from vertical direction, since events with vertical acceleration like mounts are only seen during heat. Total and north direction result as the more relevant ones.

4.2   Best configuration

The influence of segment size was not always the same when all results are compared. Nevertheless, segments of 40 s and without filtering have lead to good results using any classifier. The results over this dataset are summarized in Table 2, for all classifiers, over D_{Af} and D_{3f}. MLP and SVM achieved better results without feature selection, being MLP the best. Instead, DT results were improved after feature selection, followed by MLP. Finally, better results under almost any conditions were achieved with MLP, followed by the performance of DT. SVM showed high variance results, lowering its confidence.
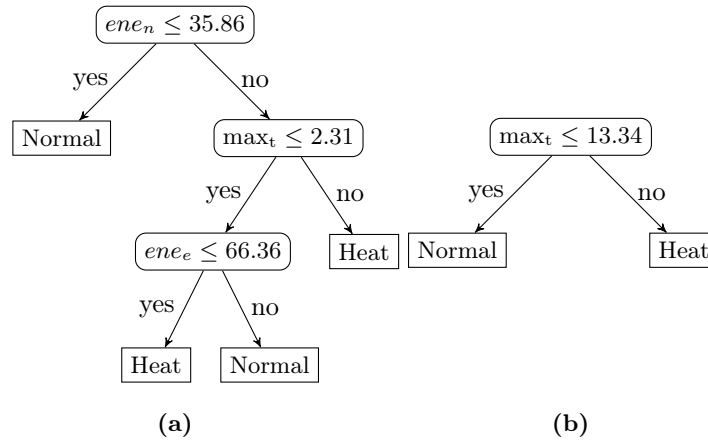
(a)   **(b)**

Fig. 6. Two examples of trained decision trees. Left tree correspond to D_6 and right tree to D_3, using all attributes.

Table 2. Comparative accuracy between classifiers with and without feature selection in D_2 (segments of 40 s, without filtering).

	MLP (%)	SVM (%)	DT (%)
All attributes	100	91.6	87.5
Selected attributes	87.5	79.1	91.6

5   Conclusions

A novel approach for heat detection was presented. This is based on commonly used classifiers and simple attributes extracted from acceleration data. The strategy involves two stages, feature extraction —where feature vectors are obtained from filtered and segmented data— and classification —where feature vectors are identified as heat or normal. Results have shown promising levels of detection, turning the proposal into a feasible method for practical applications. To our knowledge this is the first time that accelerations are used in an automatic system of heat recognition. Moreover, previous reported results using non-invasive methods have shown lower accuracy. Our method achieved levels of accuracy that are between 93% and 100%, and it has several advantages: data is gathered in a practical and not invasive way, hardware has low cost, data can be processed in seconds, and an expert can focus only on those cows that are marked as in heat.

Future work will involve the analysis of longer records and the optimization of adjustable parameters of the system.

References

1. Gray, H.G., Varner, M.A.: Signs of estrus and improving detection of estrus in cattle. Fact Sheet IRM 6, West Virginia University (1984)
2. Firk, R., Stamer, E., Junge, W., Krieter, J.: Automation of oestrus detection in dairy cows: a review. Livestock Production Science **75**(3) (2002) 219–232
3. Jónsson, R., Blanke, M., Poulsen, N.K., Caponetti, F., Højsgaard, S.: Oestrus detection in dairy cows from activity and lying data using on-line individual models. Computers and Electronics in Agriculture **76**(1) (2011) 6–15
4. Parkka, J., Ermes, M., Korpipaa, P., Mantyjarvi, J., Peltola, J., Korhonen, I.: Activity classification using realistic data from wearable sensors. IEEE Transactions on Information Technology in Biomedicine **10**(1) (2006) 119–128
5. Gyllensten, I., Bonomi, A.: Identifying types of physical activity with a single accelerometer: Evaluating laboratory-trained algorithms in daily life. IEEE Transactions on Biomedical Engineering **58**(9) (2011) 2656–2663
6. Milone, D., Rufiner, H., Galli, J., Laca, E., Cangiano, C.: Computational method for segmentation and classification of ingestive sounds in sheep. Computers and Electronics in Agriculture **65**(2) (2009) 228–237
7. Milone, D.H., Galli, J.R., Cangiano, C.A., Rufiner, H.L., Laca, E.A.: Automatic recognition of ingestive sounds of cattle based on hidden markov models. Computers and Electronics in Agriculture **87** (2012) 51–55
8. Hamalainen, W., Jarvinen, M., Martiskainen, P., Mononen, J.: Jerk-based feature extraction for robust activity recognition from acceleration data. In: 11th International Conference on Intelligent Systems Design and Applications (ISDA). (November 2011) 831–836
9. Nathan, R., Spiegel, O., Fortmann-Roe, S., Harel, R., Wikelski, M., Getz, W.M.: Using tri-axial acceleration data to identify behavioral modes of free-ranging animals: general concepts and tools illustrated for griffon vultures. The Journal of Experimental Biology **215**(6) (2012) 986–996
10. Martiskainen, P., Järvinen, M., Skön, J.P., Tiirikainen, J., Kolehmainen, M., Mononen, J.: Cow behaviour pattern recognition using a three-dimensional accelerometer and support vector machines. Applied Animal Behaviour Science **119**(1–2) (2009) 32–38
11. Fordham, D.P., McCarthy, T.T., Rowlinson, P.: An evaluation of milk temperature measurement for detecting oestrus in dairy cattle. II variations in body and milk temperature associated with oestrus. Veterinary Research Communications **11**(4) (1987) 381–391
12. Reiss, A., Hendeby, G., Stricker, D.: A competitive approach for human activity recognition on smartphones. In: ESANN 2013, ESANN (2013) 455–460
13. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: An update. SIGKDD Explor. Newsl. **11**(1) (2009) 10–18

14. Hsu, C.W., Chang, C.C., Lin, C.J.: A practical guide to support vector classification (2003)
15. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**(3) (2011) 27:1–27:27
16. Bishop, C.M.: Pattern recognition and machine learning. Information Science and Statistics. Springer, New York (2006)
17. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning **20**(3) (1995) 273–297
18. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery **2**(2) (1998) 121–167
19. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and regression trees. Chapman & Hall, New York (1993)
20. Swain, P.H., Hauska, H.: The decision tree classifier: Design and potential. IEEE Transactions on Geoscience Electronics **15**(3) (1977) 142–147
21. Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, Burlington, Mass. (2011)