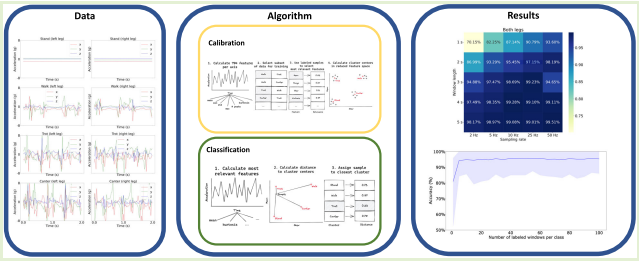
Time-Series-Based Feature Selection and Clustering for Equine Activity Recognition Using Accelerometers

Timo De Waele, Adnan Shahid, *Senior Member, IEEE*, Daniel Peralta, *Member, IEEE*, Anniek Eerdekens, Margot Deruyck, *Member, IEEE*, Frank A. M. Tuyttens, and Eli De Poorter

***Abstract*—With over 16 million horses worldwide and nearly 60 000 sport horses registered to the International Federation for Equestrian Sports database, tracking the activities and performance of these equines is becoming an important aspect in horse management. To perform this activity recognition, inertial measurement units (IMUs) are often used in combination with machine learning algorithms. These often require large labeled datasets to be trained. To this end, a data-efficient algorithm is proposed that requires only 3 min of labeled calibration data. This is achieved by combining supervised feature selection, using the tsfresh time-series feature calculation library and the Kendall rank correlation coefficient, with a distance-based clustering algorithm. The generalizability performance of the algorithm is tested by evaluating on a dataset captured with leg-mounted IMUs and on a dataset captured using a neck-mounted IMU. On both datasets, the algorithm achieved the accuracies of 95%, comparable to state-of-the-art deep learning approaches, when calibrating and evaluating using the same horse. When the algorithm was calibrated on data from multiple horses and evaluated on horses that were not in the calibration dataset, a 15% drop in classification accuracy is observed. The proposed algorithm is compared with fully supervised algorithms, such as convolutional neutral network, support vector machine, and random forest, in terms of accuracy achieved with respect to the size of the labeled data using calibration. Our approach achieved accuracies that were similar to these classical algorithms while only using 10%–5% the amount of labeled data.**



***Index Terms*—Classification, clustering, equine activity recognition, feature selection.**

Nomenclature

x_i, y_i z_i	ith raw IMU acceleration sample along the x-, y-, or z-axis, expressed in amount of g.
L	Length of the dataset of one horse consisting of the raw IMU samples.
$\|\|\bar{A}\|\|$	Mean vector norm of the dataset consisting of the raw IMU samples.
$\hat{x}_i$, $\hat{y}_i$ $\hat{z}_i$	ith normalized IMU acceleration sample.
n_c	Number of concordant pairs in the calculation of the Kendall rank correlation coefficient.
W	Window length in a number of raw IMU samples.
n_d	Number of discordant pairs in the calculation of the Kendall rank correlation coefficient.
n	Total number of items in the calculation of the Kendall rank correlation coefficient.
X	Calibration dataset windows consisting of all features calculated by the tsfresh library.
$X_{i,k}$	Value of the kth feature of the ith window in the calibration dataset.
Y	Calibration labels consisting of the numerical class for every window in X.

Manuscript received 28 February 2023; accepted 29 March 2023. Date of publication 14 April 2023; date of current version 31 May 2023. This work was supported in part by the Fund for Scientific Research Flanders, FWO-Vlaanderen, Fonds Wetenschappelijk Onderzoek (FWO)-Strategisch Basisonderzoek (SB), Belgium, under Grant 1SB7619N. The associate editor coordinating the review of this article and approving it for publication was Dr. Pedro Oliveira Conceição Junior. *(Corresponding author: Timo De Waele.)*

Timo De Waele, Adnan Shahid, Daniel Peralta, and Eli De Poorter are with the IDLab Research Group, Faculty of Engineering and Architecture, Ghent University, 9000 Ghent, Belgium (e-mail: timo.dewaele@ugent.be; adnan.shahid@ugent.be; daniel.peralta@ugent.be; eli.depoorter@ugent.be).

Anniek Eerdekens and Margot Deruyck are with the Waves Research Group, Faculty of Engineering and Architecture, Ghent University, 9000 Ghent, Belgium (e-mail: anniek.eerdekens@ugent.be; margot.deruyck@ugent.be).

Frank A. M. Tuyttens is with the Faculty of Veterinary Medicine, Ghent University, 9820 Merelbeke, Belgium (e-mail: frank.tuyttens@ugent.be).

Digital Object Identifier 10.1109/JSEN.2023.3265811

Y_i	Numerical class label for the ith window in X.
τ_k	Value of the Kendall rank correlation coefficient for the kth feature in X.
K	Total number of features calculated by tsfresh and thus present in X.
N	Number of most significant features our algorithm will select.
F	List containing the N most significant features, selected by our algorithm.
X'	Calibration dataset X in the reduced feature space consisting of the features in F.
a	Variable indicating a certain class label (0, 1, 2, or 3).
$X'_{a,i}$	ith window of the subset of the calibration dataset X' that only contain windows belonging to class a.
N_a	Total number of windows present in the calibration dataset of class a.
C_a	Cluster centroid of class a.
S	New, unseen, window that needs to be classified by our algorithm.
S_k	Value of the kth feature of window S.
$C_{a,k}$	Value of the kth feature of the cluster centroid belonging to class a.

I. Introduction

THE global equine industry consists of over 16 million horses and employs more than 1.6 million full-time workers, and the total yearly revenue of this growing sector is upward of 270 billion dollars [1]. A big part of this revenue comes from the sale of top sporting horses, with some of these horses being sold for prices upward of 10 million dollars [2]. Next to and maybe even more important than the clear financial aspects of the equine industry is the large emotional value of the horses to their owners. Thus, making sure that the horses are healthy and happy is an important aspect in horse management. Being healthy and happy, however, is not the only aspect of the horse that is being closely monitored, with nearly 60 000 sporting horses being registered to the International Federation for Equestrian Sports databases each year, and tracking the performance and training sessions of these sporting horses plays an important part in their training routines [3].

To perform this activity tracking for horses, just like with human activity recognition, a variety of sensor systems can be used. The first type of sensor that comes to mind for tracking horses is a global position system (GPS) system; however, these come with some drawbacks such as poor accuracy (in the order of a few meters), high energy consumption, and their inability to work in an indoor or covered environment. To this end, inertial measurement units (IMUs), consisting of accelerometers and gyroscopes, are being widely used for health monitoring and activity recognition tasks [4]. These work both indoors and outdoors and consume very little energy (in the order of 1.5 mW at a sampling rate of 25 Hz versus 165 mW for GPS [5], [6]). For animal behavioral analysis, they have already proven their worth by achieving the classification accuracies of 94% on a dataset consisting of data from a neck-collar-mounted IMU [7]. The dataset included data from four different gaits (standing, walking, trotting, and canter) and eating behavior. They also have shown to be a useful asset in combination with unsupervised deep learning to detect the onset of parturition in horses [8].

When using IMUs, the output of the sensors is a time-series consisting of triaxial acceleration and rotation, so an algorithm is required to interpret this data stream. While classical algorithms and statistics, such as time-series analysis and goodness-of-fit tests, have proven to work for this task, they lack generalization toward different species or sensor locations, requiring lots of manual effort and tweaking to adjust the algorithms for these different scenarios [9]. One way to solve this is by using supervised deep learning algorithms that make use of convolutional neural networks (CNNs) [10], [11]. However, these require massive labeled datasets to train, with hundreds of thousands of windows for different species and sensor locations. To reduce the need for this labeling effort, research is being directed toward more data-efficient classification algorithms and techniques, for supervised, semisupervised, and unsupervised approaches.

This article proposes a supervised classification algorithm that, given very few labeled samples, can classify the four basic gaits of horses (stand, walk, trot, and canter) with state-of-the-art accuracy. The algorithm requires only 3 min of labeled data to calculate its parameters without machine learning training while still obtaining accuracies comparable to classical fully supervised models such as CNNs specifically designed for equine gait detection and support vector machines (SVMs). The algorithm is designed by combining supervised feature selection, using the tsfresh time-series feature calculation library [12] and the Kendall rank correlation coefficient [13], with a distance-based clustering algorithm. This library can calculate 77 different time-series features, some of which can be tweaked using several parameters, bringing the total amount of features to 794. A more in-depth overview of this step is given in Section III.

The main contributions of this article are given as follows.

1) We propose a supervised classification algorithm that does not need to be trained for detecting the gait of horses and that requires just a couple of minutes of labeled data to achieve 95+% classification accuracy.
2) We evaluate the generalizability of our approach in terms of sensor location by evaluating the proposed algorithm performance on a leg-mounted IMU dataset and on a neck-mounted IMU dataset.
3) We discuss and compare individual and global approaches, where the individual approach corresponds to an approach designed specifically for an individual subject, while the global approach works for all subjects of the same species.
4) We discuss and compare the performance of the proposed algorithm with respect to several variations of input data, i.e,, sampling rate, classification window length, and number of sensors used.

5) We compare the proposed algorithm to different fully supervised approaches, such as random forest (RF), SVM, and CNN, in terms of accuracy achieved per number of labeled samples used during the training and calibration phase.

The rest of this article is structured as follows. Section II discusses the related work and Section III describes the system design and the used datasets. Section IV describes the details of the algorithm in terms of: 1) feature selection; 2) cluster mapping; and 3) classification. Section V presents the results in terms of: 1) individual-based models; 2) impact of number of sensors; 3) impact of window length and sampling rate; 4) impact of senor location; 5) global model; and 6) comparison with classical machine learning approaches. Finally, Section VI concludes this article. In addition to concluding this article, Section VI offers the directions for future research.

II. Related Work

Currently, two approaches exist for performing animal activity recognition using data from IMUs: 1) classical approaches and 2) data-efficient approaches that use a different, more compact, representation of the input data. Classical approaches make use of end-to-end deep learning classifiers such as dense neural networks and CNNs directly on the input data. Data-efficient approaches combine feature extraction, either manual or automatic, with classic machine learning algorithms such as SVM or K-nearest neighbors.

Using deep learning for equine activity recognition has been shown to yield an accuracy higher than 90%, by using a CNN where the convolutional layers automatically learned relevant features from the input data. These self-learned features were used by a couple of dense layers to perform the actual classification [14]. Using a cross-modality interaction network and a class-balanced focal loss researchers also managed to achieve 90% accuracy and an F1-score and precision of both 79% on an unbalanced dataset of both accelerometer and gyroscope data [15]. By using a bidirectional long short-term memory (Bi-LSTM) the need for feature learning/extraction is trained on data collected from the IMU of a smartphone, researchers have managed to achieve a macro average of 94% [16]. However, while often leading to the best results in terms of accuracy and precision, classical deep learning requires massive labeled training datasets, with multiple hours of data, to achieve good performance. The need for large datasets can partly be reduced by generating pseudo labels using a clustering algorithm such as K-means, and this has already shown to be successful for the case of human activity recognition based on trajectory information [17].

To reduce the need for large labeled datasets, automatic feature learning, as is done in a CNN, can be replaced by manual feature extraction. Here, humans will look into the data and manually select the best features to be used for classification. By combining this approach of manual feature extraction with a supervised classifier, researchers have managed to achieve 90+% accuracy with only a fraction of the amount of labeled data required to train a CNN [18]. However, this technique requires strong domain knowledge as well as trial-and-error to find which features and parameters work best, making it nontrivial to transfer a model that was trained for one use case to another.

Researchers have tried to combine the benefits of the automatic feature extraction from CNN with the data efficiency of manual feature extraction. By combining representation learning with an SVM for classification, they managed to achieve an F1-score of 80% for equine activity recognition [19]. The input features to the SVM were the latent variables of a sparse autoencoder that was trained using an unlabeled dataset. The SVM was then trained using these latent features with just 250 labeled samples. By combining the automatic feature learning abilities of autoencoders with a clustering objective, this approach could even be made fully unsupervised, requiring no labeled samples at all, as is shown by the deep embedded clustering (DEC) algorithm [20]. The main drawback of this approach is that it can be difficult to optimize and fine-tune, depending on the dataset and problem, especially when the initial clustering contains overlapping clusters or clusters have little space in between them.

There exists also a middle ground between fully automated feature learning using autoencoders and the manual feature selection approach. By using a feature calculation library such as tsfresh [12], hundreds of different features can be automatically calculated. A more recent proposal focused on the scalability of the feature extraction process for large amounts of long multivariate time series [21]. By using an RF to then perform feature selection, researchers have achieved good performance using tsfresh on human activity recognition tasks [22]. Another study used a combination of tsfresh and an SVM to perform the classification of human tasks based on inertial information captured by smart glasses [23]. Outside of activity recognition, the library has been used for the task of handwriting recognition using wearable sensors, achieving 98% classification accuracy for the 26 characters in the Latin alphabet [24].

Table I gives an overview of the related work in the field of animal activity recognition in terms of model type used, input type (raw input features or manually extracted features), achieved accuracy, and total length of the dataset in terms of seconds of labeled data. In contrast to classical approaches, our work achieves similar accuracies, upward of 95%, but only requires a fraction of the labeled data. Our proposed approach also achieves higher accuracies than the already existing data-efficient approaches with similar amounts of labeled data while requiring no manual feature extraction. The final benefit of our work is the lack of computationally expensive machine learning and deep learning algorithms, thus requiring fewer resources than the existing methods.

III. Pipeline and Data Description and Processing

A. Pipeline

In this section, we first present classical deep learning pipeline and then the proposed data-efficient classification pipeline for equine activity recognition. With classical deep learning approaches, as shown in Fig. 1(a), the process for creating a performant activity recognition model starts with collecting a large representative dataset containing samples of

TABLE I
COMPARISON WITH RELATED WORK, OUR WORK REQUIRES NO MANUAL FEATURE ENGINEERING, REQUIRES ONLY 200 s OF LABELED DATA WHILE ACHIEVING ACCURACIES THAT ARE SIMILAR TO NON-DATA-EFFICIENT APPROACHES

Paper	Species	Model type	Feature extraction	Accuracy	Labeled data duration
Classical approaches					
A. Eerdekens *et al.* [14]	Horse	CNN	Automatic	99.60%	633 000 seconds
A. Mao *et al.* [15]	Horse	CNN	Automatic	93.37%	116 000 seconds
J. W. Kamminga *et al.* [7]	Goat / sheep	DNN	Manual	94%	44 000 seconds
S. Luqian *et al.* [22]	Humans	RF	Automatic	89%	28 000 seconds
H.B. Davidsson *et al.* [16]	Horses	Bi-LSTM	Automatic	94%	17 000 seconds
M. Raychoudhury *et al.* [23]	Humans	SVM	Automatic	87%	/
Data efficient approaches					
M. Rana *et al.* [17]	Humans	Clustering + CNN	Manual	88.5%	6000 seconds
E. Casella *et al.* [18]	Horse	SVM	Manual	91%	135 seconds
J. W. Kamminga *et al.* [19]	Horse / goat / sheep	Autoencoder + SVM	Automatic	80%	250 seconds
This work	**Horse**	**Feature selection + clustering**	**Automatic**	**95%**	**200 seconds**

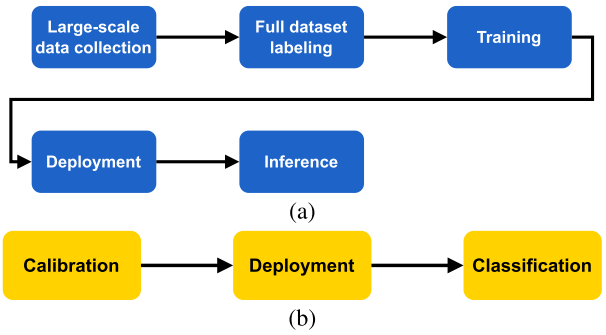


Fig. 1. Comparison between a classic deep learning pipeline for activity classification and our proposed pipeline. (a) Classic deep learning pipeline. (b) Data-efficient classification pipeline (this work).

the classes that we want to classify. This dataset then will need to be labeled manually, creating a sample-to-activity mapping. This labeled dataset is used to train a deep learning model until the required accuracy is reached. This trained model is then deployed in the field and used for the real-time classification of activities.

In this article, we introduce an algorithm that replaces the large-scale data collection, full dataset labeling, and training with one simple calibration step, and a graphical overview of this approach is shown in Fig. 1(b). The calibration step comprised of four steps: 1) 2–4 min of data collection for each activity; 2) feature extraction using the tsfresh library; 3) selecting the most relevant features using the Kendall rank correlation coefficient; and 4) cluster mapping.

B. Dataset Description

Two different datasets were used during this study: one captured by Ghent University using leg-mounted IMUs and one open-source dataset consisting of data from a single neck-mounted IMU [25].

The leg dataset consists of data captured from 13 horses that were boarded at two stables, this amount of horses exceeds or closely matches the amount of horses used in other similar studies, and the number of subjects used in the studies shown in Table I ranges from 2 to 18 horses. The data consist of a mixture of ridden and longed horses. Off-the-shelf three-axis logging accelerometers (Axivity AX3 [26]) were used as IMU. They were configured to capture data at a sampling rate of 50 Hz and with a range of ± 16g. Each horse was equipped with two IMUs, one for each of its front legs. The internal clocks of both IMUs were synchronized so the data from them could be easily combined. They were attached using Velcro to the outside of a pair of front tendon boots. The average length of the dataset per horse was 24 min, bringing the total dataset size to $13 \times 24 = 312$ min, or just over 6 h.

The neck dataset contains data of 18 horses, from which 11 were labeled. From these labeled horses, seven contained more than 5 min of data for each of the four gaits. Therefore, seven horses were used during our evaluation experiments. The IMUs were attached to the horses using neck collars and were configured to capture data at 100 Hz. To make the comparison between the leg and neck data fair, we resampled the neck data down to 50 Hz. In total, the neck dataset contained over 3 h of data per horse on average, with a total dataset size of just over 23 h.

Fig. 2 gives a visual indication where each sensor was placed. In Table II, an overview is given of the total amount of data present in both datasets for each of the four gaits of interest.

In Fig. 3, a visualization is given of a 2-s output window for the leg-mounted accelerometers for stand, walk, trot, and canter.

Fig. 2. This picture indicates all three locations on the body of the horse of the IMUs that were used in this article.

TABLE II
TOTAL DURATION FOR EACH GAIT PRESENT IN BOTH THE LEGS AND NECK DATASET

Legs		
Type	**Total duration**	**Average duration per horse**
Stand	20.3 min	1.6 min
Walk	115.2 min	8.9 min
Trot	97.0 min	7.5 min
Canter	48.4 min	3.7 min
Neck		
Type	**Total duration**	**Average duration per horse**
Stand	84.5 min	12.1 min
Walk	724.1 min	103.3 min
Trot	535.8 min	76.6 min
Canter	50.4 min	7.2 min

C. Dataset Processing

To reduce the bias introduced into the datasets by differences in height and build of each horse, we applied normalization to both the leg and neck datasets for each individual horse. Because the data consisted of 3-D acceleration vectors, we opted for vector normalization. To do this, we calculated the mean vector norm over all data of each individual horse, as shown in the following equation:

$$||\bar{A}|| = \frac{1}{L}\sum_{i=1}^{L}\sqrt{x_i^2 + y_i^2 + z_i^2} \tag{1}$$

where x_i indicates the acceleration value along the x-axis of the ith IMU sample, the same description applies to y_i and z_i, and L is the total number of IMU samples captured for each horse. Nomenclature contains an overview and description of all variables used in this article.

We then divided each axis of the 3-D acceleration vector by the mean vector norm

$$\hat{x_i} = \frac{x_i}{||\bar{A}||}, \quad \hat{y_i} = \frac{y_i}{||\bar{A}||}, \quad \hat{z_i} = \frac{z_i}{||\bar{A}||}. \tag{2}$$

In this way, the average vector norm of each horse after normalization is 1.

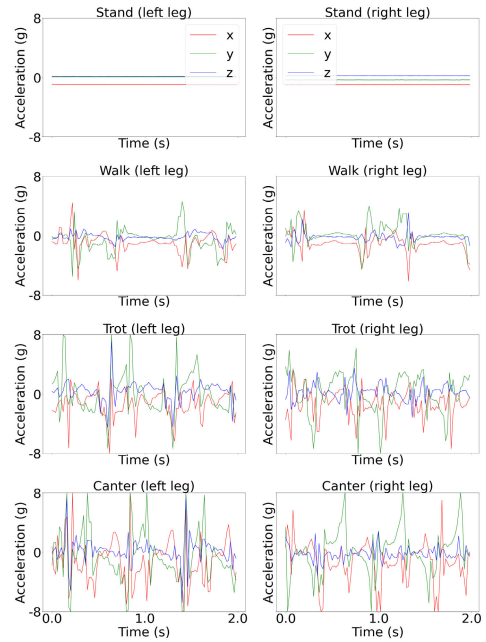


Fig. 3. Example output window of the IMU for each of the four gaits, consisting of the acceleration along all three axes for both legs.

Once we have normalized each dataset, we split it up into windows of a fixed length W, and the length of these windows depended on the experiment being performed and ranged from 1 to 5 s, as this has already proven to be a useful range for the classification of equine activities [14]. After normalizing each dataset of L samples and splitting it into windows of length W, the total number of windows for each dataset is L/W where each window now consists of three windows, each of length W, one for each axis.

With the data normalized and divided into fixed-length windows of length W, we can calculate the time-series features using the tsfresh library. This calculates 77 features for each axis of every window. Many of these features are also calculated using different parameter values and combinations, bringing the total to 794 features for each axis of every window. This is shown in Table III. Given that each accelerometer has three axes and we have either 1 or 2 of these sensors per horse, depending on the dataset, the total number of calculated features per window comes to 4764 features (3 2 × 794) for the legs dataset with two IMUs and 2382 features (3 × 794) for the neck dataset with just one IMU. An overview of the calculated features with some examples can be found in Table IV. A full overview of all extracted features and which parameters are used can be found on the tsfresh documentation page.[1]

IV. ALGORITHM

In this section, we present the proposed algorithm in three steps: 1) feature selection; 2) cluster mapping; and 3) classification.

A. Feature Selection

Before using this high-dimensional data, consisting of 794 features per axis, as the input to our clustering and

[1] https://tsfresh.readthedocs.io/en/latest/text/list_of_features.html

TABLE III
GRAPHICAL REPRESENTATION OF THE TSFRESH FEATURE CALCULATORS HIERARCHY

	77 feature calculators	794 features
tsfresh library	*autocorrelation*	$lag = 0$ $lag = 1$ ...
	fft_aggregated	$aggtype = centroid$ $aggtype = variance$ ...
	range_count	$min = -1,\ max = 1$ $min = 0,\ max = 10^{12}$ ...
	...	...

TABLE IV
OVERVIEW OF THE FEATURES THAT ARE EXTRACTED BY THE TSFRESH LIBRARY

Type	Features
Statistics	Minimum, maximum, mean, standard deviation,...
Shape	Length, trend, number of peaks, skewness, symmetry, ...
Frequency	CWT coefficients FFT coefficients, FFT statistics, ...
Entropy	Binned entropy, FFT entropy, permutation entropy, ...
Other	Autocorrelation, C3 non-linearity, Friedrich coefficients, ...

classification algorithm, we must first reduce the number of features. This should be done to avoid the curse of dimensionality, in which the volume of the search space for the clustering algorithm increases exponentially as the number of dimensions increases [27]. To reduce the amount of dimensions, a feature selection technique is applied that uses the Kendall rank correlation coefficient to indicate the significance of each feature to the classification labels [13].

The Kendall rank correlation coefficient τ is calculated as follows:

$$\tau = \frac{n_c - n_d}{\binom{n}{2}} \tag{3}$$

where n_c stands for the number of concordant (α, β) pairs, n_d stands for the number of discordant (α, β) pairs, and n is the total number of (α, β)-tuples. We call a pair (α_i, β_i) and (α_j, β_j) concordant if either $\alpha_i > \alpha_j \wedge \beta_i > \beta_j$ or $\alpha_i < \alpha_j \wedge \beta_i < \beta_j$ holds. Otherwise, we call the pair discordant. A visualization of this concept is given in Fig. 4, where green points and red points indicate all points that are, respectively, concordant and discordant with (α_i, β_i). In our use case, this translates as follows. First, we have to translate our labels to numbers; we use 0 for standing, 1 for walking, 2 for trotting, and 3 for cantering; and we call the vector containing all numerical labels for the windows used during calibration as Y. Y_i is then the numerical label (0, 1, 2, or 3) corresponding to the ith window for each axis. The vector containing all features for every window used during calibration is named X. Here, $X_{i,k}$ indicates the kth feature of the ith window. For each feature, we calculate the number of concordant and discordant pairs as given in the following

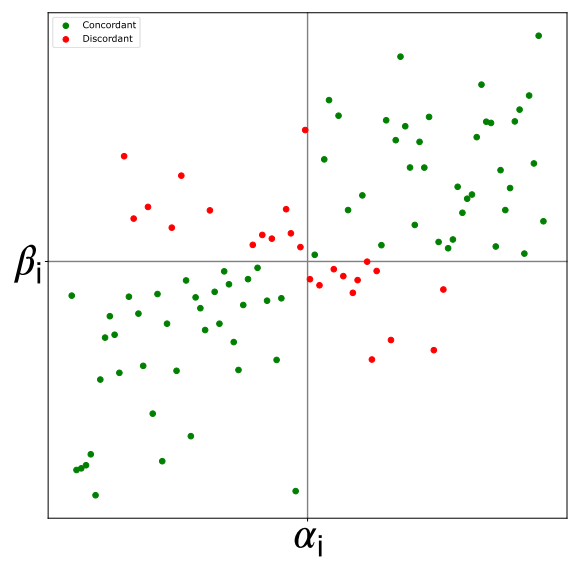


Fig. 4. Visualization of the concept of concordant and discordant pairs with respect to the point (α_i, β_i).

equation:

$$\begin{aligned}\text{concordant}_{i,k} = &((X_{i,k} < X_{j,k}) \wedge (Y_i < Y_j)) \\ &\vee ((X_{i,k} > X_{j,k}) \wedge (Y_i > Y_j))\end{aligned} \tag{4}$$

$$\begin{aligned}\text{discordant}_{i,k} = &((X_{i,k} < X_{j,k}) \wedge (Y_i > Y_j)) \\ &\vee ((X_{i,k} > X_{j,k}) \wedge (Y_i < Y_j)).\end{aligned} \tag{5}$$

Once the number of concordant and discordant pairs is calculated for each feature, it can be used to calculate the feature significance with respect to the labels. How close the value of the calculated Kendall rank correlation coefficient τ_k

Algorithm 1 Algorithm to Compute the Kendall Tau Coefficient

Input:

- **X**: list of all features calculated using tsfresh for every window of each axis in the calibration dataset
- **K**: Total number of features present in X.
- **Y**: list containing the numerical labels for every window in the calibration dataset.
- **N**: number of features to select.

Output:

- **F**: List containing the N most significant features.

Step 1: Calculate the Kendall Tau coefficient and corresponding p-value for every feature using the calibration dataset

```
for k in 1..K do
    n_c = 0
    n_d = 0
    for i = 1.. | X | do
        for j = i + 1.. | X | do
            if (X_{i,k} < X_{j,k} ∧ Y_i < Y_j) ∨ (X_{i,k} > X_{j,k} ∧ Y_i > Y_j) then
                n_c += 1
            else
                n_d += 1
    τ_k = (n_c − n_d) / (n · (n − 1) / 2)
    p_k = CalculatePValue(τ_k);
```

Step 2: Select the N features with the lowest p-value as F.

for a feature k is to zero indicates the correlation between the feature k and the labels Y_i, with a value of zero indicating no correlation at all. The higher (lower) τ_k is, the more positively (negatively) correlated the corresponding feature is to the label. To use the value of τ_k for feature selection purposes, CalculatePValue implements a normal approximation statistical test, where $\tau_k = 0$ means no correlation between the feature and the labels and is picked as the null hypothesis [28]. A low p-value for this test indicates a low probability of observing the obtained results if the null hypothesis was true. In other words, a low p-value indicates a high probability of correlation between the feature and the labels. A simple algorithm of this approach for feature selection can be found in Algorithm 1. For the actual calculation of these values, we used the implementation from the target_real_feature_real_test function in the tsfresh library.

B. Cluster Mapping

To perform the actual classification in this reduced feature space consisting of just the selected most significant features F, we used a simple clustering algorithm that does not require a training phase. Before applying this clustering algorithm, we first standardized the selected features using the Z-score, and this brings each feature to a mean value of 0 and a standard deviation of 1 [29]. During calibration, the centroids are calculated for every gait class $a \in \{0, 1, 2, 3, 4\}$ in Y. The equation for calculating these cluster centroids is given in the following equation:

$$C_a = \frac{1}{N_a} \sum_{i=1}^{N_a} X'_{a,i} \tag{6}$$

where N_a corresponds to the number of windows for the ath class, X' is the calibration dataset X in the reduced feature space consisting of features F, and $X'_{a,i}$ is the ith window in the subset of X' that only contains windows of class a. Algorithm 2 shows the clustering initialization step of the calibration phase of our algorithm.

C. Classification

When a new window S, containing W accelerometer samples of an unknown equine gait, needs to be classified, we calculate the Euclidean distance between the window and each cluster centroid in feature space F and assign the window to the class of the closest cluster centroid (7). The equation to find the numerical class a that belongs to the cluster centroid that is closest to the window S is shown in (8). In Fig. 5, a visualization of this approach is given. S indicates the new window that needs to be classified, and C_a indicates the centroids of each label cluster. As C_1 is the closest cluster centroid to this sample, based on the Euclidean distance, it is assigned to the class that corresponds to C_1

$$d(S, C_a) = \sqrt{\sum_{k=1}^{|F|} (S_k - C_{a,k})^2} \tag{7}$$

Algorithm 2 Algorithm to Calculate the Cluster Centroids for Every Gait Class in the Calibration Dataset

Input:

- $\boldsymbol{X'}$: Calibration dataset in feature space F.
- $\mathbf{Y}$: list containing the numerical labels for every window in the calibration dataset.

Output:

- $\boldsymbol{C_a}$: Location of the cluster centroid in feature space F for every class a that is in Y.

for a *in* `unique(Y)` **do**
  for k *in* F **do**
    $C_{a,k} = \frac{1}{\mid X'_{a,k} \mid} \sum_{i=1}^{|X'_{a,k}|} X'_{i,k}$

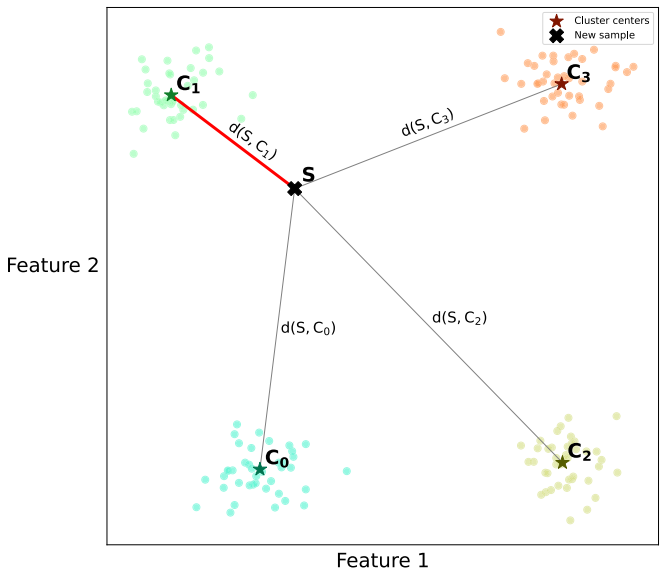


Fig. 5. Visualization of the classification algorithm.

$$a = \underset{a}{\operatorname{argmin}}\, d(S, C_a). \tag{8}$$

The proposed data-efficient algorithm is less complex compared to classical machine learning algorithms such as CNN, SVM, and RF because it does not require model training and it only requires 3 min of labeled data while still achieving the same level of performance.

V. Results

For evaluation of our approach, we used both an individual approach and a global approach. For the individual approach, we divided the dataset of each individual horse into a calibration dataset containing a few labeled windows and the evaluation dataset containing the remaining data. We then used the labeled windows from the calibration dataset to select the most relevant features and assign the cluster centroids. Thus, each horse has its own unique parameters derived from its own individual calibration dataset. For this approach, we present the results for the impact of: 1) number of labeled windows used and number of features used (Section V-A); 2) using the full x, y, and z signal for calculating the tsfresh features or using an aggregated version using the vector norm (Section V-B); 3) the window length and sampling rate (Section V-C); and 4) number and location of the sensors (Section V-D). For the global approach, we split the entire dataset into a part containing horses used for calibration and a part containing data from horses for evaluation. In this way, the algorithm's parameters are set using data from horses in the calibration dataset and the evaluation dataset consists of data from horses that are unseen by the algorithm. For the calibration and evaluation, we always used 5-s windows resampled down from the original 50 to 10 Hz unless otherwise stated. During calibration, we used ten labeled windows per class and selected the top 100 most relevant features unless otherwise stated as these values have shown to be where the performance of the algorithm is reaching a maximum (Section V-E). Finally, we also compared this global approach with several classical approaches such as SVMs, RFs, and CNNs (Section V-F).

A. Individual-Based Models

First, we looked at the influence of the number of selected features and the number of labeled samples used in the algorithm had on the accuracy of our approach. To do this, we selected different amounts of labeled windows per class for each horse to use during calibration. All other data for that horse is then used as validation data and was unseen by our algorithm.

In Fig. 6, we plotted the mean accuracy and minimum and maximum accuracies over all 13 horses with respect to the number of features that were selected and that thus were used for the clustering algorithm. For this step, we set the number of labeled windows per class that were used for each horse fixed to ten samples. We can clearly observe that at around 25 selected features, a peak is reached, after this the accuracy drops off again but reaches a stable plateau of 95% at 75 selected features. The peak at 25 with a subsequent small drop in accuracy could be attributed to the curse of dimensionality as the cluster becomes more sparsely populated as the number of dimensions increases.

Fig. 7 shows how the accuracy corresponds to the number of labeled windows per class used to find the most relevant features and calculate the cluster centroids. For this evaluation, we fixed the number of selected features at 100. Here, the accuracy reaches the plateau of 95% accuracy at just five labeled windows per class. This indicates that for the individual models, an accuracy of 95% can be achieved while only requiring five labeled windows per class and thus 20 labeled windows in total because of the four gait classes that are used.

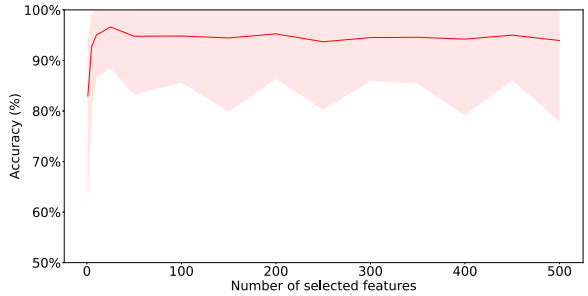


Fig. 6. Accuracy range and mean in a function of number of selected features (one model per horse, 10-Hz sampling rate, 5-s window length, and ten labeled windows per class).

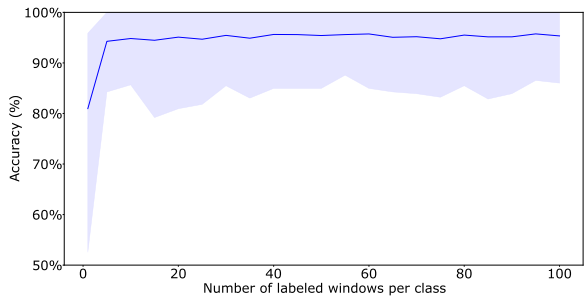


Fig. 7. Accuracy range and mean in a function of number of labeled windows per class used for calibration (one model per horse, 10-Hz sampling rate, 5-s window length, and 100 selected features).

With each window consisting of 5 s, this results in a calibration dataset size of 100 s.

As the dataset contains a slight imbalance between the four different gait types, we also calculated the balanced accuracy [30]. The formula for this balanced accuracy is given in (11). For a configuration of ten labeled windows per class and 100 selected features, this balanced accuracy was 96.8%

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{9}$$

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \tag{10}$$

$$\mathrm{balanced\_accuracy} = \frac{\mathrm{TPR} + \mathrm{TNR}}{2}. \tag{11}$$

B. Impact of Number of Sensors

To reduce both the equipment cost and the computational cost, it could prove beneficial to use data from only one sensor attached to one leg instead of one sensor attached to each of the two front legs. Another adjustment that could prove useful for further reducing the computational cost of the algorithm could be using the acceleration vector norm instead of the three separate x, y, and z acceleration values. To calculate this vector norm, we use (1), and this reduces the amount of data by a factor 3 as the values for each of the three axes are now replaced with just the singular vector norm value. In Fig. 8, a boxplot is given that compares the performance of using data from both legs to using data from only one leg, for both the full x, y, and z data or using the acceleration vector norm. From this figure, it is clear that using the acceleration vector norm instead of the full acceleration results in a wider

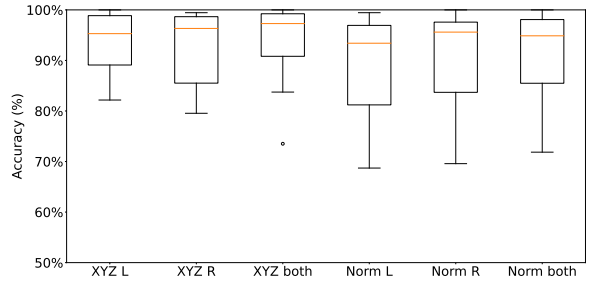


Fig. 8. Accuracy range for models using one leg or both legs and also for models using either the raw *XYZ* inputs or the acceleration vector norm.

range of accuracy values; however, the median accuracy over all 13 horses remained similar. For one leg versus two legs, we can also note a small difference in range between the different accuracies, with the one-leg approach resulting in a larger range of values. We can thus conclude that, when the highest achievable accuracy is required, using the full xyz vector and a sensor on both legs is recommended. However, when the cost of the system and battery life constraints become an issue, it might prove beneficial to take an accuracy penalty of 5%–10% and only use a sensor on one leg and/or use the vector norm instead of the full xyz vector.

C. Impact of Window Length and Sampling Rate

Fig. 9 shows two confusion matrices, Fig. 9(a) shows the results for the two legged approach, and Fig. 9(b) shows the one results for the one-leg approach. The values for the one-leg approach are obtained by taking the average of the accuracies for both the left and the right leg. In these confusion matrices, the average accuracies are given for all 13 horses, using the individual model approach, when we vary both the sampling rate and the sample window length. For both one and two legs, it can be seen that when either the window length or the sampling rate is increased, the accuracy increases as more data become available to the model. However, this increase in accuracy also has an impact on the computational cost of the algorithm. Our time series will increase in size as the sampling rate increases, as more xyz acceleration samples get added per unit of time with a higher sampling rate. This will increase the computational cost of the feature calculators. When we increase the window length, we will also increase the size of our time series, but we will also perform fewer classifications as there will be more time between the subsequent windows. This could be either positive for the computational cost or negative, depending on whether the reduced cost of fewer classifications per unit of time outweighs the added cost of longer windows to the feature calculators. In order to achieve the best results, we can observe that a minimum of 3-s windows and a 10-Hz sampling rate are required.

D. Impact of Sensor Location

To evaluate the generalization ability of our proposed algorithm, we also evaluated our approach on a publicly available dataset containing equine activities captured by a neck-mounted IMU. This dataset contained labeled data of

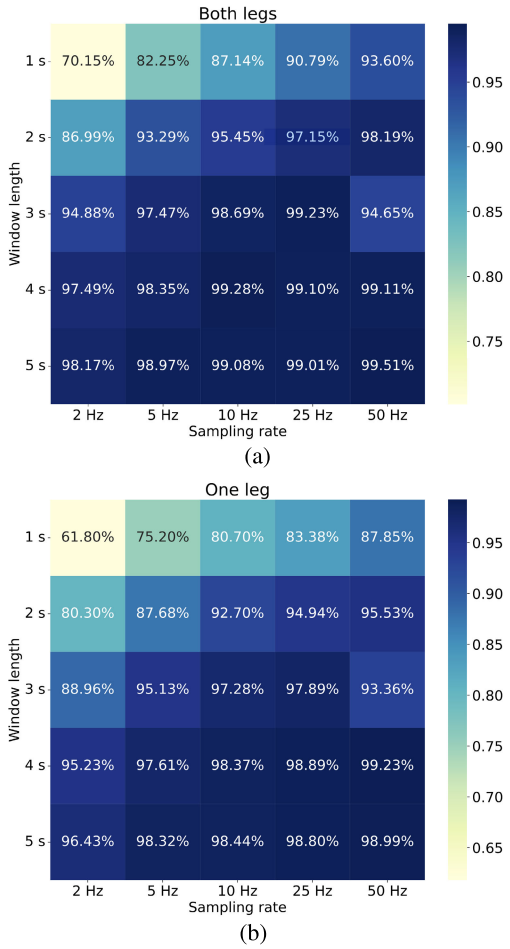


Fig. 9. Heatmap showing the impact of the sampling rate and window length on the accuracy for both a model using both legs and a model using only one leg. (a) Accuracies of a model that uses both legs for different window lengths and sampling rates. (b) Accuracies of a one-leg model for different window lengths and sampling rates.

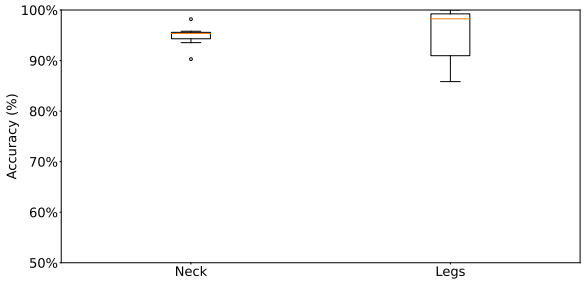


Fig. 10. Boxplot of the accuracy of a neck-mounted sensor and leg-mounted sensors.

11 horses, but only 7 were used in this study as the other four lacked sufficient data for all four gaits. We subsampled both the neck and leg dataset down to 10 Hz and used 5-s windows, as Section V-C has proven that this will give us a very high accuracy while still limiting the computational time in comparison to using the full 50-Hz signal. In Fig. 10, we show the boxplots for the accuracies over all horses in each of the two datasets. It can be seen that the average performance is similar for both datasets. However, the dataset containing leg data has a larger range of accuracy values over all horses, whereas on the neck dataset, our approach performs similarly

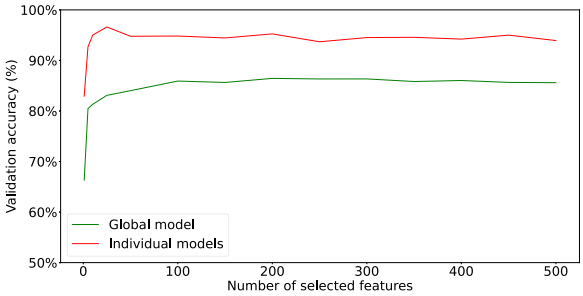


Fig. 11. Average accuracy of a global model and the average accuracy of one individually trained model per horse in terms of number of selected features.

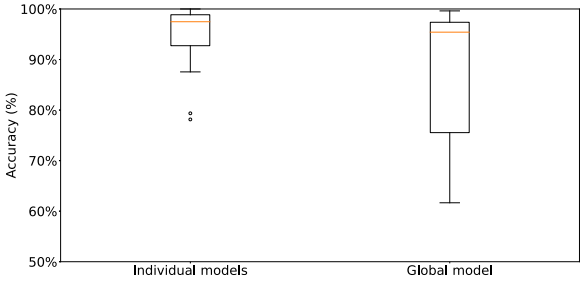


Fig. 12. Boxplot of the accuracy of a global model and one individually trained model per horse.

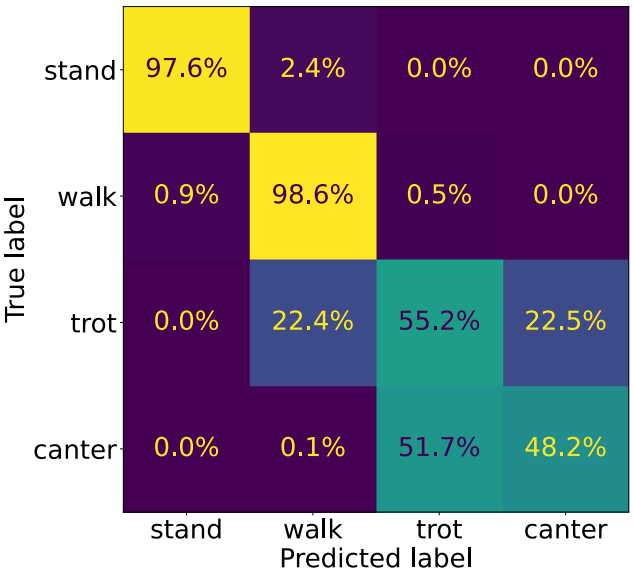


Fig. 13. Confusion matrix showing the mean accuracies of the global model for the four different types of gait.

over all horses. Intuitively, we would suspect that the legs would give much better information for detecting the gait of the horse in comparison to the neck. This unexpected result could be attributed to the fact that there are fewer usable horses in the neck dataset, and evaluating on this dataset thus will correspond less to a general dataset or to the quality of the data and the labeling itself.

E. Global Model

The drawback of using an individualized approach where each horse has its own parameters is that these parameters need to be recalibrated every time a different horse needs to be tracked. A solution to this would be to find parameters that work for all horses. To evaluate this global model approach,

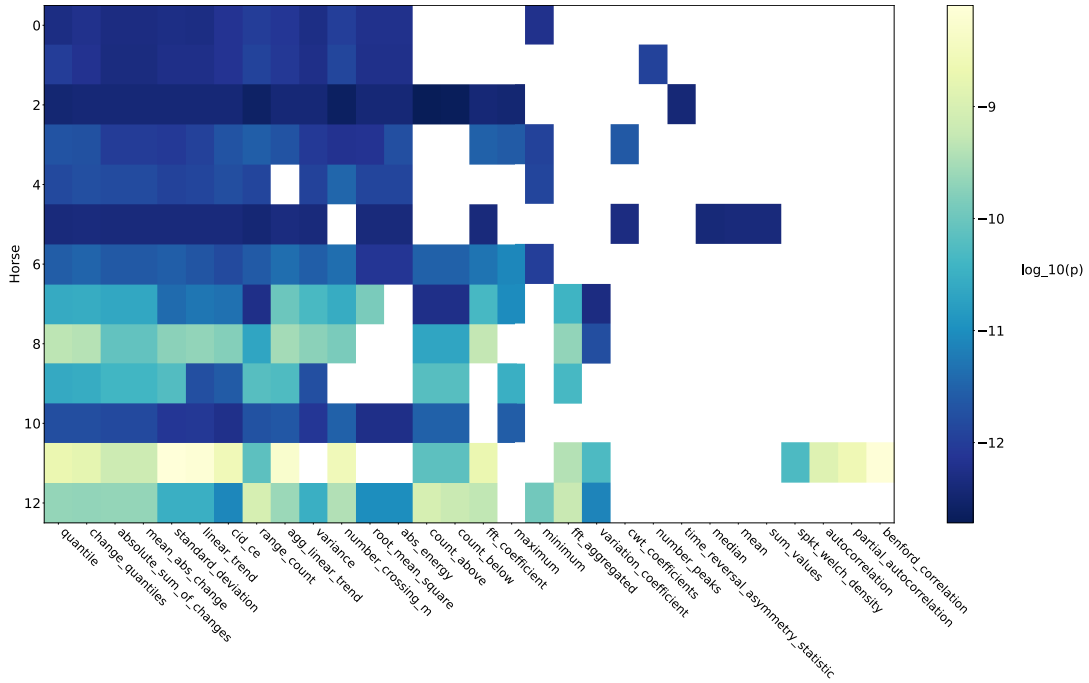


Fig. 14. Overview of the selected feature types per horse, color indicates the base 10 log of the *p*-value of the average feature significance.

we used threefold cross validation where we split the entire 13 horses leg dataset into threefold each containing 2/3 of the horses for calibration and 1/3 of the horses to evaluate our approach on. To select our subset of labeled windows for feature selection and clustering, we shuffle all windows from the calibration dataset and randomly select the samples from this shuffled dataset containing data from several different horses. To further evaluate the stability of this global approach we performed ten iterations for each fold, so in total, we performed 30 calibration and evaluation runs.

In Fig. 11, we plotted the average accuracy with respect to the number of selected features for both the individual and the global approach. In Fig. 12, a boxplot is given showing the spread of the accuracy values for both approaches over all runs. From these graphs, it is clear that trying to find a global model will result in a drop in performance, with a lower average accuracy as well as a lot more variation in the performance of the model over different folds and iterations. These results indicate that no general parameters that would work for all horses could be found by our algorithm and thus no general model that only needs to be calibrated once and can subsequently be used for all horses exists with our algorithm.

When we look at the confusion matrix in Fig. 13, we observe that almost all wrong classifications by this global model occur for the trot and canter classes.

To investigate why this difference in performance between an individual and a global approach occurred, we looked at what classes of features got selected for each horse in the individual approach. A small overlap in selected features between different horses might indicate that the most ideal features are highly horse dependent, making a global set of optimal features difficult to obtain.

In Fig. 14, we plotted the average p-value of the Kendall rank correlation coefficient for each class of features selected for every horse when we select the 100 most relevant features. From this, we can see that for the first seven horses, there is a strong overlap between the selected features and all features have very low p-values, indicating a strong correlation between the features and the labels. However, for the other six horses, we notice less overlap between the features and also significantly higher p-values, indicating less correlation between the features and the labels. As the data for the first seven horses were captured at a different day and location as the data from the other six horses, this difference could be attributed to a difference in data or labeling quality. When we look at the boxplot for a global approach in Fig. 15, we can see that is intuition is correct as the accuracy when we calibrate and evaluate on horses 7–12 is significantly lower than when we calibrate and evaluate using horses 0–6. To further find out what caused this difference in performance, we plotted the data distribution for the trot and canter class for both groups of horses using boxplots. This is shown in Fig. 16, and it is clear that the distributions between both groups are significantly different. The difference in performance between both groups and the inability to find global parameters for all horses in our dataset indicates that our approach is very sensitive to small perturbations of the data. Special care should thus be taken during data capturing to make sure that both training and evaluation data are captured in exactly the same manner and follows the same distribution.

F. Comparison With Classical Machine Learning

Finally, we compared our data-efficient approach with different classical machine learning classifiers: an SVM, an RF, and a recent, state-of-the-art, CNN developed specifically for

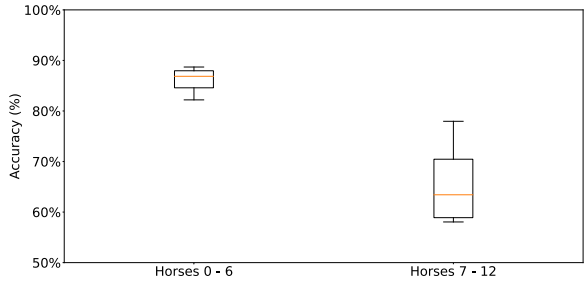


Fig. 15. Boxplot of the accuracy of a global model for horses 0–6 and horses 7–12.

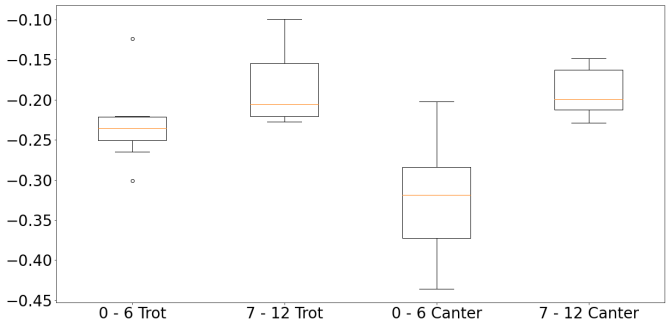


Fig. 16. Boxplot of the distribution of the mean values per horse for each axis and each window for both canter and trot.

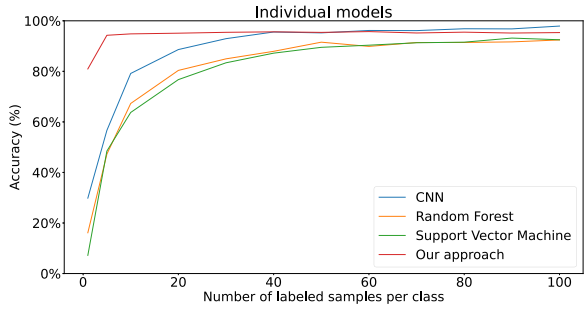


Fig. 17. Comparison of the accuracy of several machine learning techniques with respect to the number of labeled samples per class used during training and calibration.

equine gait detection [14]. For the RF 100 subtrees and the Gini impurity, a selection criterion was used. For the SVM classifier, the radial basis function kernel was used.

Fig. 17 plots the accuracy curve for all four approaches, using an individual approach, with respect to the number of labeled windows per class used during the training and calibration. From this, it can be seen that our approach has an edge on the other models up to around 40 labeled windows per class when the CNN starts outperforming it. The RF and SVM, however, do not achieve equal or higher performance than our approach in the 1–100 labeled samples per class range that was evaluated.

Fig. 18 shows the same type of graph but now for a global approach instead of the individual approach. For the global approach, it can be seen that our approach only outmatches the other models up to five labeled samples per class. This lies in line with what already was discovered earlier, our approach has difficulty with finding parameters that work for all horses in the dataset.

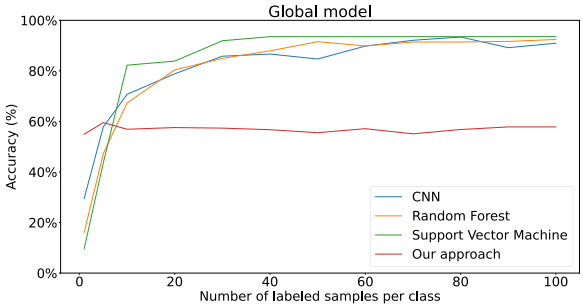


Fig. 18. Comparison of the accuracy of several machine learning techniques with respect to the number of labeled samples per class used during training and calibration.

VI. Conclusion

In this article, a data-efficient solution for equine gait detection is proposed, which uses a combination of supervised feature selection and a supervised clustering approach. By making use of the tsfresh Python library, containing close to 800 feature calculators and combining it with the Kendall rank correlation coefficient, a supervised feature significance test, the need for both automated feature learning through neural networks, or the need for manual feature engineering was removed. This eliminated the need for domain expertise as well as complicated parameter optimization and tuning, as is required for feature engineering and deep learning. This approach managed to achieve accuracies that were similar to classical machine learning algorithms such as CNNs, SVMs, and RFs while only requiring a fraction of the labeled data to train/calibrate. We also proved that our approach could easily be adjusted for different sensor locations, such as when the sensor is placed on the neck of the horse. How the performance of the approach changes when the dimensionality of the input data is reduced by only using the vector norm or just one leg instead of two legs was also investigated, and the different average accuracies over a combination of sampling rate and classification window sizes were also shown. However, the drawback of this approach is that the most optimal parameters in terms of selected features and cluster centroid locations differ from horse to horse. To this end, these parameters should be set for each horse individually, a potential time and labor-intensive job. When we attempted to find global parameters that could work for a variety of different horses, we noticed a 15% drop in average accuracy as well as a much wider range in terms of stability over different iterations and horses. When we further investigated this behavior, we noticed a difference between both which feature classes got selected as well as in the p-values of the selected feature classes between the first six and last seven horses in the dataset. Diving deeper into the data, we observed a significant difference in the data distribution between the two groups of horses. As these came from two different locations and days, it could be that this occurred due to slight differences in the mounting procedure of the accelerometers. This result showed that while our approach achieves good performance, matching or surpassing other methods, it is very sensitive to small perturbations in the data used.

Further research should focus on finding improvements to increase the performance and stability of the approach when

calibrated and evaluated over multiple different horses. This could be either done by investigating and increasing the data quality and/or further enhancements to the algorithm, either by replacing or improving upon the clustering algorithm using deep learning, or by improving on the calculated features by extending them with learned features through the use of either autoencoders or CNNs. Another track could be to find optimizations to the individual calibration phase, and one possible optimization could be to calculate and select the most relevant features using a large calibration dataset and only using data from individual horses to find the cluster centroids for each class.

Declaration of Competing Interests

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 article.

Acknowledgment

This work was executed within the imec.icon project Hoof-MATE, a research project bringing together academic researchers and industry partners. The Hoof-MATE project was co-financed by imec and received project support from Flanders Innovation & Entrepreneurship under Project HBC.2018.0536.

References

[1] P. Cross, *Global Horse Statistics Internal*. Gulep, ON, Canada: HiPoint Agro Bedding Corp, Feb. 2019.

[2] (2010). *Totilas Sold to Paul Schockemöhle of Germany*. Accessed: Mar. 8, 2022. [Online]. Available: https://dressage-news.com/2010/10/14/totilas-sold-to-paul-schockemohle-of-germany/

[3] Fédération Equestre Internationale. (2022). *WELcome to the FEI Database*. [Online]. Available: https://data.fei.org/Default.aspx

[4] M. M. Rodgers, V. M. Pai, and R. S. Conroy, "Recent advances in wearable sensors for health monitoring," *IEEE Sensors J.*, vol. 15, no. 6, pp. 3119–3126, Jun. 2015.

[5] B. Van Herbruggen, J. Fontaine, A. Eerdekens, M. Deruyck, W. Joseph, and E. D. Poorter, "Feasibility of wireless horse monitoring using a kinetic energy harvester model," *Electronics*, vol. 9, no. 10, p. 1730, Oct. 2020.

[6] R. Jurdak, P. Corke, D. Dharman, and G. Salagnac, "Adaptive GPS duty cycling and radio ranging for energy-efficient localization," in *Proc. 8th ACM Conf. Embedded Netw. Sensor Syst.*, Nov. 2010, pp. 57–70.

[7] J. W. Kamminga, H. C. Bisby, D. V. Le, N. Meratnia, and P. J. M. Havinga, "Generic online animal activity recognition on collar tags," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput. ACM Int. Symp. Wearable Comput.*, Sep. 2017, pp. 597–606.

[8] T. D. Waele et al., "Internet of animals: Unsupervised foaling detection based on accelerometer data," in *Proc. IEEE 7th World Forum Internet Things (WF-IoT)*, Jun. 2021, pp. 315–320.

[9] W. Gomaa, "Statistical and time series analysis of accelerometer signals for human activity recognition," in *Proc. 14th Int. Conf. Comput. Eng. Syst. (ICCES)*, Dec. 2019, pp. 351–356.

[10] S. Ha and S. Choi, "Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 381–388.

[11] M. Kim, C. Y. Jeong, and H. C. Shin, "Activity recognition using fully convolutional network from smartphone accelerometer," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2018, pp. 1482–1484.

[12] M. Christ, N. Braun, J. Neuffer, and A. W. Kempa-Liehr, "Time series feature extraction on basis of scalable hypothesis tests (tsfresh—A Python package)," *Neurocomputing*, vol. 307, pp. 72–77, Sep. 2018.

[13] H. Abdi, "The Kendall rank correlation coefficient," in *Encyclopedia of Measurement and Statistics*. Thousand Oaks, CA, USA: Sage, 2007, pp. 508–510.

[14] A. Eerdekens, M. Deruyck, J. Fontaine, L. Martens, E. D. Poorter, and W. Joseph, "Automatic equine activity detection by convolutional neural networks using accelerometer data," *Comput. Electron. Agricult.*, vol. 168, Jan. 2020, Art. no. 105139.

[15] A. Mao, E. Huang, H. Gan, R. S. V. Parkes, W. Xu, and K. Liu, "Cross-modality interaction network for equine activity recognition using imbalanced multi-modal data," *Sensors*, vol. 21, no. 17, p. 5818, Aug. 2021.

[16] H. B. Davíosson, T. Rees, M. R. Ólafsdóttir, and H. Einarsson, "Efficient development of gait classification models for five-gaited horses based on mobile phone sensors," *Animals*, vol. 13, no. 1, p. 183, Jan. 2023.

[17] M. Rana, A. Rahman, and D. Smith, "A semi-supervised approach for activity recognition from indoor trajectory data," 2023, *arXiv:2301.03134*.

[18] E. Casella, A. R. Khamesi, and S. Silvestri, "Smartwatch application for horse gaits activity recognition," in *Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, Jun. 2019, pp. 409–416.

[19] J. W. Kamminga, D. V. Le, and P. Havinga, "Towards deep unsupervised representation learning from accelerometer time series for animal activity recognition," in *Proc. 6th Workshop Mining Learn. From Time Ser., (MiLeTS)*, 2020, pp. 1–8.

[20] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 478–487.

[21] F. J. Baldán, D. Peralta, Y. Saeys, and J. M. Benítez, "SCMFTS: Scalable and distributed complexity measures and features for univariate and multivariate time series in big data environments," *Int. J. Comput. Intell. Syst.*, vol. 14, no. 1, pp. 1–15, Dec. 2021.

[22] S. Luqian and Z. Yuyuan, "Human activity recognition using time series pattern recognition model-based on tsfresh features," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, Jun. 2021, pp. 1035–1040.

[23] M. Raychoudhury, H. Yu, and J. D. Kiper, "ActiviSee: Activity transition detection for human users through wearable sensor-augmented glasses," in *Proc. Adjunct 10th Int. Workshop Hum. Activity Sens. Corpus Appl. (HASCA) Conjunct With ACM (UbiComp)*, 2023, pp. 1–8.

[24] S. K. Singh and A. Chaturvedi, "Leveraging deep feature learning for wearable sensors based handwritten character recognition," *Biomed. Signal Process. Control*, vol. 80, Feb. 2023, Art. no. 104198.

[25] J. W. Kamminga, L. M. Janßen, N. Meratnia, and P. J. M. Havinga, "Horsing around—A dataset comprising horse movement," *Data*, vol. 4, no. 4, p. 131, Sep. 2019.

[26] *AX3 User Manual*. Accessed: Apr. 20, 2023. [Online]. Available: https://axivity.com/userguides/ax3/

[27] M. Verleysen and D. François, "The curse of dimensionality in data mining and time series prediction," in *Proc. Int. Work-Conf. Artif. Neural Netw.* Cham, Switzerland: Springer, 2005, pp. 758–770.

[28] H. Abdi, "The Kendall rank correlation coefficient," in *Encyclopedia of Measurement and Statistics*. Thousand Oaks, CA, USA: Sage, 2007, pp. 508–510.

[29] H. Abdi, "Z-scores," *Encyclopedia Meas. Statist.*, vol. 3, pp. 1055–1058, Jan. 2007.

[30] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, "The balanced accuracy and its posterior distribution," in *Proc. 20th Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 3121–3124.

Timo De Waele received the M.Sc. degree in computer science from Ghent University, Ghent, Belgium, in 2020, where he is pursuing the Ph.D. degree with the iWINE-UGent/IDLab/imec Research Group.

In 2021, he became a Ph.D. Fellow of the Research Foundation—Flanders (FWO-V), Brussel, Belgium. His scientific work mainly focuses on developing unsupervised and semisupervised machine learning algorithms using data from biosignals, such as electrocardiograms and movement data from accelerometers and gyroscopes.

Adnan Shahid (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in computer engineering from the University of Engineering and Technology at Taxila, Taxila, Pakistan, in 2006 and 2010, respectively, and the Ph.D. degree in information and communication engineering from Sejong University, Seoul, South Korea, in 2015.

From March 2015 until August 2015, he worked as a Postdoctoral Researcher at Yonsei University, Seoul. From September 2015 to June 2016, he worked as an Assistant Professor at Taif University, Taif, Saudi Arabia. He is currently working as a Senior Researcher at IDLab, which is a core research group of imec with research activities embedded within Ghent University and the University of Antwerp, Antwerp, Belgium. He is the lead of "artificial intelligence (AI)/machine learning (ML) for wireless," which is a research group at iWINE-UGent/IDLab/imec. He is leading the Erasmus+ DigiHealth-Asia project for the term 2021–2024 and the European Space Agency (ESA) MRC100 Project for the term 2022–2023. He is also involved in the imec ICON 5GECO Project for the term 2022–2023 and the MSCA Staff Exchanges EVOLVE Project for the term 2022–2025. He has been the Technical Manager of the ESA CODYSUN Project. He has also been involved in several projects: DARPA Spectrum Collaboration Challenge (SC2); European H2020 Research Projects, such as eWINE and WiSHFUL; and national projects, such as SAMURAI, IDEAL-IOT, Cognitive Wireless Networking Management, Smartwaterways, and UWB-AAA. His research interests include AI/ML for wireless communications and networks, resource optimization, interference management, self-organizing networks, the Internet of Things, localization, and 5G/6G networks.

Dr. Shahid is a Voting Member and the Secretary of IEEE P1900.8—Standard for Training, Testing, and Evaluating Machine-Learned Spectrum Awareness Models. He is also serving as an Associate Editor for various journals, such as IEEE ACCESS and the *Journal of Networks and Computer Application (JNCA)* (Elsevier).

Daniel Peralta (Member, IEEE) received the M.Sc. degree in soft computing and intelligent systems and the Ph.D. degree in computer science from the University of Granada, Granada, Spain, in 2012 and 2016, respectively.

He is currently a Postdoctoral Researcher at IDLab, Department of Information Technology, Ghent University and imec, Ghent, Belgium. He has published more than 20 articles in international journals. His research interests include machine learning, time-series analysis, biological imaging data analysis, biometrics, large-scale datasets, and parallel and distributed computing.

Anniek Eerdekens received the M.Sc. degree in biomedical engineering from Ghent University, Ghent, Belgium, in 2019, where she is pursuing the Ph.D. degree with the Department of Information Technology.

She began working as a Research Assistant with the UGent-INTEC (Ghent University–Department of Information Technology) WAVES Research Group, Ghent, in August 2019, specifically advanced machine learning models for detecting animal behavior. Her research focuses on animal Internet of Things (IoT).

Margot Deruyck (Member, IEEE) received the M.Sc. degree in computer science engineering and the Ph.D. degree from Ghent University, Ghent, Belgium, in 2009 and 2015, respectively.

Since 2009, she is a member of the WAVES group (Department of Information Technology), Ghent University, where she became an Assistant Professor in Advanced Network Planning for (beyond) 5G and Internet of Things (IoT) applications in 2021. She is a Postdoctoral Fellow of the Research Foundation—Flanders (FWO-V), Brussel, Belgium. Her scientific work focuses on green wireless access networks (minimal power consumption and/or human exposure), unmanned aerial vehicle (UAV)-based wireless network design, and animal IoT.

Frank A. M. Tuyttens received the B.Sc. (Hons.) degree in biology from Edinburgh University, Edinburgh, U.K., in 1994, and the D.Phil. degree in zoology from Oxford University, Oxford, U.K., in 1999.

He is the Head of the Farm Animal Welfare and Behavior Research Group, Flemish Research Institute of Agriculture, Fisheries and Food (ILVO), Belgium. Since 2011, he has been appointed as a Visiting Professor at the Faculty of Veterinary Medicine, Ghent University (UGent), Ghent, Belgium, for lecturing and researching on farm animal ethology and welfare. He is a principal investigator of many national and regional research projects and currently participates in several EU research consortia (e.g., PPILOW, CHICKENSTRESS, and aWISH). He has published more than 180 A1 papers and is currently (co)supervising 12 Ph.D. students. His main research and teaching interest is on monitoring and improving the welfare of farm animals, including cattle, pigs, poultry, and rabbits.

Prof. Tuyttens is a member of the Steering Group of the Flemish Council for Animal Welfare, the ILVO Ethics Commission for Animal Experiments, and various scientific societies [e.g., International Society for Applied Ethology (ISAE), European Federation of Animal Science (EAAP), Universities Federation for Animal Welfare (UFAW), and Animal Welfare Research Network (AWRN)] and an Editor of *Animal*.

Eli De Poorter received the master's degree in computer science engineering from Ghent University, Ghent, Belgium, in 2006, and the Ph.D. degree from the Department of Information Technology, Ghent University, in 2011.

He is a Professor at the IDLab Research Group, Ghent University and imec, Ghent. Since 2017, he has been with the IMEC Research Institute, Ghent. His team performs research on wireless communication technologies, such as (indoor) localization solutions, wireless Internet of Things (IoT) solutions, and machine learning for wireless systems. He performs both fundamental and applied research. For his fundamental research, he is currently the coordinator of several research projects (H2020, SBO, FWO, GOA, and so on) and has over 200 publications in international journals or in the proceedings of international conferences. For his applied research, he collaborates with (Flemish) industry partners to transfer research results to industrial applications as well as to solve challenging industrial research problems.