

Automatic detection of abnormal behaviour of equines

Anniek Eerdeken

Student number: 01713069

Supervisors: Prof. dr. ir. Wout Joseph, Prof. dr. ir. Eli De Poorter
Counsellors: Dr. ir. Margot Deruyck, Ir. Said Benaissa, Prof. dr. ir. David Plets,
Ir. Jaron Fontaine, Ir. Ben Van Herbruggen

Master's dissertation submitted in order to obtain the academic degree of
Master of Science in Biomedical Engineering

Academic year 2018-2019

Automatic detection of abnormal behaviour of equines

Anniek Eerdeken

Student number: 01713069

Supervisors: Prof. dr. ir. Wout Joseph, Prof. dr. ir. Eli De Poorter
Counsellors: Dr. ir. Margot Deruyck, Ir. Said Benaissa, Prof. dr. ir. David Plets,
Ir. Jaron Fontaine, Ir. Ben Van Herbruggen

Master's dissertation submitted in order to obtain the academic degree of
Master of Science in Biomedical Engineering

Academic year 2018-2019

Preface

With the uprising of health monitoring systems, introducing such technologies in and on a horse is the next logical step. When looking at the financial and emotional aspects of owning a horse, automatic detection of abnormal behaviours would bring piece of mind for the horse owner since he would know that his beloved horse is in good condition all the time. Colic is the most common emergency in equine practise and early identification is critical to obtain a successful outcome in the healing process.

To ensure early detection of colics, an accurate behaviour recognition algorithm is of greatest importance. In this master's dissertation a machine learning algorithm is introduced to classify horse's abnormal behaviour based on accelerometer data. Seven types of activities of six horses are detected and classified. The influence of several parameters that are of importance for both high classification accuracy and long battery life of a wearable device is investigated. Based on this investigation optimal measurement settings are proposed. In addition, data is gathered in a variety of conditions to ensure the realtime evaluation of activity patterns in real world circumstances.

I would like to thank my supervisors Prof. dr. ir. Wout Joseph and Prof. dr. ir. Eli De Poorter for their expertise and encouraging words. Also, I am very grateful for the extraordinary opportunity of doing a Phd after my studies.

Next to them, I would like to thank my counsellors and in particular Dr. ir. Margot Deruyck and Ir. Jaron Fontaine for their guidance, help and feedback in order to complete this dissertation. They have made writing this master's dissertation a very enjoyable experience.

Also, my gratitude goes out to the horses who participated in this study and did their uttermost best during all the experiments.

At last, a special thanks goes out to my boyfriend Massimo, my parents and his parents who always have always supported me in every possible way.

Anniek Eerdeken, June 2019

Permission for Use of Content

The author gives permission to make this master's dissertation available for consultation and to copy parts of this master's dissertation for personal use.

In the case of any other use, the copyright terms have to be respected, in particular with regard to the obligation to state expressly the source when quoting results from this master's dissertation."

Anniek Eerdeken, June 2019

Automatic detection of abnormal behaviour of equines

Anniek Eerdekens

Supervisors: Prof. dr. ir. Wout Joseph, Prof. dr. ir. Eli De Poorter

Counsellors: Dr. ir. Margot Deruyck, Ir. Said Benaissa, Prof. dr. ir. David Plets, Ir. Jaron
Fontaine, Ir. Ben Van Herbruggen

Master's dissertation submitted in order to obtain the academic degree of
Master of Science in Biomedical Engineering

Academic year 2018-2019

Faculty of engineering and architecture
Ghent University

Summary

In recent years, with a widespread of sensors embedded in all kind of mobile devices, human activity analysis is occurring more often in several domains like healthcare monitoring and fitness tracking. This trend did also enter the equestrian world but existing technologies fall short. In this research, a deep learning-based approach for activity detection of equines is proposed to classify abnormal behaviours of equines related to colic. We propose using Convolutional Neural Networks by which features are extracted automatically by using strong computing capabilities. In this way information about the global form of the time series is preserved. Furthermore, we investigate the impact of the sampling frequency, the time series length and the type of underground on which the data is gathered on the recognition accuracy. The model is evaluated on three types of datasets that are compiled of labeled accelerometer data gathered from six different subjects performing seven different activities. The results show that the proposed model demonstrates high performance while requiring low computational cost, low sampling rates and no manual feature engineering.

Keywords

Equines, colic, behaviour detection, wearable devices, machine learning, convolutional neural networks

Automatic detection of abnormal behaviour of equines

Anniek Eerdekens

Supervisors: Prof. dr. ir. Wout Joseph (promotor), Prof. dr. ir. Eli De Poorter (promotor), Dr. ir. Margot Deruyck, Ir. Said Benaissa, Prof. dr. ir. David Plets, Ir. Jaron Fontaine, Ir. Ben Van Herbruggen

Abstract—In this research a behaviour detection algorithm is proposed to detect abnormal activities of equines related to colic. The limits of the trade off between sampling rate, length of measured time and accuracy of the model are extended and the model is tested in a variety of conditions to asses the use in real life circumstances.

Keywords—Equines, colic, behaviour detection, wearable devices, machine learning, convolutional neural networks

I. INTRODUCTION

HORSES have a high value on both a financial and emotional level and therefore it would be a great advantage for the owner to know that his beloved horse is in good condition all the time. Monitoring behaviours can yield important information about the health and welfare of horses. For example, colic is the most common emergency in equine practice and early identification and referral of horses with a surgical lesion is critical to obtain a successful outcome [1].

Today, different technologies exist to detect various parameters such as activity, elevation, heart rate and so on from what conclusions can be drawn regarding the behaviour of the horse but they have failed on the level of accuracy in combination with ease of use and comfort for the horse. The use of small devices such as accelerometers in combination with an accurate classification algorithm could provide an excellent solution for this problem.

In this abstract, we will first propose a classification algorithm based on a convolutional neural network and solve the problem for the early detection of abnormal behaviour of equines. The effect of variations in sampling rate, the length of the measured time interval and various types of underground on the model accuracy was studied.

II. MATERIALS AND METHOD

A. Animals and training arena

The study is carried out in Zutendaal and is conducted with six adult horses of different breeds. All details about the subjects can be found in Table I. The exercising for data recording is carried out by the owners or familiar riders at local training arena with a size of 25 m x 38 m and a track surface of sand mixed with GEOPAT polyflakes.

B. Data collection procedure

All six subjects, while wearing two single triaxial Axivity AX3 accelerometers (Axivity Ltd, Newcastle, United Kingdom) as depicted in Figure 1 are exercised in the different gaits walk, trot and canter for about 10 min each; exercise is either ridden or longed. The gait walk is also measured on a field and hard



Fig. 1. Orientation of accelerometer.

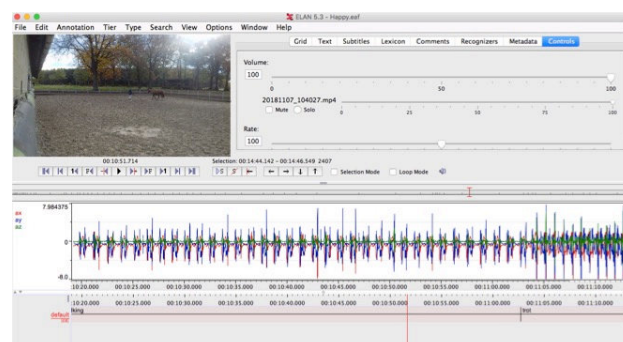


Fig. 2. Screenshot of ELAN software with the video and accelerometer data.

underground for horse 2. Horse 2 and 4 performed in addition other activities like rolling, pawing and flank watching. Observations on the activities of the horses are made with video recordings at the same time as data from the sensors is collected. Table II lists the considered activities in this study with their descriptive definitions and the number of samples taken. All the data is labelled based on the video recordings since it is difficult to use direct observation in combination with training of the horse. ELAN is a tool that allows such type of labelling procedure and is used by animal scientists for the video analysis and codification of images [2], [3]. Annotations can be made by selecting the length of the segment where the behaviour is performed and typing the annotation as depicted in Figure 2.

C. The Datasets

The annotations together with corresponding time intervals are exported into a CSV file and merged with the accelerometer dataset. No features are extracted from the data before presenting it to the machine learning model. A convolutional Neu-

Subject number	Breed class	Height at withers (cm)	Gender	Age	Condition	Shoeing
1	Horse	172	Mare	7	Healthy	Barefoot
2	Horse	167	Gelding	11	Healthy	Barefoot
3	Horse	181	Mare	17	Lame	Barefoot
4	Horse	168	Mare	19	Healthy	Barefoot
5	Friesian Horse	159	Mare	12	Healthy	Shoed
6	Pony	116	Gelding	15	Healthy	Barefoot

TABLE I

PARTICIPATING HORSES (N = 6) WITH BREED CLASS, HEIGHT AT WITHERS, GENDER, AGE, CONDITION AND TYPE OF SHOEING.

Observed activities	Description	Number of samples
Stand	The horse is standing on at least three legs with no movement to another place.	92121
Walk	The horse performs a four beat gait with its legs following this sequence: left hind leg, left front leg, right hind leg, right front leg, leaving three feet on the ground.	406939
Trot	The horse performs a two beat diagonal gait where the diagonal pairs of legs move forward at the same time with a moment of suspension between each beat.	327015
Canter	The canter is a three beat gait. This gait starts with the hind leg then leads to the front in a rocking motion. This gait has a period of suspension after each stride.	110706
Roll	The horse starts in a lying position on the side called lateral recumbency and rotates the body over its back, alternately from one side to another, remaining parallel to the performing surface.	11884
Paw	The horse scrapes the ground with a forelimb.	5948
Flank watching	The horse looks at its side or flank.	4462

TABLE II

DESCRIPTION OF THE OBSERVED ACTIVITIES [4], [5].

ral Network (CNN) has the advantage of automatic features extraction by using strong computing capabilities. Deep learning-based classifiers can learn features and achieve better accuracy [6]. These sensor data is sampled at five different sampling rates i.e., 25 Hz, 50 Hz, 100 Hz, 200 Hz and 1600 Hz. Each AX3 was set to record with a range of $\pm 8g$ for all the datasets except for one high sampling rate measurement at 1600 Hz the range is increased to $\pm 16g$ since this measurement was necessary for another research topic. Table III gives an overview of the time measured, the number of samples, the number of subjects and the number of behaviours at each sampling rate. The class proportions of the seven studied activities are not distributed evenly. The class walk is mainly present with a share of 42.43% followed by trot, canter, stand, roll, paw and flank watching with a proportion of 34.10%, 11.54%, 9.61%, 1.24%, 0.62% and 0.47%, respectively.

III. MACHINE LEARNING MODEL

A multilayer convolutional network, is used with two convolutional layers, which are followed by max-pooling layers, and two fully connected layers. The output of the last fully-connected layer is fed to a 7-way softmax layer which produces a distribution over the seven class labels: stand, walk, trot, canter, roll, paw, flank watching.

The first convolutional layer filters the $n \times 6 \times 1$ input acceleration data with 64 kernels of size 3×1 and stride 1. The L2 regularization technique is used in this layer with a weight de-

cay coefficient of 0.01 [7]. After the first convolutional layer a zero-padding is used such that the output has the same length as the original input. Then a max-pooling operation is done. The second convolutional layer takes as input the (pooled) output of the first convolutional layer and filters it with 16 kernels of size 5×2 and stride 1. Both layers contain an activation layer using rectified units (ReLU) and dropout of 0.55 is used [8]. The Adam optimizer is used for training the neural network through back propagation. Training is done for 400 epochs, with an early stopping criterion of halting training when there is no increase in accuracy during the last 60 epochs [9]. Table IV shows the experimental setup.

IV. RESULTS

Figure 3 illustrates exemplar two second data windows of the four gaits and the other behaviours, from the left and right accelerometer worn on the lateral side of the tendon boot.

To train the convolutional neural network, separate training and validation sets are needed and can be selected in various ways. First, a training and validation set are obtained by automatically splitting the training and the validation data with a fixed ratio of 66/34 referred to as the 'First dataset'. Secondly, the 50 Hz dataset, which contains every behaviour, is resampled to 25 Hz, 100 Hz and 200 Hz and merged with the original dataset at that sampling rate referred to as the 'Second dataset'. The model can then be assessed for any behaviour at each sampling rate. Again automatic split testing is used to obtain the

	25 Hz	50 Hz	100 Hz	200 Hz	1600 Hz
Time measured [s]	2752	5492	3006	2560	417
Number of samples	68800	274580	300640	511920	666952
Number of subjects	3	6	3	3	1
Number of behaviours	4	7	4	5	4

TABLE III

TOTAL TIME OF MOVEMENT DATA, NUMBER OF SAMPLES, NUMBER OF SUBJECTS AND NUMBER OF BEHAVIOURS FOR EACH SAMPLING RATE OF THE MERGED ACCELEROMETER DATA.

Parameter	Value
The size of the input vector	n
The number of input channels	6
The number of feature maps	64-16
Filter size	$3 \times 1 - 5 \times 2$
Stride	1
Pooling size	$2 \times 3 - 3 \times 1$
Activation function	ReLU and Softmax
Weight decay	0.01 (L2 regularization)
The probability of dropout	0.55
Maximum epochs	400
Optimization (back propagation)	Adam optimizer

TABLE IV
EXPERIMENTAL SETUP.

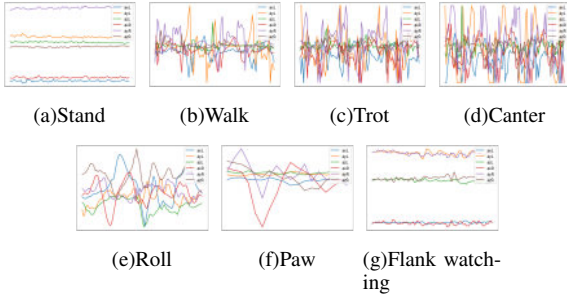


Fig. 3. Typical accelerometer patterns of (a) stand, (b) walk, (c) trot, (d) canter, (e) roll, (f) paw and (g) flank watching in a 2 s window. The blue, yellow, green lines represent X,Y,Z signals from the left accelerometer and the red, purple and brown lines represent X,Y,Z signals from the right accelerometer, respectively.

training and validation set. Finally, the separation of the training and validation data is attained manually and as a result the model is not validated on data from a seen horse referred to as the 'Third dataset'. In this case data from the lame horse is used to validate our model while it is trained on healthy horses, to further assess the generalization of the model.

A. Effects of the sampling rate

In Figure 4 the performance of the CNN with increasing sampling rate is depicted for the three datasets. The number between the brackets indicates the number of behaviours that are taken into account in the training and validation of the CNN. For all datasets the accuracy increases when the sampling rate is increased from 25 Hz to 100 Hz. From 100 Hz to 200 Hz, the accuracy for two out of three datasets decreases slightly. As can

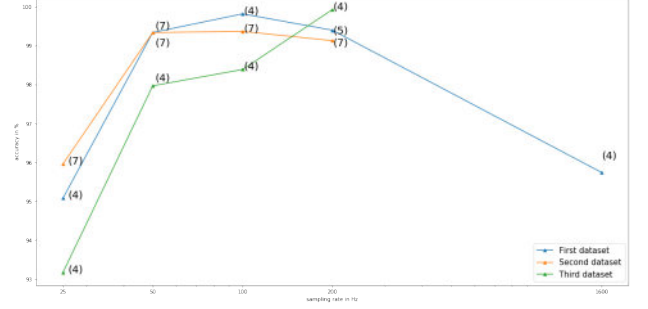


Fig. 4. Mean performance of convnet with increasing sampling rate presented on a logarithmic scale for three datasets.

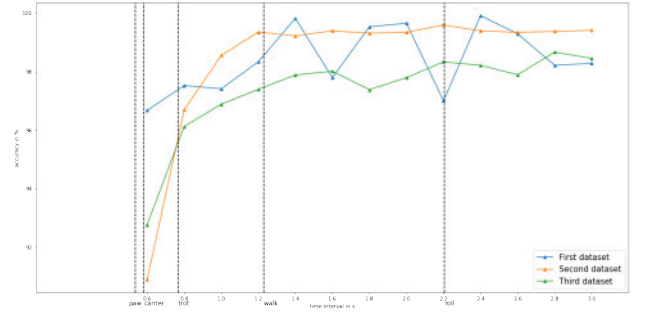


Fig. 5. Mean performance of convnet with increasing time interval for the three datasets.

be concluded from this graph, for a sampling rate of 25 Hz, the CNN performs the best when all behaviours and all horses are taken into account. The CNN validated on the data of the lame horse performs the least in the sampling rate range from 25 Hz to 100 Hz. At a sampling rate of 200 Hz, the CNN performs best when the training and validation data are split up by hand and the model trained on all behaviours performs the least. Increasing the sampling rate to 1600 Hz reduces the overall mean validation accuracy of the first dataset to 95.74%.

B. Effect of the time interval

In Figure 5 the mean performance of the CNN with increasing time interval is plotted for the three datasets for sampling rates between 25 Hz and 200 Hz. The mean duration of the behaviours are annotated with black striped lines except for the flank-watching movement since the mean duration of this behaviour lies outside the investigated time intervals at 4.866 s. If the time interval decreases from 1.2 s (a full walk cycle) to 0.6 s, the overall mean validation accuracy of the three datasets decreases. In all cases, the largest contributor to a lower accuracy

is the misclassification of 'canter' and to a lesser extent 'roll', 'paw' and 'flank-watching'. As can be noticed the mean accuracy plot for the third dataset lies lower than those of the first and second dataset due to more spread out accuracies at higher time intervals for sampling rates between 50 Hz and 200 Hz.

C. Combination of the time interval and sampling rate

Figure 6 presents the accuracy surface plots for the three datasets as function of time interval and sampling rate generated through the obtained datapoints indicated as blue dots. The surface plot is obtained by fitting a polynomial of degree two through the datapoints. The low predicted accuracies are indicated with the colour blue and the high ones with the colour red. For the first dataset, the 1600 Hz datapoints are left out of the interpolation. The combinations that are the least performing for the three datasets are observed in the region where both sampling rate and time interval are low.

As can be seen from the contourplots shown in Figure 7 a 100% is reached in the red region. As indicated with yellow cross markers, the combinations that gain an accuracy of 100% at the lowest sampling rate and the shortest time interval are for the first dataset observed in the region where the value of the sampling rate ranges between 64 Hz at a time interval of 2.05 s and 170 Hz at a time interval of 0.85 s, for the second dataset in the region where the value of the sampling rate ranges between 36 Hz at a time interval of 2.4 s and 170 Hz at a time interval of 1 s, for the third dataset in the region where the value of the sampling rate ranges between 90 Hz at a time interval of 2.3 s and 170 Hz at a time interval of 1 s.

D. Effects of the underground

Model accuracy for the class walk is studied for four different surfaces: dry sand mixed with polyflakes, wet sand mixed with polyflakes, meadow and hard underground. The normalized confusion matrices are depicted in Figure 8.

As can be seen from the normalized confusion matrices for different time intervals, the class walk on a wet underground and on a dry underground get classified with an accuracy above 98% for every time interval. The class walk on a hard underground reaches accuracies higher than 86%. The class walk on a field swings between 15% and 86% classification accuracy. As can be concluded from the results presented in the normalized confusion matrices, the data gathered from different undergrounds is significantly different so that the model could conclude from which underground the data was gathered for three out of four undergrounds.

Normalized confusion matrices with all activities included are shown in Figure 9. As can be concluded from the confusion matrices, at small time intervals, more misclassification is taking place than at higher time intervals. 'Walk-F' is performing the worst with accuracies swinging between 1% and 74%. The other 'walk classes' get classified with high accuracies between 93% and 100% at higher time intervals. All the walk movements get classified as walk, independent of the underground, at any time interval. The other movements that are now included get classified in a few cases as one of the 'walk classes'.

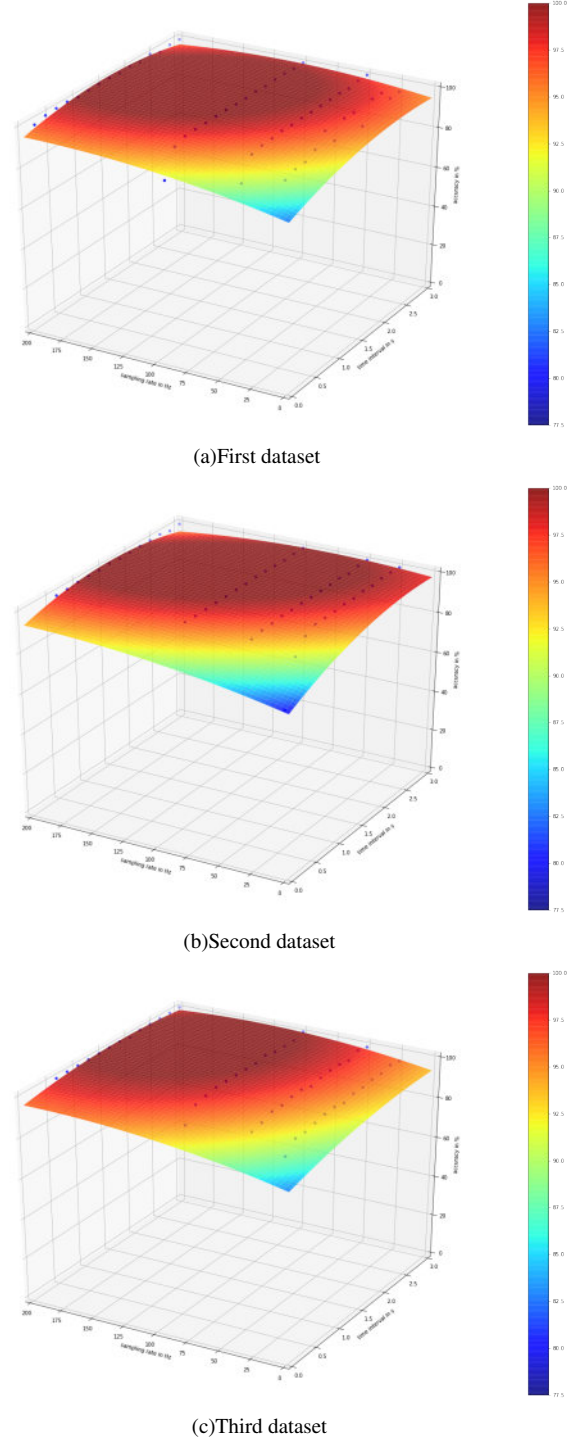
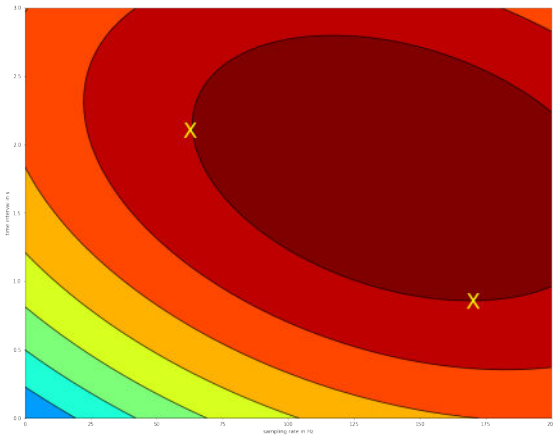
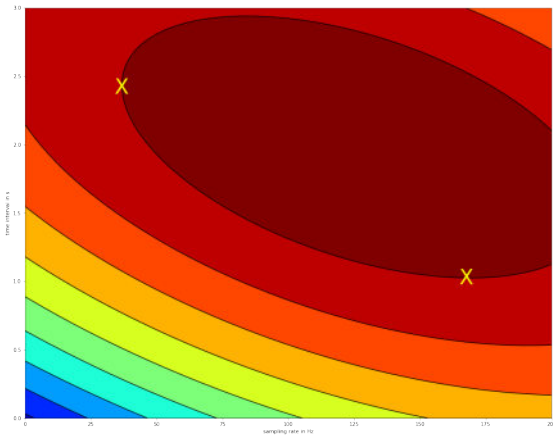


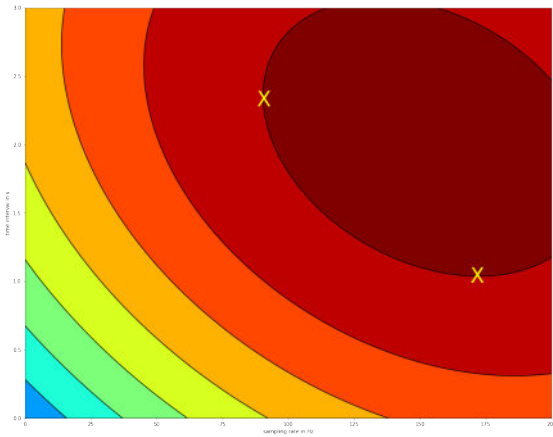
Fig. 6. Accuracy surface plot as function of the sampling rate and the length of the time interval for three datasets.



(a)First dataset

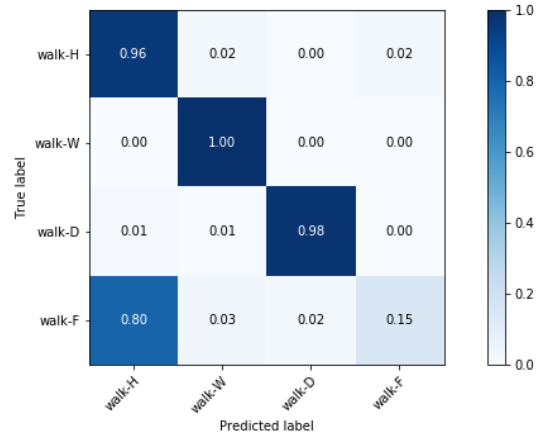


(b)Second dataset

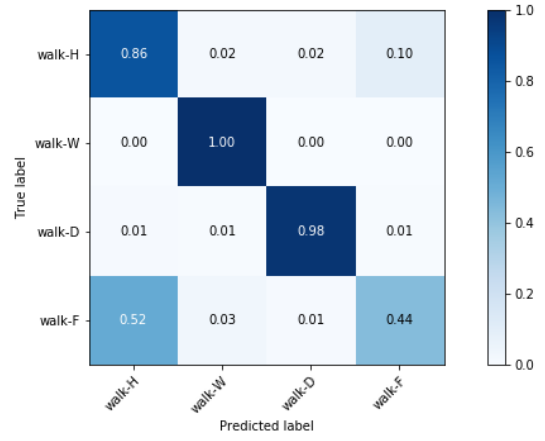


(c)Third dataset

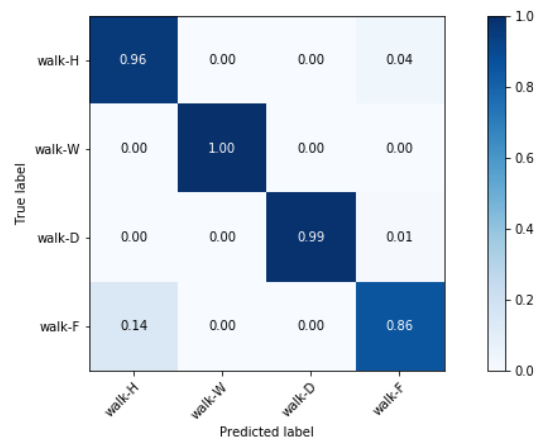
Fig. 7. Accuracy contour plot as function of the sampling rate and the length of the time interval for three datasets.



(a) $n = 0.6$ s



(b) $n = 1.2$ s



(c) $n = 2.4$ s

Fig. 8. Normalized confusion matrix for training and test set at a sampling rate of 50 Hz for different time intervals and four types of underground (H= hard, W = wet, F= field and D = dry).

V. CONCLUSION

In this research we proposed a solution for a horse activity recognition problem that is based on Convolutional Neural Networks with the use of accelerometer time series. High accuracies can already be reached using short recognition intervals and small sampling rates and requiring no feature engineering. Due to a relatively shallow architecture, the proposed algorithm has a small running time and can be efficiently executed on wearable devices in real time. To evaluate the performance of the considered approach we tested it on three datasets. The obtained results demonstrate that the proposed CNN-based model establishes high accuracies in all cases. The experiment has further emphasized an architecture that can be applied not only to different subjects, but can be used in different measurement conditions. Future works will include experimenting with more behaviours, lower sampling rates and reduction in the number of accelerometer axes. Moreover, we need further study for the analysis of the features extracted automatically by the convent and compare them with the well-known hand-crafted features. Further study on the characteristics of the used CNN and utilizing larger dataset should be conducted.

REFERENCES

- [1] Vanessa L Cook and Diana M Hassel. Evaluation of the colic in horses: decision for referral. *The Veterinary clinics of North America. Equine practice*, 30(2):383–98, 2014.
- [2] Hennie Brugman, Albert Russel, and Xd Nijmegen. Annotating multimedia/multi-modal resources with elan. In *LREC*, 2004.
- [3] Katja Liebal, Bridget M Waller, Katie E Slocombe, and Anne M Burrows. *Primate communication: a multimodal approach*. Cambridge University Press, 2013.
- [4] Horse gaits. https://en.wikipedia.org/wiki/Horse_gait. Accessed: 2019-04-02.
- [5] Gila Abells Sutton, Roe Dahan, Dan Turner, and Ora Paltiel. A behaviour-based pain scale for horses with acute colic: scale construction. *The Veterinary Journal*, 196(3):394–401, 2013.
- [6] Xin Zhao, Haikun Wei, Hai Wang, Tingting Zhu, and Kanjian Zhang. 3d-cnn-based feature extraction of ground-based cloud images for direct normal irradiance prediction. *Solar Energy*, 181:510–518, 2019.
- [7] Arafat Abu Mallouh, Zakariya Qawaqneh, and Buket D Barkana. Utilizing cnns and transfer learning of pre-trained models for age range classification from unconstrained face images. *Image and Vision Computing*, 2019.
- [8] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- [9] Charissa Ann Ronao and Sung-Bae Cho. Human activity recognition with smartphone sensors using deep learning neural networks. *Expert systems with applications*, 59:235–244, 2016.

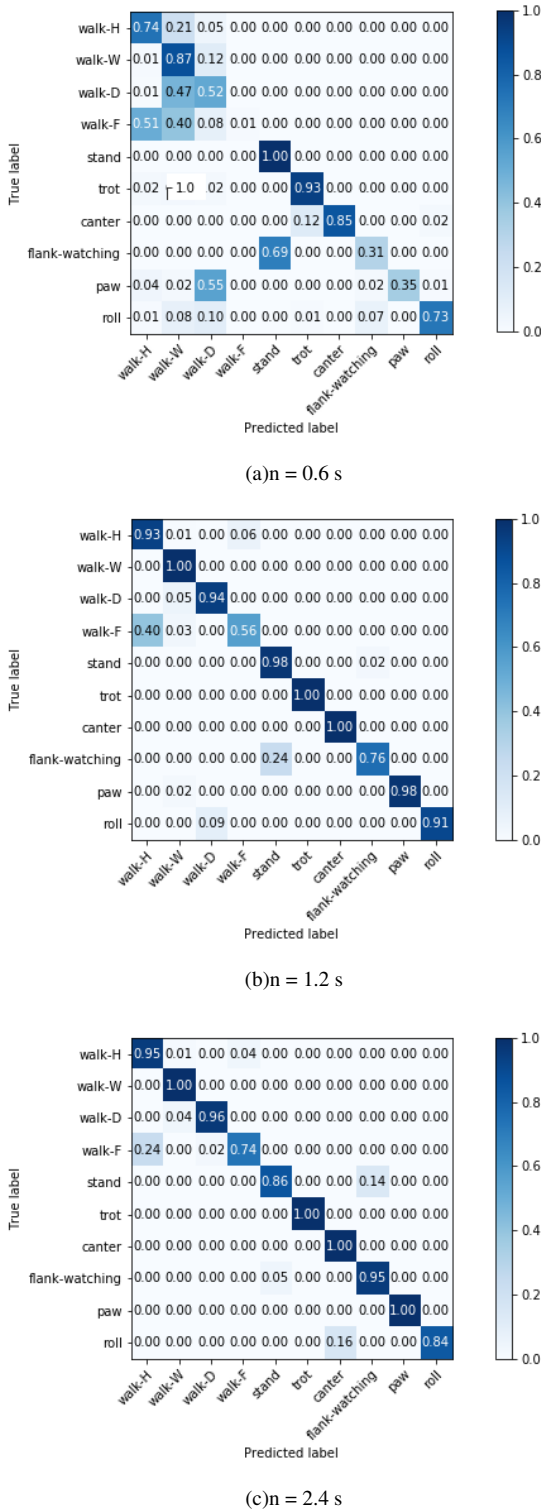


Fig. 9. Normalized confusion matrix for training and test set at a sampling rate of 50 Hz for different time intervals and four types of underground (H= hard, W = wet, F= field and D = dry) including all activities.

Table of Contents

Preface	ii
Permission for Use of Content	iii
Abstract	iv
Extended abstract	v
Table of Contents	xii
List of Figures	xiii
Abbreviations	xvi
1 Introduction	1
2 Methodology	5
2.1 Animals and training arena	5
2.2 Data collection procedure	5
2.2.1 Activities' observation	6
2.2.2 Accelerometer data	7
2.3 Processing and classification of accelerometer data	8
2.4 Machine learning model	18
2.4.1 The datasets	18
2.4.2 Further preprocessing	20
2.4.3 The architecture	21
3 Results	30
3.1 Effects of the sampling rate of the accelerometer	41
3.2 Effects of the time interval of the accelerometer data	52
3.2.1 Model complexity	57
3.3 Combination of time interval and sampling rate	59
3.4 Effects of the underground	62

4 Discussion	71
5 Conclusion and Future work	73
Bibliography	75
A AX3 data sheet	78
B Manual	87
B.1 Introduction	87
B.2 Setup and configure AX3 sensors for recording	87
B.3 Datamarkers	88
B.4 Extracting data from Axivity AX3 accelerometer	88
B.4.1 Connect accelerometer and download data	89
B.5 Preprocessing data for Elan	90
B.5.1 Import the CSV file in Excel	90
B.5.2 Preprocess data of two accelerometers	92
B.6 Video annotation	94
B.6.1 Add files to Elan	94
B.7 Annotate file in Excel	97
B.8 Uploading data to algorithm and adjusting parameters	102
B.8.1 Reading the CSV file of the code with split testing	102
B.8.2 Reading the CSV file of the code without split testing	102
B.8.3 Adjusting parameters	103
B.9 Conclusion	105
C Tables	106

List of Figures

1.1 Birth alarm [1].	3
1.2 Equestic saddle clip [2].	3
2.1 Observed behaviours (a) roll, (b) paw and (c) flank watching [3, 4, 5].	6
2.2 AX3 Axivity 3-Axis Logging Accelerometer [6].	7
2.3 Tendon boot with accelerometer [7].	7
2.4 Orientation of accelerometer.	8
2.5 Segment of the original and the interpolated signal together with the cross-correlation as a function of lag.	9
2.6 Screenshot of ELAN software showing the video and accelerometer data.	10
2.7 2s of observed movements stand, walk, trot, canter, paw, roll and flank watching. The red and blue lines represent left and right acceleration signals, respectively.	11
2.8 Observed movements stand, walk, trot, canter, paw, roll and flank watching. The red, blue lines represent left and right acceleration signals, respectively.	14
2.9 Boxplot of acceleration sum vector of activities for the 3 breed classes horse (n = 4), Friesian horse (n = 1) and Pony (n = 1), given as boxplots with medians, interquartile and absolute ranges.	17
2.10 Bar chart of the movement class proportions of the studied movements of all horses.	20
2.11 The shuffle and filter step when preprocessing the dataset.	20
2.12 Architecture of a traditional CNN [8].	21
2.13 The convolution step whit input $I \times I$, filter $F \times F$ and output $O \times O \times K$	22
2.14 The max-pooling step with input $I \times I$, filter $F \times F$ and output $O \times O$	23
2.15 The zero-padding technique.	24
3.1 Typical accelerometer patterns of (a) stand, (b) walk, (c) trot, (d) canter, (e) roll, (f) paw and (g) flank watching in a 2 s window. The blue, yellow, green lines represent X,Y,Z signals from the left accelerometer and the red, purple and brown lines represent X,Y,Z signals from the right accelerometer , respectively.	30
3.2 Training and validation data for three datasets at sampling rate 25 Hz, 50 Hz, 100 Hz and 200 Hz.	41
3.3 Performance of convnet with increasing sampling rate.	42

3.4 Confusion matrix for training and test set at a sampling rate of 25 Hz for different time intervals.	43
3.5 The training accuracy and validation accuracy results for a sampling rate of 25 Hz at low accuracy and high accuracy time intervals.	44
3.6 The training loss and validation loss results for a sampling rate of 25 Hz at low accuracy and high accuracy time intervals.	44
3.7 Performance of convnet with increasing sampling rate from 25 Hz to 50 Hz.	45
3.8 Confusion matrix for training and test set at a sampling rate of 50 Hz for a time interval of 0.6 s.	45
3.9 Performance of convnet with increasing sampling rate from 50 Hz to 100 Hz.	46
3.10 Performance of convnet with increasing sampling rate from 100 Hz to 200 Hz.	46
3.11 Confusion matrix for training and test set at a sampling rate of 200 Hz for different time intervals.	47
3.12 Performance of convnet with increasing sampling rate from 200 Hz to 1600 Hz.	47
3.13 Confusion matrix for training and test set at a sampling rate of 1600 Hz for different time intervals.	48
3.14 Mean overall accuracy of convnet with increasing sampling rate.	48
3.15 Performance of convnet with increasing sampling rate.	49
3.16 Confusion matrix for training and test set at a sampling rate of 25 Hz for different time intervals.	49
3.17 Performance of convnet with increasing sampling rate.	50
3.18 Confusion matrix for training and test set at a sampling rate of 25 Hz for different time intervals.	51
3.19 Trot training data, canter training data and canter validation data.	51
3.20 Performance of convnet with increasing sampling rate presented on a logarithmic scale for three datasets.	52
3.21 Performance of convnet with increasing time interval.	53
3.22 Confusion matrix for training and test set at a sampling rate of 50 Hz for time intervals of 0.4 s and 0.6 s and 100 Hz for time intervals of 0.2 s and 0.4 s.	54
3.23 Performance of convnet with increasing time interval.	55
3.24 Performance of convnet with increasing time interval.	55
3.25 Mean performance of convnet with increasing time interval for the three datasets.	56
3.26 Performance of convnet with increasing time interval for three datasets for different sampling rates.	56
3.27 Minimum number of samples necessary to reach an overall validation accuracy higher than 98% for the three datasets.	57
3.28 Performance of convnet with increasing number of samples for sampling rates from 25 Hz to 1600 Hz presented on a logarithmic scale.	58
3.29 Performance of convnet with increasing number of samples for sampling rates from 25 Hz to 200 Hz.	58

3.30 Performance of convnet with increasing number of samples for sampling rates from 25 Hz to 200 Hz.	59
3.31 Accuracy surface plot as function of the sampling rate and the length of the time interval for three datasets.	60
3.32 Accuracy contour plot as function of the sampling rate and the length of the time interval for three datasets.	61
3.33 Contour plots for three datasets with accuracy levels of 98%, 99% and 100%. . . .	61
3.34 Normalized confusion matrix for training and test set at a sampling rate of 50 Hz for different time intervals and four types of underground (H= hard, W = wet, F= field and D = dry).	63
3.35 Normalized confusion matrix for training and test set at a sampling rate of 50 Hz for different time intervals and four types of underground (H= hard, W = wet, F= field and D = dry) including four other behaviours.	65
3.36 Normalized confusion matrix for training and test set at a sampling rate of 50 Hz for different time intervals and four types of underground (H= hard, W = wet, F= field and D = dry) including an extra horse.	67
3.37 Normalized confusion matrix for training and test set at a sampling rate of 50 Hz for different time intervals and four types of underground (H= hard, W = wet, F= field and D = dry) including extra behaviours of horse 4.	69

Abbreviations

CNNs	Convolutional Neural Networks
a_{xL}	acceleration in x-direction of left accelerometer
a_{yL}	acceleration in y-direction of left accelerometer
a_{zL}	acceleration in z-direction of left accelerometer
a_{xR}	acceleration in x-direction of right accelerometer
a_{yR}	acceleration in y-direction of right accelerometer
a_{zR}	acceleration in z-direction of right accelerometer
A_{sum}	acceleration sum vector
min	minimum
Q1	first quartile
Q3	third quartile
max	maximum
rms	root mean square
std	standard deviation
CONV	convolutional layer
ZP	zero-padding
POOL	pooling layer
FC	fully connected layer
RELU	The rectified linear unit layer

Chapter 1

Introduction

Horses have a high value on both a financial and emotional level. The purchase price of a horse depends upon the pedigree, the age, the health condition and the level of training. It can vary between €3750 for a three-year old riding horse for leisure training and competition up to 15 million € for a world top show jumping athlete such as Palloubet d'Halong. In addition, maintenance costs such as (riding) equipment, food, stabling, meadows, blacksmith, veterinarian, competitions and so on place a financial burden on the owner of the horse. Besides the financial aspect, there is often a strong emotional bond between owner and horse. It would be a great advantage for the owner to know that his beloved horse is in good condition all the time.

Monitoring behaviours can yield important information about the health and welfare of horses. For example, colic is the most common emergency in equine practice with approximately 4 out of every 100 horses having an episode of colic each year showing one or a combination of the following signs of pain [9],[10]:

- Depression: A horse that is quieter than normal, dull, less responsive and disinterested in the environment, may have ears pulled back or down but ears are not pricked up forward and do not move readily to surrounding noise. May have lowered head carriage.
- Flank watching: A horse that glances at its side or flank.
- Weight shifting: A horse that moves his weight from one side to the other, usually with the forelimbs.
- Restlessness: A horse that does not stand quietly but moves, apparently aimlessly, and appears agitated. Movements may be jerky with wide excursions of the head.
- Kicking abdomen: A horse that kicks in the direction of its abdomen.
- Pawing: Scraping the ground with a forelimb.
- Stretching: Most commonly, taking a stance as a male horse would to urinate, but not urinating.

- Sternal recumbency: Lying on ground but on the sternum with the legs tucked under the body.
- Lateral recumbency: Lying on ground but on the side.
- Attempting to lie down: A horse that buckles the legs (crouches) and looks like it will lie down, but does not, or lies down but gets up immediately.
- Collapse: Attempting to or succeeding to suddenly drop to the ground, usually as a prelude to rolling.

From the horses that are evaluated by a veterinarian in private practice, approximately 7% to 10% have a lesion that requires surgical correction [11]. Early identification and referral of horses with a surgical lesion is critical to obtain a successful outcome [12]. According to a study of 1847 Swedish warmblood horses, diseases of the digestive system dominated by colic of an undefined nature (seven of ten) are one of the most common causes of death (5–6%) [13]. This time-sensitive event emphasizes the need for early detection. Changes in activity can also help horse owners in predicting foaling moments. For instance, a significant increase in the total amount of daily activity is found on days two and one prenatal compared to day three prenatal [14]. From 30 minutes to 6 hours before foaling the mare is restless, walks around with a raised tail, and urinates small amounts frequently. She shows signs resembling to those of colic such as alternately lying down and standing up [15].

Today, different technologies exist to detect various parameters such as activity, elevation, heart rate and so on from which conclusions can be drawn regarding the behaviour of the horse. The key requirements for such a device are accuracy, ease of use, comfort for the horse and long range availability of data. Existing solutions have failed on at least one out of these four key features. For example, video monitoring with *EquiView360* uses cameras that are installed in the stable to detect changes in the horse's behaviour. This technology is very accurate but limited to one location for each camera and data is only available on a short range. Another technology called *Birth alarm*, a foaling alarm system, is based on gyroscopes which measure the level of elevation. An advantage of this measurement device is that a long range is achieved, since the cellular network is used to communicate but they have a low accuracy and are uncomfortable for the horse to wear as is depicted in Figure 1.1. A full overview of comparable products and their shortcomings can be found in Table C.1.



Figure 1.1: Birth alarm [1].

As in this study, also accelerometers are used already to detect the activity of the horse. For example, the *Equestic saddle clip* which is depicted in figure 1.2 detects the activity of the horse but the accuracy is low since the positioning is inconsistent and the range of the data is short since it works with bluetooth connection. Another study revealed that accelerometer data allows the determination of gaits by definition of distinct acceleration value ranges for stand, walk, trot and gallop [16].



Figure 1.2: Equestic saddle clip [2].

Thus the use of small devices such as accelerometers in combination with an accurate classification algorithm could provide an excellent solution for accurately detecting the horse's behaviour in a comfortable manner. The aim of this study is to automatically classify the horse's behaviour based on machine learning algorithms by using accelerometer data. In recent years, convolutional neural networks (CNNs) have shown excellent performance on classification problems when large-scale

labelled datasets are available [17]. Studies demonstrated that deep learning models are able to learn and discriminate among human activities ranging from sitting, walking, climbing upstairs, walking downstairs and falling, among others [18].

In this study, accelerometer data of seven activities of six different horses is gathered by attaching the accelerometers to the front legs of the subjects. The horses are exercised under different conditions, to examine the behaviour event detection algorithms in a variety of settings. After preprocessing the gathered data, labelling based on video recordings is done. The final data set is presented to a multilayer convolutional neural network. The model trains itself first by using the training set and afterwards validates itself by the validation or test set. This apportion of the data in the training and test set is done both randomly and manually. Different parameters of the CNN such as filter size, stride and number of layers were fine-tuned to gain high model accuracies. Also regularization techniques such as dropout, L2 regularization and softmax were added, ensuring that the algorithm is better at making (correct) classifications and has the ability to generalize. After optimizing all the model parameters, the effect of variations in sampling frequency, length of the time interval and underground on the model accuracy was studied.

Chapter 2

Methodology

2.1 Animals and training arena

The study is carried out in Zutendaal and is conducted with six adult horses of different breeds. All details about the subjects can be found in Table 2.1. This variety in horses is important for the generalization of the machine learning model, since the gathered data will differ. For example, the mean acceleration values per second during the gaits trot and walk are higher for ponies than for horses [16]. Also, lame horses have asymmetrical gait patterns because they consistently shorten the cranial (forward) phase of stride [19].

Subject number	Breed class	Height at withers (cm)	Gender	Age	Condition	Shoeing
1	Horse	172	Mare	7	Healthy	Barefoot
2	Horse	167	Gelding	11	Healthy	Barefoot
3	Horse	181	Mare	17	Lame	Barefoot
4	Horse	168	Mare	19	Healthy	Barefoot
5	Friesian Horse	159	Mare	12	Healthy	Shoed
6	Pony	116	Gelding	15	Healthy	Barefoot

Table 2.1: Participating horses ($n = 6$) with breed class, height at withers, gender, age, condition and type of shoeing.

The exercising for data recording is carried out by the owners or familiar riders at local training arena with a size of 25 m x 38 m and a track surface of sand mixed with GEOPAT polyflakes. A minority of the data is gathered on a field and a hard underground.

2.2 Data collection procedure

All six subjects, while wearing two single triaxial Axivity AX3 accelerometers (Axivity Ltd, Newcastle, United Kingdom) are exercised in the different gaits walk, trot and canter for about 10 min each; exercise is either ridden or longed. The gait walk is also measured on a field and hard

underground for horse 2. Horse 2 and 4 performed in addition other activities like rolling, pawing and flank watching.

2.2.1 Activities' observation

Observations on the activities of the horses are made with video recordings at the same time as data from the sensors is collected. Table 2.2 lists the considered activities in this study with their descriptive definitions and the number of samples taken. In Figure 2.1 the expressed activities occurring during an episode of colic are depicted. All the data is labelled based on the video recordings since it is difficult to use direct observation in combination with training of the horse.

The number of samples for the gait movements is significantly higher than for the other behaviours rolling, pawing and flank watching which are not performed on command but by provocation. For example, the horse starts scraping over the floor with a forelimb when he stands on the grooming area and becomes impatient. On the other hand, to stimulate rolling, the horse is left free in the training arena after a workout. The horse might be itchy due to the sweat on its pelt which can induce the wanted behaviour. To provoke flank watching, a piece of carrot is held near the flank of the horse until he grabs it and returns to a forward head position.

Observed activities	Description	Number of samples
Stand	The horse is standing on at least three legs with no movement to another place.	92121
Walk	The horse performs a four beat gait with its legs following this sequence: left hind leg, left front leg, right hind leg, right front leg, leaving three feet on the ground.	406939
Trot	The horse performs a two beat diagonal gait where the diagonal pairs of legs move forward at the same time with a moment of suspension between each beat.	327015
Canter	The canter is a three beat gait. This gait starts with the hind leg then leads to the front in a rocking motion. This gait has a period of suspension after each stride.	110706
Roll	The horse starts in a lying position on the side called "lateral recumbency" and rotates the body over its back, alternately from one side to another, remaining parallel to the performing surface.	11884
Paw	The horse scrapes the ground with a forelimb.	5948
Flank watching	The horse looks at its side or flank.	4462

Table 2.2: Description of the observed activities [20], [9].



Figure 2.1: Observed behaviours (a) roll, (b) paw and (c) flank watching [3], [4], [5].

2.2.2 Accelerometer data

Each horse wears two single triaxial Axivity AX3 accelerometers, a device for the measurements of different physical parameters, providing integrated temperature, light and acceleration sensors, as depicted in Figure 2.2, which are validated for activity recognition. The data sheet of the accelerometer can be found in Appendix A. Accelerometers fitted to the lateral side of the tendon boot with a size of 23 x 32.5 x 7.6 mm and a weight of 11 g are used, as shown in figure 2.3 log data with configurable sampling rates ranging from 12.5 Hz to 3200 Hz.



Figure 2.2: AX3 Axivity 3-Axis Logging Accelerometer [6].



Figure 2.3: Tendon boot with accelerometer [7].

The data logger is powered by a 150 mAh lithium-polymer battery, rechargeable via USB connection, which enables measurements over 30 days at 12.5 Hz and 14 days at 100 Hz. Acceleration is measurable on x-, y-, z-axes with a maximum sensitivity of $\pm 16g$ [$g = m/s^2$]. Setup and configuration of the AX3 sensors for recording is done with the AX3 OMGUI Configuration and Analysis Tool, which is an open source application. Data is recorded on an integrated memory with a capacity of 512 MB. It was transferred to a computer after recording via USB connection and stored in a CWA-file. An attachment convention for device orientation assists in consistent and comparable datasets being gathered. The orientation of the accelerometer respected for all horses is depicted in figure 2.4 with the USB port configured to point towards the ground as is suggested by the AX3 user manual. For successful data capture the AX3 is securely fastened with the use of VELCRO stick on circles to the tendon boot with minimal room for vibration, slip or twist;

this helps preserve only the motions of the horse are captured. The AX3 has a built in, real-time clock (RTC) and calendar which provides the time base for the recorded acceleration data.



Figure 2.4: Orientation of accelerometer.

2.3 Processing and classification of accelerometer data

Before presenting the accelerometer data to the machine learning model, the raw data needs to be transformed into an understandable format. The representation and quality of the instance data is a factor that affects the success of machine learning on a given task. If there is much irrelevant and redundant information present or noisy and unreliable data, then knowledge discovery during the training phase of the model is more difficult. The product of data preprocessing is the final training set. Often, preprocessing can have a significant impact on generalization performance of a supervised ML algorithm [21].

The AX3 logs data internally in a binary packed format, named Continuous Wave Accelerometer (CWA) format. To preprocess the obtained data it is necessary to convert this file to a Comma Separated Value (CSV) format which can be done with the OMGUI software. To combine the accelerometer data of the two sensors for recognizing motion states, these two datasets are aligned according to their timestamps. A minor deviation of the timestamps of 300 to 600 ms is noticed during the preprocessing phase which is corrected based on the video-recordings. In addition, the actual sampling rate for both the left and right sensor is lower than the configured sampling rate and is found to have a deviation of $1.25\% \pm 0.37\%$ and $1.56\% \pm 0.35\%$, respectively. A full overview of the configured and actual sampling rate for each dataset can be found in Table C.2. It therefore becomes necessary to interpolate the values for the intervening timestamps. A segment of 200 samples of the original and the interpolated signal as a function of time together with the cross-correlation as a function of lag is depicted in Figure 2.5. The cross-correlation coefficient gives information about the degree of similarity between two time series while lag can be considered when time is under investigation. Here cross-correlation, is a dimensionless coefficient that ranges between -1 and 1. A value of 0 means there is no cross-correlation, a value of 1 means that there

is perfect cross-correlation, a value of -1 means a perfect anti cross-correlation. For this particular case a mean cross-correlation coefficient of 0.84 is found when taking 20 segments with a length of 200 samples into account, which indicates a high similarity between the two time series. Also the highest cross-correlation is found at zero lag which means that the signals have the highest similarity without shifting them.

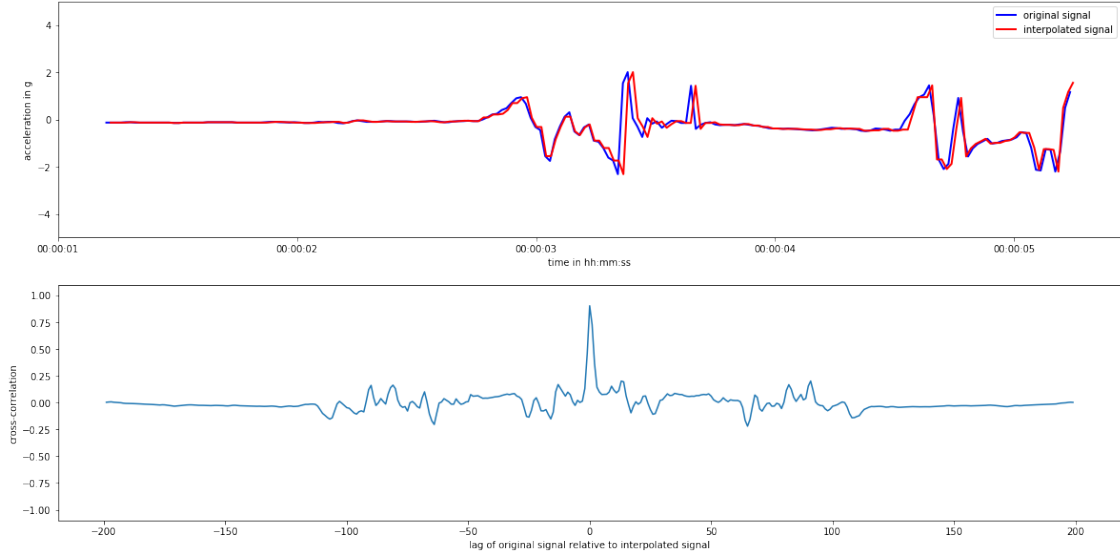


Figure 2.5: Segment of the original and the interpolated signal together with the cross-correlation as a function of lag.

The experiments require combining the observation of specific behaviours and the identification of the accelerometer data measured by the sensors attached to those subjects involved in a particular behavioural pattern. This is the most reliable way to obtain a complete scenario of an event of interest in order to develop an algorithm.

ELAN is a tool that allows such type of labelling procedure because it is a multimodal and multipurpose annotation program and is used by animal scientists for the video analysis and codification of images [22], [23]. It was also proposed as a potential efficient tool for combining audio and video analysis [24]. ELAN is a highly specialised software that allows downloading of the video file and the correspondent acceleration data that have to be labelled.

Annotations can be made by selecting the length of the segment where the behaviour is performed and typing the annotation (Figure 2.6). Mainly, behaviours such as stand, walk, trot and canter are observed and annotated and to a lesser extend roll, paw and flank watching. Changes of pace, for example canter-to-trot transitions, get a separate label, to acquire a reliable training set for the algorithm. A full list of the annotations used to label the data is displayed in Table 2.3. The annotations together with corresponding the time intervals are exported into a CSV file and merged with the accelerometer data set.

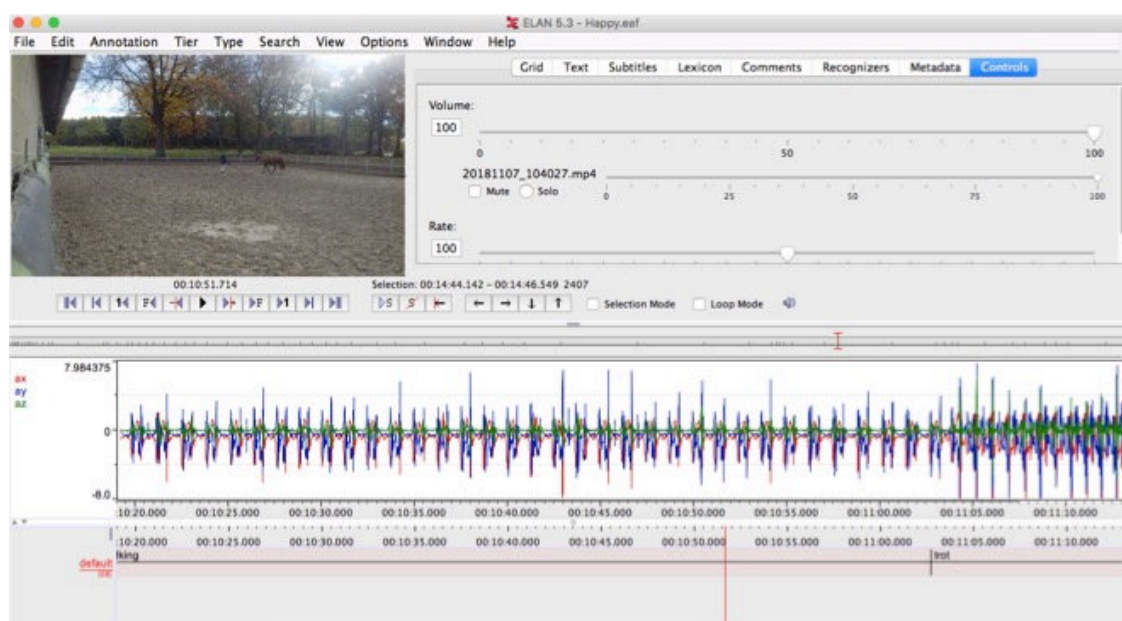
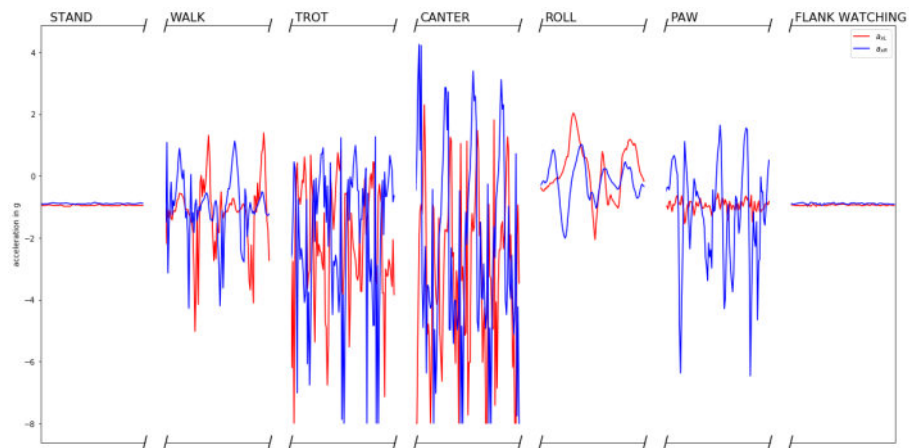


Figure 2.6: Screenshot of ELAN software showing the video and accelerometer data.

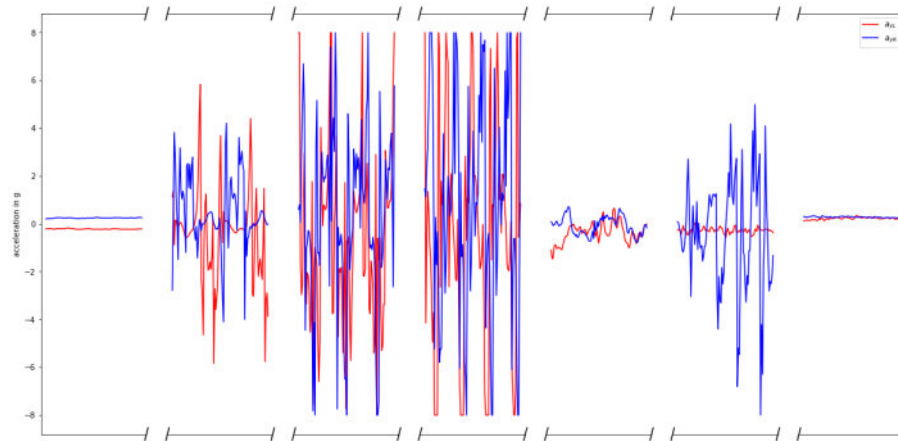
Annotations for model	Other annotations		
stand	stand-to-walk	small-step	turn
walk	walk-to-stand	not-visible	various-movements
trot	walk-to-trot	cross-canter	head-movement
canter	trot-to-walk	contra-canter	flank-watching
roll	trot-to-canter	checking-accelerometers	not-defined
paw	canter-to-trot	step-backwards	shake
not-defined	walk-to-canter	standing-up	protest
smelling-ground	rider-movement	wild-movements	kick-backwards
stand-to-trot	itching-to-walk	shaking-to-walk	

Table 2.3: A full list of annotations.

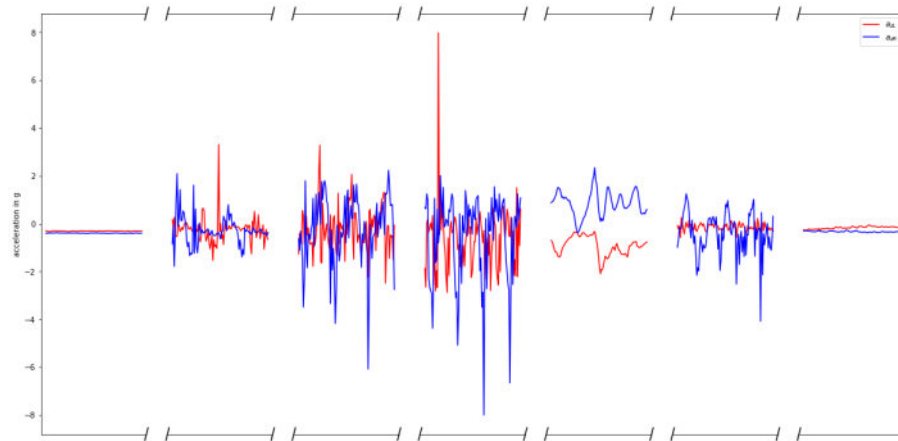
Not only the video-recordings give information about the performed activity but also the variability can be seen visually in the amplitude and shape of the acceleration signals for some of the activities. In figure 2.7 different segments of the acceleration in the three directions of the activities are displayed for the left and right leg.



(a) x-direction



(b) y-direction



(c) z-direction

Figure 2.7: 2s of observed movements stand, walk, trot, canter, paw, roll and flank watching. The red and blue lines represent left and right acceleration signals, respectively.

For both sensors, increasing variations are registered with stand, walk, trot and canter. Due to gravitation on the vertical axis, AX3 data loggers measured acceleration values of -1 for stand. Furthermore, during the different gaits and other behaviours, the AX3 data loggers measure following accelerations in the three directions:

- $a_x < -1$ during upward movement and acceleration $a_x > -1$ during downward movement,
- for the right accelerometer $a_y < 0$ during forward movement and $a_y > 0$ during backward movement, for the left accelerometer, $a_y > 0$ during forward movement and $a_y < 0$ during backward movement,
- $a_z < 0$ during outward movement and $a_z > 0$ during inward movement.

Not only the variability of the signal but also the shape gives an indication which movement is performed by the subject. In Figure 2.8 segments of the acceleration in the three directions of the activities as defined in Table 2.2 are depicted.

Stand appears as a constant signal in every direction.

For the gait *walk* the acceleration in the x-direction a_x shows the alternating movement of the left and right leg. As the horse walks, the signal of the right accelerometer decreases since the leg moves upward (in the negative x-direction) as annotated in figure 2.8(a) with (A). Then the signal increases as the horse accelerates its leg downward (B). The moment the leg touches the ground, due to action and reaction forces, a small negative acceleration peak can be noticed due to upward movement of the leg from the impact (C). A_y of the left accelerometer becomes at first positive during the lifting (D), since the horse accelerates in the forward direction (in the positive y-direction). Then the horse brings its leg down and the signal becomes negative so acceleration in the backward direction occurs (E). At the end of the movement a small forward flexion of the lower limb can be noticed as a small positive acceleration (F). A_y of the right accelerometer is a mirror image around the horizontal axis since the positive y-axis for the is in the backward direction in contrast to the positive y-direction of the left accelerometer which is pointing forward. The z-signal is positive during upward movement, so the leg performs a small inward acceleration (G). During the downward movement, this signal becomes negative, so a small outward acceleration happens (H).

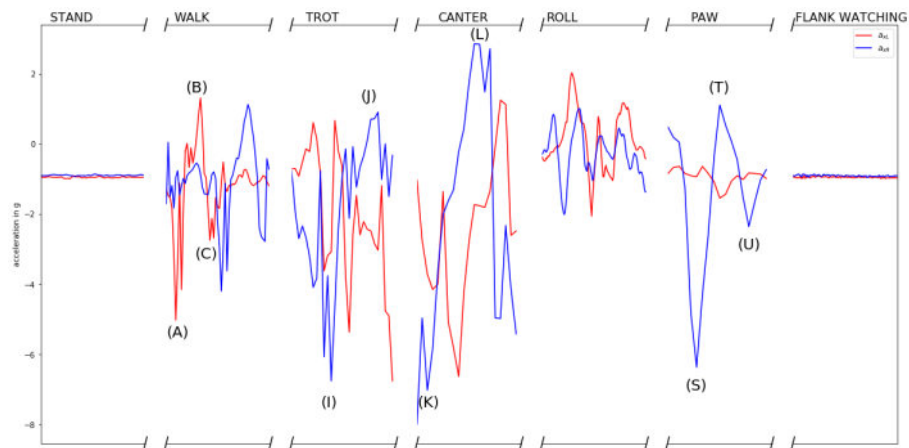
The *trot* signal shows the same characteristics as the walk signal, but the signals from the left and right leg are more intertwined. Again, during upward movement of the leg, the x-signal is decreasing below -1 (I) and during downward movement a signal higher than -1 is produced (J). The x-signal of the left leg is the mirror image of the signal of the right leg. The a_y of the left and right leg lie more or less on top of each other because the moment one leg moves forward, the other leg moves backward and since the direction of the y-axis is opposite the left leg produces a positive signal during the forward movement and the right leg produces a positive signal during the backward movement.

Canter shows similarity of the signals with the one of the inside leg lagging behind to the other of the outside leg (in this case the right leg). First the outside leg shows a decreasing signal for the x-direction as the horse lifts the leg in the upward direction (K), then the leg accelerates downward which produces a positive signal (L). In the y-direction, the signals are again each others mirror image, with the positive signal from the left leg and the negative signal from the right leg indicating forward movement (M). During downward movement of the legs, a backward acceleration occurs expressed as a negative signal from the left leg and a positive signal from the right leg (N). The z-signal is first positive indicating an inward movement (O) and then negative due to an outward movement (P). For the outside right leg, this is more pronounced.

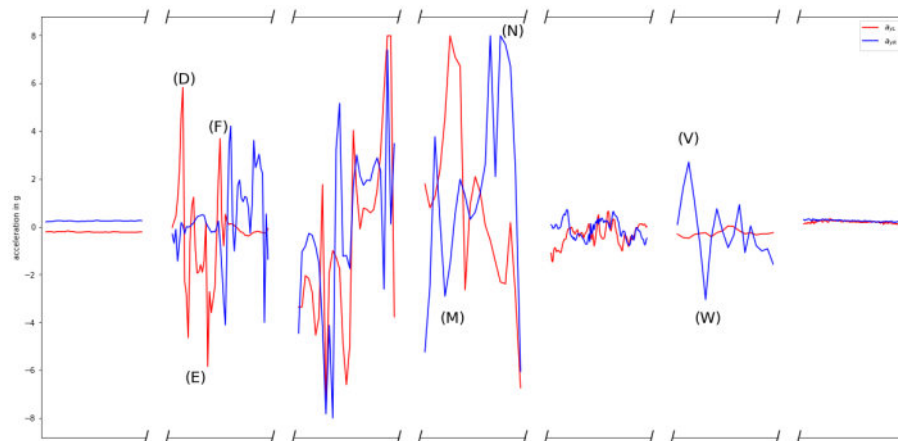
During *rolling*, the legs of the horse perform a half circular motion in sync with one leg lagging behind on the other leg. First there is acceleration in the upward direction and afterwards in the downward direction. Since the horse raises its legs in the air, the upward direction of the accelerometer is now positive. In the y-direction, almost no acceleration is detected since there is no forward and backward movement of the legs. The signal in the z-direction for the right leg is positive due to an outward movement (Q) and is negative for the left accelerometer due to an inward movement (R).

The act of *pawing* shows a constant signal for one leg and an alternating signal for the other leg. First the signal in the x-direction becomes more negative than -1 as the horse lifts its right leg upward (S). Then the horse brings the leg down, which makes the signal more positive (T). At last, due to the action and reaction forces, contact with the ground produces a small negative signal which indicates acceleration in the upward direction (U). The y-signal of the right accelerometer becomes positive during the leg lift, since the leg accelerates in the forward direction (V). During downward movement, acceleration in the backward direction occurs and the y-signal becomes negative (W). Again, the z-signal is positive during upward movement, so the leg performs a small inward acceleration (X). During downward movement, this signal becomes negative, so a small outward acceleration happens (Y).

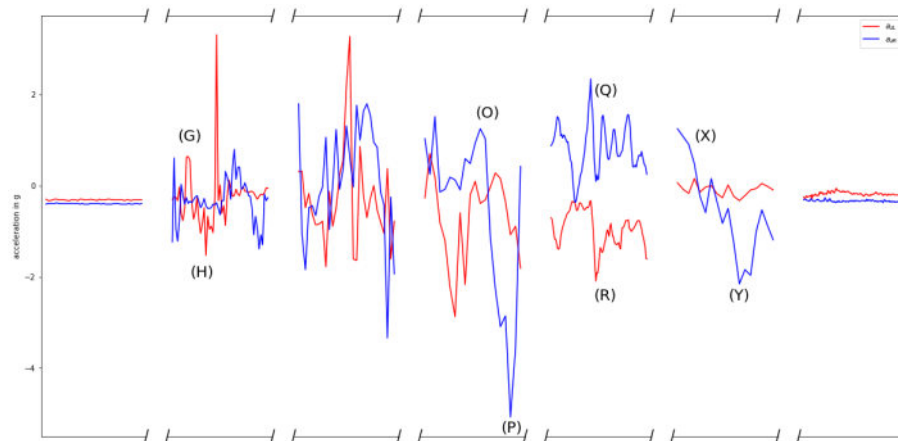
The signal of *flank watching* appears as a constant signal but compared to the signal measured for stand some extra vibrations can be noticed. In this particular case, the horse watches his left flank. This means that the horse rotates mainly over its left front leg. This can be also noticed in the a_{yL} -signal, since it is a higher than when the horse stands fully still. This means that some movement in the forward direction is performed. Looking at the a_{zL} -signal, increasing variations can be noticed in the signal compared to the case where the horse stands still.



(a) x-direction



(b) y-direction



(c) z-direction

Figure 2.8: Observed movements stand, walk, trot, canter, paw, roll and flank watching. The red, blue lines represent left and right acceleration signals, respectively.

Feature extraction

Feature selection aims to find the most important information to save computational efforts and data storage. For threshold-based movement detection algorithms, the classification of behaviours is achieved by comparing and selecting the optimal threshold from statistical features of the motion dynamics.

Now, a closer look is taken at the magnitude of the acceleration to see if thresholds can be deduced from it, since this could reduce the number of parameters taken into account by the model from three to one. Triaxial accelerometers produce three separated accelerometer data time series, one time series for acceleration on each axis. From this data, the magnitude of the acceleration can be calculated, also called the acceleration sum vector. The acceleration sum vector A_{sum} can be calculated as follows from the accelerations along x, y, and z axes [25]:

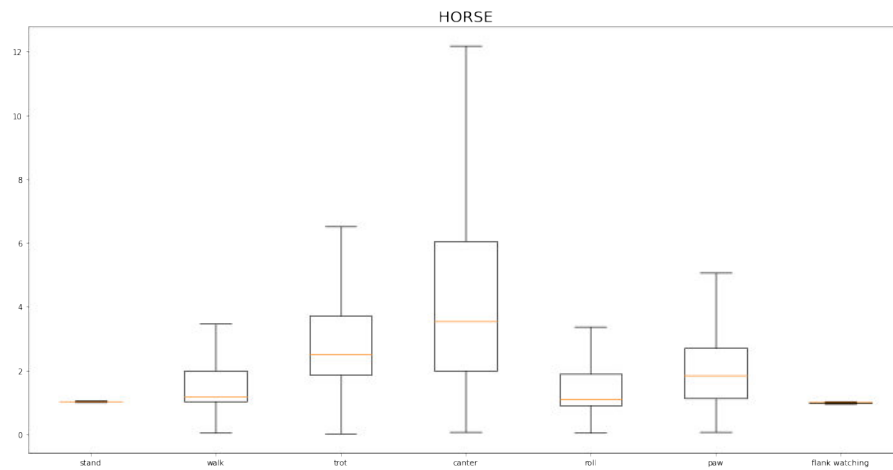
$$A_{sum} = \sqrt{a_x^2 + a_y^2 + a_z^2} \quad (2.1)$$

where, a_x is the acceleration along the x-axis, a_y is the acceleration along the y-axis, and a_z is the acceleration along the z-axis.

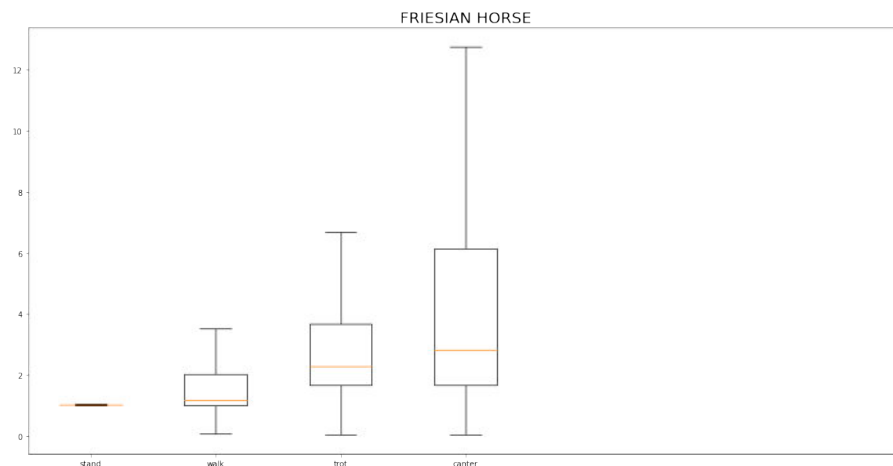
From this acceleration sum vector, eight statistical features are presented in Table 2.4 e.g. minimum (outliers excluded), first quartile, median, third quartile, maximum (outliers excluded), mean, root mean square, and standard deviation are derived directly from the sum vector A_{sum} of the activities for the 3 breed classes horse ($n = 4$), Friesian horse ($n = 1$) and Pony ($n = 1$), and given as boxplots in Figure 2.9

Features of the acceleration sum vector A_{sum}								
	Horse							
	Min	Q1	Median	Q3	Max	Mean	Rms	Std
Stand	1.00	1.02	1.03	1.04	1.08	1.03	1.03	0.02
Walk	0.62	1.02	1.19	2.00	5.90	1.72	2.07	1.15
Trot	0.65	1.86	2.51	3.73	9.49	3.09	3.62	1.90
Canter	0.55	1.98	3.54	6.06	11.32	4.25	5.07	2.77
Roll	0.34	0.92	1.12	1.90	6.51	1.60	1.99	1.19
Paw	0.26	1.13	1.84	2.72	7.65	2.13	2.55	1.41
Flank watching	0.97	0.99	1.00	1.01	1.05	1.00	1.00	0.02
	Friesian horse							
	Min	Q1	Median	Q3	Max	Mean	Rms	Std
Stand	1.01	1.02	1.03	1.04	1.06	1.03	1.03	0.01
Walk	0.44	1.01	1.18	2.02	7.12	1.81	2.30	1.42
Trot	0.43	1.68	2.28	3.68	10.07	2.97	3.63	2.09
Canter	0.23	1.67	2.83	6.16	11.26	3.97	4.99	3.03
	Pony							
	Min	Q1	Median	Q3	Max	Mean	Rms	Std
Stand	1.00	1.02	1.03	1.04	1.06	1.03	1.03	0.01
Walk	0.60	1.03	1.18	1.92	5.26	1.72	2.06	1.13
Trot	0.61	2.00	2.66	4.30	8.81	3.32	3.82	1.89
Canter	0.67	2.03	3.23	6.30	11.31	4.26	5.12	2.84

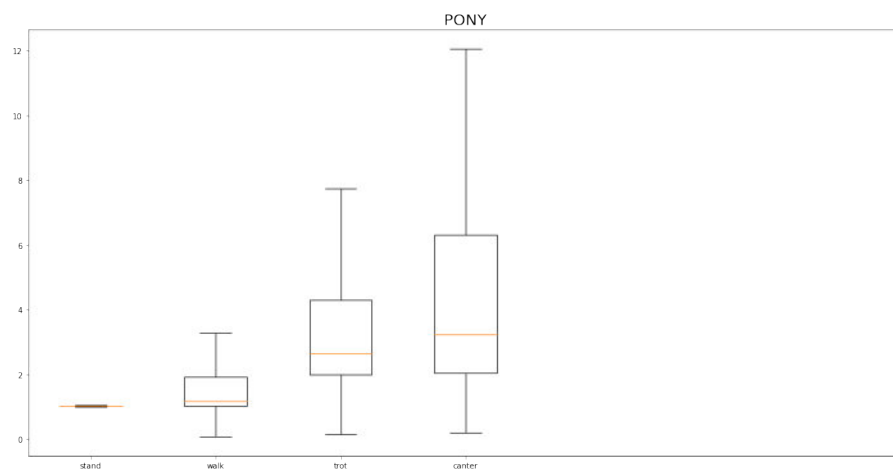
Table 2.4: Acceleration sum vector features per sample [g] measured by the Axivity AX3 accelerometer for stand, walk, trot, canter, roll, paw and flank watching; minimum, first quartile, median, third quartile, maximum, mean, root mean square, and standard deviation out of all measurements.



(a) Horse



(b) Friesian horse



(c) Pony

Figure 2.9: Boxplot of acceleration sum vector of activities for the 3 breed classes horse ($n = 4$), Friesian horse ($n = 1$) and Pony ($n = 1$), given as boxplots with medians, interquartile and absolute ranges.

Taking breed classes into account, for both sensors, increasing variations, means and medians are registered with stand, walk, trot and canter. Taking a closer look at the boxplots, as depicted in Figure 2.9, acceleration value ranges per sample for the different movements show overlaps and, therefore, no interval limits can be determined. Fixed threshold-based techniques are thus not suitable for detecting the different types of movements. For that reason, no features are extracted from the data before presenting it to the machine learning model. Also, a convolutional Neural Network (CNN) has the advantage of automatic features extraction by using strong computing capabilities. Deep learning-based classifiers can learn features and achieve better accuracy. For example, [26] uses the deep CNN features for ground-based cloud image classification. The results show that the cloud classification accuracy of CNN improved significantly, demonstrating the superiority of CNN over hand-engineered features. Besides high accuracy and good generalization, one main advantage of this way of working is that after a deep learning model is designed, it is trained in an end-to-end fashion, thus completely removing the need of manual feature engineering [27].

2.4 Machine learning model

In this research, convolutional neural networks, also known as CNNs, are used for behaviour classification. Among deep learning algorithms, CNNs are a good choice in this setting due to their specific architecture because CNNs train filters that are applied to small segments of the data, and therefore they are able to capture local data patterns and their variations. Additionally, the amount of computations and running time of CNNs is significantly lower compared to other deep learning algorithms due to a small number of connections and high parallelism of the network [27]. The gained datasets are further preprocessed and splitted in training and test data, either automated or manually. The architecture and hyperparameters of the CNN are adjusted to obtain high accuracies and regularization techniques are applied to acquire model generalization.

2.4.1 The datasets

Accelerometer tri-axial sensor data is collected from six subjects who performed seven different activities while the accelerometers were attached to their front legs. Each AX3 was set to record with a range of $\pm 8g$ for all the datasets except for one high sampling rate measurement at 1600 Hz the range is increased to $\pm 16g$ since this measurement was necessary for another research topic. These sensor data is sampled at 5 different sampling rates. Table 2.5 gives an overview of each gathered dataset for each subject number with the number of samples, the type of exercise and the type of underground.

Dataset	Subject number	sampling rate	Number of samples	Type of exercising	Underground
1	1	25 Hz	10899 (29.61% walk, 34.10% trot, 9.27% canter, 9.80% stand)	Longed	Dry sand mixed with polyflakes
2	1	25 Hz	13262 (65.61% walk, 25.32% trot, 3.20% canter, 5.53% stand)	Ridden	Dry sand mixed with polyflakes
3	1	50 Hz	22725 (41.50% walk, 29.32% trot, 3.60% canter, 16.01% stand)	Longed	Wet sand mixed with polyflakes
4	1	50 Hz	35383 (72.16% walk, 21.22% trot, 4.14% stand)	Ridden	Wet sand mixed with polyflakes
5	1	100 Hz	101649 (24.47% walk, 39.54% trot, 14.56% canter, 1.72% stand)	Longed	Dry sand mixed with polyflakes
6	2	25 Hz	21766 (47.54% walk, 16.77% trot, 3.70% canter, 17.06% stand)	Longed	Dry sand mixed with polyflakes
7	2	25 Hz	10658 (44.39% walk, 30.02% trot, 15.0% canter, 4.18% stand)	Ridden	Dry sand mixed with polyflakes
8	2	50 Hz	18940 (39.24% walk, 37.47% trot, 5.93% canter, 11.84% stand)	Longed	Wet sand mixed with polyflakes
9	2	50 Hz	21559 (38.97% walk, 24.09% trot, 24.49% canter, 5.63% stand)	Ridden	Wet sand mixed with polyflakes
10	2	50 Hz	17996 (24.79% flank-watching)	None	Dry sand mixed with polyflakes
11	2	50 Hz	13201 (79.72% walk, 1.78% stand)	None	Hard underground
12	2	50 Hz	15298 (35.27% walk)	None	Field
13	2	100 Hz	87539 (28.20% walk, 28.99% trot, 13.15% canter, 11.16% stand)	Longed	Dry sand mixed with polyflakes
14	2	100 Hz	(43.38% walk, 37.22% trot, 9.02% canter, 7.50% stand)	Ridden	Dry sand mixed with polyflakes
15	2	200 Hz	93603 (26.90% walk, 23.99% trot, 9.37% canter, 18.69% stand)	Longed	Dry sand mixed with polyflakes
16	2	200 Hz	98197 (38.18% walk, 41.47% trot, 9.21% canter, 8.56% stand)	Ridden	Dry sand mixed with polyflakes
17	2	1600 Hz	666952 (25.09% walk, 29.06% trot, 17.20 % canter, 9.65% stand)	Longed	Dry sand mixed with polyflakes
18	3	25 Hz	12306 (44.77% walk, 24.35% trot, 6.40% canter, 11.36% stand)	Longed	Dry sand mixed with polyflakes
19	3	50 Hz	19159 (42.99% walk, 27.30% trot, 6.58% canter, 9.23% stand)	Longed	Dry sand mixed with polyflakes
20	3	100 Hz	53769 (25.09% walk, 21.29% trot, 6.93% canter, 2.37% stand)	Longed	Dry sand mixed with polyflakes
21	3	200 Hz	75848 (51.02% walk, 25.62% trot, 4.41% canter, 5.08% stand)	Longed	Dry sand mixed with polyflakes
22	4	50 Hz	32130 (24.04% walk, 23.04% trot, 11.95% canter, 21.34% stand)	Longed	Dry sand mixed with polyflakes
23	4	50 Hz	29571 (43.08% walk, 31.73% trot, 12.33% canter, 6.86% stand)	Ridden	Dry sand mixed with polyflakes
24	4	50 Hz	4863 (26.97% walk, 4.91% trot, 23.60% stand, 27.20% roll)	None	Dry sand mixed with polyflakes
25	4	50 Hz	1420 (37.18% walk, 6.27% stand, 35.99% roll)	None	Dry sand mixed with polyflakes
26	4	50 Hz	9080 (32.15 % paw)	None	Hard underground
27	4	50 Hz	7178 (11.63 % walk, 12.38 % roll, 1.53 % paw)	None	Dry sand mixed with polyflakes
28	4	200 Hz	90646 (22.45% walk, 33.22 % trot, 14.29% canter, 7.48 % stand)	Longed	Dry sand mixed with polyflakes
29	4	200 Hz	114264 (43.47% walk, 24.35 % trot, 13.61% canter, 4.83 % stand)	Ridden	Dry sand mixed with polyflakes
30	4	200 Hz	32998 (15.85 % roll)	None	Dry sand mixed with polyflakes
31	4	200 Hz	6486 (26.73 % walk, 51.37 % roll)	None	Dry sand mixed with polyflakes
32	5	50 Hz	114264 (43.64% walk, 38.80 % trot, 7.79% canter, 3.57 % stand)	Ridden	Dry sand mixed with polyflakes
33	6	50 Hz	44813 (30.90% walk, 15.69 % trot, 4.72% canter, 8.32 % stand)	Longed	Dry sand mixed with polyflakes

Table 2.5: Datasets with subject number, sampling rate, number of samples as well as exercise type and track surface at data collection.

As indicated earlier, the class proportions of the seven studied activities are not distributed evenly. In Figure [2.10](#) a bar chart of the proportions of the studied movements of all horses is depicted. The class walk is mainly present with a share of 42.43% followed by trot, canter, stand, roll, paw and flank watching with a proportion of 34.10%, 11.54%, 9.61%, 1.24%, 0.62% and 0.47%, respectively.

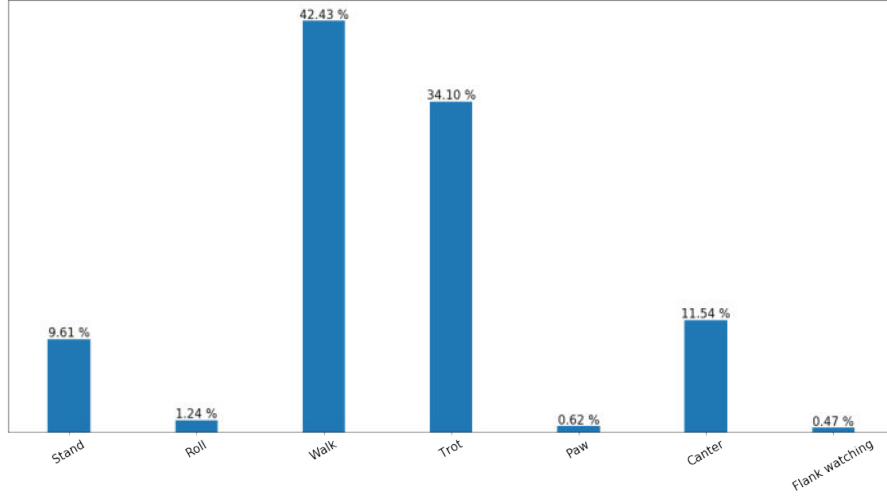


Figure 2.10: Bar chart of the movement class proportions of the studied movements of all horses.

2.4.2 Further preprocessing

As illustrated in figure 2.11 sensor data is separated into windows of n values, which can be varied according to the sampling rate of the dataset. This list with chunks of data with length n is then shuffled making sure that the model remains general and overfits less. Afterwards, the data chunks are filtered so that the n -real value vector stands for one activity at the time. Finally, the data is normalized per batch of measurements along every axis.

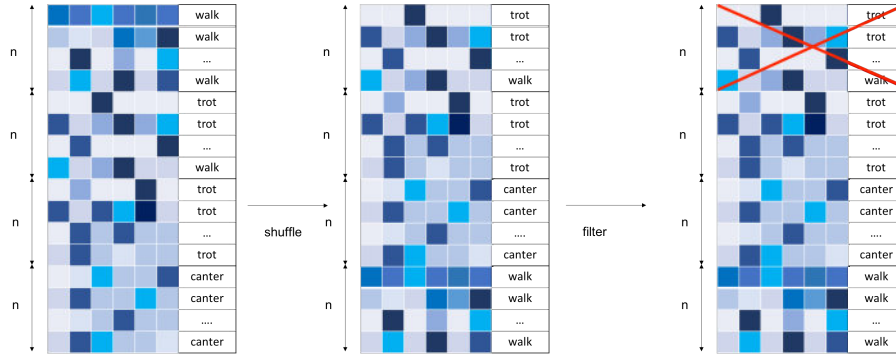


Figure 2.11: The shuffle and filter step when preprocessing the dataset.

The normalized data is separated in training and test data. First, split testing is used, this means that the apportion of the data into training and test sets is done at random. In this study, an 66-34 split ratio is used as found in the literature [28]. Secondly, the training and test set are uploaded separately to the algorithm.

2.4.3 The architecture

Now the overall architecture of the CNNs is described. CNNs are a specific type of neural networks that are generally composed of different layers as shown in Figure 2.12.

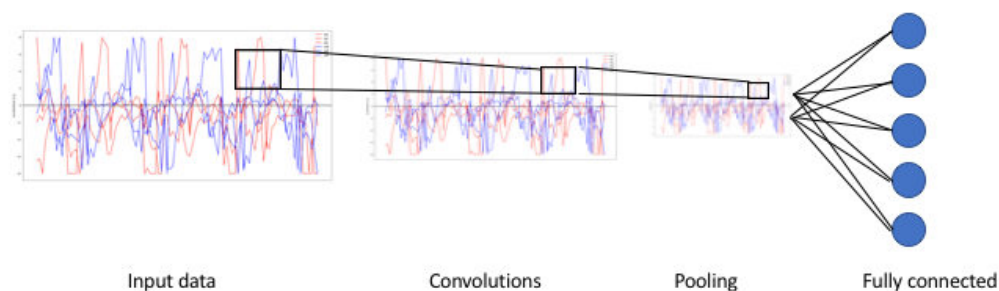


Figure 2.12: Architecture of a traditional CNN [8].

In machine learning, given an observation of an input, a probability distribution over a set of classes, is generated as an probability feature vector with dimensions number of classes $\times 1$. During the training phase, input data with certain dimensions is transformed to a vector with this size with the help of convolution, pooling and fully connected layers.

Types of layers and hyper parameters

The convolution layer (CONV) uses filters, also called kernels, that execute convolution operations as it is scanning the input data I with respect to its dimensions. Its hyper parameters include the filter size F and stride S . The resulting output O is called feature map or activation map.

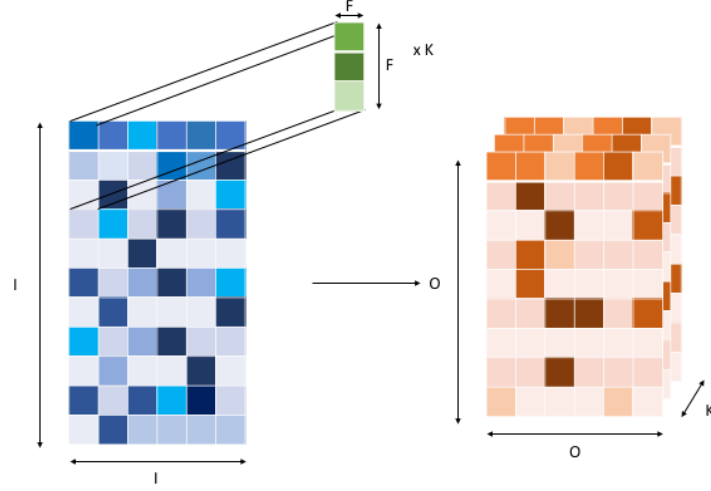


Figure 2.13: The convolution step with input $I \times I$, filter $F \times F$ and output $O \times O \times K$.

An exemplar convolution step is depicted in Figure 2.13. Each filter is a matrix of trainable weights. As the filter moves over the data, a convolution operation is performed between the data and each filter. The result is a matrix with the element-wise product and sum of the filter matrix and the input data. When training data, these weights change, and so when it is time to evaluate accelerometer data, these weights return high values if it thinks it is seeing a pattern it has seen before. The combinations of high weights from various filters let the network predict the behaviour based on accelerometer data. These weights of the kernels are the parameters that need to be trained and are updated in a step called backpropagation. The goal of backpropagation is to optimize the weights so that the neural network can learn how to correctly map arbitrary inputs to outputs. For the convolutional layer, the number of parameters that need to be trained is calculated as follows:

$$(F \times F + 1) \cdot K \quad (2.2)$$

The stride S indicates the number of pixels by which the window moves after each operation.

The pooling layer (POOL) is a downsampling operation as depicted in Figure 2.14 which significantly reduces computational complexity, typically applied after a convolution layer. It partitions the input data I into a set of non-overlapping rectangles and, for each such sub-region, outputs a value. The idea is that the exact location of a feature is less important than its rough location relative to other features. The output O is called the pooled feature map. Its hyperparameters include again the filter size F and stride S . The pooling operation used in this research, max-pooling, is characterized by outputting the maximum value among a set of nearby inputs. The max-pooling layers do not have any weight to be updated so there are no parameters trained in this layer.

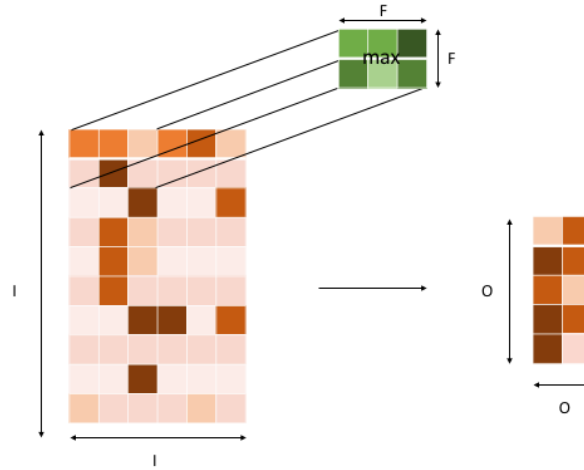


Figure 2.14: The max-pooling step with input $I \times I$, filter $F \times F$ and output $O \times O$.

The convolution layer and the pooling layer can be fine-tuned with respect to the hyperparameters e.g. the dimensions of the filter and the stride.

The fully connected layer (FC) operates on a flattened input (features from the stacked convolutional and pooling layers are flattened to form feature vectors) where each input is connected to all neurons. These layers are generally found towards the end of CNN architectures and can be used to optimize objectives such as class scores.

Zero-padding

The zero-padding (ZP) technique as depicted in Figure 2.15, adds symmetrically zeroes to the input matrix, during the convolution, to keep the size of the convolutional output and input equal. It helps to maintain desirable volume sizes and to preserve the border data. It is popular to apply the zero-padding technique for the 2-D CNN, and those models with the zero-padding can achieve the state-of-art performance in their tasks [29].

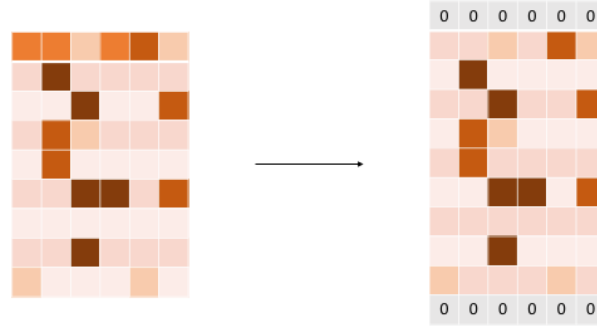


Figure 2.15: The zero-padding technique.

Activation functions

Activation functions are used to introduce non-linear properties to a neural network. It means that the CNN can successfully predict the class of a function which is divided by a decision boundary which is not linear. An activation function squashes the output value of a node into a range. In this study, two types of activation functions are used: Rectified Linear Unit and Softmax.

The ReLu function is defined as follows [30]:

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases} \quad (2.3)$$

The ReLu function takes an input 'x' and returns 'x' if it is positive, otherwise it returns zero. A ReLu function is used here because combining deep convolutional neural networks with ReLUs allows a faster training than for example their equivalents with hyperbolic functions [31].

The Softmax function is defined as follows [32]:

$$\begin{pmatrix} p_1 \\ \dots \\ p_n \end{pmatrix} \text{ with } p_i = \frac{e^{x_i}}{\sum_1^n e^{x_j}} \quad (2.4)$$

The softmax function does exponentiation and normalization over the feature vector input of scores x via the sum of exponents and outputs a vector of output probability p at the end of the architecture. It squashes a vector of size n between 0 and 1. A combination of a fully-connected layer and a softmax classifier, also called a softmax layer acting as topmost layer, can be utilized to recognize distinct activities.

An interesting parameter can be calculated from the output of the softmax function i.e., the negative log-likelihood, also called the cross-entropy loss. The cross-entropy loss is defined as

follows [33]:

$$L(p_i) = -\ln(p_i) \quad (2.5)$$

This is summed for all the correct classes. The loss becomes high at smaller probabilities and decreases at larger values. Because the loss function is the summation of the probabilities of all the correct classes, whenever the network assigns a high probability at the correct class, the loss is low, but when the network assigns a low probability at the correct class, the loss is high.

Regularization techniques

Regularization techniques are used to control the overfitting of data during training because it is a considerable drawback since generalization of the model is endangered. Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data. In this model, two regularization techniques are used: dropout and L2 regularization.

Dropout modifies the network itself to avoid overfitting. In this work, different dropout ratios were tested at different layers of the network to prevent the model from overfitting. It works by randomly and temporarily deleting a node in the network during the training phase to inhibit the neurons from adjusting exceedingly well to the training data. As the neurons were dropped out, the connecting weights will be ruled out from updating. In this way, the dropped out neurons do not contribute to the forward pass and do not take part in back-propagation. This compels the network to learn from the flawed patterns and thus improve the generalization of the model [34]. In our proposed CNN architecture, dropout is applied after the max-pooling layers.

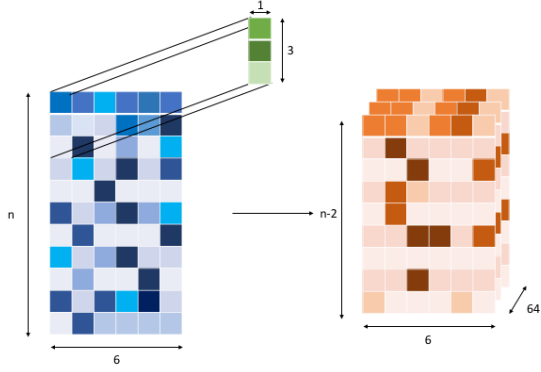
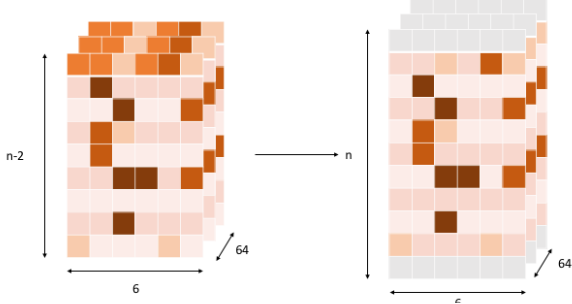
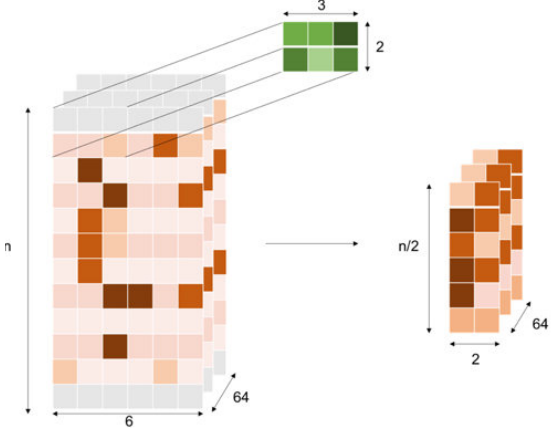
L2 regularization, also named weight decay, is a regularization method that adds an extra term $\lambda \sum_w w^2$ into the loss function that penalizes large weights [32]:

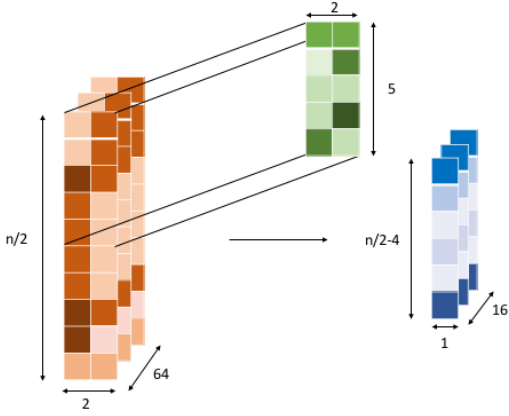
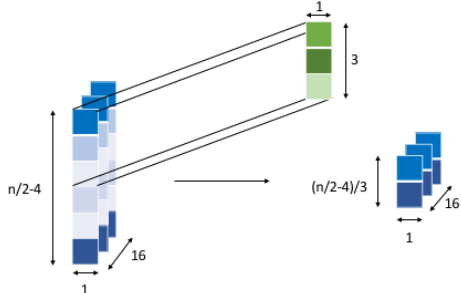
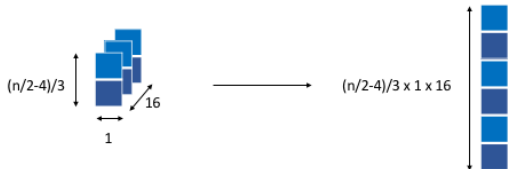
$$L = L_0 + \lambda \sum_w w^2 \quad (2.6)$$

where L_0 is the unregularized cost function, and λ is the weight decay coefficient.

In this study, a multilayer convolutional network, is used with two convolutional layers, which are followed by max-pooling layers, and two fully connected layers. The output of the last fully-connected layer is fed to a 7-way softmax layer which produces a distribution over the seven class labels: stand, walk, trot, canter, roll, paw, flank watching. Features are automatically extracted from the preprocessed time-series sensor data. A detailed overview of all the layers, with the feature map input and output size and the number of trainable parameters can be found in Table 2.6. The first convolutional layer filters the $n \times 6 \times 1$ input acceleration data with 64 kernels of size 3×1 and stride 1. The L2 regularization technique is used in this layer with a weight decay coefficient of 0.01 [35]. After the first convolutional layer a zero-padding is used such that the output has the same length as the original input. Then a max-pooling operation is done. The second convolutional layer takes as input the (pooled) output of the first convolutional layer and

filters it with 16 kernels of size 5×2 and stride 1. Both layers contain an activation layer using rectified units (ReLU) and dropout of 0.55 is used [34]. The Adam optimizer is used for training the neural network through back propagation. Training is done for 400 epochs, with an early stopping criterion of halting training when there is no increase in accuracy during the last 60 epochs [32].

Layer	Illustration	Input and output size	Number of parameters
CONV		$n \times 6 \times 1$ $(n-2) \times 6 \times 64$	256
ZP		$(n-2) \times 6 \times 64$ $n \times 6 \times 64$	0
POOL		$n \times 6 \times 64$ $\frac{n}{2} \times 2 \times 64$	0

		$\frac{n}{2} \times 2 \times 64$	
DROPOUT			0
		$\frac{n}{2} \times 2 \times 64$	
CONV		$\frac{n}{2} \times 2 \times 64$ $(\frac{n}{2} - 4) \times 1 \times 64$	10256
POOL		$(\frac{n}{2} - 4) \times 1 \times 16$ $(\frac{\frac{n}{2}-4}{3}) \times 1 \times 16$	0
DROPOUT		$(\frac{\frac{n}{2}-4}{3}) \times 1 \times 16$	0
		$(\frac{\frac{n}{2}-4}{3}) \times 1 \times 16$	
FLATTEN		$(\frac{\frac{n}{2}-4}{3}) \times 1 \times 16$ $16 \frac{(\frac{n}{2}-4)}{3}$	0

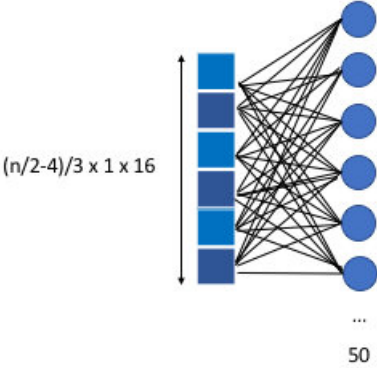
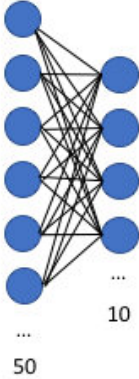
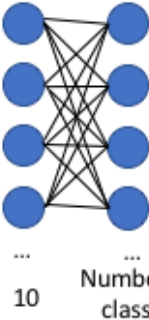
FC	<div>  <div> <div>$(n/2-4)/3 \times 1 \times 16$</div> <div>...</div> <div>50</div> </div> </div>	<div> $16 \frac{(\frac{n}{2}-4)}{3}$ </div> <div>50</div>	<div> $(16 \frac{(\frac{n}{2}-4)}{3} + 1) \cdot 50$ </div>
FC	<div>  <div>...</div> <div>10</div> <div>...</div> <div>50</div> </div>	<div>50</div> <div>10</div>	<div>510</div>
SOFTMAX	<div>  <div>...</div> <div>10</div> <div>...</div> <div>Number of classes</div> </div>	<div>10</div> <div>7</div>	<div>77</div>

Table 2.6: Overview of model architecture.

Table [2.7](#) shows the experimental setup.

Parameter	Value
The size of the input vector	n
The number of input channels	6
The number of feature maps	64-16
Filter size	$3 \times 1 - 5 \times 2$
Stride	1
Pooling size	$2 \times 3 - 3 \times 1$
Activation function	ReLu and Softmax
Weight decay	0.01 (L2 regularization)
The probability of dropout	0.55
Maximum epochs	400
Optimization (back propagation)	Adam optimizer

Table 2.7: Experimental setup.

Chapter 3

Results

The deep network in this work is developed and evaluated in Python language using Keras with Tensorflow as backend. The experiments are performed on a MacBook Pro with an Intel Core i5, CPU (2.5 GHz), 4 GB 1600 MHz DDR3 memory and an Intel HD Graphics 4000 1536 MB graphics card. Each training epoch took between approximately 1 s to 30 s to complete.

Figure 3.1 illustrates exemplar two second data windows of the four gaits and the other behaviours, from the left and right accelerometer worn on the lateral side of the tendon boot. Standing data typically appear as constant signals indicating less movement (figure 3.1(a)) while faster gait data consist of increasing fluctuating movements (figure 3.1(b), 3.1(c) and 3.1(d)).

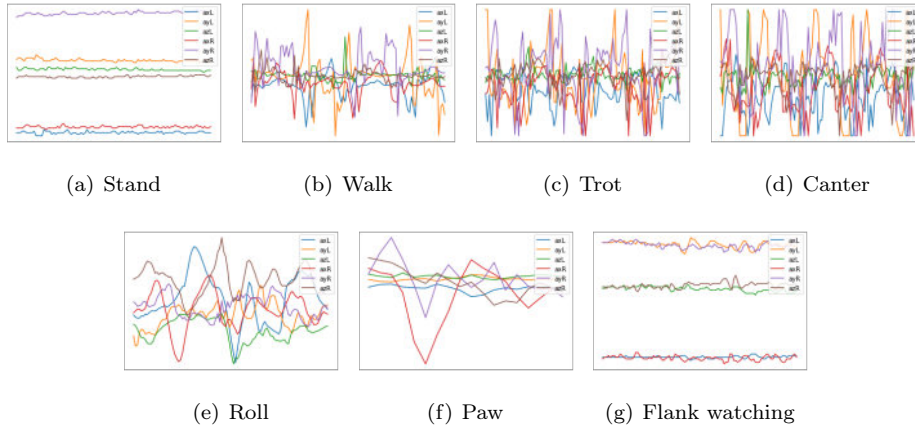


Figure 3.1: Typical accelerometer patterns of (a) stand, (b) walk, (c) trot, (d) canter, (e) roll, (f) paw and (g) flank watching in a 2 s window. The blue, yellow, green lines represent X,Y,Z signals from the left accelerometer and the red, purple and brown lines represent X,Y,Z signals from the right accelerometer , respectively.

To train the convolutional neural network, separate training and validation sets are needed and can be selected in various ways. First, a training and validation set are obtained by automatically

splitting the training and the validation data with a fixed ratio of 66/34 referred to as the 'First dataset'. In this case it is very likely that validation data of a particular behaviour of one horse is already seen by the model during the training phase. If the model then selects the wrong features, validation accuracies on an unseen horse could be low and the model would lack the ability to generalise. Secondly, the 50 Hz dataset, which contains every behaviour, is resampled to 25 Hz, 100 Hz and 200 Hz and merged with the original dataset at that sampling rate referred to as the 'Second dataset'. The model can then be assessed for any behaviour at each sampling rate. Again automatic split testing is used to obtain the training and validation set. Finally, the separation of the training and validation data is attained manually and as a result the model is not validated on data from a seen horse referred to as the 'Third dataset'. In this case data from the lame horse is used to validate our model while it is trained on healthy horses, to further asses the generalization of the model.

First dataset

The acquired training dataset consists of the merged accelerometer data from the measurements at equal sampling rate. The number of labelled rows, the total time of movement data and the number of different subjects in the final dataset is displayed in Table 3.1. 40% of the total measured time of movement data is measured at a sampling rate of 50 Hz from six different subjects performing seven different behaviours. Only datasets 11 and 12 as depicted in Table 2.5 are not taken into account since those were preprocessed at a later stage.

	25 Hz	50 Hz	100 Hz	200 Hz	1600 Hz
Time measured [s]	2752	5492	3006	2560	417
Number of samples	68800	274580	300640	511920	666952
Number of subjects	3	6	3	3	1
Number of behaviours	4	7	4	5	4

Table 3.1: Total time of movement data, number of samples, number of subjects and number of behaviours for each sampling rate of the merged accelerometer data.

The training data for the CNN is obtained by automatically splitting the dataset in two parts: two thirds for training and one third for testing. The per class and overall validation accuracy is presented in Table 3.2 for a time interval ranging from 0.2 s to 3 s. At lower sampling rates i.e., 25 Hz and 50 Hz, the network cannot be trained at short time segments since the size of the input vector becomes too small for the designed neural network.

		25 Hz	50 Hz	100 Hz	200 Hz	1600 Hz
n = 0.2 s	Stand	-	-	99.63	100.00	100.00
	Walk	-	-	97.30	98.66	97.78
	Trot	-	-	90.67	98.58	96.11
	Canter	-	-	32.72	96.99	98.18

	Roll	-	-	-	70.83	-
	Paw	-	-	-	-	-
	Flank watching	-	-	-	-	-
	Validation accuracy	-	-	85.00	97.98	97.53
n = 0.4 s	Stand	-	100.00	100.00	100.00	100.00
	Walk	-	99.11	99.58	100.00	100.00
	Trot	-	91.21	98.30	99.67	100.00
	Canter	-	24.56	91.64	99.53	96.30
	Roll	-	37.70	-	82.50	-
	Paw	-	-	-	-	-
	Flank watching	-	12.86	-	-	-
	Validation accuracy	-	87.39	97.82	99.42	99.28
n = 0.6 s	Stand	100.00	98.99	100.00	100.00	100.00
	Walk	97.18	99.74	99.81	100.00	100.00
	Trot	88.28	97.77	99.81	100.00	100.00
	Canter	0.00	93.69	99.36	98.72	100.00
	Roll	-	85.71	-	96.55	-
	Paw	-	83.33	-	-	-
	Flank watching	-	86.36	-	-	-
	Validation accuracy	89.22	97.94	99.69	99.82	100.00
n = 0.8 s	Stand	100.00	99.42	100.00	100.00	100.00
	Walk	99.81	100.00	100.00	100.00	100.00
	Trot	98.57	98.82	99.74	100.00	100.00
	Canter	0.00	98.91	100.00	100.00	100.00
	Roll	-	77.41	-	100.00	-
	Paw	-	0.00	-	-	-
	Flank watching	-	94.44	-	-	-
	Validation accuracy	91.20	98.97	99.90	100.00	100.00
n = 1 s	Stand	100.00	98.80	100.00	100.00	100.00
	Walk	100.00	100.00	100.00	100.00	100.00
	Trot	97.21	99.33	99.71	100.00	97.78
	Canter	0.00	98.63	100.00	100.00	92.00
	Roll	-	90.48	-	72.22	-
	Paw	-	95.24	-	-	-
	Flank watching	-	88.00	-	-	-
	Validation accuracy	91.37	99.17	99.87	99.24	97.14
n = 1.2 s	Stand	100.00	99.16	100.00	100.00	100.00
	Walk	100.00	100.00	100.00	100.00	100.00
	Trot	99.37	100.00	100.00	100.00	100.00
	Canter	0.00	99.07	100.00	98.39	100.00

	Roll	-	94.12	-	90.00	-
	Paw	-	94.12	-	-	-
	Flank watching	-	100.00	-	-	-
	Validation accuracy	94.00	99.69	100.00	99.63	100.00
n = 1.4 s	Stand	100.00	100.00	100.00	100.00	100.00
	Walk	100.00	100.00	100.00	100.00	100.00
	Trot	100.00	100.00	100.00	100.00	100.00
	Canter	100.00	100.00	100.00	100.00	100.00
	Roll	-	84.21	-	92.31	-
	Paw	-	90.91	-	-	-
	Flank watching	-	92.86	-	-	-
	Validation accuracy	100.00	99.49	100.00	99.78	100.00
n = 1.6 s	Stand	100.00	100.00	100.00	100.00	100.00
	Walk	100.00	99.79	100.00	100.00	100.00
	Trot	100.00	100.00	100.00	100.00	100.00
	Canter	0.00	100.00	98.31	100.00	75.00
	Roll	-	93.33	-	83.33	-
	Paw	-	100.00	-	-	-
	Flank watching	-	84.62	-	-	-
	Validation accuracy	92.05	99.57	99.78	99.75	95.24
n = 1.8 s	Stand	100.00	98.48	100.00	100.00	100.00
	Walk	100.00	100.00	100.00	100.00	100.00
	Trot	100.00	100.00	100.00	99.20	100.00
	Canter	96.43	100.00	98.28	97.44	90.00
	Roll	-	81.82	-	83.33	-
	Paw	-	100.00	-	-	-
	Flank watching	-	92.31	-	-	-
	Validation accuracy	99.76	99.46	99.74	99.15	98.21
n = 2 s	Stand	100.00	98.53	100.00	100.00	100.00
	Walk	100.00	100.00	100.00	100.00	100.00
	Trot	99.11	100.00	100.00	100.00	100.00
	Canter	95.65	100.00	100.00	100.00	71.43
	Roll	-	90.91	-	88.88	-
	Paw	-	100.00	-	-	-
	Flank watching	-	83.33	-	-	-
	Validation accuracy	99.46	99.46	100.00	99.68	95.83
n = 2.2 s	Stand	100.00	100.00	100.00	100.00	100.00
	Walk	100.00	100.00	100.00	100.00	100.00
	Trot	100.00	100.00	99.30	100.00	100.00
	Canter	0.00	100.00	100.00	100.00	50.00

	Roll	-	71.43	-	50.00	-
	Paw	-	100.00	-	-	-
	Flank watching	-	91.67	-	-	-
	Validation accuracy	89.82	99.55	99.68	98.93	90.70
n = 2.4 s	Stand	100.00	100.00	100.00	100.00	100.00
	Walk	100.00	100.00	100.00	100.00	100.00
	Trot	100.00	100.00	100.00	100.00	100.00
	Canter	100.00	100.00	100.00	100.00	85.71
	Roll	-	50.00	-	100.00	-
	Paw	-	100.00	-	-	-
	Flank watching	-	75.00	-	-	-
	Validation accuracy	100.00	99.66	100.00	100.00	97.50
n = 2.6 s	Stand	100.00	100.00	100.00	100.00	100.00
	Walk	100.00	100.00	100.00	100.00	100.00
	Trot	100.00	100.00	100.00	100.00	100.00
	Canter	100.00	100.00	100.00	89.66	25.00
	Roll	-	60.00	-	50.00	-
	Paw	-	75.00	-	-	-
	Flank watching	-	75.00	-	-	-
	Validation accuracy	100.00	98.91	100.00	98.26	82.86
n = 2.8 s	Stand	100.00	98.11	100.00	100.00	100.00
	Walk	100.00	100.00	100.00	100.00	100.00
	Trot	100.00	100.00	99.06	100.00	100.00
	Canter	0.00	100.00	100.00	93.33	40.00
	Roll	-	100.00	-	100.00	-
	Paw	-	100.00	-	-	-
	Flank watching	-	100.00	-	-	-
	Validation accuracy	94.42	99.80	99.57	99.05	90.63
n = 3 s	Stand	100.00	100.00	100.00	100.00	100.00
	Walk	100.00	100.00	100.00	100.00	100.00
	Trot	100.00	100.00	98.91	100.00	100.00
	Canter	0.00	97.22	100.00	90.00	66.67
	Roll	-	100.00	-	100.00	-
	Paw	-	100.00	-	-	-
	Flank watching	-	100.00	-	-	-
	Validation accuracy	94.82	99.79	99.53	98.97	96.55

Table 3.2: Accuracies for each behavioural class and overall accuracy for each sampling rate.

Second dataset

Not all movements are measured at any sampling rate, therefore the resampling method is proposed to resolve the imbalanced dataset by processing the dataset and producing a balanced dataset. This technique consists of two implementations: oversampling and undersampling. The 50 Hz dataset, which contains every behaviour, is resampled to the three different sampling rates i.e., 25 Hz, 100 Hz and 200 Hz and merged with the original dataset at the same sampling rate. The total time measured, the original number of samples and the number of samples after resampling can be found in Table 3.3. The number of subjects is six and the number of behaviours is seven.

	25 Hz	50 Hz	100 Hz	200 Hz
Time measured [s]	8244	5492	8498	8052
Original number of samples	343380	274580	575220	786500
Number of samples after resampling	206090	274580	849800	1610240
Number of subjects			6	
Number of behaviours			7	

Table 3.3: Total time of movement data and number of samples for each sampling rate of the merged accelerometer data including resampled datasets.

The per class and overall validation accuracy is presented in Table 3.4 for a time interval ranging from 0.6 s to 3 s.

		25 Hz	50 Hz	100 Hz	200 Hz
n = 0.6 s	Stand	100.00	98.99	100.00	93.37
	Walk	99.64	99.74	100.00	99.81
	Trot	22.29	97.77	100.00	99.35
	Canter	0.00	93.69	96.61	99.39
	Roll	39.29	85.71	100.00	96.77
	Paw	0.00	83.33	100.00	91.11
	Flank watching	0.00	86.36	40.00	91.37
	Validation accuracy	67.59	97.94	99.28	98.75
n = 0.8 s	Stand	100.00	99.42	98.81	97.13
	Walk	100.00	100.00	99.92	99.84
	Trot	80.14	98.82	99.48	99.33
	Canter	78.02	98.91	98.37	98.82
	Roll	0.00	77.41	88.89	98.00
	Paw	0.00	0.00	0.00	0.00
	Flank watching	0.00	94.44	78.94	58.62
	Validation accuracy	89.89	98.97	99.10	98.78
n = 1 s	Stand	100.00	98.80	98.54	97.36
	Walk	99.91	100.00	99.90	100.00

	Trot	94.23	99.33	99.48	99.71
	Canter	92.17	98.63	99.17	99.55
	Roll	60.00	90.48	91.30	87.88
	Paw	61.54	95.24	100.00	92.86
	F flank watching	61.54	88.00	77.78	90.63
	Validation accuracy	96.55	99.17	99.22	99.24
n = 1.2 s	Stand	100.00	99.16	98.81	99.43
	Walk	100.00	100.00	100.00	100.00
	Trot	98.95	100.00	99.84	99.82
	Canter	98.78	99.07	98.96	98.32
	Roll	68.42	94.12	86.67	95.00
	Paw	88.24	94.12	88.89	92.86
	F flank watching	92.31	100.00	83.33	71.43
	Validation accuracy	99.12	99.69	99.37	99.23
n = 1.4 s	Stand	100.00	100.00	97.80	95.90
	Walk	99.87	100.00	99.86	99.85
	Trot	98.73	100.00	99.60	99.57
	Canter	97.39	100.00	98.81	99.21
	Roll	100.00	84.21	100.00	100.00
	Paw	100.00	90.91	90.00	92.31
	F flank watching	57.14	92.86	88.89	80.00
	Validation accuracy	98.95	99.49	99.33	99.10
n = 1.6 s	Stand	99.25	100.00	100.00	99.22
	Walk	100.00	99.79	100.00	100.00
	Trot	98.84	100.00	99.58	100.00
	Canter	96.49	100.00	99.30	99.07
	Roll	92.31	93.33	84.62	95.83
	Paw	90.00	100.00	80.00	83.33
	F flank watching	83.33	84.62	93.33	84.21
	Validation accuracy	99.01	99.57	99.50	99.48
n = 1.8 s	Stand	100.00	98.48	98.61	95.89
	Walk	100.00	100.00	100.00	100.00
	Trot	99.68	100.00	99.45	99.19
	Canter	100.00	100.00	99.00	98.04
	Roll	100.00	81.82	50.00	90.48
	Paw	100.00	100.00	100.00	100.00
	F flank watching	60.00	92.31	100.00	90.91
	Validation accuracy	99.40	99.46	99.38	99.00
n = 2 s	Stand	99.03	98.53	100.00	100.00
	Walk	100.00	100.00	100.00	100.00

	Trot	99.68	100.00	100.00	99.33
	Canter	97.65	100.00	99.07	98.97
	Roll	100.00	90.91	80.00	94.44
	Paw	100.00	100.00	100.00	100.00
	F flank watching	50.00	83.33	92.31	73.68
	Validation accuracy	99.10	99.46	99.72	99.06
n = 2.2 s	Stand	100.00	100.00	100.00	98.55
	Walk	100.00	100.00	100.00	100.00
	Trot	99.66	100.00	100.00	99.63
	Canter	100.00	100.00	98.98	100.00
	Roll	100.00	71.43	100.00	92.86
	Paw	100.00	100.00	100.00	100.00
	F flank watching	66.67	91.67	75.00	84.62
	Validation accuracy	99.70	99.55	99.69	99.42
n = 2.4 s	Stand	100.00	100.00	98.55	100.00
	Walk	100.00	100.00	100.00	100.00
	Trot	100.00	100.00	99.65	100.00
	Canter	100.00	100.00	98.94	98.67
	Roll	90.91	50.00	81.82	100.00
	Paw	100.00	100.00	90.00	80.00
	F flank watching	33.33	75.00	100.00	62.50
	Validation accuracy	99.23	99.66	99.32	99.36
n = 2.6 s	Stand	100.00	100.00	100.00	100.00
	Walk	100.00	100.00	100.00	100.00
	Trot	100.00	100.00	98.49	100.00
	Canter	100.00	100.00	98.81	100.00
	Roll	100.00	60.00	62.50	100.00
	Paw	100.00	75.00	100.00	83.33
	F flank watching	85.71	75.00	100.00	66.67
	Validation accuracy	99.88	98.91	99.00	99.58
n = 2.8 s	Stand	100.00	98.11	98.21	98.53
	Walk	100.00	100.00	100.00	100.00
	Trot	100.00	100.00	99.19	98.96
	Canter	100.00	100.00	100.00	100.00
	Roll	100.00	100.00	87.50	92.31
	Paw	100.00	100.00	100.00	40.00
	F flank watching	71.43	100.00	100.00	40.00
	Validation accuracy	99.74	99.80	99.46	98.48
n = 3 s	Stand	100.00	100.00	100.00	100.00
	Walk	100.00	100.00	100.00	100.00

Trot	100.00	100.00	100.00	100.00
Canter	96.61	97.22	94.29	96.88
Roll	100.00	100.00	100.00	90.91
Paw	100.00	100.00	100.00	100.00
Flank watching	40.00	100.00	100.00	60.00
Validation accuracy	99.28	99.79	99.41	99.17

Table 3.4: Accuracies for each behavioural class and overall accuracy for each sampling rate.

Third dataset

The datasets are splitted manually into a training set and a validation set. The model is trained on all the datasets available at a certain sampling rate except for the lame horse 3, which is used to validate the CNN. At 1600 Hz, one dataset is available from horse 2 so the algorithm is not validated for this sampling rate. Furthermore, horse 3 performed only four movements i.e., 'stand', 'walk', 'trot' and 'canter' and solely those are thus investigated. In Table 3.5 the time measured, the number of samples and the number of subjects is presented for the training dataset and the validation dataset.

		25 Hz	50 Hz	100 Hz	200 Hz
Training data	Time measured [s]	2261	5623	2469	2180
	Number of samples	56525	281130	246880	435960
	Number of subjects	2	5	2	2
Validation data	Time measured [s]	492	383	538	379
	Number of samples	12306	19159	53769	75848
	Number of subjects			1	
	Number of behaviours			4	

Table 3.5: Total time of movement data, number of samples and number of subjects for each sampling rate of the training dataset and the validation dataset.

The per class and overall validation accuracy is presented in Table 3.6 for a time interval ranging from 0.6 s to 3 s.

		25 Hz	50 Hz	100 Hz	200 Hz
n = 0.6 s	Stand	100.00	100.00	100.00	100.00
	Walk	100.00	100.00	99.05	100.00
	Trot	75.52	87.95	81.36	99.35
	Canter	0.00	76.32	84.91	100.00
	Validation accuracy	86.17	94.42	90.63	99.81

n = 0.8 s	Stand	100.00	100.00	100.00	100.00
	Walk	99.25	100.00	99.35	100.00
	Trot	95.10	88.52	97.67	100.00
	Canter	0.00	89.29	87.50	100.00
Validation accuracy		91.57	95.57	97.32	100.00
n = 1 s	Stand	100.00	100.00	100.00	100.00
	Walk	97.20	100.00	100.00	100.00
	Trot	97.37	94.90	96.00	100.00
	Canter	0.00	91.30	100.00	100.00
Validation accuracy		91.36	97.68	98.47	100.00
n = 1.2 s	Stand	100.00	100.00	100.00	100.00
	Walk	98.29	100.00	100.00	100.00
	Trot	98.92	97.53	98.80	100.00
	Canter	0.00	76.47	100.00	100.00
Validation accuracy		92.47	97.56	99.52	100.00
n = 1.4 s	Stand	100.00	100.00	100.00	100.00
	Walk	99.32	100.00	100.00	100.00
	Trot	98.72	95.65	98.53	100.00
	Canter	0.00	100.00	100.00	100.00
Validation accuracy		93.53	98.56	99.43	100.00
n = 1.6 s	Stand	100.00	100.00	100.00	100.00
	Walk	100.00	100.00	100.00	100.00
	Trot	100.00	96.42	98.28	100.00
	Canter	0.00	100.00	100.00	100.00
Validation accuracy		93.87	98.86	99.32	100.00
n = 1.8 s	Stand	100.00	100.00	100.00	100.00
	Walk	100.00	100.00	100.00	100.00
	Trot	98.28	94.00	97.87	100.00
	Canter	0.00	90.91	100.00	100.00
Validation accuracy		92.92	97.40	99.18	100.00
n = 2 s	Stand	100.00	100.00	100.00	100.00
	Walk	100.00	100.00	100.00	100.00
	Trot	98.15	95.65	97.67	100.00
	Canter	0.00	100.00	100.00	75.00
Validation accuracy		94.24	98.58	99.10	99.22
n = 2.2 s	Stand	100.00	100.00	100.00	100.00
	Walk	100.00	100.00	100.00	100.00
	Trot	97.67	97.56	100.00	100.00
	Canter	0.00	100.00	100.00	100.00

Validation accuracy		94.16	99.17	100.00	100.00
n = 2.4 s	Stand	100.00	100.00	100.00	100.00
	Walk	100.00	100.00	100.00	100.00
	Trot	100.00	97.30	96.97	100.00
	Canter	0.00	100.00	100.00	100.00
Validation accuracy		94.89	99.10	98.84	100.00
n = 2.6 s	Stand	100.00	100.00	100.00	100.00
	Walk	100.00	100.00	100.00	100.00
	Trot	100.00	93.94	96.55	100.00
	Canter	0.00	100.00	100.00	100.00
Validation accuracy		94.89	97.94	98.72	100.00
n = 2.8 s	Stand	100.00	100.00	100.00	100.00
	Walk	100.00	100.00	100.00	100.00
	Trot	100.00	100.00	96.30	100.00
	Canter	0.00	100.00	100.00	100.00
Validation accuracy		96.09	100.00	98.57	100.00
n = 3 s	Stand	100.00	100.00	100.00	100.00
	Walk	100.00	100.00	100.00	100.00
	Trot	100.00	100.00	100.00	100.00
	Canter	0.00	80.00	100.00	100.00
Validation accuracy		95.08	98.73	100.00	100.00

Table 3.6: Accuracies for each behavioural class and overall accuracy for each sampling rate.

In Figure 3.2 validation and training data is depicted for four sampling rates i.e., 25 Hz, 50 Hz, 100 Hz and 200 Hz of the three datasets.

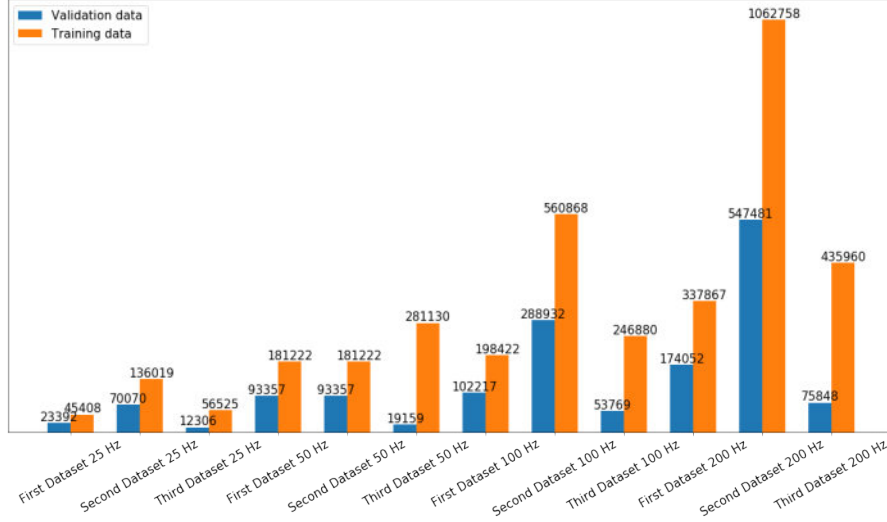


Figure 3.2: Training and validation data for three datasets at sampling rate 25 Hz, 50 Hz, 100 Hz and 200 Hz.

In Section 3.1 the effects of the sampling rate on the model accuracy is discussed, Section 3.2 takes a closer look at the effects of the length of the time interval and in Section 3.3 the effect of the combination of the time interval and sampling rate is examined for the validation accuracies obtained in Table 3.2, 3.4 and 3.6. In the last Section 3.4 the effect of the underground on the classification accuracy of the class walk is studied.

3.1 Effects of the sampling rate of the accelerometer

A lower sampling rate could extend the battery lifetime by reducing storage load and minimizing both sensing and transmitting energies. For this purpose, it is important to investigate the relation between the sampling rate and the performance of the algorithm.

First dataset

In Figure 3.3 the overall validation accuracy as function of the sampling rate is depicted for time intervals ranging from 0.2 s to 3 s.

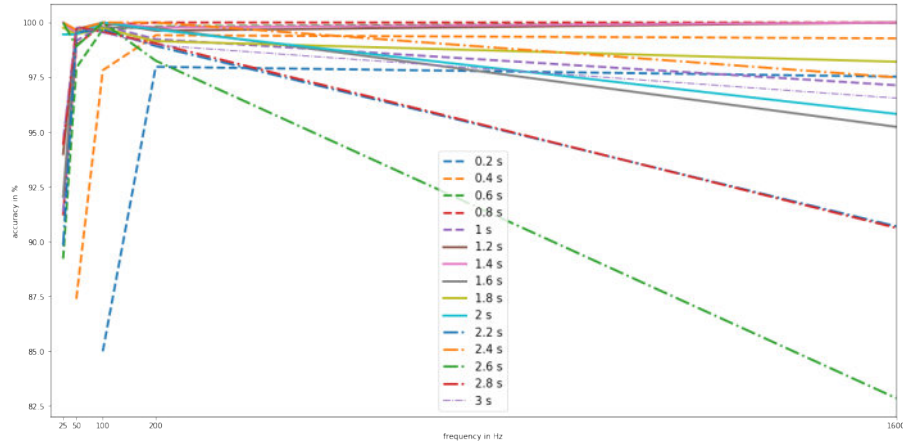


Figure 3.3: Performance of convnet with increasing sampling rate.

At lower sampling rates, for smaller time intervals, the accuracy decreases except for the following time intervals (1.4 s, 1.8 s, 2 s, 2.4 s, 2.6 s) for which the accuracy remains more or less constant from 25 Hz to 200 Hz. When taking a closer look at the class accuracies which can be found in Table 3.2, the main cause of the lower accuracies at a sampling rate of 25 Hz is due to the poor classification of the gait canter which is at 8 out of 13 time intervals zero. When looking at the 8 confusion matrices as partially depicted in Figure 3.4 for this poorly classified canter instances, one can see that the class canter is in all the matrices primarily classified as trot and to a lesser extend as walk. The class canter has a share of (7%) of the total dataset of 25 Hz and is thus the least present so fewer instances are available for training.

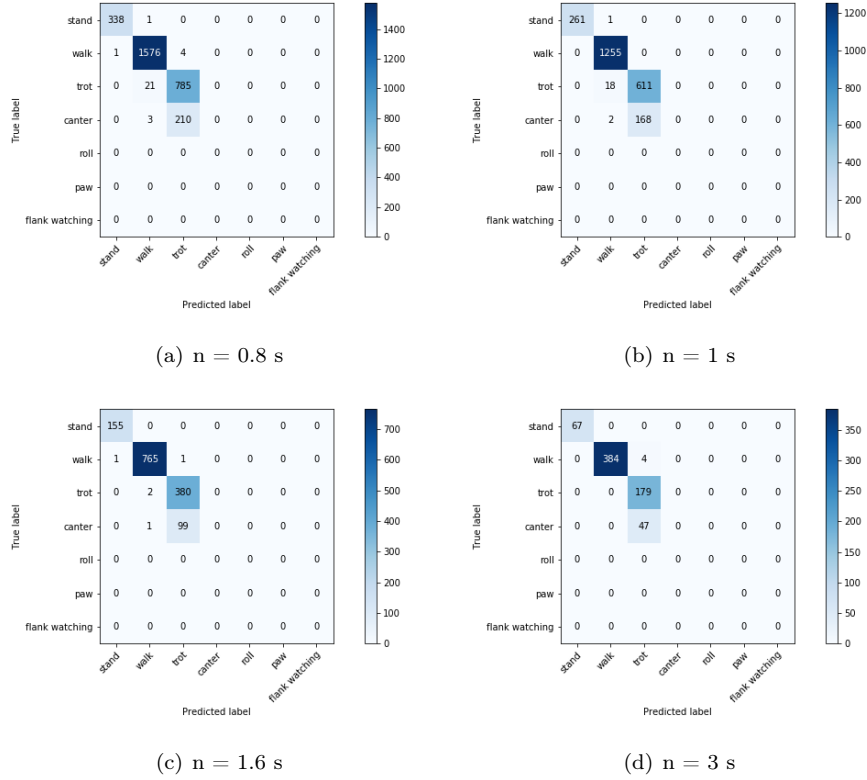


Figure 3.4: Confusion matrix for training and test set at a sampling rate of 25 Hz for different time intervals.

For the other 5 of the 13 time intervals, the class canter reaches an accuracy of 95% or above. In Figure 3.5 the training and validation accuracy plots for two high scoring (1.4 s and 2.4 s) and two low scoring (0.8 s and 1.6 s) time intervals are depicted. The validation accuracy for $n=0.8$ s highly fluctuates and for $n=1.6$ s reaches a plateau. For the latter time interval the validation curve intersects the training accuracy curve. For the two high performing time intervals, the accuracy of the training and validation set gradually increase which means that the algorithm learns generalizable features. In Figure 3.6 the loss plots are depicted for the two high scoring and two low scoring time intervals. Again, high fluctuations can be noticed in the validation loss plot for the time interval of 0.8 s. The loss plot of the low accuracy time interval of 1.6 s shows a gradually decrease together with the training loss. The same applies for the high accuracy time intervals.

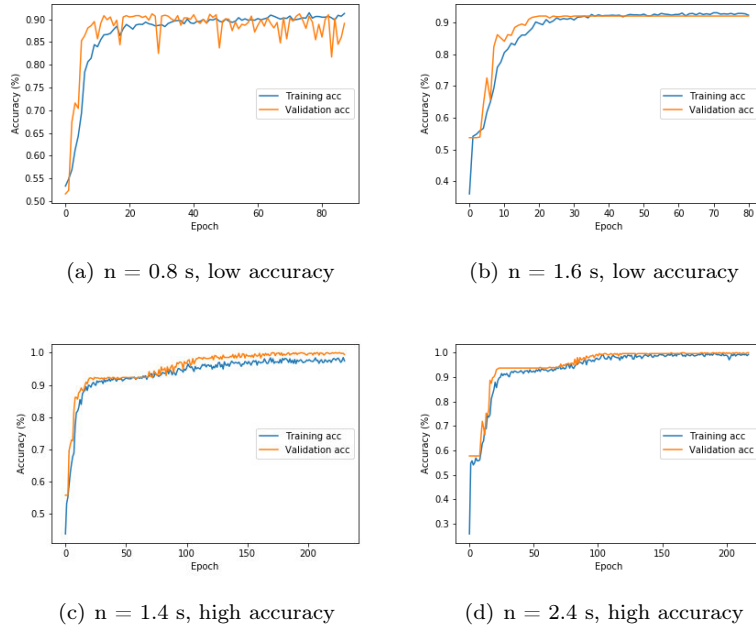


Figure 3.5: The training accuracy and validation accuracy results for a sampling rate of 25 Hz at low accuracy and high accuracy time intervals.

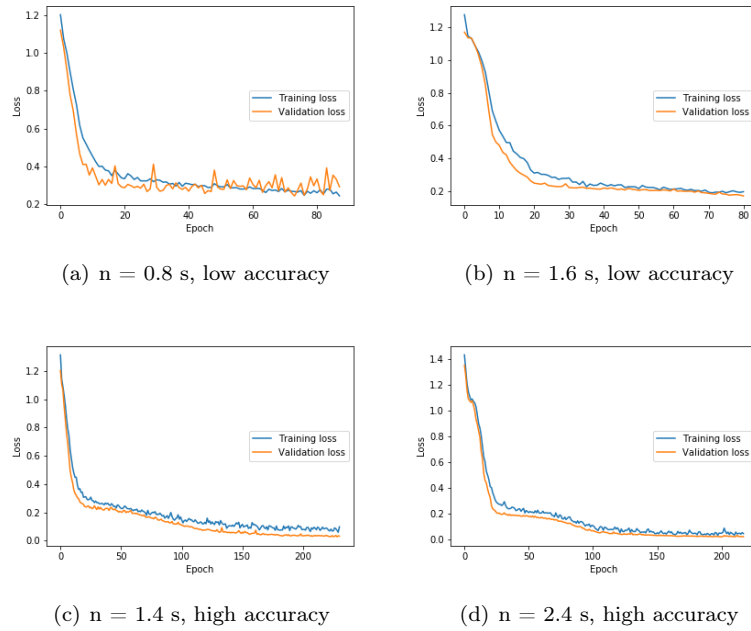


Figure 3.6: The training loss and validation loss results for a sampling rate of 25 Hz at low accuracy and high accuracy time intervals.

From 25 Hz to 50 Hz the overall accuracy increases or remains constant for most of the time

intervals as is depicted in Figure 3.7. For time intervals greater than 0.6 s accuracies of 98.91% and above are reached.

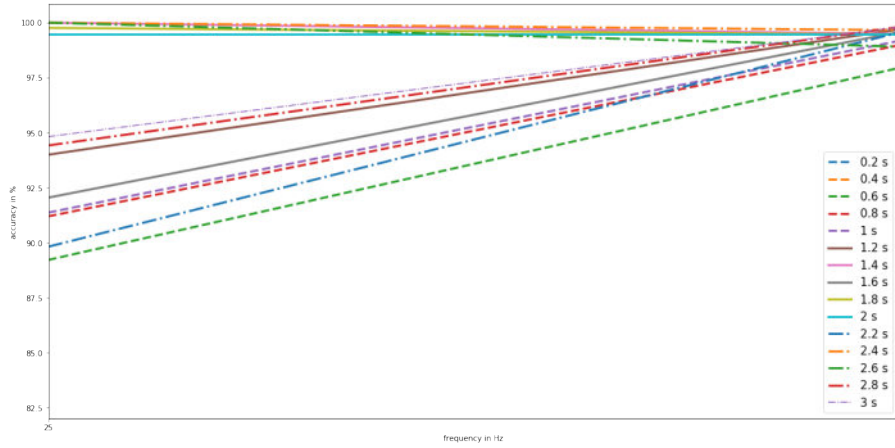
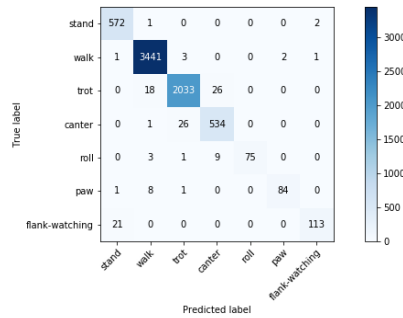


Figure 3.7: Performance of convnet with increasing sampling rate from 25 Hz to 50 Hz.

In the 50 Hz dataset, all behaviours are present. The classes 'roll', 'paw' and 'flank watching' perform the least, while the classes stand, walk, trot and canter for most of the time intervals reach accuracies of 100%. In Figure 3.8 the confusion matrix of a time interval of 0.6 s is presented. The classes 'roll' (1.3%), 'paw' (1.3%) and 'flank watching' (2%) are least present in the merged dataset of 50 Hz. If the true label of an instance is flank watching, it is partially predicted as stand at this time interval. This is as expected, since the data of flank watching appears as a more or less constant signal such as the signal from the class stand. The class roll gets in some instances the label canter and the paw movement is mixed in some instances with walk.



(a) $n = 0.6$ s

Figure 3.8: Confusion matrix for training and test set at a sampling rate of 50 Hz for a time interval of 0.6 s.

When increasing the sampling rate to 100 Hz, the accuracies increase for most of the time intervals as is depicted in Figure 3.9

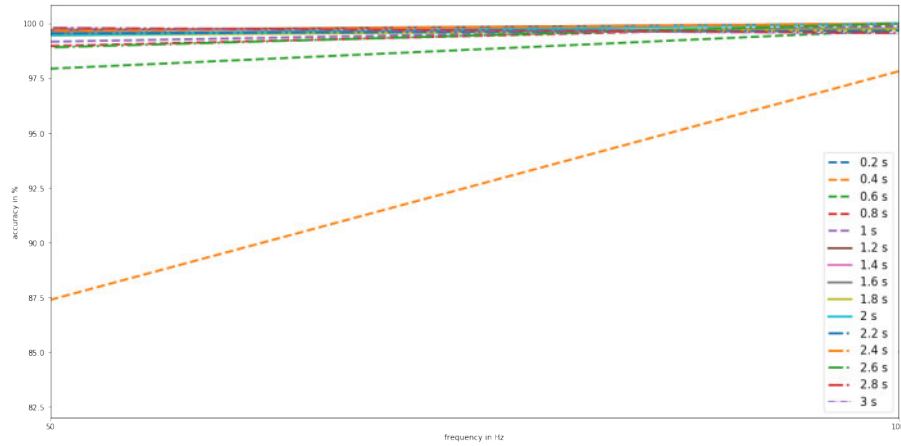


Figure 3.9: Performance of convnet with increasing sampling rate from 50 Hz to 100 Hz.

In the 100 Hz dataset, four classes are present i.e. 'stand', 'walk', 'trot' and 'canter'. The overall accuracy is greater than 99.53 % at time intervals above 0.4 s. The classes stand, walk, trot and canter get classified with an accuracy of 100% in 14 out of 15, 12 out of 15, 7 out of 15 and 10 out of 15 cases, respectively.

If the sampling rate increases from 100 Hz to 200 Hz, the overall validation accuracy decreases in 11 of the 15 cases on average with 0.47%.

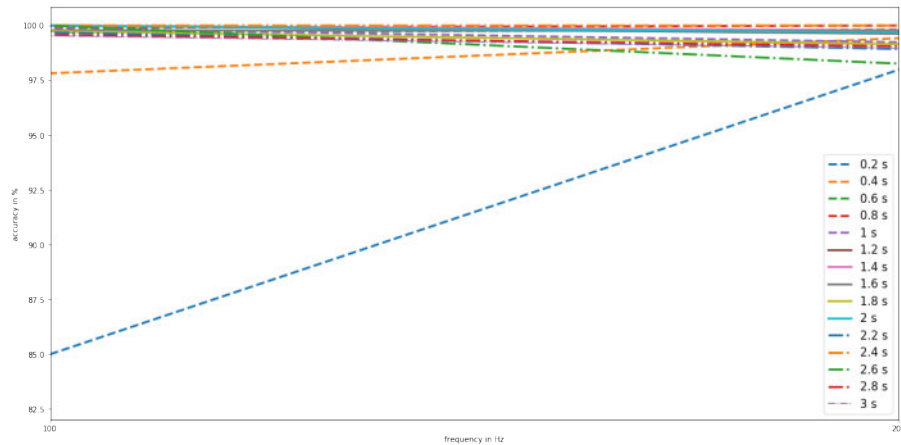


Figure 3.10: Performance of convnet with increasing sampling rate from 100 Hz to 200 Hz.

In the 200 Hz dataset, the movement 'roll' is present with a share of 2.1% in contrast to the 100 Hz dataset which does not contain this movement. The least performing classes are 'roll' and 'canter'. When taking a closer look at the confusion matrices, as partially depicted in Figure 3.11 the class 'canter' gets in some cases wrongly predicted as the class 'trot'. If the class roll gets wrongly predicted, then in most cases it is as 'canter' and 'trot'.

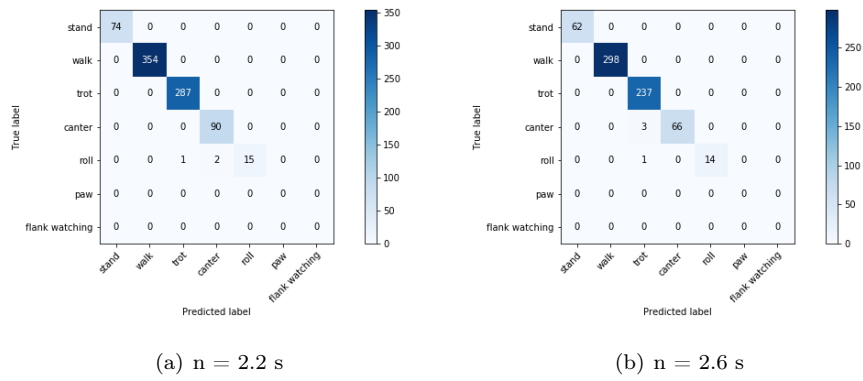


Figure 3.11: Confusion matrix for training and test set at a sampling rate of 200 Hz for different time intervals.

The validation accuracy decreases in 11 out of 15 cases when increasing the sampling rate from 200 Hz to 1600 Hz as is depicted in Figure 3.12.

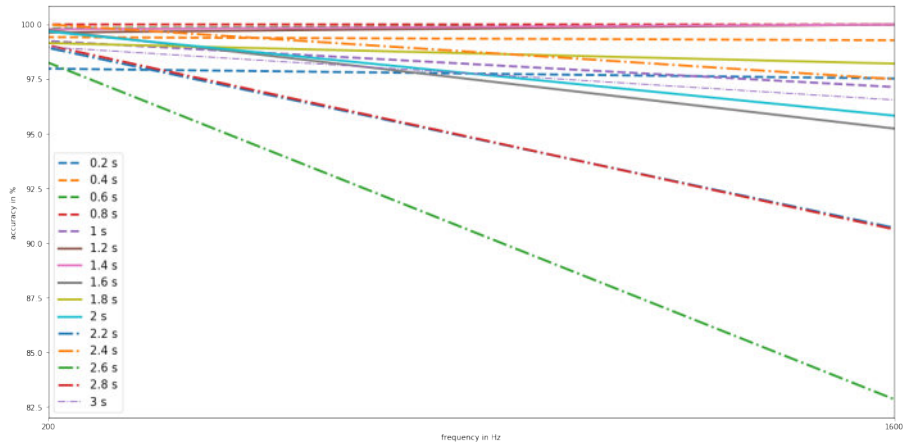


Figure 3.12: Performance of convnet with increasing sampling rate from 200 Hz to 1600 Hz.

The dataset of 1600 Hz contains four behaviours i.e., stand, walk, trot and canter of one horse. The classification accuracy of 'canter' is the lowest performing and is misclassified as trot as one can see from the confusion matrices depicted in Figure 3.13.

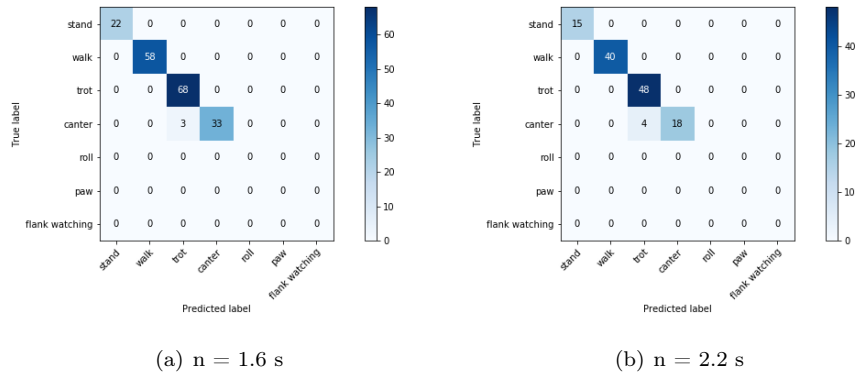


Figure 3.13: Confusion matrix for training and test set at a sampling rate of 1600 Hz for different time intervals.

The mean performance of the convolutional neural network is depicted in Figure 3.14 as a thick line.

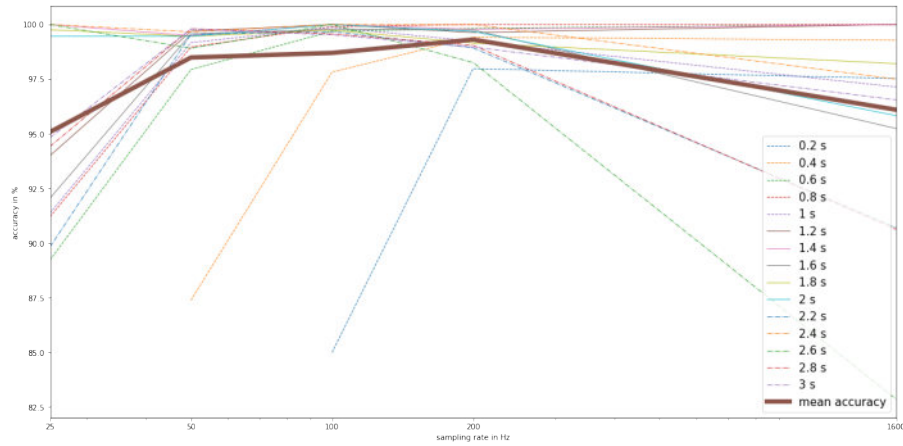


Figure 3.14: Mean overall accuracy of convnet with increasing sampling rate.

The sampling rate of 200 Hz gained the highest on average accuracy of 99.31%. The worst performing sampling rate is 25 Hz with an average accuracy of 95.08%. However, the classification accuracy was still over 82.86% for all sampling rates and all time intervals.

Second dataset

In Figure 3.15 the overall validation accuracy as function of the sampling rate is depicted for time intervals ranging from 0.6 s to 3 s. The thick line represents the mean overall accuracy.

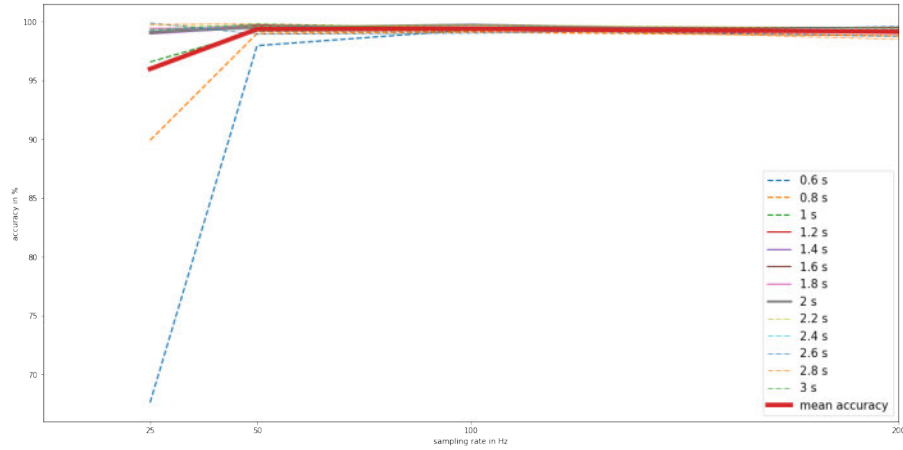


Figure 3.15: Performance of convnet with increasing sampling rate.

The sampling rate of 100 Hz gained the highest on average accuracy of 99.37%. The worst performing sampling rate is 25 Hz with a mean on average accuracy of 95.96%. The lowest measured classification accuracy is 67.59%.

At 25 Hz, the worst performing classes at low time intervals are 'canter', 'roll', 'paw' and 'flank watching'. At higher time intervals ($n \geq 1.4$ s) the classification of flank watching stays unstable with validation accuracies swinging between 33.33% and 85.71%. When taking a closer look at the confusion matrices as depicted in Figure 3.16 it can be seen that 'flank watching' gets classified as 'stand'.

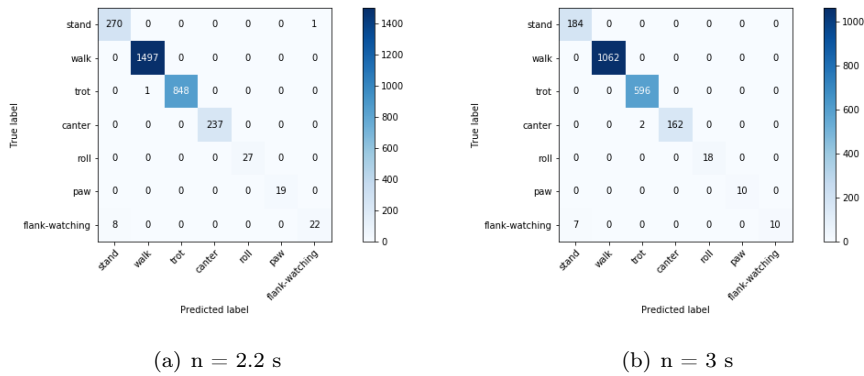


Figure 3.16: Confusion matrix for training and test set at a sampling rate of 25 Hz for different time intervals.

When the sampling rate is increased to 50 Hz also the overall mean validation accuracy increases from 95.96% to 99.34%. At this sampling rate, the gait classes perform well at any time interval. Also the other behaviours 'roll', 'paw' and 'flank-watching' reach high accuracies in most time intervals with a few exceptions.

Increasing the sampling rate from 50 Hz to 100 Hz means a small increase in accuracy from 99.34% to 99.37%. Again, the gait classes are validated with high accuracies and the other behaviours reach high accuracies with a few exceptions.

Increasing the sampling rate from 100 Hz to 200 Hz leads to a minor decrease in the mean validation accuracy from 99.37% to 99.13%. Again, the gait classes are performing well for any time interval, the other behaviours reach also high accuracies for most time intervals with a few exceptions.

Third dataset

In Figure 3.17 the overall validation accuracy as function of the sampling rate is depicted for time intervals ranging from 0.6 s to 3 s. The thick line represents the mean overall accuracy.

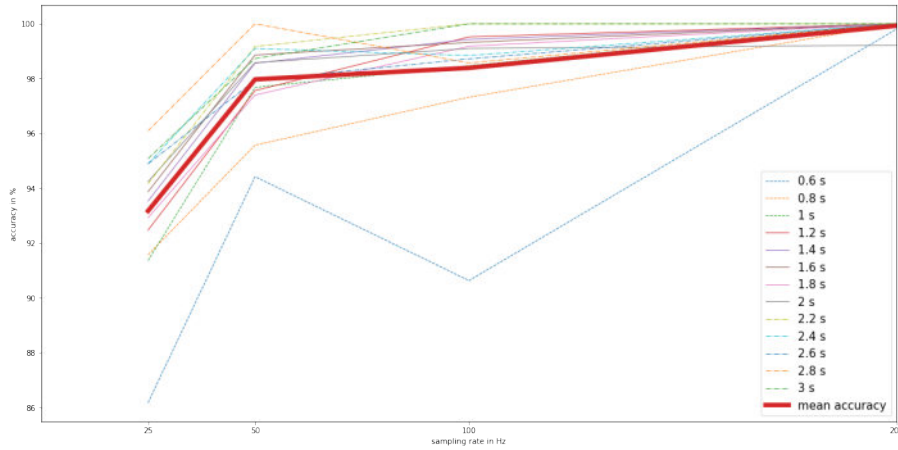


Figure 3.17: Performance of convnet with increasing sampling rate.

The sampling rate of 200 Hz gained the highest on average accuracy of 99.93% .The worst performing sampling rate is 25 Hz with an average accuracy of 93.17%. However, the classification accuracy was still over 86.17% for all sampling rates and all time intervals.

When taking a closer look at the behaviour class accuracies at the sampling rate of 25 Hz it becomes clear that the canter gait performs the worst. In all time intervals, canter did not get classified right on one occasion and thus has an accuracy of 0%. If a closer look is taken at the confusion matrices as some of them depicted in Figure 3.18, it can be seen that the class canter gets classified in any case as trot.

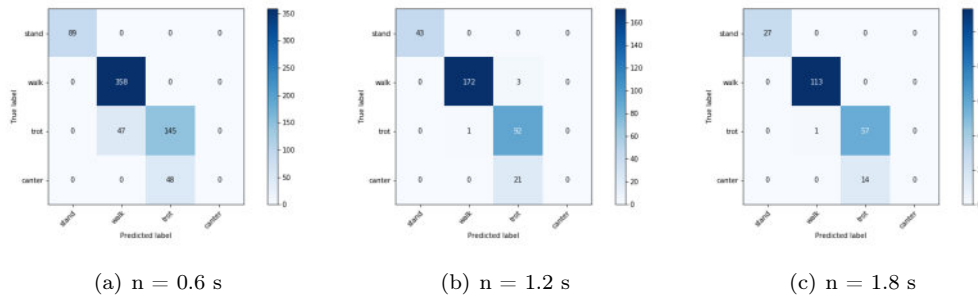


Figure 3.18: Confusion matrix for training and test set at a sampling rate of 25 Hz for different time intervals.

Since this misclassification occurred throughout the whole dataset, another look at the original trot and canter signal was taken from horse 1 (one of the horses on which the model was trained) and at the original canter signal from horse 3 (the horse on which the model was validated) since it is possible that something went wrong during the preprocessing. Segments of those signals are depicted in Figure 3.19.

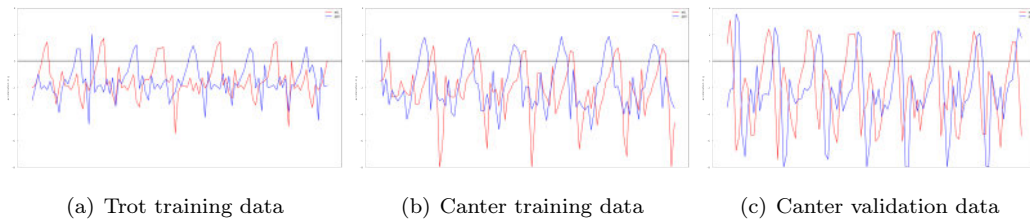


Figure 3.19: Trot training data, canter training data and canter validation data.

After inspecting these segments visually, it is clear that the classification done during the preprocessing phase is correct since the shape of the signal is resembling this of a canter signal as discussed in Section 2.3. The model trained itself on the wrong features for this sampling rate.

When the sampling rate is increased to 50 Hz, the mean overall validation accuracy is 97.97%. Stand and walk get on any instance classified right. The class trot and canter reach overall validation accuracies between 87.95% to 100% and between 76.32% and 100%, respectively.

At sampling rate of 100 Hz, the overall mean classification accuracy gets at 98.39%. Again, stand gets classified right on every occasion. On most instances, 11 out of 13, the class walk and canter achieve an accuracy of 100%. The class trot has an accuracy between 81.36% and 100%.

At 200 Hz the overall classification accuracy is 99.93%. In 11 of the 13 time intervals, all data is classified 100% correct by the algorithm.

Comparison of performance of CNN for the three datasets

In Figure 3.20 the mean performance of the CNN with increasing sampling rate is depicted on for the three datasets for a time interval from 0.6 s to 3 s. The scale used for the sampling rate is logarithmic. The number between the brackets indicates the number of behaviours that are taken into account in the training and validation of the CNN.

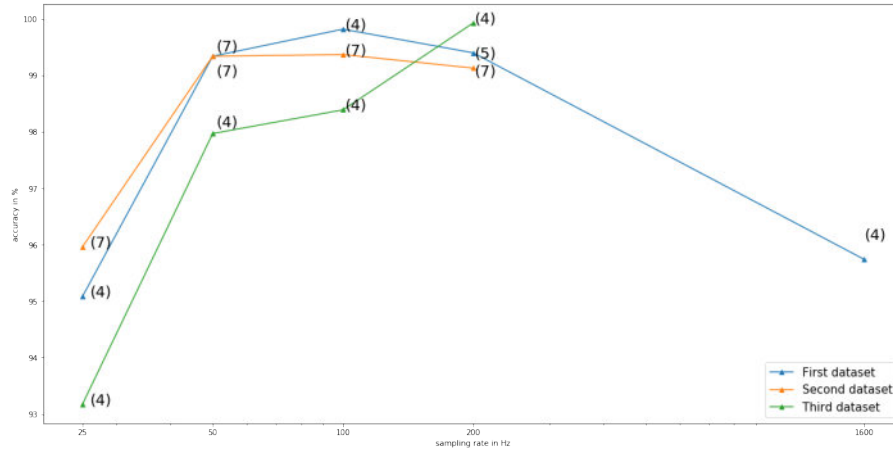


Figure 3.20: Performance of convnet with increasing sampling rate presented on a logarithmic scale for three datasets.

For all datasets the accuracy increases when the sampling rate is increased from 25 Hz to 100 Hz. However, the increase in performance from 50 Hz to 100 Hz is much smaller. From 100 Hz to 200 Hz, the accuracy for two out of three datasets decreases. For the third dataset an increase in accuracy can be noticed.

As can be concluded from this graph, for a sampling rate of 25 Hz, the CNN performs the best when all behaviours and all horses are taken into account. At a sampling rate of 100 Hz, the dataset with a lower number of behaviours and horses outperforms the dataset with seven behaviours and six horses. The CNN validated on the data of the lame horse performs the least in the range of 25 Hz to 100 Hz. At a sampling rate of 200 Hz, the CNN performs best when the training and validation data are split up by hand. The model trained on all behaviours performs the least.

3.2 Effects of the time interval of the accelerometer data

In this Section the influence of the length of the time interval on the model accuracy is investigated since a lower time interval and thus a lower number of samples can reduce the computational efforts of the convolutional neural network. It is important to take the mean duration of each behaviour in consideration since this could affect the performance of the model. From the videofiles combined with the accelerometer data, the mean duration of each gait was calculated by taking 10 samples according to their description in Table 2.2.

Subject number	Walk [s]	Trot [s]	Canter [s]	Roll [s]	Paw [s]	Flank watching [s]
1	1.2731	0.7736	0.5783			
2	1.270	0.8346	0.643			4.866
3	1.2430	0.7998	0.5362			
4	1.2295	0.7218	0.604	2.206	0.5369	
5	1.179	0.8095	0.592			
6	1.183	0.669	0.5405			
Mean	1.2296	0.76805	0.5823	2.206	0.5369	4.866

Table 3.7: Mean duration of behaviour for the participating horses.

First dataset

In Figure 3.21 the performance of the convnet as a function of the time interval for different sampling rates is depicted. The mean duration of the behaviours are annotated with black striped lines except for the flank-watching movement since the mean duration of this behaviour lies outside the investigated time intervals at 4.866 s.

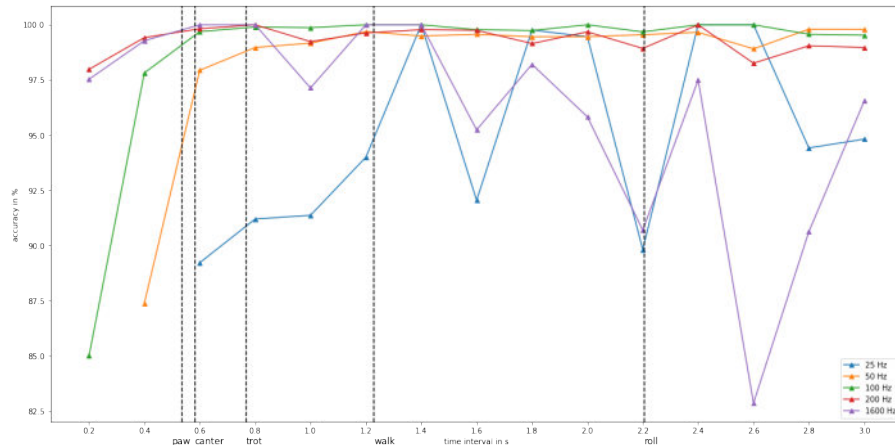


Figure 3.21: Performance of convnet with increasing time interval.

From the figure it can be seen that the performance of the model is lower with decreasing time intervals in particular for sampling rates between 25 Hz and 100 Hz. If we take a closer look at the confusion matrices of these time intervals and these sampling rates as depicted in Figure 3.22 it becomes clear that at a lower time interval, the gaits trot and canter are the worst performing classes but when the time interval is increased, and thus the number of samples, the classification of trot and canter improves. For the dataset of 50 Hz also other movements like roll, paw and flank watching are present. Their classification scores are improving with increasing the time interval from 0.4 s to 0.6 s. The sampling rates of 200 Hz and 1600 Hz reach already accuracies of 97.53% at the time interval of 0.2 s. For sampling rates between 50 Hz and 200 Hz the accuracies

fluctuate between 97.94% and 100.00% after time intervals greater than 0.6 s while the accuracies for sampling rates of 25 Hz and 1600 Hz fluctuate between 82.86% and 100% so the model seems to be less stable for these sampling rates. The maximum number of epochs was not adjusted for the sampling rate of 1600 Hz and since the input vector has a larger dimension it could be possible that longer training was needed to converge to an optimum and reach higher accuracies.

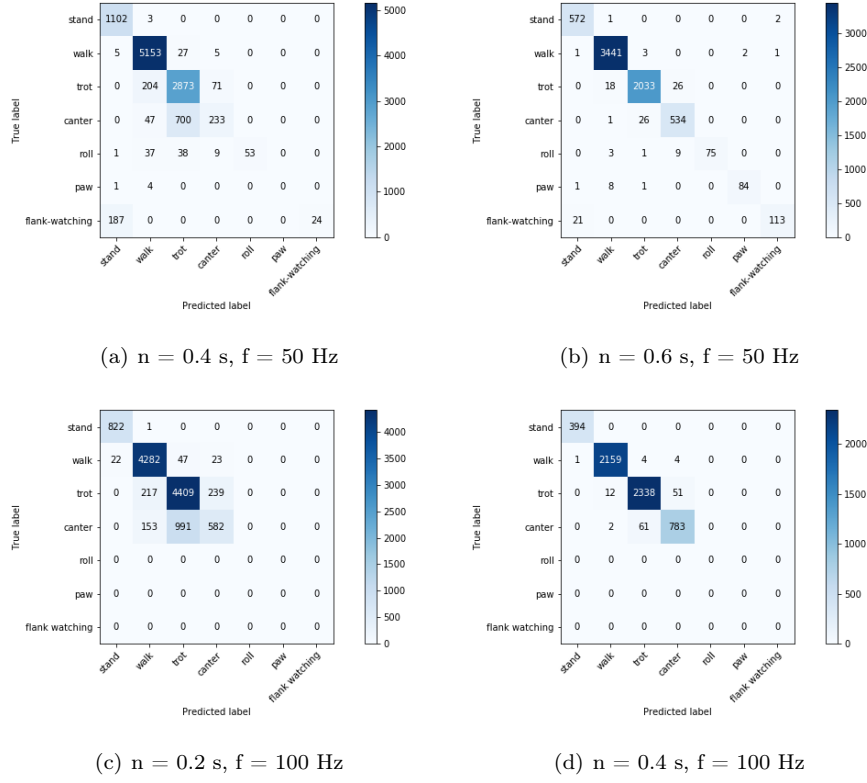


Figure 3.22: Confusion matrix for training and test set at a sampling rate of 50 Hz for time intervals of 0.4 s and 0.6 s and 100 Hz for time intervals of 0.2 s and 0.4 s.

Second dataset

In Figure 3.23 the performance of the convnet as a function of the time interval for different sampling rates is depicted.

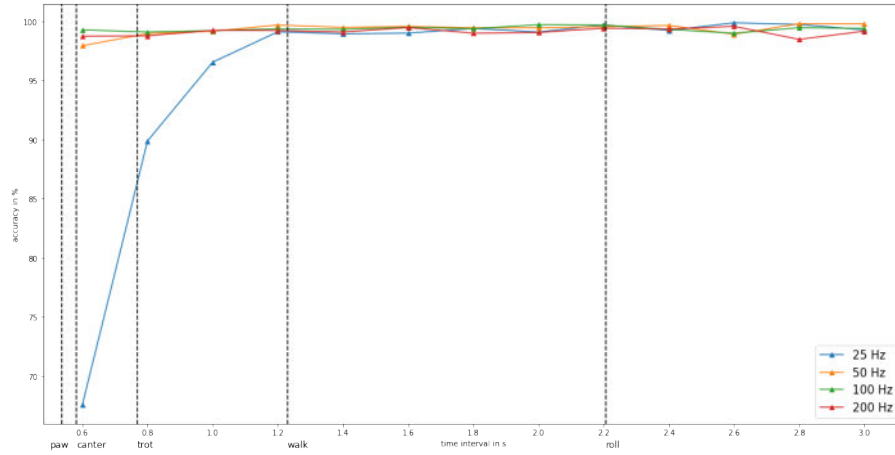


Figure 3.23: Performance of convnet with increasing time interval.

Most sampling rates reach already high accuracies (≥ 97.94) at small time intervals ($n = 0.6$ s). However, the lowest sampling rate of 25 Hz reaches higher accuracies when the time interval is greater than a full walk cycle ($n = 1.2$ s).

Third dataset

In Figure 3.24 the performance of the convnet as a function of the time interval for different sampling rates is depicted.

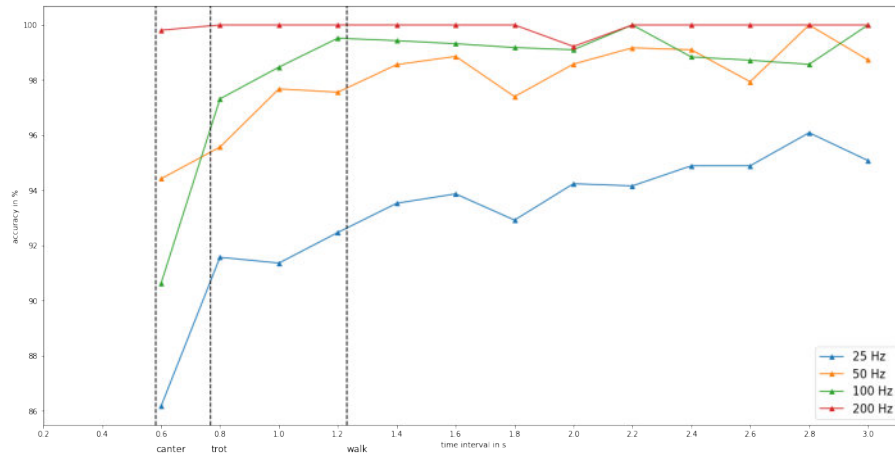


Figure 3.24: Performance of convnet with increasing time interval.

If the time intervals decrease, for sampling rates between 25 Hz and 100 Hz, it can be seen that the overall validation accuracies also decrease. The accuracy at 25 Hz increases from 0.6 s to 3 s but still stays significantly lower than the accuracies reached at sampling rates between 50 Hz and 200 Hz. At sampling rates of 50 Hz and 100 Hz, the accuracy increases between 0.6 s and 1.2

s. After 1.2 s when the time interval is greater than a full walk cycle, the accuracies stay more or less constant. At a sampling rate of 200 Hz, the accuracy stays high and more or less constant at time intervals ranging between 0.6 s and 3 s.

Comparison of performance of CNN for the three datasets

In Figure 3.25 the mean performance of the CNN is plotted for the three datasets for sampling rates between 25 Hz and 200 Hz.

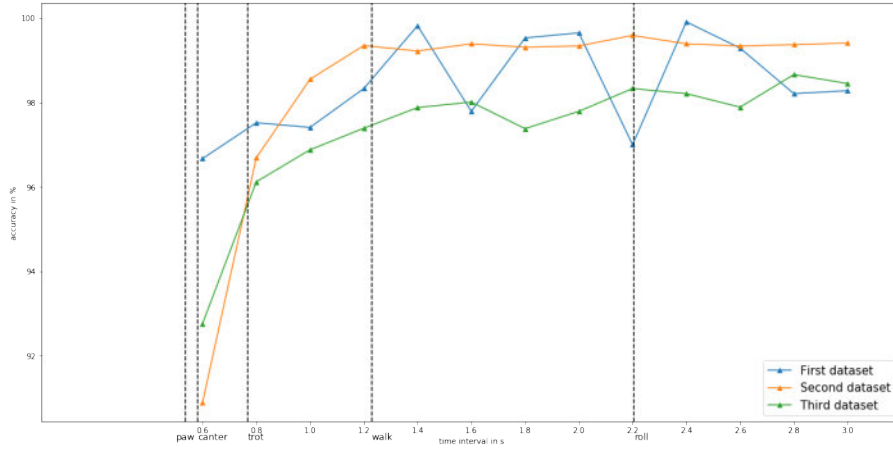


Figure 3.25: Mean performance of convnet with increasing time interval for the three datasets.

If the time interval decreases from 1.2 s (a full walk cycle) to 0.6 s, the overall mean validation accuracy of the three datasets decreases. In all cases, the largest contributor to a lower accuracy is the misclassification of 'canter' and to a lesser extend 'roll', 'paw' and 'flank-watching'. As can be noticed the mean accuracy plot for the third dataset lies lower than those of the first and second dataset due to more spread out accuracies at higher time intervals for sampling rates between 50 Hz and 200 Hz as is shown in Figure 3.26. The fluctuations in the mean accuracy of the first dataset are due to the fluctuations of the validation accuracy when the sampling rate is 25 Hz.

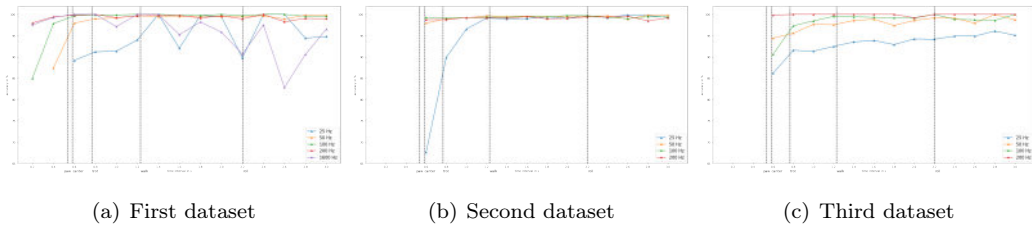


Figure 3.26: Performance of convnet with increasing time interval for three datasets for different sampling rates.

3.2.1 Model complexity

Every time interval corresponds to a number of samples depending on the sampling rate. The size of the input vector and thus the number of samples presented to the model influences the number of trainable parameters and thus the computational cost. The number of trainable parameters for each layer can be found in Table 2.6. The total number of trainable parameters N_{par} is calculated by summing all the trainable parameters for each layer [36]:

$$N_{par} = 11099 + \left(16 \frac{\left(\frac{n}{2} - 4\right)}{3} + 1\right) \cdot 50 \quad (3.1)$$

with n the number of samples.

The minimum number of samples for each dataset necessary to obtain an accuracy higher than 98% is shown in Figure 3.27 for sampling rates between 25 Hz and 100 Hz. The sampling rate of 200 Hz is not taken into account, since the minimum accuracy is already reached at the start of the measured time interval for the three datasets and thus this would give a distorted image. For the third dataset an accuracy above 98% is not reached at 25 Hz.

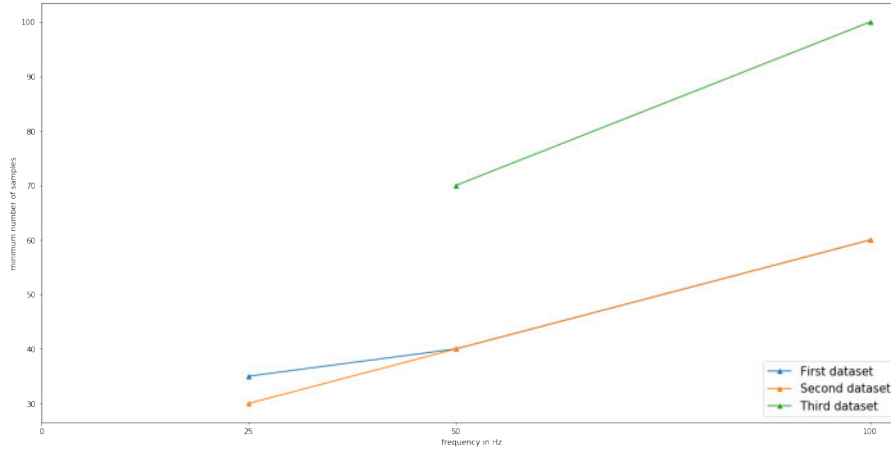


Figure 3.27: Minimum number of samples necessary to reach an overall validation accuracy higher than 98% for the three datasets.

It can be concluded, when increasing the sampling rate, the minimum number of samples necessary to obtain an accuracy higher than 98% also needs to be increased and thus the number of trainable parameters increases since there is a linear relationship between the number of trainable parameters N_{par} and number of samples n . For the third dataset, the minimum number of samples is higher than for the first and second dataset. For the first and second dataset, the minimum number of samples is more or less equal at each sampling rate. The analysis of the experimental results brings out that the first and second dataset and a lower sampling rate reduce the complexity of the function by decreasing the quantity of trainable parameters while retaining high accuracy.

First dataset

The accuracy of the model is depicted in function of the number of samples in Figure 3.28. It becomes clear that the accuracy increases with increasing number of samples for sampling rates ranging from 50 Hz to 200 Hz. At the sampling rates of 25 Hz and 1600 Hz, the accuracy is fluctuating.

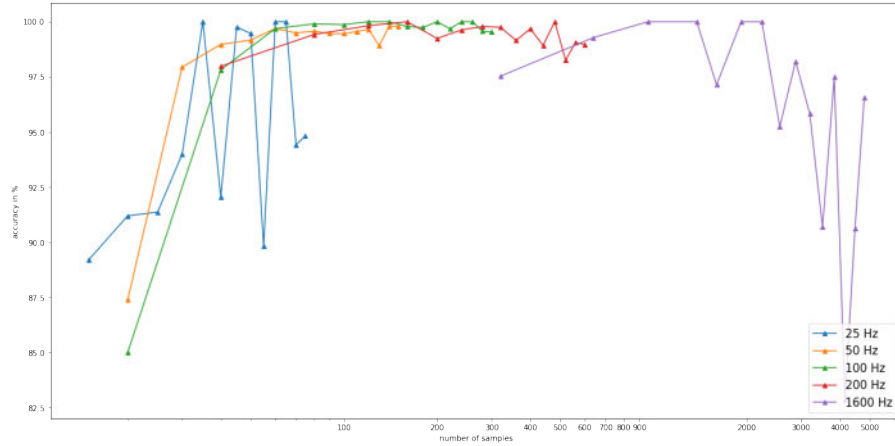


Figure 3.28: Performance of convnet with increasing number of samples for sampling rates from 25 Hz to 1600 Hz presented on a logarithmic scale.

Second dataset

In Figure 3.29 the overall model validation accuracies of the second dataset as function of the number of samples are plotted.

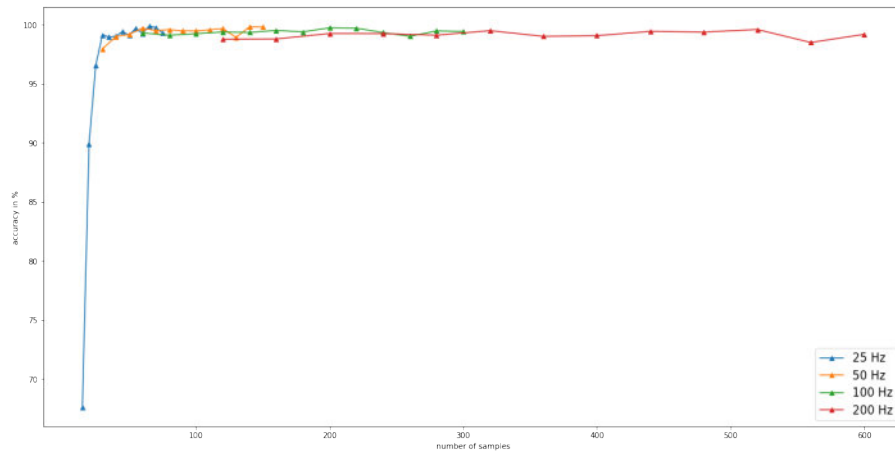


Figure 3.29: Performance of convnet with increasing number of samples for sampling rates from 25 Hz to 200 Hz.

The accuracy at a sampling rates between 50 Hz and 200 Hz are high and more or less constant

for each number of samples on which the model is trained. For a sampling rate of 25 Hz, the accuracy increases as function of the number of samples.

Third dataset

In Figure 3.30 the overall model validation accuracies are plotted in function of the number of samples.

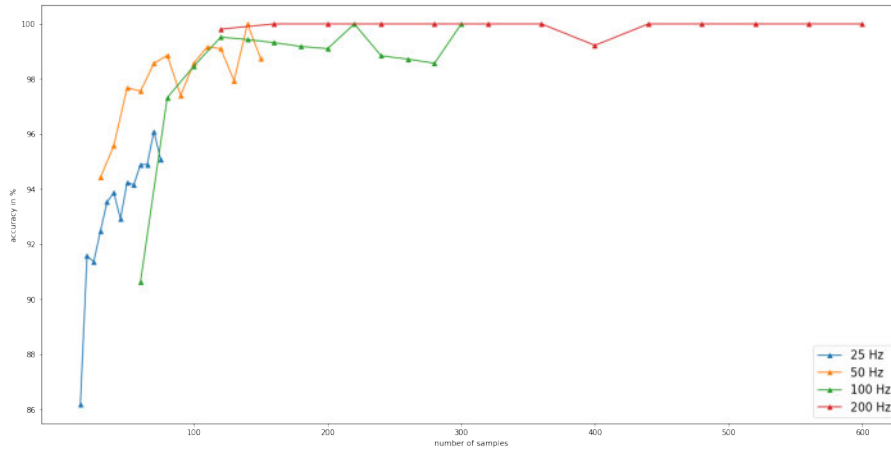


Figure 3.30: Performance of convnet with increasing number of samples for sampling rates from 25 Hz to 200 Hz.

As the number of samples increases, the overall model accuracy increases for sampling rates between 25 Hz and 100 Hz. The accuracy at a sampling rate of 200 Hz is high and more or less constant for each number of samples on which the model is trained.

3.3 Combination of time interval and sampling rate

Figure 3.31 presents the accuracy surface plots for the three datasets as function of time interval and sampling rate generated through the obtained datapoints indicated as blue dots. The surface plot is obtained by fitting a polynomial of degree two through the datapoints. The low predicted accuracies are indicated with the colour blue and the high ones with the colour red. For the first dataset, the 1600 Hz datapoints are left out of the interpolation. The combinations that are the least performing for the three datasets are observed in the region where both sampling rate and time interval are low.

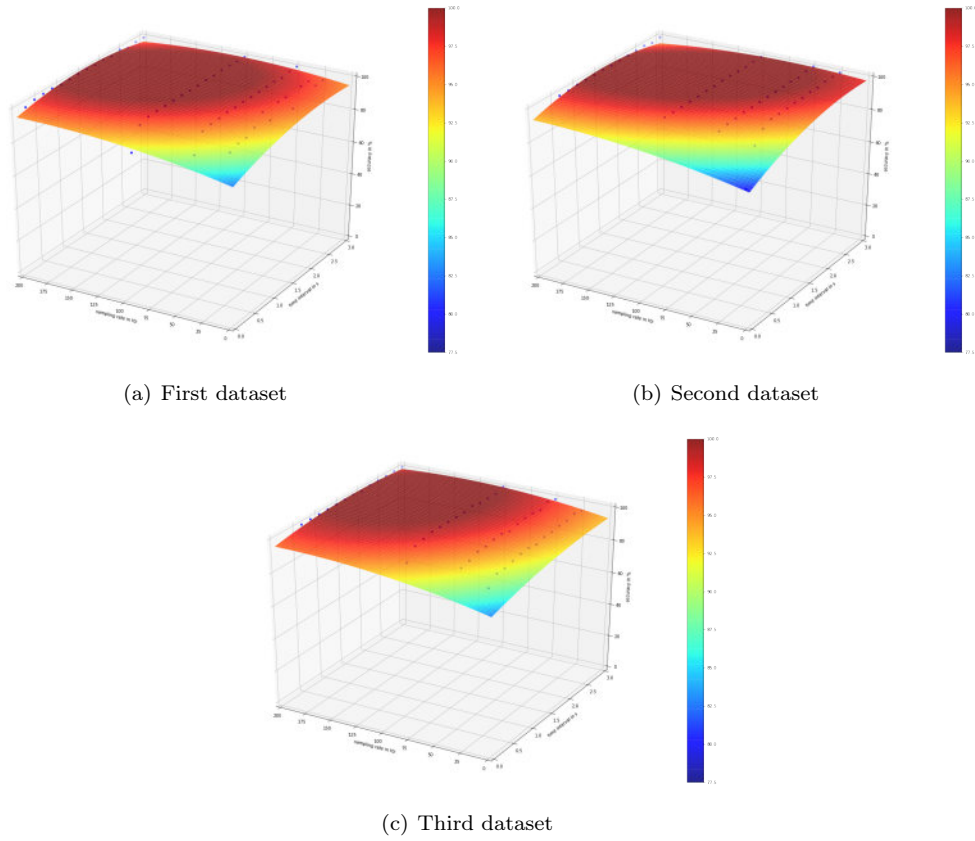


Figure 3.31: Accuracy surface plot as function of the sampling rate and the length of the time interval for three datasets.

As can be seen from the contourplots shown in Figure 3.32 a 100% is reached in the dark red region. As indicated with yellow cross markers, the combinations that gain an accuracy of 100% at the lowest sampling rate and the shortest time interval are for the first dataset observed in the region where the value of the sampling rate ranges between 64 Hz at a time interval of 2.05 s and 170 Hz at a time interval of 0.85 s, for the second dataset in the region where the value of the sampling rate ranges between 36 Hz at a time interval of 2.4 s and 170 Hz at a time interval of 1 s, for the third dataset in the region where the value of the sampling rate ranges between 90 Hz at a time interval of 2.3 s and 170 Hz at a time interval of 1 s.

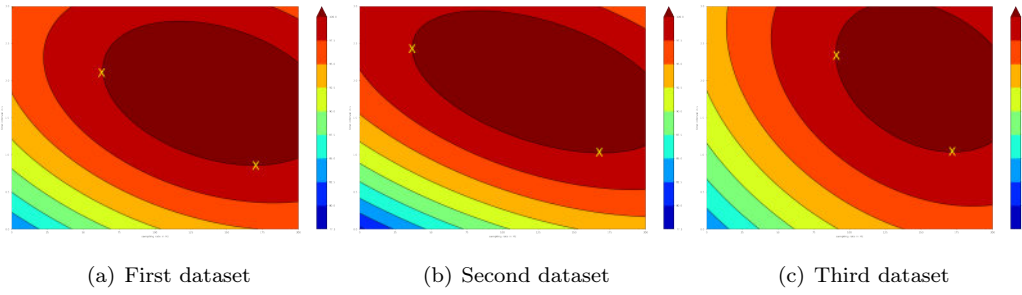


Figure 3.32: Accuracy contour plot as function of the sampling rate and the length of the time interval for three datasets.

Depending on the wanted level of accuracy, the behaviour detected from a seen or unseen horse and the wanted number of behaviours to be recognized different optimal time interval and sampling rates can be recommended. To investigate this the contour plots for three levels of accuracy for the different datasets are placed on top of each other as is depicted in Figure

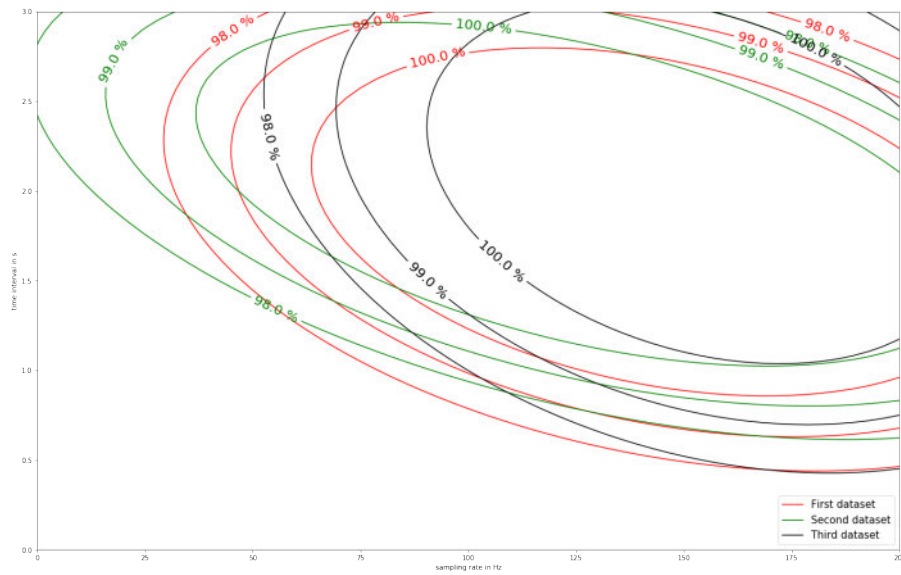


Figure 3.33: Contour plots for three datasets with accuracy levels of 98%, 99% and 100%.

If the accelerometers would be placed on an unseen horse by a model trained by dataset three, then the recommended time interval and sampling rate for different levels of accuracy are depicted in Table 3.8. If the accelerometers would be placed on a seen horse or the movement of an unseen horse would resemble the data in the training set, even lower sampling rates and smaller time intervals could be reached and more behaviours can be predicted as is depicted in Table 3.8. For the recommendations only sampling rates are taken into account bigger than or equal to those used in the experiments. Also, the lowest sampling rate is selected before reducing the length of time interval and thus model complexity.

	Seen (seven behaviours)			Unseen (four behaviours)		
	98%	99%	100%	98%	99%	100%
f (Hz)	25	25	36.5	52.5	69	90
n (s)	1.8	2.1	2.4	2.5	2.4	2.3

Table 3.8: Time interval and sampling rate recommendations for a seen horse and an unseen horse for three levels of accuracies.

3.4 Effects of the underground

Model accuracy for the class walk is studied for four different surfaces: dry sand mixed with polyflakes, wet sand mixed with polyflakes, meadow and hard underground. To draw straightforward conclusions, at first only one horse is taken in to consideration. Secondly, four other movement classes are taken into account, to see if the model still predicts the walk classes right. Thirdly, the movements of horse 1 are added to the dataset which also performs movements on wet and dry underground. At last, the movements of all subjects are added. The movement walk is labelled for all datasets according to the underground, with a separate character, 'H', 'W', 'D' and 'F' for hard underground, wet underground, dry underground and field, respectively.

The datasets from horse 2 with numbers 8, 9, 11, 12 and 13 as displayed in Table 2.5 are selected for the first experiment. Horse 2 performed the gait walk on all four types of underground. The total time of movement data and the number of samples for each underground can be found in Table 3.9. The configured sample rate is for three out of the four undergrounds 50 Hz. For the dry underground, a measurement at 100 Hz is resampled to 50 Hz.

	Time measured [s]	Original number of samples
Hard underground	264	13201
Field	306	15298
Wet underground	810	40499
Dry underground	875	87539

Table 3.9: Total time of movement data and number of samples for each underground.

The normalized confusion matrices for five different time intervals are depicted in Figure 3.34

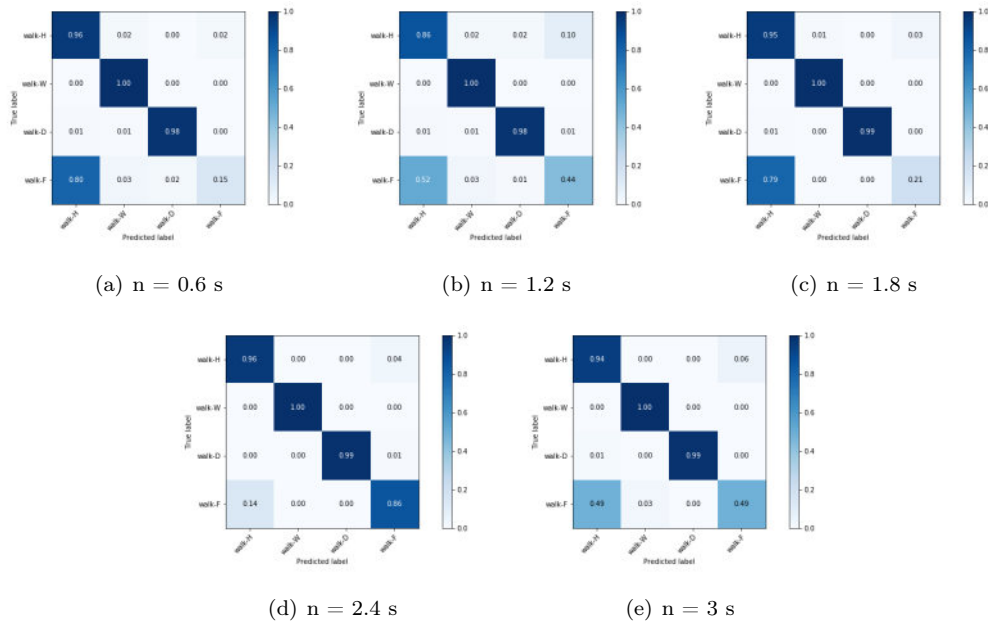


Figure 3.34: Normalized confusion matrix for training and test set at a sampling rate of 50 Hz for different time intervals and four types of underground (H= hard, W = wet, F= field and D = dry).

As can be concluded from the results presented in the normalized confusion matrices, the data gathered from different undergrounds is significantly different so that the model could conclude from which underground the data was gathered for three out of four undergrounds. Otherwise, the walk data from every class with a different underground would have to be more mixed up. Only the data gathered from the field gets often classified as data gathered from a hard underground and visa versa. The class walk on a wet underground and on a dry underground get classified with an accuracy above 98% for every time interval. The class walk on a hard underground reaches accuracies higher than 86%. The class walk on a field swings between 15% and 86% classification accuracy.

Now extra movements of horse 2 are taken into account, i.e. 'stand', 'trot', 'canter' and 'flank-watching'. The first three movements were already present in the used dataset. To add the last movement an extra dataset needs to be included, dataset number 10 as depicted in Table 2.5 which is measured on a dry underground. In Table 3.10 the number of samples and the measured time is depicted.

	Time measured [s]	Original number of samples
Hard underground	264	13201
Field	306	15298
Wet underground	810	40499
Dry underground	1235	105535

Table 3.10: Total time of movement data and number of samples for each underground including extra behaviours.

The normalized confusion matrices are shown in Figure [3.35](#).

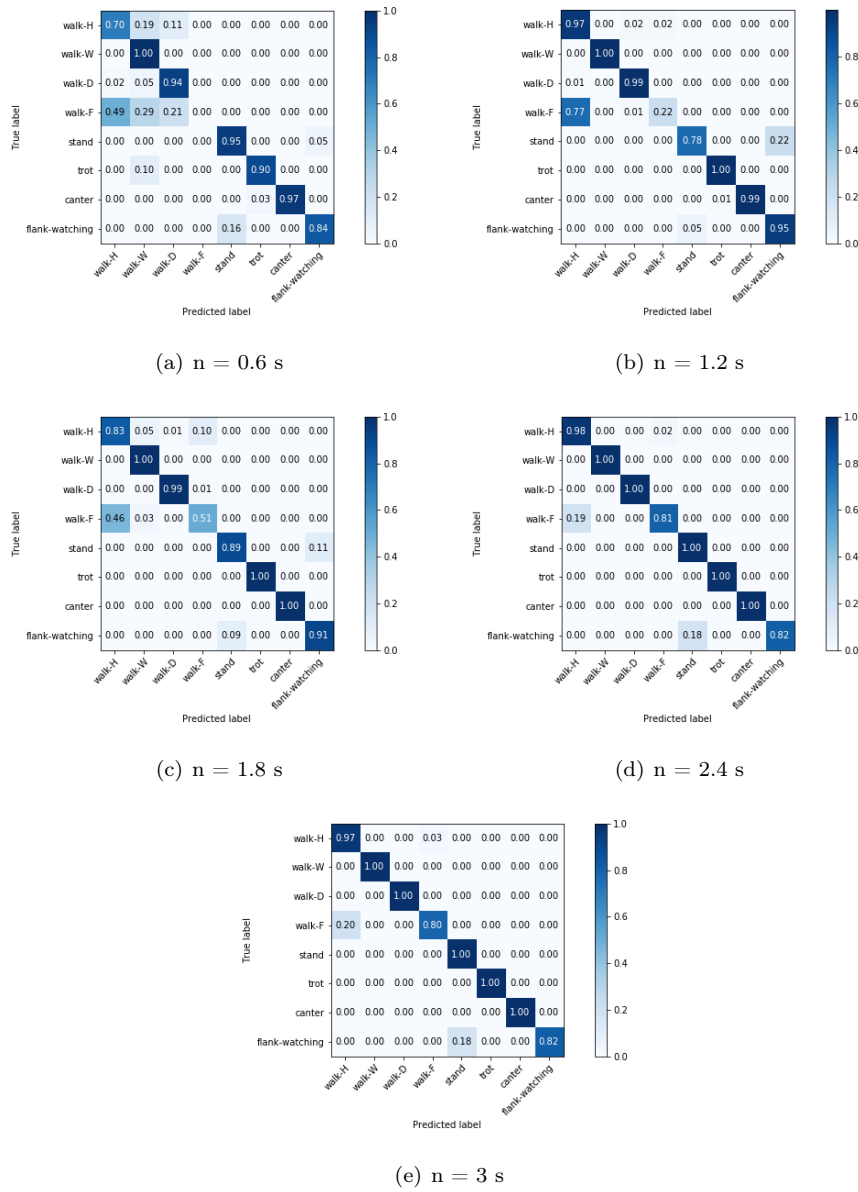


Figure 3.35: Normalized confusion matrix for training and test set at a sampling rate of 50 Hz for different time intervals and four types of underground (H= hard, W = wet, F= field and D = dry) including four other behaviours.

As can be seen from the normalized confusion matrices, the class walk on a wet underground and on a dry underground again reaches high accuracies above 94% for every time interval. The class walk on a hard underground reaches accuracies higher than 70%. The class walk on a field swings between 0% and 81% classification accuracy. But walk on every type of underground gets in any case classified as walk for any time interval. So the type of underground does not affect the overall classification accuracy of the behaviour walk. At lower time intervals, the behaviour trot gets classified as walk on wet underground for 10% of the instances. But as discussed in the previous

Section, small time intervals gain lower accuracies for sampling rates between 25 Hz and 100 Hz. At higher time intervals, none of the other behaviours get classified in one of the 'walk' classes. So adding the extra behaviours does not influence our model accuracy.

Now, extra movement data of horse 1 is added to our dataset. Two datasets where horse 1 performed movements on a wet underground (datasets 3 and 4 as depicted in Table 2.5), measured at a sampling rate of 50 Hz and another dataset where the same horse was longed on a dry underground (dataset 5 as depicted in Table 2.5), measured at a sampling rate of 100 Hz. This dataset is resampled to a sampling rate of 50 Hz. The total number of samples and the time measured can be found in Table 3.11.

	Time measured [s]	Original number of samples
Hard underground	264	13201
Field	306	15298
Wet underground	1972	98607
Dry underground	2251	207184

Table 3.11: Total time of movement data and number of samples for each underground including two subjects.

The normalized confusion matrices are depicted in Figure 3.36.

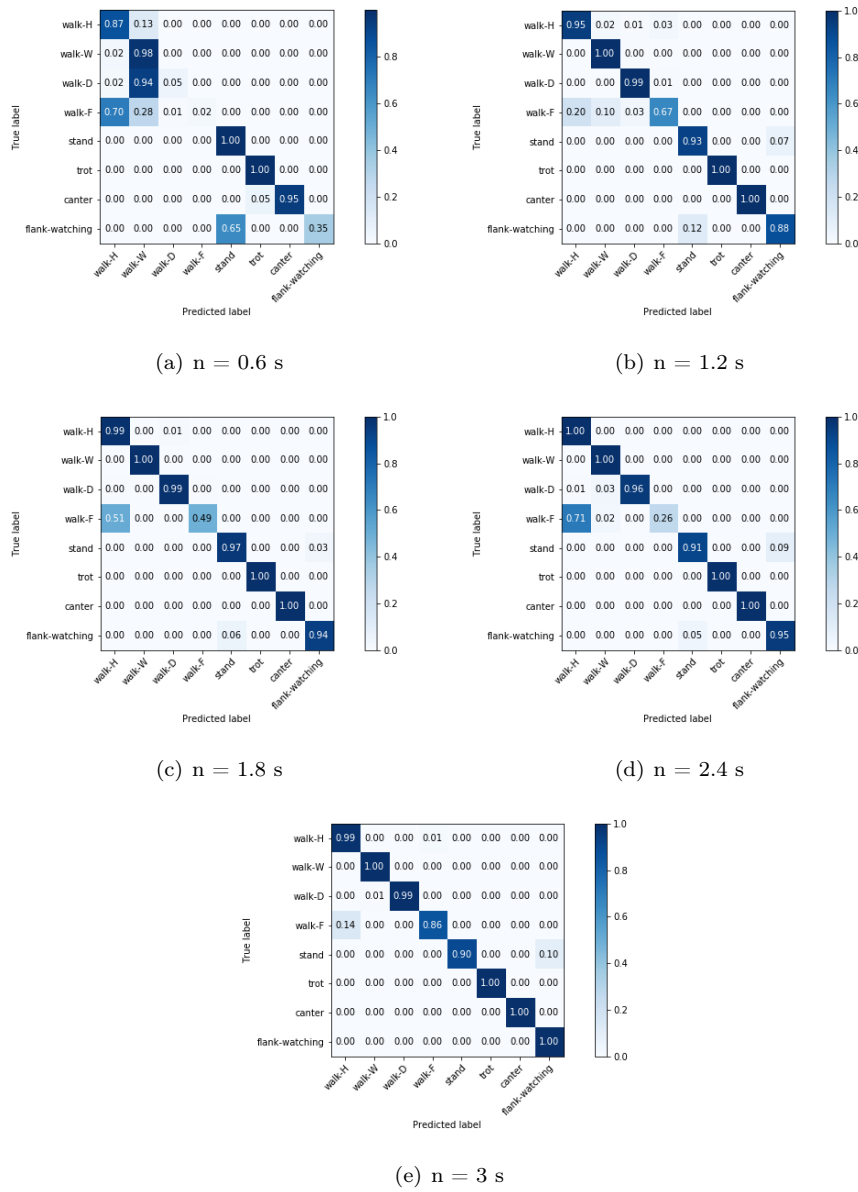


Figure 3.36: Normalized confusion matrix for training and test set at a sampling rate of 50 Hz for different time intervals and four types of underground (H= hard, W = wet, F= field and D = dry) including an extra horse.

As can be seen from the normalized confusion matrices, now the class walk on a wet underground and on a hard underground reaches high accuracies above 87% for every time interval. The class walk on an dry underground and field are misclassified at the smallest time interval of 0.6 s. But when the time interval is increased, the class 'walk-H', 'walk-W' and 'walk-D' reach high accuracies. The class 'walk-H' is even performing better at higher time intervals than when only one horse was present in the dataset. The class 'walk-F' performs the worst at all time intervals. This label mostly gets swapped with the label 'walk-H'. Again walk on any type of underground

gets in any case classified as walk for any time interval. So the type of underground does not affect the overall classification accuracy of the behaviour walk. Also other movements, do not get classified as the behaviour walk, so adding the extra behaviours and the extra horse does not decrease our overall model accuracy for other behaviours and even increases the accuracy for the class 'walk-H'.

Extra datasets (24-27 as depicted in Table 2.5) of horse 4 are added, so that all the behaviours are available in the final dataset. Three out of four datasets are measured on a dry underground. One dataset (26) is measured on a hard underground. The total number of samples and the time measured can be found in Table 3.12.

	Time measured [s]	Original number of samples
Hard underground	446	22281
Field	306	15298
Wet underground	810	40499
Dry underground	1504	118996

Table 3.12: Total time of movement data and number of samples for each underground including extra behaviours.

The normalized confusion matrices are depicted in Figure 3.37.

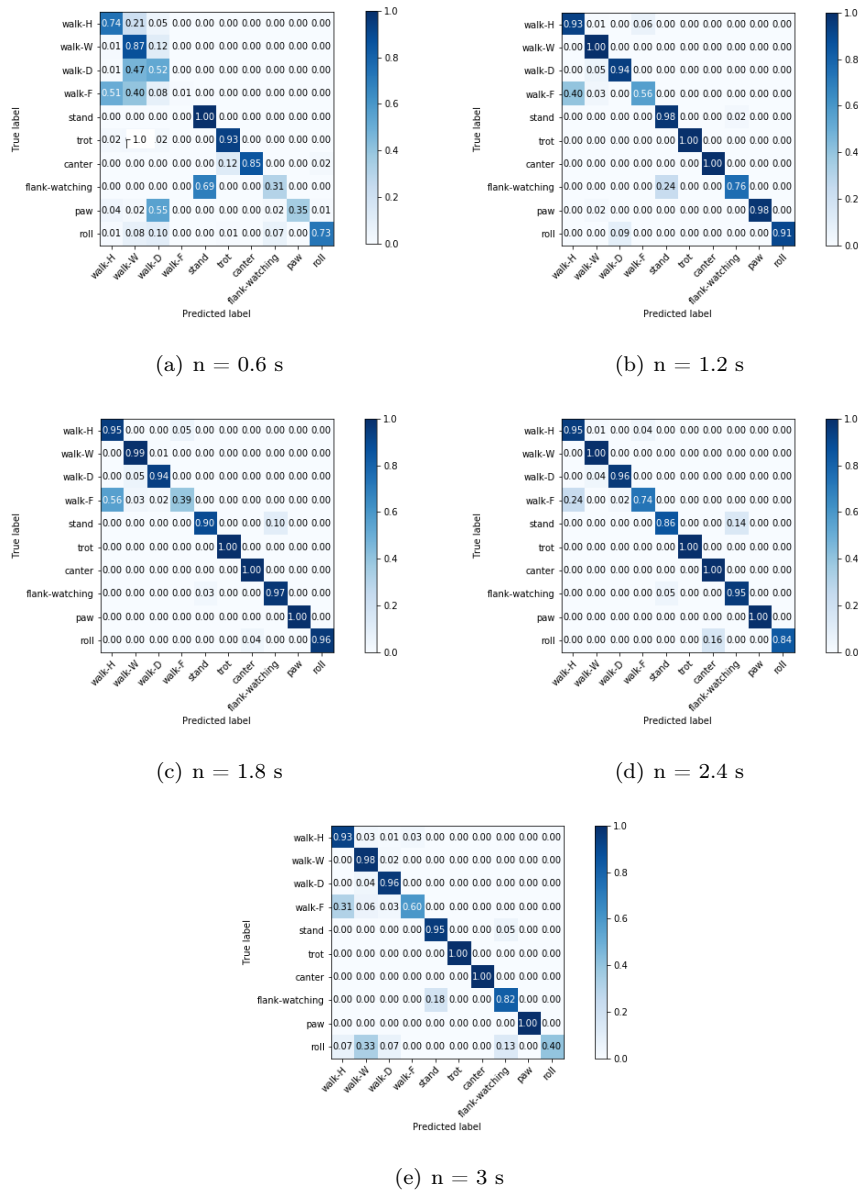


Figure 3.37: Normalized confusion matrix for training and test set at a sampling rate of 50 Hz for different time intervals and four types of underground (H= hard, W = wet, F= field and D = dry) including extra behaviours of horse 4.

As can be concluded from the confusion matrices, at small time intervals, more misclassification is taking place than at higher time intervals. 'Walk-F' is performing the worst with accuracies swinging between 1% and 74%. The other 'walk classes' get classified with high accuracies between 93% and 100% at higher time intervals. All the walk movements get classified as walk, independent of the underground, at any time interval. The other movements that are now included get classified in a few cases as one of the 'walk classes'.

From the obtained results, it can be concluded that the walk gait on any underground gets classified as walk independent from the number of horses and the number of movements present in the training dataset. Data from three of the four undergrounds get classified at the right underground in many cases if the time interval is above 1.2 s. This means that a difference in accelerometer data is measured from different undergrounds and is also recognized by the algorithm. When adding extra movements, those get in a few cases classified in one of the 'walk classes'.

Chapter 4

Discussion

The aim of this study was to automatically classify horse's abnormal behaviours based on machine learning algorithms by using the wearable sensor data. The challenging classification task is successfully tackled using a multilayer convolutional network with two convolutional layers, which are followed by max-pooling layers, two fully connected layers and a softmax layer. Since convolutional neural networks are flexible to extend to more behaviours, first gait data was gathered since gaits are performed by a horse on command in contrast to other behaviours. Afterwards, three extra behaviours were added to the training and validation datasets, which horses are performing during an episode of colic, i.e. rolling, pawing and flank watching. Due to limited data availability of the extra behaviours, validation of the algorithm could be done on seen data for all the behaviours and on unseen data for the gaits.

A good wearable device would have an optimum of both high classification accuracy and long battery lifetime. Battery life time is dependent on the sampling rate and the length of the time interval since a reduction of both parameters would reduce storage load, calculation cost and minimizing both sensing and transmitting energies. Also, decreasing the time interval and the sampling rate within applicable measurement settings (25 Hz - 100 Hz) reduces the classification accuracy. If the sampling rate is decreased from 100 Hz to 25 Hz, the overall mean validation accuracy decreases for the first, second and third dataset from 99.82% to 95.08%, from 99.37% to 95.96% and from 98.39% to 93.17%, respectively. If the time interval is decreased from 1.2 s to 0.6 s, the mean validation accuracy decreases for the first, second and third dataset from 98.33% to 96.67%, 99.35% to 90.89% and 97.39% to 92.76%, respectively.

As wearable technology allows for gait analyses among unseen horses, the use of automated accelerometer-based behaviour event detection methods may be helpful in the real-time evaluation of behaviour patterns on a variety of horses without the necessity to calibrate them when applied to an unseen horse. The algorithm was validated on gait data of an unseen lame horse which is a challenging task because lame horses have asymmetrical gait patterns. Overall validation accuracies of 100% are no exception. A sampling rate of 52.5 Hz and time interval of 2.5 s is

recommended to reach an accuracy of 98 %. Also, the presented model could be used to validate seven behaviours of seen horses or an unseen horse with resembling movements of those in the training set with an accuracy of 98% when the sampling rate is set to 25 Hz and the time interval is 1.8 s.

To test real life circumstances the model is validated on data gathered in a variety of conditions. The results show that the classification accuracy of the walk gait is independent of the type of underground. The classification was consistent across several walk conditions, suggesting that the behaviour event detection algorithms can be used in a variety of settings. Not only was the walk gait classified at every instance as walk, but also three out of four undergrounds reached high classification accuracies specific for their type of underground proving that the model recognized the type of underground the horse was walking on. As wearable technology allows for walk gait analyses among different types of underground, the use of automated accelerometer-based behaviour event detection methods may be helpful in the real-time evaluation of behaviour patterns in a variety of conditions.

Adding the extra behaviours changes the performance of the gait classes due to the presence of similar data features between classes that can trigger false positives. For example, in the first dataset, stand gets classified with an accuracy of 100% at higher time intervals for most sampling rates if only the gaits are present in the validation dataset. If the three extra behaviours are added, the classification accuracy of stand decreases. Stand events get classified as flank-watching and visa versa.

Chapter 5

Conclusion and Future work

In this research we proposed a solution for a horse activity recognition problem that is based on Convolutional Neural Networks with the use of accelerometer time series. It has the benefits of using short recognition intervals of size up to 1.8 s and small sampling rates up to 25 Hz and requiring no feature engineering. Due to a relatively shallow architecture, the proposed algorithm has a small running time and can be efficiently executed on wearable devices in real time.

To evaluate the performance of the considered approach we tested it on three datasets. The obtained results demonstrate that the proposed CNN-based model establishes high accuracies at a lot of time intervals and sampling rates. A reduction in the sampling rate and time interval length did reduce the overall classification accuracy of the model. The experiment has further emphasized an architecture that can be applied not only to different subjects, but can be used in different measurement conditions.

Future work will include capturing and analyzing more behaviours related to horses experiencing an episode of colic like: kicking the abdomen, stretching and attempting to lie down. Also, further investigation needs to be done concerning the eating behaviour since this could give extra information about the well-being of the horse. In addition, activity measurements could be performed to conclude if a horse is agitated or depressed.

In this study, sampling rates up to 25 Hz were used. The predicted contour accuracy plots indicated that high accuracies could be reached at even lower sampling rates up to 12.5 Hz. Those were not investigated yet so no conclusions could be drawn. Therefore, further study at lower sampling rates is needed. Available datasets can be resampled and re-analyzed.

In this masters's dissertation, three the axis of the two accelerometers were taken into account. A reduction in the number of accelerometer axes could reduce computational cost, storage load and energy use. To study the effects of reducing the number of the accelerometer axes on the classification accuracy, we could use one axis or two axes instead of three axes and use this as an input for the classification algorithm.

Also, we need further study for the analysis of the features extracted automatically by the convent and compare them with the well-known hand-crafted features. Further study on the characteristics of the used CNN and utilizing larger dataset should be conducted.

Bibliography

- [1] Birth alarm. https://www.gallaghereurope.com/en_export_ge/birth-alarm. Accessed: 2019-05-29.
- [2] Equestic saddle clip. <http://www.maddelin.be/nl/equestic-eq-saddleclip.html>. Accessed: 2019-05-29.
- [3] Horse rolling. <https://www.horseandhound.co.uk/tag/colic-in-horses>. Accessed: 2018-05-30.
- [4] Horse pawing. <http://www.intakevets.co.uk/colic-advice>. Accessed: 2018-05-30.
- [5] Horse flank watching. <https://extension.umn.edu/horse-health/colic-your-horse>. Accessed: 2018-05-30.
- [6] Ax3 axivity 3-axis logging accelerometer. <https://axivity.com/product/ax3>. Accessed: 2019-04-02.
- [7] Tendon boot with accelerometer. <https://equiporiumstore.com/products/tr-pro-tendon-boot>. Accessed: 2019-04-02.
- [8] Xusheng Lei and Zhehao Sui. Intelligent fault detection of high voltage line based on the faster r-cnn. *Measurement*, 138:379–385, 2019.
- [9] Gila Abells Sutton, Roe Dahan, Dan Turner, and Ora Paltiel. A behaviour-based pain scale for horses with acute colic: scale construction. *The Veterinary Journal*, 196(3):394–401, 2013.
- [10] Josie L Traub-Dargatz, Christine A Koprak, Ann Hillberg Seitzinger, Lindsey P Garber, Kim Forde, and Nathaniel A White. Estimate of the national incidence of and operation-level risk factors for colic among horses in the united states, spring 1998 to spring 1999. *Journal of the American Veterinary Medical Association*, 219(1):67–71, 2001.
- [11] CJ Proudman. A two year, prospective survey of equine colic in general practice. *Equine Veterinary Journal*, 24(2):90–93, 1992.
- [12] Vanessa L Cook and Diana M Hassel. Evaluation of the colic in horses: decision for referral. *The Veterinary clinics of North America. Equine practice*, 30(2):383–98, 2014.

- [13] Lena Wallin, Erling Strandberg, Jan Philipsson, and Göran Dalin. Estimates of longevity and causes of culling and death in swedish warmblood and coldblood horses. *Livestock production science*, 63(3):275–289, 2000.
- [14] Claudia Giannetto, Marilena Bazzano, Simona Marafioti, Cristiano Bertolucci, and Giuseppe Piccione. Monitoring of total locomotor activity in mares during the prepartum and postpartum period. *Journal of Veterinary Behavior: Clinical Applications and Research*, 10(5):427–432, 2015.
- [15] Patrick T Colahan, IG Mayhew, AM Merritt, JN Moore, et al. *Equine medicine and surgery. Volumes I and II*. Number Fourth Edition. American Veterinary Publications Inc., 1991.
- [16] Joan-Bryce Burla, Anic Ostertag, Heike Schulze Westerath, and Edna Hillmann. Gait determination and activity measurement in horses using an accelerometer. *Computers and electronics in agriculture*, 102:127–133, 2014.
- [17] Terry Taewoong Um, Vahid Babakeshizadeh, and Dana Kulić. Exercise motion classification from large-scale wearable sensor data using convolutional neural networks. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 2385–2390. IEEE, 2017.
- [18] Daniele Ravi, Charence Wong, Benny Lo, and Guang-Zhong Yang. A deep learning approach to on-node sensor data analytics for mobile or wearable devices. *IEEE journal of biomedical and health informatics*, 21(1):56–64, 2017.
- [19] Elizabeth J Davidson. Lameness evaluation of the athletic horse. *Veterinary Clinics: Equine Practice*, 34(2):181–191, 2018.
- [20] Horse gaits. https://en.wikipedia.org/wiki/Horse_gait. Accessed: 2019-04-02.
- [21] SB Kotsiantis, Dimitris Kanellopoulos, and PE Pintelas. Data preprocessing for supervised leaning. *International Journal of Computer Science*, 1(2):111–117, 2006.
- [22] Hennie Brugman, Albert Russel, and Xd Nijmegen. Annotating multi-media/multi-modal resources with elan. In *LREC*, 2004.
- [23] Katja Liebal, Bridget M Waller, Katie E Slocombe, and Anne M Burrows. *Primate communication: a multimodal approach*. Cambridge University Press, 2013.
- [24] Andrea Ravignani and W Tecumseh Fitch. Sonification of experimental parameters as a new method for efficient coding of behavior. In *Proceedings of Measuring Behavior*, volume 2012, pages 376–379. Citeseer, 2012.
- [25] Said Benaissa, Frank AM Tuytens, David Plets, Toon de Pessemier, Jens Trogh, Emmeric Tanghe, Luc Martens, Leen Vandaele, Annelies Van Nuffel, Wout Joseph, et al. On the use of on-cow accelerometers for the classification of behaviours in dairy barns. *Research in veterinary science*, 2017.

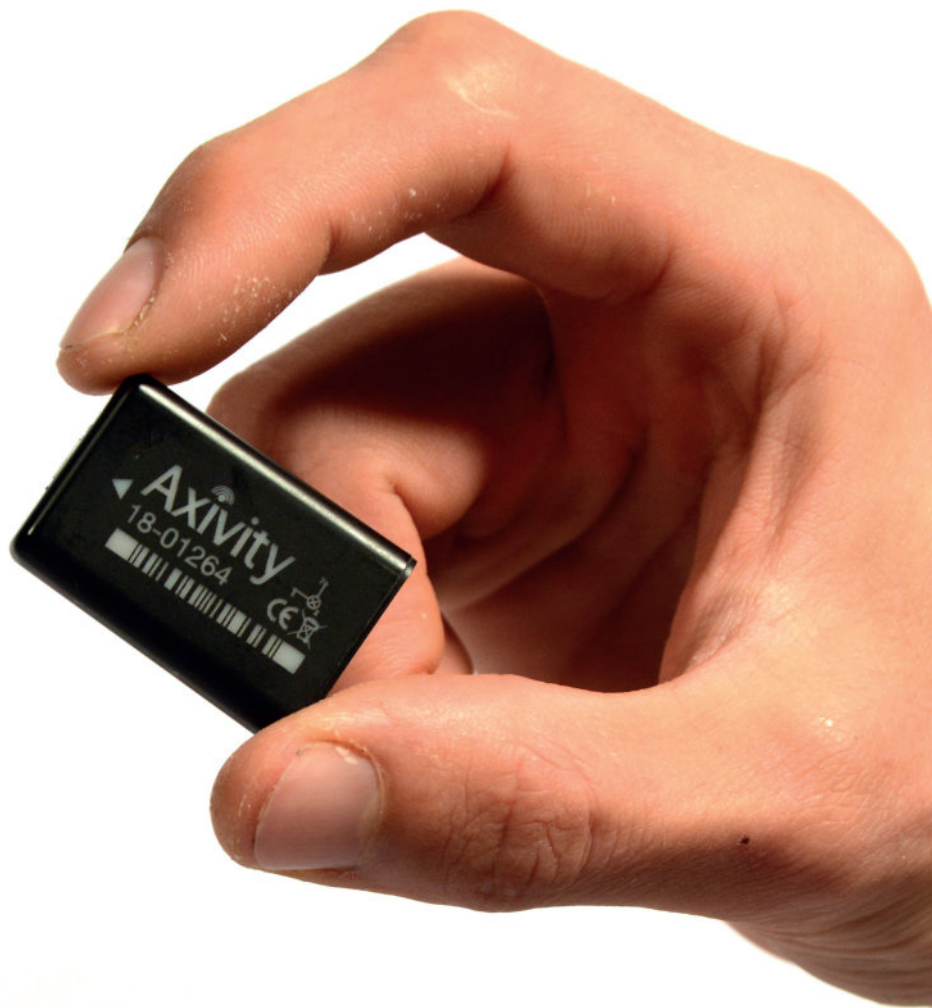
- [26] Xin Zhao, Haikun Wei, Hai Wang, Tingting Zhu, and Kanjian Zhang. 3d-cnn-based feature extraction of ground-based cloud images for direct normal irradiance prediction. *Solar Energy*, 181:510–518, 2019.
- [27] Andrey Ignatov. Real-time human activity recognition from accelerometer data using convolutional neural networks. *Applied Soft Computing*, 62:915–922, 2018.
- [28] Baris Bozkurt, Ioannis Germanakis, and Yannis Stylianou. A study of time-frequency features for cnn-based automatic heart sound classification for pathology detection. *Computers in biology and medicine*, 100:132–143, 2018.
- [29] Ji Gan, Weiqiang Wang, and Ke Lu. A new perspective: Recognizing online handwritten chinese characters via 1-dimensional cnn. *Information Sciences*, 478:375–390, 2019.
- [30] Muhammad Tahir, Hilal Tayara, and Kil To Chong. ipseu-cnn: Identifying rna pseudouridine sites using convolutional neural networks. *Molecular Therapy-Nucleic Acids*, 16:463–470, 2019.
- [31] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [32] Charissa Ann Ronao and Sung-Bae Cho. Human activity recognition with smartphone sensors using deep learning neural networks. *Expert systems with applications*, 59:235–244, 2016.
- [33] Zhihua Cui, Lei Du, Penghong Wang, Xingjuan Cai, and Wensheng Zhang. Malicious code detection based on cnns and multi-objective algorithm. *Journal of Parallel and Distributed Computing*, 2019.
- [34] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- [35] Arafat Abu Mallouh, Zakariya Qawaqneh, and Buket D Barkana. Utilizing cnns and transfer learning of pre-trained models for age range classification from unconstrained face images. *Image and Vision Computing*, 2019.
- [36] Anju Unnikrishnan, V Sowmya, and KP Soman. Deep alexnet with reduced number of trainable parameters for satellite image classification. *Procedia computer science*, 143:931–938, 2018.
- [37] Ax3 user manual. <https://axivity.com/userguides/ax3/using/>. Accessed: 2018-10-01.
- [38] Ax3 omgui. <https://github.com/digitalinteraction/openmovement/wiki/AX3-GUI>. Accessed: 2018-10-01.

Appendix A

AX3 data sheet

AX3 Data Sheet

3-Axis Logging Accelerometer



Version: 1.3
Date: 05/02/2015

www.axivity.com

AX3

3-Axis Logging Accelerometer

Description

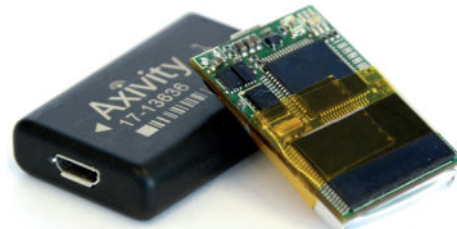
The AX3 is a low cost logging 3-axis accelerometer. At the heart of the sensor is a non-volatile flash memory chip linked by a USB enabled microcontroller. A temperature sensor, ambient light sensor, real time clock (RTC) and lithium polymer battery are also integrated into the sealed polycarbonate puck. The charge time is approximately 90 minutes and the sensor will record for up to 21 days of continuous data. The device is suitable for use in a variety of environments, is water resistant up to 1.5 meters and is CE safety mark approved.

Applications

- Human movement science
- Sports research
- Instrumented environments
- Digital interaction
- Activity recognition

Summary

- 3-axis accelerometer
- Light sensor
- Temperature sensor
- 512MB memory
- 14 days recording at 100Hz
- Rechargeable lithium polymer battery
- Water resistant and CE marked
- Configurable logging options



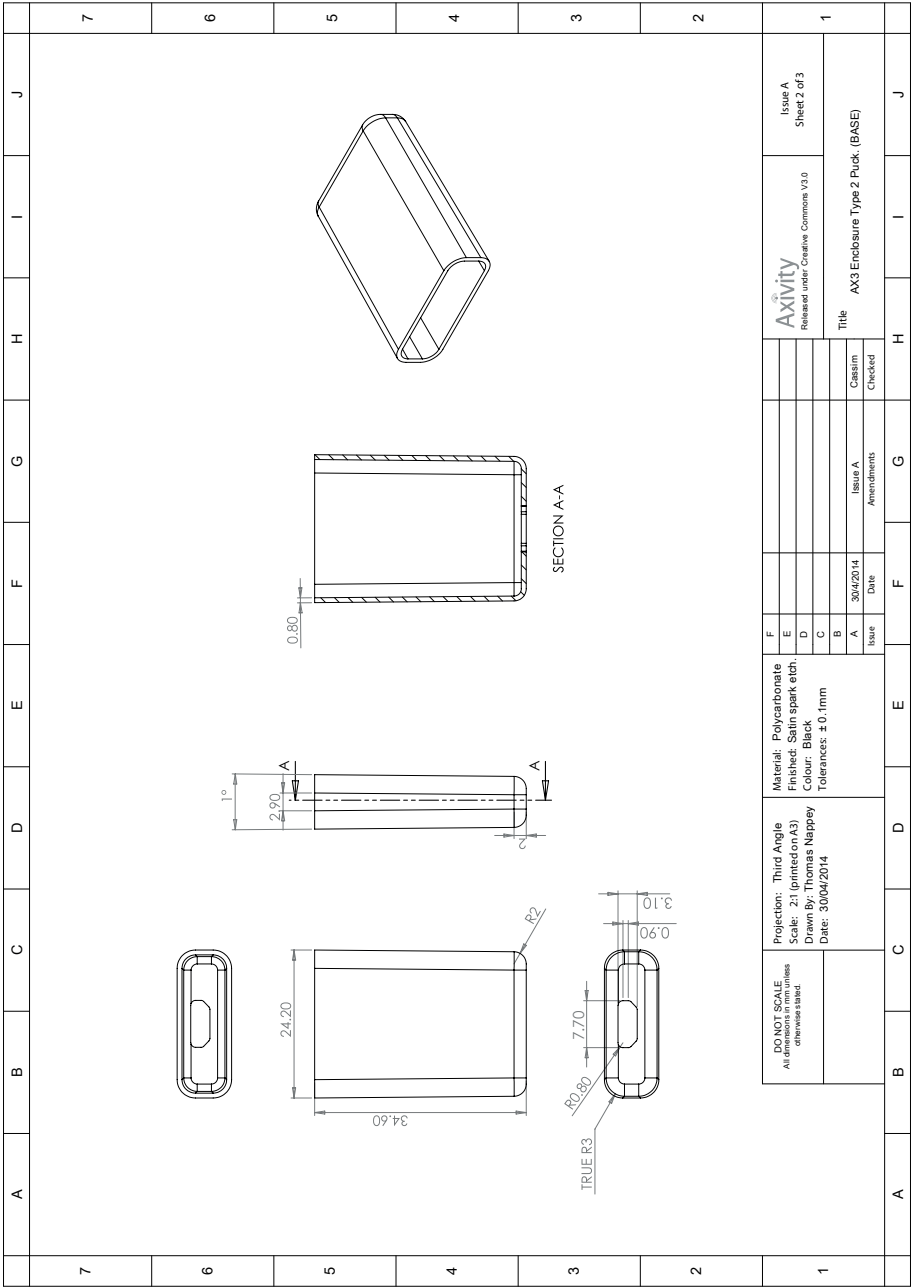


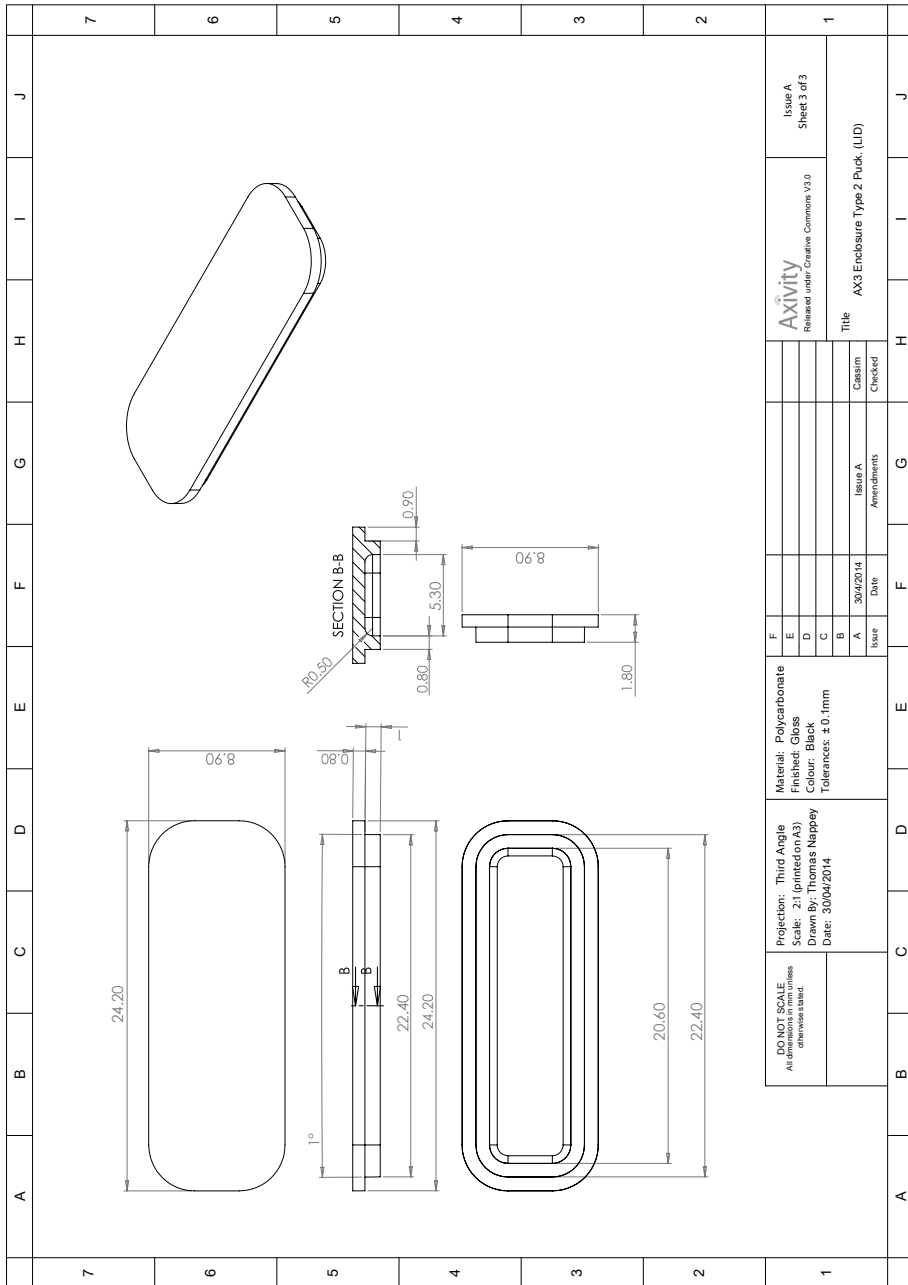
Specification: AX3

PARAMETER	VALUE	NOTES
Puck Size	23x32.5x8.9 mm	
Puck Weight	11g	
Enclosure Material	Polycarbonate	
Battery Capacity	150mAh	Rechargeable lithium polymer
Battery Charge Current	150mA	
Connectivity	Micro USB	
ENVIRONMENTAL		
Moisture Ingress	Water-resistant to 1.5m	IPx8
Dust Ingress	Dust tight	IP6x
Operating Temperature	0 - 65°C (not charging)	40°C if charging
TYPICAL CAPABILITIES		
Memory	512MB NAND flash non-volatile	
Logging Frequencies	Configurable 12.5Hz - 3200Hz	
Maximum Logging Periods	30 days at 12.5 Hz or 14 days at 100Hz	
REAL TIME CLOCK		
Type	Quartz real time clock	
Frequency	32.768KHz	
Precision	± 50ppm (typical)	
ACCELEROMETER		
Sensor Type	MEMS	
Range	±2/4/8/16g	Configurable
Resolution	upto 13-bit	Configurable
LIGHT		
Sensor Type	APDS9007	Logarithmic light sensor
Wave Length	470-650 nm	Matched to human eye
Range	3-1000 LUX	At sensor
Digital format	10 bit	
TEMPERATURE		
Sensor Type	MCP9700	Linear thermistor
Range	0 - 40°C	
Resolution	0.3°C	
Accuracy	1°C typical (4°C max)	

Dimensions: AX3



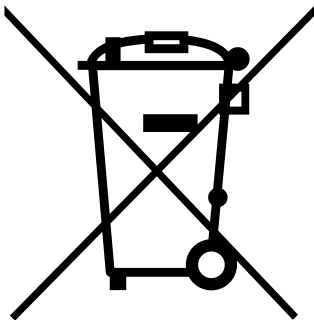
[illegible]





Certification:

The AX3 is certified to the following:

Certification	Test
	<p>The product is compliant with the Directive 2004/108/EC; the relevant Declaration of Conformity is available from Axivity</p> <p>The product has been tested to BS EN 61000-6-1 :2007 and BS EN 61000-6-3 :2007 (Electromagnetic compatibility (EMC), Generic standards, Immunity for residential, commercial and light-industrial environments).</p>
	<p>The product has an ingress protection rating as defined in IEC 60529 to level 68. Due to the nature of the housing (potted enclosure) the device was passed on the basis that it was fully functional both before and after each testing criterion.</p>
	<p>In accordance with the European Directive 2002/96/EC on Waste Electrical and Electronic Equipment (WEEE), the product must not be disposed of in the normal unsorted municipal waste stream. Instead, it is the user's responsibility to dispose of this product by returning it to a collection point or directly to Axivity. Separate collection of this waste helps optimize the recovery and recycling of any reclaimable materials and also reduces the impact on human health and the environment. For more information concerning the correct disposal of this product, please contact your local authority or our issuing authority</p> <p>The lithium polymer cell has met the acceptance criterion for the UN Recommendations on the Transport of Dangerous Goods relating to lithium batteries, reference Para 38.3 of Manual tests and Criteria document No. ST/SG/AC.10.11/Rev.4:2003</p>

**Disclaimer:**

Information in this document is believed to be accurate and reliable. However, the manufacturer does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. The manufacturer reserves the right to make changes to information published in this document, including without limitation, specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof. The manufacturer's products are not designed, authorized or warranted to be suitable for use in applications where failure or malfunction can reasonably be expected to result in personal injury, death or severe property or environmental damage. The manufacturer accepts no liability for inclusion and/or use of its products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Copyright (c) 2009-2015, Newcastle University, UK.
All rights reserved.
Licensed under Creative Commons 3.0 Attribution License (BY),
<http://creativecommons.org/licenses/by/3.0/>

Appendix B

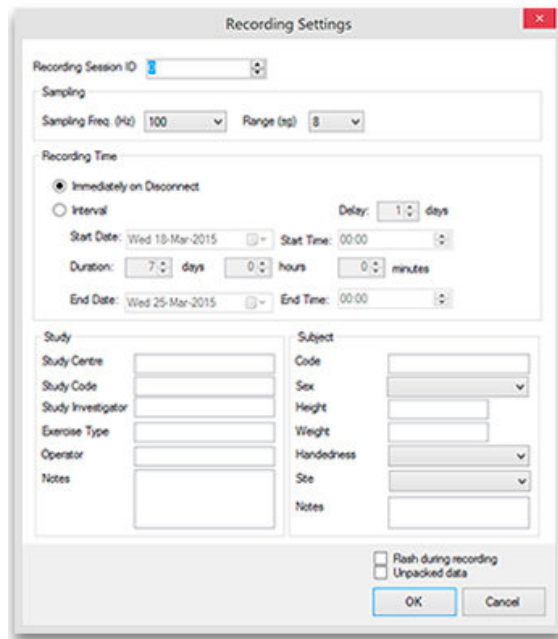
Manual

B.1 Introduction

In this manual an overview is given on how to setup and configure the AX3 sensors for recording. Secondly, an explanation is given on how to extract gathered data from the Axivity AX3 accelerometer. Thirdly, the data of two accelerometers is preprocessed so that it can be uploaded to a video annotation tool named ELAN. Then, it is explained how the data file is annotated using ELAN. Afterwards, it is shown how the obtained annotations are exported to label the gathered data. And finally, it is explained how the data can be uploaded to the algorithm with split testing and without split testing. Also, a brief discussion is given on which parameters can be adjusted to fine tune the model.

B.2 Setup and configure AX3 sensors for recording

To setup and configure AX3 sensors for recording it is necessary to download the OMGUI software. The OMGUI software is available from the open source GitHub repository and can be downloaded here at the following link: <https://github.com/digitalinteraction/openmovement/blob/master/Downloads/AX3/AX3-GUI-37.zip?raw=true>. Connect the device and ensure it appears in the Device Browser Pane. To remove any existing data that may be stored on the device, ensure the device is highlighted in the Device Browser Pane and left click the Clear button in the Device Toolbar. Then click the record button. This button is used to configure the device to record. The button brings up the recording window.



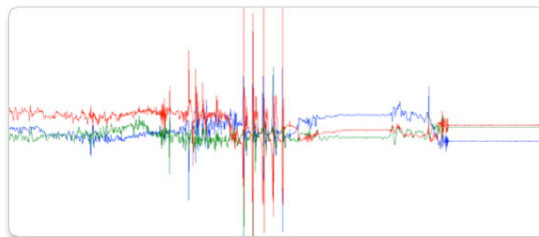
The image shows a 'Recording Settings' dialog box with the following fields and options:

- Recording Session ID:** A text input field.
- Sampling:**
 - Sampling Freq. (Hz):** A dropdown menu set to 100.
 - Range (g):** A dropdown menu set to 8.
- Recording Time:**
 - Immediately on Disconnect:** A radio button that is selected.
 - Interval:** A radio button that is not selected.
 - Delay:** A dropdown menu set to 1 days.
 - Start Date:** A date picker set to Wed 18-Mar-2015.
 - Start Time:** A time picker set to 00:00.
 - Duration:**
 - days:** A spinner set to 7.
 - hours:** A spinner set to 0.
 - minutes:** A spinner set to 0.
 - End Date:** A date picker set to Wed 25-Mar-2015.
 - End Time:** A time picker set to 00:00.
- Study:**
 - Study Centre: Text input field.
 - Study Code: Text input field.
 - Study Investigator: Text input field.
 - Exercise Type: Text input field.
 - Operator: Text input field.
 - Notes: Text area.
- Subject:**
 - Code: Text input field.
 - Sex: Dropdown menu.
 - Height: Text input field.
 - Weight: Text input field.
 - Handedness: Dropdown menu.
 - Site: Dropdown menu.
 - Notes: Text area.
- Checkboxes:**
 - ☐ Flush during recording
 - ☐ Unpacked data
- Buttons:** OK and Cancel.

Set the desired sampling frequency and the sampling range. Set the record time to "immediately on disconnect". In this configuration the sensor will start recording as soon as it is unplugged from the computer.

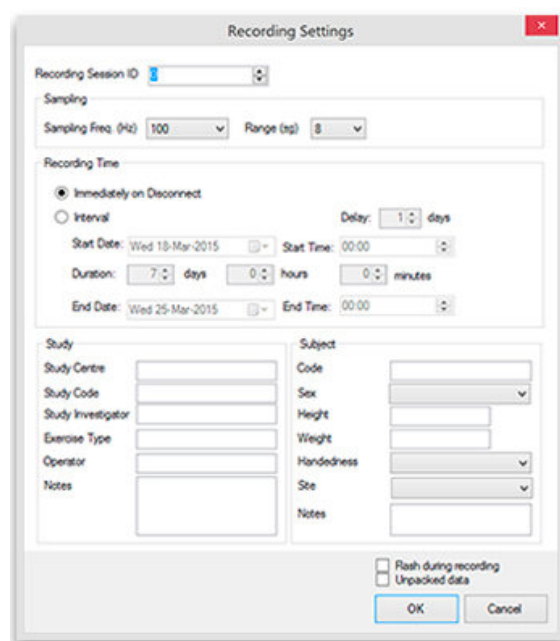
B.3 Datamarkers

When capturing experimental data it is often useful to place markers in the data. Such markers can retroactively be used to identify certain event start and stops and make it easier to preprocess the datafile. A popular choice for creating a data marker is to subject the device to a short impulse force. Such forces can be generated through clapping with the device held in the hand. The graph below is a graphical representation of a data marker generated by five short claps [37].



B.4 Extracting data from Axivity AX3 accelerometer

The AX3 logs data internally in a binary packed format. This format is named Continuous Wave Accelerometer (CWA) format. This format is very efficient for storing large amounts of data but is not natively supported by many applications. Therefore it is necessary to convert the CWA file to a Comma Separated Value (CSV) format. The OMGUI software supplied with the sensors provides provision to convert the raw CWA files into either of these formats with a variety data interpolation and timestamp formatting options [38].



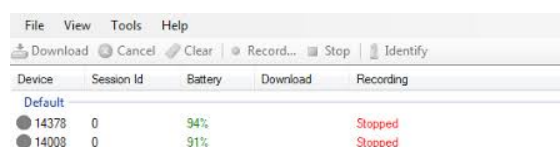
The 'Recording Settings' dialog box contains the following fields and controls:

- Recording Session ID:** A text input field with a dropdown arrow.
- Sampling:**
 - Sampling Freq. (Hz):** A dropdown menu set to 100.
 - Range (g):** A dropdown menu set to 8.
- Recording Time:**
 - Immediately on Disconnect:** A radio button that is selected.
 - Interval:** An unselected radio button.
 - Delay:** A dropdown menu set to 1 days.
 - Start Date:** A date picker set to Wed 18-Mar-2015.
 - Start Time:** A time picker set to 00:00.
 - Duration:**
 - Days:** A spinner set to 7.
 - Hours:** A spinner set to 0.
 - Minutes:** A spinner set to 0.
 - End Date:** A date picker set to Wed 25-Mar-2015.
 - End Time:** A time picker set to 00:00.
- Study:**
 - Study Centre: Text input
 - Study Code: Text input
 - Study Investigator: Text input
 - Exercise Type: Text input
 - Operator: Text input
 - Notes: Text area
- Subject:**
 - Code: Text input
 - Sex: Dropdown menu
 - Height: Text input
 - Weight: Text input
 - Handedness: Dropdown menu
 - Site: Dropdown menu
 - Notes: Text area
- Checkboxes:**
 - ☐ Flush during recording
 - ☐ Unpacked data
- Buttons:** OK and Cancel.

B.4.1 Connect accelerometer and download data

Connect the device and ensure it appears in the device browser pane. Any connected devices will appear in the device browser pane marked as having data. When highlighted, data on these devices can be previewed in the data preview window. There is a zoom tool and highlight tool in the selection tools section. In the preview filters section, a number of checkboxes exist to help the user visualize the data. On mousing over the data, the user is able to get a precise data preview at any given instant in the recording.

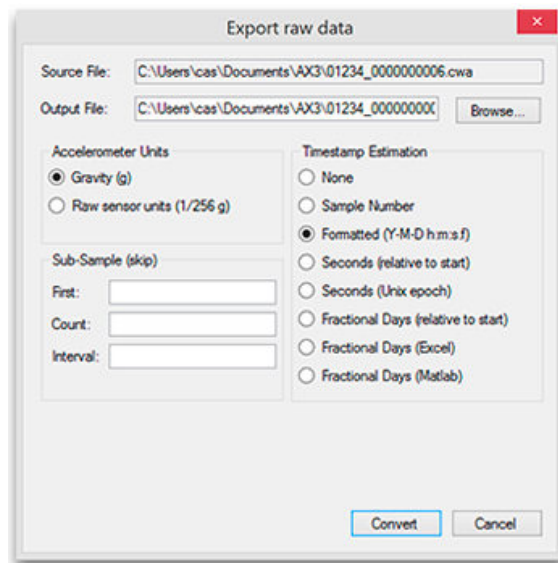
Recorded data stored on the sensor can be downloaded to the working folder by clicking the download button. Firstly, stop the data capturing by clicking on the stop button.



Device	Session Id	Battery	Download	Recording
Default				
14378	0	94%		Stopped
14008	0	91%		Stopped

In order to use the raw data with a third-party software package, such as Excel, it must first be converted into a suitable format. This can be done using the export window. To investigate this, select the file in the files tab within the local files pane and click export button.

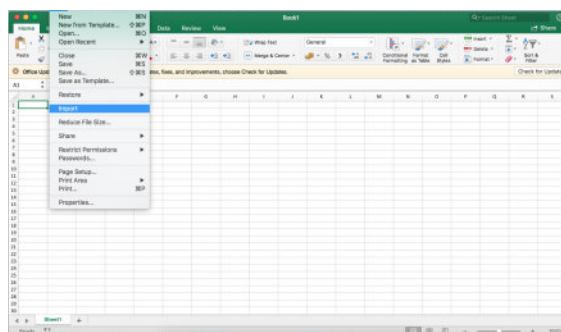
The following window will appear. Set the configuration as shown below and click the OK button to generate the file.



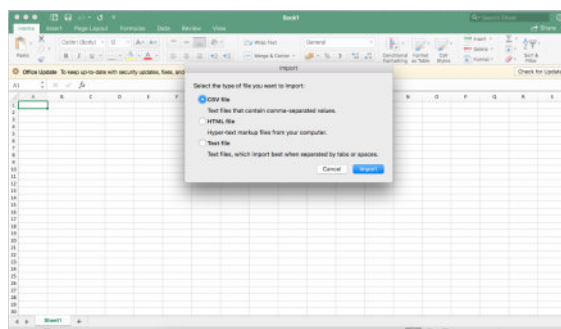
B.5 Preprocessing data for Elan

B.5.1 Import the CSV file in Excel

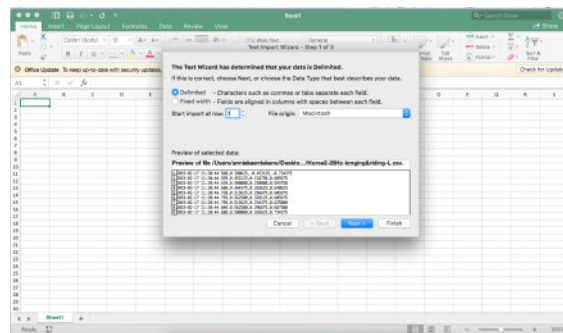
Start Excel with a blank workbook open. Select 'File' from the menu, and then 'Import'. (If the menu options are greyed out this could be because there is no workbook open).



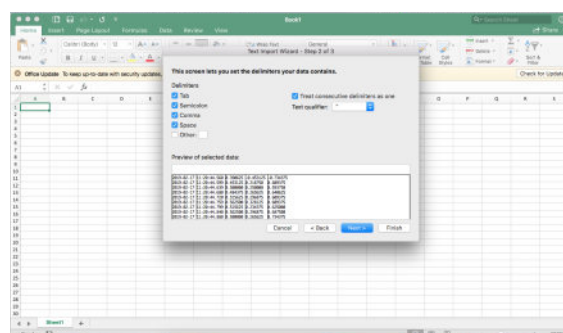
Browse for the *.csv file you want to open, and click 'Import'.



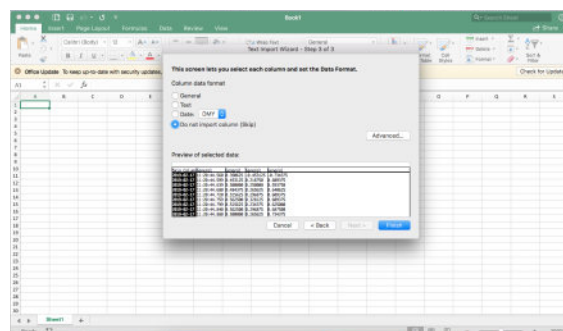
In the Text import wizard, ensure the 'Delimited' option is selected. Click Next.



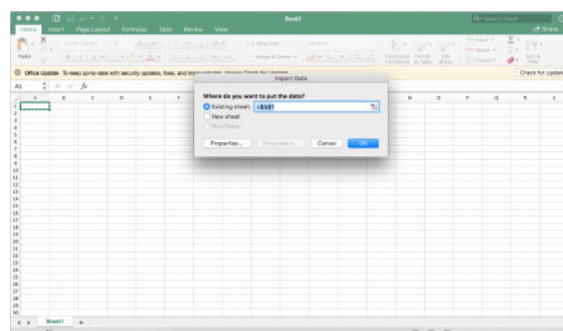
In the delimiters section, set the configuration as shown below and click next.



In the data format section, set the configuration as shown below. The first column is highlighted and will be skipped. Click Finish.

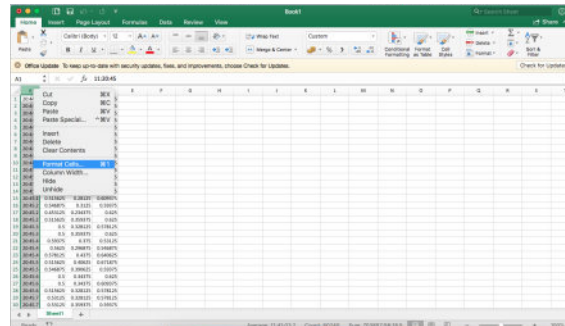


The following question will appear 'Where do you want to put the data?'. Any cell on the blank worksheet can be clicked to determine where the data should go. Click 'OK' to proceed.

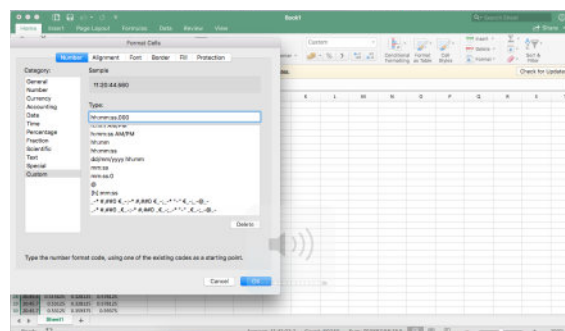


B.5.2 Preprocess data of two accelerometers

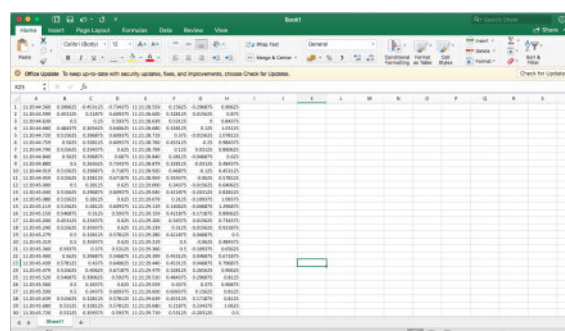
Format a column to display the time by right-clicking on the selected column and selecting "Format Cells".



On the "Number" tab, under "Category" select "Time". On the right hand side under "Type", choose the format hh:mm:ss and format it as hh:mm:ss.000, then click "OK".



Do the same for the second accelerometer data and place the data next to the first accelerometer data.



The accelerometer times need to be aligned. Look at the start time of the video and search for this time in the file.

Select the cells above this start time you want to delete. Right click and select delete cells.

The delete dialog box opens, showing several options for filling in the gaps. Choose shift cells up.

Do the same for the second accelerometer.

Insert a column and subtract the start time from the time cells.

Apply this to the whole column and do the same for the right accelerometer.

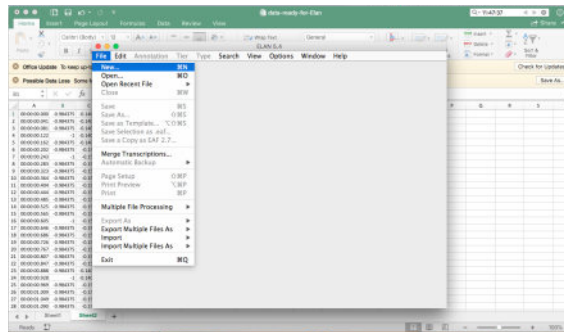
Save the file in the CSV file format.

B.6 Video annotation

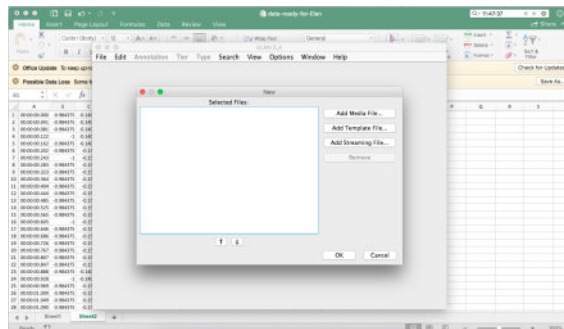
Video annotation is done with the open source software ELAN. It can be downloaded from this link: <https://tla.mpi.nl/tools/tla-tools/elan/download/>. ELAN is a highly specialized software that allows downloading of the video file and the correspondent acceleration data that have to be labelled. Annotations can be made by selecting the length of the segment where the behaviour is performed and typing the annotation. Once the CSV file is obtained in the right format, the video and the data stream can then be aligned.

B.6.1 Add files to Elan

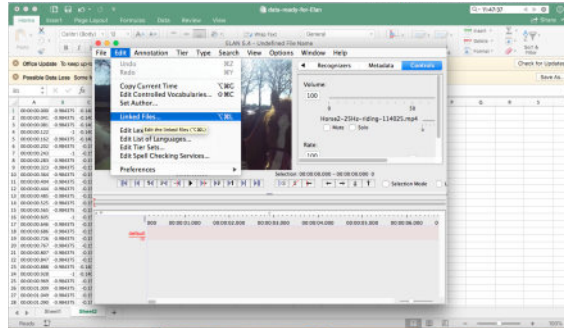
Open Elan, select File from the menu bar and then click New.



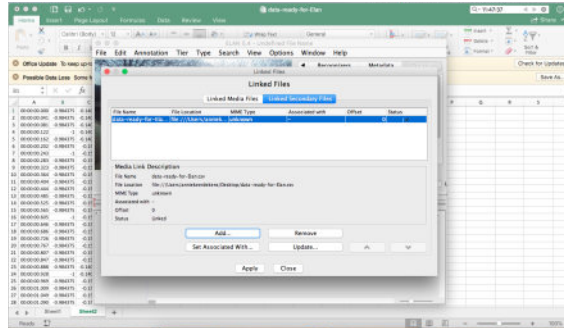
Then click in the dialogbox on add Media File and select the right videofile.



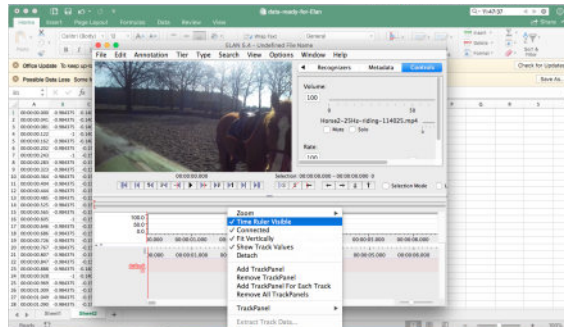
Click on Edit and choose linked files.



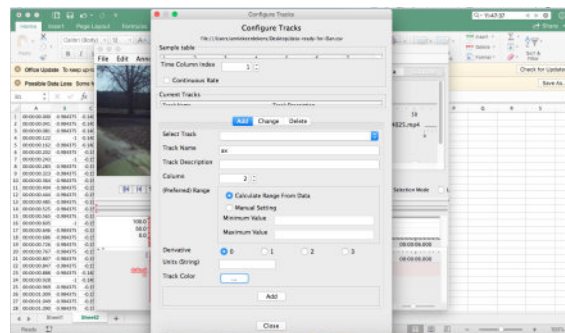
Click on Linked Secondary Files and add the CSV file with the accelerometer data.



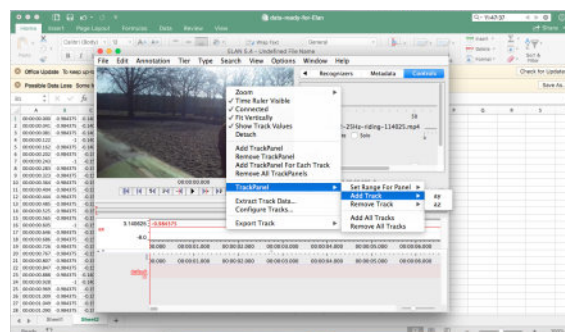
On the track view, right click and select Configure tracks.



A dialogbox opens and configure it as displayed in the figure below. Here a_{xL} is displayed. Choose as time column index for the left accelerometer 1. When configuring the right accelerometer data, you should adjust this to 6. The data of a_{xL} is in column 3. Do the same for track a_{yL} (in column 4) and a_{zL} (in column 5). And repeat this process with adjusted column numbers for the right accelerometer data. Also select for each of the data an other color. For this experiment it was only necessary to use the a_{xL} and a_{xR} data to annotate the file. Click close.

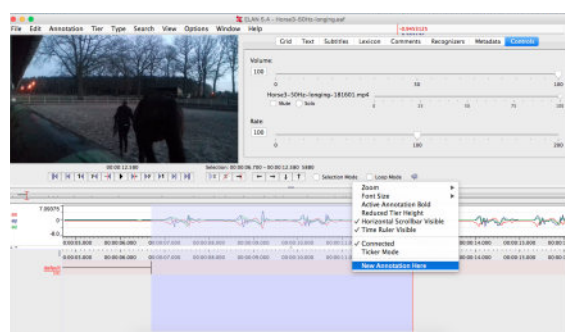


Now only a_x is displayed. Right click on the track view and select Trackpanel and then Add track.

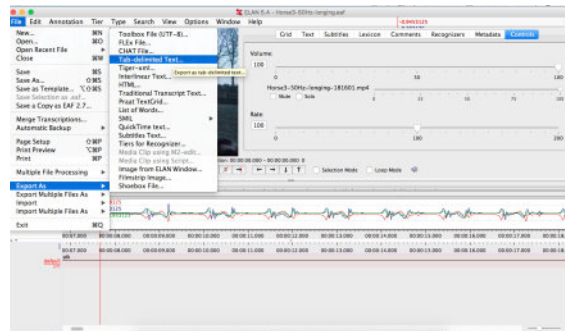


Look for the datamarker in accelerometer data and on the video. Look if they are appearing on the same moment for both accelerometers. Sometimes there is still a time difference between the video file and the data. Then some extra preprocessing might be done to cut off the right amount of time from the data file. If the accelerometer data and video file are aligned then the annotation process can be started.

Select the data to annotate, right click on the annotation panel and select new annotation.



When all annotations are done, those need to be exported as a CSV file. Click on file, export as Tab-delimited text.



B.7 Annotate file in Excel

First cut of the data at the end of the file in Excel when the video recording was stopped. Then you will notice that there is probably a difference in length for each accelerometer.

 The screenshot shows an Excel spreadsheet with a table of data points. The table has columns for time, amplitude, and other parameters. The data points are listed in rows, with the first row starting at 00:00:00.000 and the last row ending at 00:00:00.000. The table is titled "SLAN 6.4 - Internal Data Connections" and includes a "Security Warning" message.

This is because the actual sampling rates differ from the configured sampling rates. Some extra preprocessing is necessary. Insert an extra cell next to the time cells. The function MROUND is used. This function returns a number rounded to the desired multiple. The values of the time column can be brought to the closest multiple of the sampling rate. In this file a sampling rate of 50Hz was used, so the multiple was "00:00:00.020".

 The screenshot shows an Excel spreadsheet with a table of data points. The table has columns for time, amplitude, and other parameters. The data points are listed in rows, with the first row starting at 00:00:00.000 and the last row ending at 00:00:00.000. The table is titled "SLAN 6.4 - Internal Data Connections" and includes a "Security Warning" message.

Do the same for the other time column.

Excel spreadsheet showing a table with columns A through R. The table contains numerical data, including dates and times, and some cells are highlighted in green. The formula bar shows a VLOOKUP function.

Then the shortest rounded column should be copied.

Excel spreadsheet showing a table with columns A through R. The table contains numerical data, including dates and times, and some cells are highlighted in green. The formula bar shows a VLOOKUP function.

And also the acceleration data, which belongs to this column.

Excel spreadsheet showing a table with columns A through R. The table contains numerical data, including dates and times, and some cells are highlighted in green. The formula bar shows a VLOOKUP function.

Now the VLOOKUP function needs to be used to retrieve data from a specific column in a table. Apply the VLOOKUP function as displayed in the next image.

Excel spreadsheet showing a table with columns A through R. The table contains numerical data, including dates and times, and some cells are highlighted in green. The formula bar shows a VLOOKUP function.

Do the same for the other 2 columns, adjusting the third argument to 4 and 5.

Security Metrics - Export Data Configuration (View: Detailed)																										
Please select the data source, metrics, and filters for the report. If you are using the default (Live) feed, you can use the default metrics. If you are using a custom feed, you can use the custom metrics.																										
DEED																										

Import the CSV file with annotations in Excel.

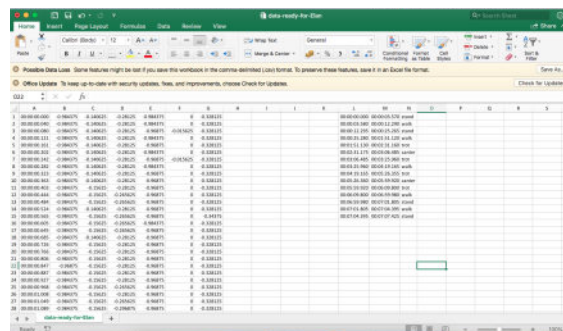
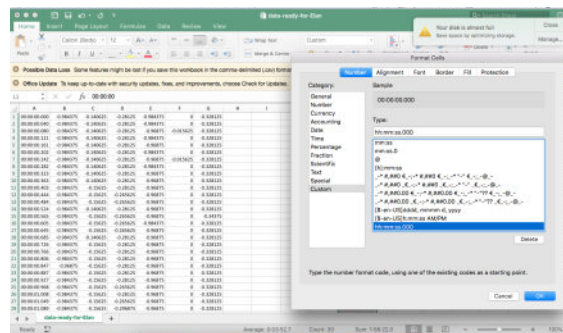
Microsoft Excel 'Data' ribbon screenshot. The 'Data' tab is active, showing the 'Data Tools' group. The 'Data Validation' icon is highlighted with a red box. The ribbon includes tabs for 'Formulas', 'Data', 'Review', and 'View'. The 'Data' tab is active, showing options like 'Sort & Filter', 'Data Tools', and 'Data Validation'. The 'Data Validation' icon is a small icon with a checkmark and a red X, located in the 'Data Tools' group. Below the ribbon, a message bar states: 'Positive Data Series: Some features might not be visible if you use the version in the column named (data) below. To prepare these features, use a 64-bit Excel file format.' To the right of the message bar, there is a link 'Check for updates'.

Delete all the columns as shown in the figure below.

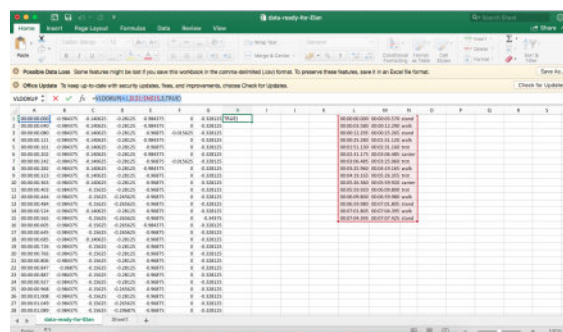
[illegible]

Format the first 2 columns to display the time by right-clicking on the selected column and selecting "Format Cells".

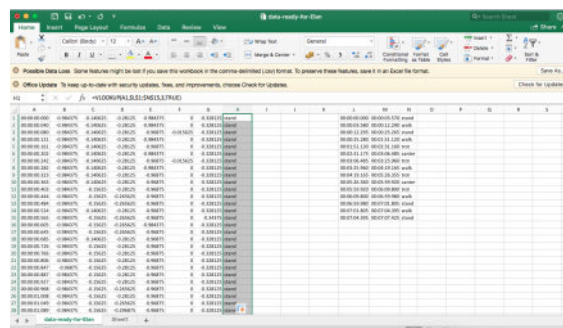
[illegible]



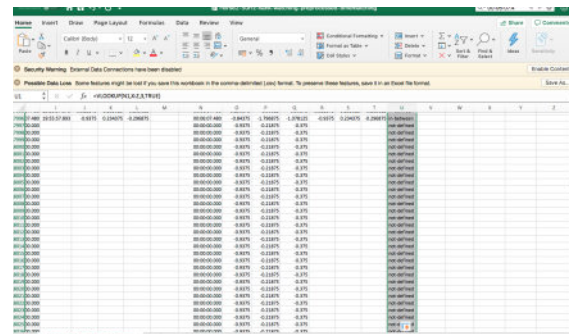
Use the VLOOKUP function in excel to annotate the rows as is displayed in the next image.



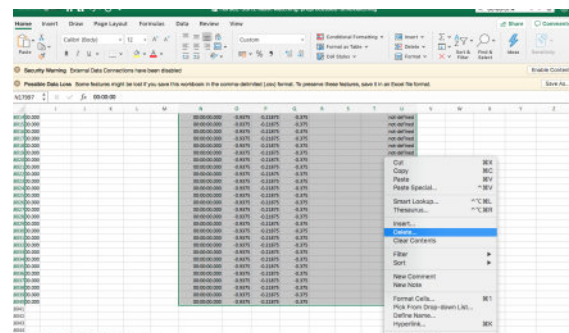
Apply this to the whole column.



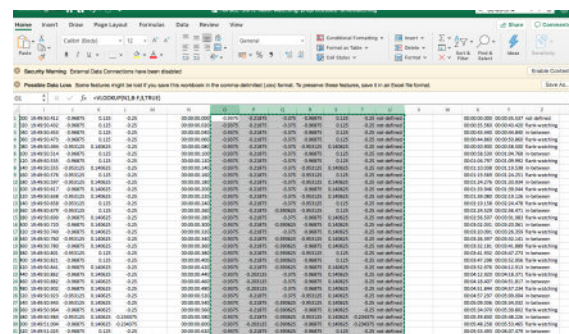
Check the end of the columns since sometimes too much cells are labeled as is displayed in this figure.



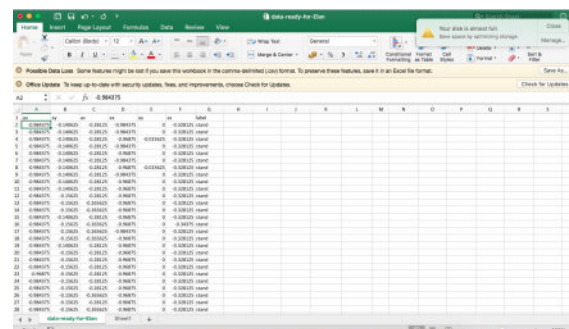
Delete these rows.



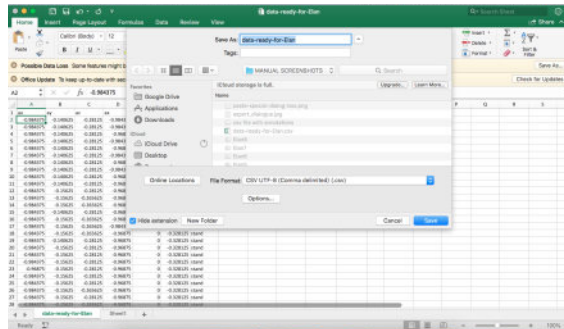
Copy paste the columns with the accelerometer data and labels to another sheet.



Insert a row with the names of the columns as is indicated in the next figure.

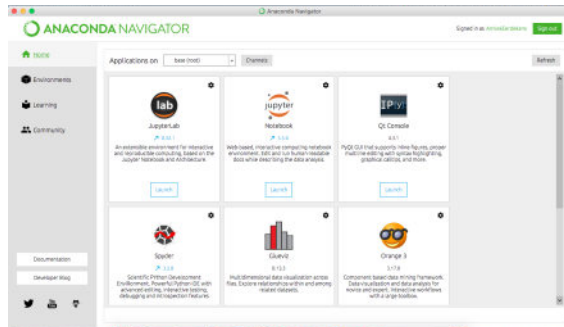


Save this file as a CSV file format.



B.8 Uploading data to algorithm and adjusting parameters

Install Anaconda using the following url: <https://www.anaconda.com/>. Open Anaconda and launch the notebook.



Upload the python code "Simple movements with split testing" and "Simple movements without split testing" together with the CSV file.

B.8.1 Reading the CSV file of the code with split testing

Open the code and insert the name of the CSV file that is uploaded.

```
# reading csv file
df = pd.read_csv('csv file with labeled data.csv', encoding='utf8')
rows = df.shape[0]
columns = df.shape[1]
print("Read data with {} rows and {} columns".format(rows, columns))
```

B.8.2 Reading the CSV file of the code without split testing

If there are two CSV files, it is also possible to split the testing and validation data based on these two files. Insert the name of the CSV files that are uploaded. Insert the name of the CSV file that contains the training data in the training part of the algorithm and the name of the CSV file that contains the validation data in the validation part of the algorithm.

```
# reading training csv file
df_train = pd.read_csv('file1_with_training_data.csv', encoding='utf8')
rows = df_train.shape[0]
```

```

columns = df_train.shape[1]
print("Read training data with {} rows and {} columns".format(rows, columns))

# reading validation csv file
df_val = pd.read_csv('file2_with_validation_data.csv', encoding='utf8')
rows = df_val.shape[0]
columns = df_val.shape[1]
print("Read validation data with {} rows and {} columns".format(rows, columns))

```

B.8.3 Adjusting parameters

Dropout

Dropout consists of setting to zero the output of each hidden neuron with probability a certain probability dr . The neurons which are “dropped out” in this way do not contribute to the forward pass and do not participate in back-propagation. The dropout parameter can be set as any number between 0 and 1. In this dissertation a dropout of 0.55 is used.

```

# Variables
dr = 0.55

# Fixed random seed
np.random.seed(1024)

```

Filter size, Pool size, ...

In the model a number of parameters can be adjusted such as filter size, pool size, zero-padding and type of regularization. Also, the number of layers can be adjusted. A detailed overview of all the parameters that can be adjusted can be found in Section [2.4.3](#).

```

# building model
model = tf.keras.models.Sequential()

model.add(layers.Conv2D(64,(3,1), name="conv1", padding='valid', activation='relu',
    kernel_regularizer= tf.keras.regularizers.l2(0.01), input_shape=(n,columns-1,1)))
model.add(layers.ZeroPadding2D((1,0)))
model.add(layers.MaxPooling2D(pool_size=(2,3)))
model.add(layers.Dropout(dr))

model.add(layers.Conv2D(16,(5,2), name="conv2", padding='valid', activation='relu'))
model.add(layers.MaxPooling2D(pool_size=(3,1)))
model.add(layers.Dropout(dr))

model.add(layers.Flatten())
model.add(layers.Dense(50,activation=tf.nn.relu))
model.add(layers.Dense(10,activation=tf.nn.relu))
model.add(layers.Dense(len_unique_classes, activation=tf.nn.softmax))

# compiling model
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

callback = []
early_stop = tf.keras.callbacks.EarlyStopping(monitor='val_acc', min_delta=0, patience=60,
    verbose=0, mode='auto')
callback.append(early_stop)

```

```

callback.append(tf.keras.callbacks.ModelCheckpoint('weightsfile.weight.h5', monitor='val_acc',
    verbose=0, save_best_only=True, mode='max'))

model.summary()

# train model
history = model.fit(x=X_train, y=Y_train, validation_data=(X_val, Y_val), epochs = 400,
    batch_size=64, callbacks=callback)
model.save('simple_model.h5')
model.load_weights('weightsfile.weight.h5')

```

Split input into measurements of n samples

In this step the input is split into measurements of n samples. Depending on the measurement sampling rate, the number of seconds that each sample presents can be calculated. This parameter can be adjusted and depends on the duration of a movement.

```

# split input into measurements of n samples
n = 200 # number of rows for t = n/f [s] measurements
list_df = [df[i:i+n] for i in range(0, df.shape[0], n)] # split into n chunks
list_df = list_df[:-1] # remove last measurement that probably is not divided by n
np.random.shuffle(list_df) # shuffle data
print("Read {} measurements with dimensions {}".format(len(list_df[:]), list_df[0].shape))

```

Filtering data

Adjust the names under filter_movements to the labels used to describe the movements specified in the uploaded CSV file.

```

# filtering data
arr_Y_r = []
filter_movements = ['stand', 'walk', 'canter', 'trot'] # if empty, take every class
selected_dfs = []
for i in range(0, len(list_df)):
    classes = list_df[i]['label'].unique()
    if (classes.size == 1 and (len(filter_movements) == 0 or classes[0] in filter_movements)):
        selected_dfs.append(list_df[i].values[:, 0:6]) # X values
        arr_Y_r.append(list_df[i].values[0, 6]) # Y values

unique_classes = np.unique(arr_Y_r)
len_unique_classes = len(unique_classes)
arr_Y_r = np.array(arr_Y_r[:])
print("Using {} of measurements".format(len(selected_dfs)))
print("Using {} classes {}".format(len_unique_classes, unique_classes))

```

Splitting training and validation data

In this step, the experimental data is split into two sets randomly. In this case the training set took 66% of the data, and the testing set took the remaining 34% of the data.

```

X_train, X_val, Y_train, Y_val = train_test_split(arr_X, arr_Y, test_size=0.34,
    random_state=1024)

```

Overall validation accuracy and per class accuracy

The overall validation accuracy and per class accuracy is printed after the shown piece of code.

```

cat_Y = np.argmax(Y_val, axis=1)
arr_Y_val = le.inverse_transform(cat_Y)
for movement in unique_classes:
    indexes = np.where(arr_Y_val == movement)[0]
    if (indexes.size > 0):
        # of len(indexes) > 0 als het geen numpy array is
        loss, acc = model.evaluate(X_val[indexes], Y_val[indexes], verbose=0)
        print("Class {} ({} instances) has an unseen accuracy of {:.f}%".format(movement,
            len(indexes), acc*100))
    else:
        print("The movement {} is not present in the validation set".format(movement))

loss, acc = model.evaluate(X_val, Y_val, verbose=0)
print("Best classification accuracy {:.f}%".format(acc*100))

```

Accuracy and loss plots

Validation accuracy of the CNN model for the training and validation dataset is displayed. The plots can provide an indication of useful things about the training of the model, such as: speed of convergence over epochs, indication if the model may have already converged, over/under learning of the model.

```

# list all data in history
print(history.history.keys())
# summarize history for accuracy
plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'validation'], loc='upper left')
plt.show()
# summarize history for loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'validation'], loc='upper left')
plt.show()

```

B.9 Conclusion

In this manual an overview was given on how to use the AX3 accelerometers to detect activities of horses fitted to the lateral side of the tendon boot, preprocess the captured data and using the designed algorithm. When using the accelerometers in different settings different adjustments might be necessary.

Appendix C

Tables

Categorie	Product	Health parameters	Accuracy	Practical use	Comfort horse	Range
Sport watches	Polar equine series	1 (Heart rate)	-/+	+	-	Short
	Birth Alarm	1 (Elevation)	-/+	+	-/+	Long
Birth Alarm	Breeder Alert, Equipage, EquiFone	1 (Position head)	-/+	+	+	Long
	Foal-Alert, Keros Sigloo	1 (Opening birth canal)	+	-	+	Long
	Birth Alert	1 (Temperature)	-/+	-	+	Unknown
	Wyke Foaling Alarm	1 (Perspiration)	-/+	+	-/+	Long
	Trackener	None (Distance, time, speed)	+	+ -/+	Short	
Wearable	Nightwatch	5 (Heart rate, respiration, activity, motion, posture)	-	+	+	Long
	EquiSense Motion	1 (Activity)	-	+	+	Short
	Equestic	1 (Activity)	-	+	+	Short
	Arioneo performance	2 (Heart rate, activity)	-/+	+	-/+	Short
	EquiSense Care	6 (Heart rate, respiration, perspiration, temperature, activity, sleep)	Unknown (prototype only)	+	-	Long
	SeeHorse	4 (Heart rate, respiration, temperature, activity)	-	+	+	Short
	Etrakla	1 (Heart rate)	+	+	+	Short
	Fourganza blanket	1 (Temperature)	-	+	-	Short
	Arioneo care	1 (Temperature)	+	+	-/+	Short
	GesEq	1 (Behaviour)	+	+	++	Short
Video	EquiView360					
Medical	PillCam	(Endoscopy)	+	-	++	Very short
	GIPill	1 (Temperature)	+	-	++	Very Short

Table C.1: Comparable products and their shortcomings.

Subject number	Type of exercising	Configured sample rate [Hz]	Actual sample rate left accelerometer [Hz]	Actual sample rate right accelerometer [Hz]
1	Longed	25	24.68	24.62
1	Ridden	25	24.69	24.65
1	Longed	50	49.60	49.39
1	Ridden	50	49.77	49.47
1	Longed	100	98.90	98.53
2	Longed	25	24.75	24.69
2	Ridden	25	24.74	24.66
2	Longed	50	49.68	49.53
2	Ridden	50	49.73	49.60
2	None (flank-watching)	50	49.08	48.97
2	Longed	100	99.12	98.74
2	Ridden	100	98.77	98.33
2	Longed	200	197.15	196.36
2	Ridden	200	196.77	196.15
2	Longed	1600	1587.46	1584.91
3	Longed	25	24.65	24.58
3	Longed	50	49.37	49.22
3	Longed	100	98.81	98.43
3	Longed	200	197.93	197.27
4	Longed	50	49.39	49.18
4	Ridden	50	49.31	49.13
4	None (roll)	50	49.29	49.11
4	None (roll)	50	49.32	49.13
4	None (roll)	50	49.00	48.80
4	None (paw)	50	49.27	49.09
4	Longed	200	197.22	196.43
4	Ridden	200	196.64	196.23
4	None (roll)	200	196.40	196.17
4	None (roll)	200	196.38	196.11
5	Ridden	50	49.41	49.24
6	Longed	50	49.43	49.29

Table C.2: Configured sampling rate and actual sampling rate for each dataset.

