Towards Deep Unsupervised Representation Learning from Accelerometer Time Series for Animal Activity Recognition

Jacob W. Kamminga
University of Twente
Enschede, The Netherlands
j.w.kamminga@utwente.nl

Duc V. Le
University of Twente
Enschede, The Netherlands
v.d.le@utwente.nl

Nirvana Meratnia
University of Twente
Enschede, The Netherlands
n.meratnia@utwente.nl

Paul J.M. Havinga
University of Twente
Enschede, The Netherlands
p.j.m.havinga@utwente.nl

ABSTRACT

The most compelling reason to use unsupervised representation learning as a feature extraction method for effective animal activity recognition is the ability to learn from *unlabeled* data. Obtaining labeled data is tedious, labor-intensive, and costly, while it is much easier to obtain raw unlabeled data. In this paper, we compare three unsupervised representation learning techniques with three conventional feature extraction methods that are simple and have excellent performance. To investigate the effect of the size of both *labeled* and *unlabeled* parts of the dataset on the quality of the representations, we train the representations and classifier using various sample sizes. Furthermore, we evaluate the effect of depth in feature architectures on the performance of the representation learning techniques. All evaluations are performed on two animal datasets that are diverse in terms of species, subjects, sensor-orientations, and sensor-positions. We demonstrate that unsupervised representation learning techniques approach and, in some cases, outperform engineered features in animal activity recognition.

CCS CONCEPTS

• **Theory of computation → Unsupervised learning and clustering**; • **Computing methodologies → Dimensionality reduction and manifold learning**; *Cross-validation*; • **Applied computing** → Consumer health; Health care information systems; Agriculture.

KEYWORDS

Machine Learning, Feature Learning, Representation Learning, Deep Learning, Artificial Intelligence, Animal Monitoring, Activity Recognition, Accelerometer, IMU, Sensor Orientation, Embedded Systems

MileTS '20, August 24th, 2020, San Diego, California, USA

ACM ISBN 0…$15.00
https://doi.org/0

ACM Reference Format:
Jacob W. Kamminga, Duc V. Le, Nirvana Meratnia, and Paul J.M. Havinga. 2020. Towards Deep Unsupervised Representation Learning from Accelerometer Time Series for Animal Activity Recognition. In *MileTS '20: 6th KDD Workshop on Mining and Learning from Time Series, August 24th, 2020, San Diego, California, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/0

1 INTRODUCTION

The activity of animals is a rich source of information that not only provides insights into their life and well-being but also their environment [5, 8, 9, 26, 27]. Animal activity recognition (AAR) is a relatively new field of research that supports various goals, including the conservation of endangered species and livestock's well-being. Over the last decades, the advent of small, lightweight, and low-power electronics have made it possible to attach unobtrusive sensors to animals that can measure a wide range of aspects such as location, temperature, and activity. These aspects are highly informative properties for numerous application domains, including wildlife monitoring [37], anti-poaching [17], and livestock management [24].

A significant challenge in AAR is the acquisition of labeled data. It is difficult to find and observe (wild) animals. Therefore, it is often hard to collect videos (ground truth data) of collared animals. Moreover, the synchronization of videos with sensor data and manual annotation is laborious, expensive, and tedious. Because of this, the proportion of labeled movement data collected in the wild is usually small. Activity datasets are naturally imbalanced because some activities are either not frequently performed, or challenging to observe. It is a lot easier to collect unlabeled data than labeled data. Therefore, we investigate unsupervised representation learning from AAR time series data.

Effectively, less labeled data is required when the data representation is more discriminative. Researchers in the field of human activity recognition (HAR) have investigated unsupervised feature learning with small datasets and demonstrated promising performances [10, 22, 23]. However, HAR and AAR are different. The most significant differences between HAR and AAR are the type of activities, the sensor location, and movement patterns between the activities in humans and quadruped animals. In animal monitoring applications, the sensors are mostly worn by animals on collars around the neck, and the sensor orientation is not fixed. Furthermore, AAR is often used in remote sensing applications, and the

tags need to be small and lightweight. Therefore, the energy requirements are strict, and the AAR system must be resource-efficient in terms of energy, computation, and memory. There are only a few recent papers on the subject of deep learning (DL) in relation to AAR [6, 30, 35]. To the best of our knowledge, DL, and more specifically, unsupervised representation learning using time series inertial measurement unit (IMU) data, has not been researched for AAR so far.

In this paper, we focus on unsupervised representation learning to improve AAR that aims to automatically recognize the activity from motion data –on the animal– while the activities are performed (online). Specifically, we use time series motion data recorded through an accelerometer because this is a lightweight and energy-efficient sensor. We propose to use offline unsupervised learning on raw accelerometer time-series data to train a feature extraction method. The trained architecture can then be implemented on the animal tag along with an energy-efficient online classifier. Our primary goal is to compare and analyze the quality of unsupervised representation learning techniques and evaluate their effect on the performance of AAR. Specifically, our main objective is to compare learned representations, from unlabeled data, that are expressive, orientation-independent, and discriminate various activities. Lee et al. introduced convolutional deep belief network (CDBN) in [20] for efficient hierarchical feature detection in images. In a later work, the authors applied CDBN to audio data and empirically evaluated them on various audio classification tasks [21]. The authors showed that the learned features corresponded to phonemes and demonstrated that the feature representations learned from unlabeled audio data had an excellent performance for multiple audio classification tasks. We hypothesize that components in time series motion data are built up to a given activity, similar to how phonemes build-up to words. Therefore, we evaluated a CDBN for AAR task. We compare the CDBN with simple principal component analysis (PCA) and a deep net consisting of two stacked sparse auto encoders (SAEs). As a baseline, we compare the learned representations with conventional engineered time- and frequency-domain summary statistics. We assess the performance by training and testing a multi-class support vector machine (SVM) with the features derived using each extraction method. Furthermore, the quality of the feature extraction methods is affected by various factors. In this work, we investigated two: (i.) the size of the labeled and unlabeled dataset, (ii.) the number of layers (depth) of the representation learning architectures.

2 METHODOLOGY

The overall methodology of the experiments is shown in Figure 1.

First, raw annotated data from the accelerometer was pre-processed and transformed into an orientation independent 3D acceleration vector. The dataset acquisition and preprocessing are discussed in more detail in Section 3. Second, various representations were extracted from the 3D vector. We discuss each representation architecture in more detail in Section 2.1. The different representations were used separately and subsequently to describe the data used to train and test a SVM classifier. To minimize the influence of the classifier, we use the same type of classifier throughout our evaluations. We used SVM because this classifier is generally robust to the higher

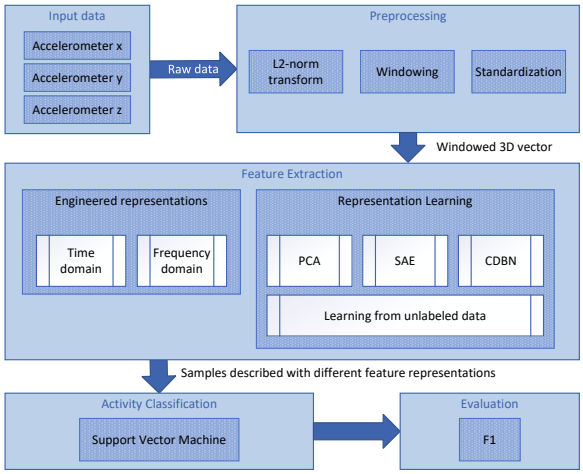

Figure 1: Experiment methodology

dimensionality of the data [19]. The SVM was implemented with the LibSVM library (version 3.23) [7] using a radial basis function. During each experiment iteration a grid search was performed to find optimal values for the parameters $C \in \{2^{-5}, 2^{-3}, 1, 2^3, 2^5\}$ and $\gamma \in \{2^{-4}, 2^{-2}, 2^{-1}, 1, 2^2, 2^4\}$. We sampled our dataset with increasing sample sizes utilizing stratified random sampling and random under-sampling. The experiments were repeated multiple times for each subsample (batch) of training data to take variability into account. For each sample size, the trained representation architecture was used to extract features from the training data and train the SVM as a supervised learning algorithm. Finally, the performance of the SVM was assessed using the test data. We evaluated the performance using the F_1 measure. To address heterogeneity, we used leave-one-subject-out cross-validation. For each fold, all data from one subject was only used as test data, while data from the remaining subjects was used as training data. We did not use the test data for representation learning because we assumed that this would not be available when AAR is deployed on unseen animals. The procedure above was repeated for various sizes of labeled and unlabeled data.

2.1 Feature Extraction Methods

In this section, we briefly describe each feature extraction method and the settings that were used in their implementation.

2.1.1 Engineered Representations. The shallow statistical features do not utilize the unlabeled part of the data because they cannot be learned. For each window of both training and test data, the features described in Table 2 (Supplement A.1) were calculated. A selection was made that consisted of 21 time and frequency-domain features that are typically used for activity recognition [2, 17, 32, 38].

2.1.2 Principal component analysis (PCA). Principal component analysis (PCA) [31] is a commonly used and well-established dimensionality reduction and decorrelation technique that is often used in activity recognition [28]. PCA is a basic form of shallow representation learning since it automatically discovers a compact and descriptive subset of representation from the raw data without

relying on expert domain knowledge [28]. The 21 most significant components identified in the training data were retained and used to describe the training and testing data.

2.1.3 Deep net of Sparse Auto Encoders. An SAE is an unsupervised learning algorithm that is trained using backpropagation. SAEs learn to compress data from the input layer into a shortcode and then decompress it into something that resembles the original data. The encoder part of the SAE is trained with unlabeled training data during the training phase. The training and test sets are given as inputs to the encoder part of the network and transformed with the learned code $h_{W,b}(x)$ to obtain the feature vector used for classification. We utilized a deep net that consisted of two sequentially connected SAEs. The deep net was trained using greedy layer-wise training [4]. Please see Supplement A.2 for implementation details.

2.1.4 Convolutional deep belief network (CDBN). CDBNs are generative probabilistic models composed of one visible layer and multiple hidden layers [20]. Each hidden neuron learns a statistical relationship between the neurons in the lower layer; the higher layer representations usually become more complex [21]. The CDBN was implemented by adopting the code that was available from [20] and [21] so that it could be used with our activity datasets. The CDBN was trained using greedy layer-wise training [4]. Please see Supplement A.3 for implementation details.

2.2 Effect of Size of the Labeled and Unlabeled Data

In machine learning, the use of more training data generally improves the performance of a learning task such as activity recognition (AR) [1, 34]. We evaluated the effect of the size of both *labeled* and *unlabeled* data on the classifier's performance for each feature extraction method.

*2.2.1 Effect of **Labeled** Data size on Representation Quality.* We investigated the quality of the representations using different amounts of labeled data. Figure 3 shows the methodology of this experiment. During each fold, 100 % of available training data was used for representation learning. In theory, good representations allow a classifier to separate activities with only a small amount of labeled training data.

*2.2.2 Effect of **Unlabeled** Data size on Representation Quality.* To verify that the representation learning techniques are learning and improving as more data became available, we fixed the size of the labeled dataset and analyzed the effect of the amount of available *unlabeled* data on the quality of the representation learning. Figure 4 shows the methodology of this experiment. We chose a labeled dataset size where the F_1 performance of all architectures was above 50 % in the previous experiments. We fixed the amount of labeled data to 50 samples per class and utilized the rest of the data in incremental subset sizes as unlabeled data. This size is not too big so that the effect of additional unlabeled size would become insignificant against the large labeled dataset. Not all architectures were able to deal with the imbalance with this sample size of labeled data. Therefore, we used random under-sampling for the smaller labeled dataset sizes.

2.3 Effect of Depth

Deep architectures promote the re-use of features and can potentially lead to increasingly more abstract features at higher layers [3]. To analyze to what degree the additional depth in the representation learning techniques contributes to the AAR performance, we repeated the experiments using features derived from intermittent layers of the representation learning techniques. The SAE deep net and CDBN are both hierarchical DL representation architectures with two layers. In each experiment, we trained the SVM classifier three times using either the representation from the first (pooling) layer activations $L1$, the second (pooling) layer $L2$, and the concatenation of both layer activations $L1 + L2$.

3 DATA DESCRIPTION AND PREPROCESSING

We used our open-access real-world datasets from goats [13, 15] and horses [16, 18] comprising multiple subjects, diverse sensor-orientations, and various activities. The datasets comprise a diverse set of animals, e.g., some goats were from a different subspecies than others, and our horse dataset contains data from large horses and smaller ponies from different breeds. Furthermore, we used a single sensor of which the position around the neck and orientation was not fixed [17]. A more detailed description of the datasets is attached as Supplement C. Because all activities were not exercised by all subjects, we used data from 6 subjects and 5 classes so that the experiments could be evaluated through leave-one-subject-out cross-validation.

3.1 Pre-processing

We used a low dimensional $(1 \times n)$ vector as input for the representation learning frameworks, where n is the size of the window. The magnitude of the 3D vector (ℓ^2-norm) of accelerometer data is theoretically orientation-independent [17]. This vector is defined as:

$$M(t) = \sqrt{s_x(t)^2 + s_y(t)^2 + s_z(t)^2}, \tag{1}$$

where, s_x, s_y, and s_z are the three respective axes of the sensor. The transformation effectively reduces the input dimensionality with a factor of 3, reducing resource requirements. The activity time series were segmented with a window size of 2 seconds, and 50 % overlap. We scaled the data through a Z-transformation, obtaining a standard score with zero-mean and unit variance.

4 EVALUATION

In this section, we present the evaluation results. All analysis was done in Matlab [25]. The F_1 measure was used as the evaluation metric.

4.1 Varying Size of the Labeled Dataset

Figures 2a and 2b show the experiment results of each feature extraction method for the goat and horse dataset, respectively. The percentages on the x-axis between the two figures vary slightly because we used leave-one-subject-out cross-validation and random under-sampling. When using leave-one-subject-out cross-validation, the size of the dataset changes per fold. The number of samples between the brackets represent the average number of samples over all folds per percentage.

With small dataset sizes, the performance of all representations increased rapidly and flattened out as the labeled dataset size becomes approximately 1000 samples. The performance did not change much between 1 % and 19 %, and these were therefore not included in the figures for clarity. For both datasets, the sampled dataset ranging from 0.01 % to 1 % size had a balance score of 1 due to random under-sampling. The balance score is discussed in Supplement C. The larger sample sizes had average balance scores of 0.71 and 0.83 for the goat and horse datasets, respectively. Most likely the imbalance is the cause for the dip in performance and high standard deviation for the SAE and PCA representations at the 18 % - 76 % sample sizes in the goat dataset.

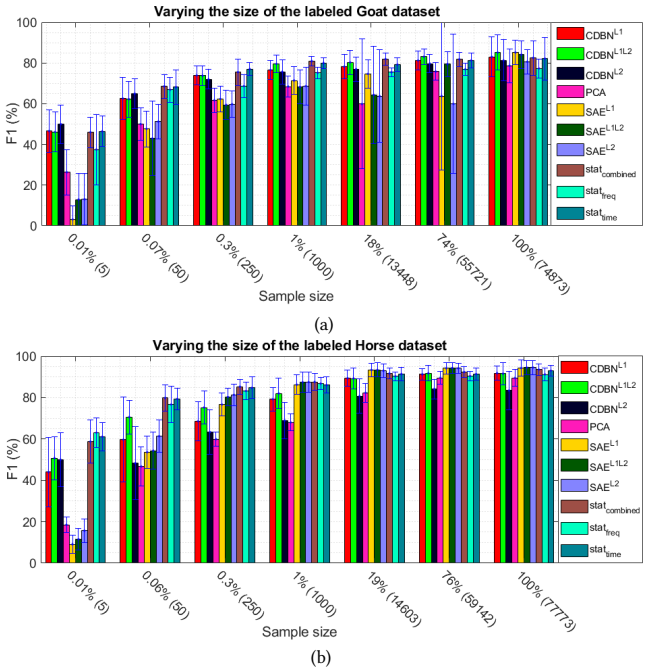

Figure 2: AAR performance using different sizes of labeled data. (a) Goats (b) Horses

Legend: **CDBN**: convolutional deep belief network, **PCA**: principal component analysis, **SAE**: sparse auto encoder, $\mathbf{stat_{freq}}$: spectral features, $\mathbf{stat_{time}}$: temporal features, $\mathbf{stat_{combined}}$: spectral and temporal features, **L1**: first layer, **L2**: second layer, **L1L2**: first and second layer concatenated

4.2 Varying Size of the Unlabeled Dataset

Tables 1a and 1b, show the experiment results for the Goat and Horse datasets, respectively. The results show that the overall performance of the learned representations gradually increases with the size of the unlabeled dataset. The performance over the sample size increased mostly in the goat dataset and more slowly in the horse dataset. The results are further discussed in the following section.

5 DISCUSSION

The results in Figure 2 show that the CDBN is more robust to smaller dataset sizes and a higher imbalance in the dataset than the other representation learning architectures. The 2nd layer representation performed better than using only the 1st layer for smaller sizes of the goat dataset. In the horse dataset and larger sizes of the labeled goat dataset, the concatenation of both layers ($L1 + L2$) performed the best. Thus, even when the second layer by itself did not perform better than using only the first layer, the concatenation often resulted in better representations. The CDBN representations outperformed statistical features for the goat dataset and performed slightly worse with the horse dataset. Although statistical features generally outperformed the learned representations in smaller labeled dataset samples, our results show that CDBN can automatically learn representations that are almost as good as engineered features that rely on domain knowledge. With larger labeled dataset sample sizes, representation learning techniques slightly outperformed summary statistics.

Table 1: AAR performance using different sizes of unlabeled data. Each second column denotes the standard deviation σ over multiple folds and batches. (a) Goats (b) Horses

(a)

sample size	0.3% (250)		1% (1000)		18% (13398)		74% (55510)		100% (74623)	
representation	F_1	σ	F_1	σ	F_1	σ	F_1	σ	F_1	σ
$CDBN^{L1}$	67,19	6,18	66,30	5,66	72,56	9,63	73,56	5,72	**79,55**	10,87
$CDBN^{L1L2}$	**68,10**	4,96	**67,02**	5,20	**76,28**	8,51	**75,40**	5,56	79,46	11,33
$CDBN^{L2}$	53,20	5,07	47,19	17,04	74,19	5,26	72,29	7,08	77,80	10,95
PCA	59,23	5,72	62,14	6,50	60,57	5,39	61,65	5,57	62,03	0,18
SAE^{L1}	65,62	6,20	62,07	8,14	61,79	5,49	60,43	7,47	67,89	6,23
SAE^{L1L2}	62,63	7,92	61,13	8,63	59,40	6,63	59,61	7,09	65,31	8,56
SAE^{L2}	57,90	6,53	60,90	8,66	59,57	8,48	60,71	7,55	64,34	7,64

(b)

sample size	0.3% (250)		1% (1000)		19% (14553)		76% (58935)		100% (77523)	
representation	F_1	σ	F_1	σ	F_1	σ	F_1	σ	F_1	σ
$CDBN^{L1}$	65,17	4,40	66,78	5,71	72,14	5,11	72,62	5,69	70,01	6,86
$CDBN^{L1L2}$	76,45	6,81	76,16	5,90	75,30	6,12	76,87	5,48	78,00	7,90
$CDBN^{L2}$	60,59	10,83	61,15	10,20	62,43	7,64	63,24	9,98	65,79	9,57
PCA	59,18	4,78	59,13	4,48	58,96	3,62	58,99	3,68	56,30	5,06
SAE^{L1}	78,47	4,83	77,63	5,71	75,14	4,53	76,39	6,70	74,19	5,75
SAE^{L1L2}	**80,01**	5,09	**78,74**	5,56	76,40	4,91	80,05	4,68	79,77	6,90
SAE^{L2}	76,45	4,70	76,04	4,71	**77,26**	4,63	**81,50**	5,24	**80,86**	9,71

Tables 1a and 1b show that when increasing the amount of unlabeled data, the rate of improvement was the largest for the CDBN and the smallest for PCA representation. Although Table 1a shows that the standard deviation for the CDBN increased with more unlabeled data, the worst case result for CDBN at 100 % (70.49 %) was still the best compared to the other representations.

We observe a significant difference between the relative performances of CDBN and SAE between Table 1a and Table 1b. On the Goat dataset, CDBN outperforms SAE, while on the Horse dataset, the opposite is true. We think that these effects may be caused by the fact that the number of used sensors was different for the datasets. The goat dataset used six sensors around the neck, and the horse dataset one. Consequently, increasing the sample size of the horse data also increases the relative diversity of the data while the diversity of the goat dataset increases less because the same activities were recorded with more sensors. SAEs perform well on smaller datasets that are sparse and more diverse, while CDBN perform well on dense and large datasets. Furthermore, SAEs are more capable to to deal with overfitting because of the sparsity constraint. These properties may explain the reverse order of SAE and CDBN performance improvement between the two datasets.

Furthermore, Table 1b shows that the performance of PCA slightly decreases as more unlabeled data is available while the opposite occurs with the goat dataset. Because PCA is defined as a linear transformation, the increased diversity of the data distribution may have a negative impact on its performance and vice versa.

The extra layers did not seem to contribute much when only a small amount of unlabeled data was available. For the horse dataset, the concatenation of both CDBN layers gained in performance as more unlabeled data became available. The PCA representations did not benefit from more unlabeled data and seemed to be restricted in performance due to the size of the labeled dataset sample. Although the performance increased very slowly, deep representations for AAR increase in performance as more unlabeled data becomes available. The performance of the learned representations is excellent, especially given that both architectures were only trained with unlabeled data from limited sensor modalities without fine-tuning to the datasets. We expect fine-tuning using a small amount of the labeled data improves the representations even more.

The results presented in this paper are average F_1 performances for all activity classes in the datasets. In the following, we briefly discuss the classification performance of the individual activities using the different representations. For the goat dataset, *stationary* and *walking* overall obtained the best F_1 score for all representations and *eating* generally the lowest. Stationary is defined as no or little motion and is both easy to classify, and it is the majority class comprising 41 % of the goat dataset. For the horse dataset, *trotting* generally had the best F_1 score and *eating* the lowest. Walking and trotting are periodic and relatively simple activities that are easier to classify. Eating is a complex and more subtle activity that includes eating different types of food such as fresh grass and hey and is, therefore, harder to classify.

6 CONCLUSION

We demonstrated that unsupervised representation learning techniques approaches and, in some cases, outperforms engineered features in AAR. When fixing the amount of labeled data and varying the size of the unlabeled dataset, our results show that the CDBN architecture benefited the most from an increasing amount of unlabeled data, especially in the more diverse and imbalanced dataset. The most significant difference between the CDBN and other architectures is that it is a generative model. Based on our findings, we believe that generative models, such as the CDBN, are compelling research directions in unsupervised representation learning for AR. Our results show that concatenating the 1st and 2nd layer representations often results in better classification performance. Our results indicate that deep representations provide more robustness to the imbalance in smaller labeled datasets. An important lesson is that the effect of class imbalance in datasets is not only crucial in labeled data but also affects the performance of unsupervised representation learning, especially as the unlabeled dataset grows. Therefore, class imbalance in unlabeled data for representation learning is an important research problem. Although 'simple' – engineered – time-domain features match or even outperform *unsupervised* representation learning algorithms at this point, we believe that representation learning outperforms engineered representations as the task becomes increasingly difficult, we find better design solutions, and the unlabeled dataset size grows. The performance of CDBN motivates further analysis with generative models such as generative adversarial network (GAN) and variational auto encoder (VAE).

ACKNOWLEDGMENTS

This research was supported by the Smart Parks Project, which involves the University of Twente, Wageningen University & Research, ASTRON Dwingeloo, and Leiden University. The Smart Parks Project is funded by the Netherlands Organisation for Scientific Research (NWO).

REFERENCES

[1] Michele Banko and Eric Brill. 2001. Mitigating the Paucity-of-Data Problem: Exploring the Effect of Training Corpus Size on Classifier Performance for Natural Language Processing. In HLT '01 Proceedings of the first international conference on Human language technology research. Association for Computational Linguistics, San Diego, 1–5. https://doi.org/10.3115/1072133.1072204

[2] Ling Bao and Stephen S. Intille. 2004. Activity Recognition from User-Annotated Acceleration Data. In Pervasive Computing. Springer Berlin Heidelberg, Berlin, Heidelberg, 1 – 17. https://doi.org/10.1007/b96922

[3] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2012. Representation Learning: A Review and New Perspectives. CoRR 1206.5538, 1993 (2012), 1–30. https://doi.org/10.1145/1756006.1756025

[4] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. 2007. Greedy Layer-Wise Training of Deep Networks. In Advances in Neural Information Processing Systems 19. The MIT Press, Canada, 153–160. https://doi.org/10.7551/mitpress/7503.003.0024

[5] Greg Bishop-Hurley, Dave Henry, Daniel Smith, Ritaban Dutta, James Hills, Richard Rawnsley, Andrew Hellicar, Greg Timms, Ahsan Morshed, Ashfaqur Rahman, Claire D'Este, and Yanfeng Shu. 2014. An investigation of cow feeding behavior using motion sensors. In 2014 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings. IEEE, Montevideo, 1285–1290. https://doi.org/10.1109/I2MTC.2014.6860952

[6] Ella Browning, Mark Bolton, Ellie Owen, Akiko Shoji, Tim Guilford, and Robin Freeman. 2018. Predicting animal behaviour using deep learning: GPS data alone accurately predict diving in seabirds. Methods in Ecology and Evolution 9, 3 (2018), 681–692. https://doi.org/10.1111/2041-210X.12926

[7] Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2 (2011), 27:1–27:27. Issue 3. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[8] Steven J. Cooke, Scott G. Hinch, Martin Wikelski, Russel D. Andrews, Louise J. Kuchel, Thomas G. Wolcott, and Patrick J. Butler. 2004. Biotelemetry: A mechanistic approach to ecology. Trends in Ecology and Evolution 19, 6 (2004), 334–343. https://doi.org/10.1016/j.tree.2004.04.003

[9] Jamali Firmat Banzi. 2014. A Sensor Based Anti-Poaching System in Tanzania National Parks. International Journal of Scientific and Research Publications 4, 4 (2014), 1–7.

[10] Henry Friday, Ying Wah, Mohammed Ali Al-garadi, Uzoma Rita, Henry Friday Nweke, Ying Wah Teh, Mohammed Ali Al-garadi, Uzoma Rita Alo, Henry Friday, Ying Wah, Mohammed Ali Al-garadi, and Uzoma Rita. 2018. Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges. Expert Systems with Applications 105 (2018), 233–261. https://doi.org/10.1016/j.eswa.2018.03.056

[11] LLC Gulf Coast Data Concepts. 2019. Human Activity Monitor: HAM. online. http://www.gcdataconcepts.com/ham.html

[12] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li Shiuan Peh, and Daniel Rubenstein. 2002. Energy-efficient computing for wildlife tracking. ACM SIGOPS Operating Systems Review 36, 5 (2002), 96. https://doi.org/10.1145/635508.605408

[13] Jacob W. Kamminga. 2018. Dataset: Multi Sensor-Orientation Movement Data of Goats. https://doi.org/10.17026/dans-xhn-bsfb

[14] Jacob W. Kamminga. 2019. Dataset: Horsing Around – A Dataset Comprising Horse Movement. 4TU.Centre for Research Data. https://doi.org/10.4121/uuid:2e08745c-4178-4183-8551-f248c992cb14

[15] Jacob W. Kamminga, Helena C. Bisby, Duc V. Le, Nirvana Meratnia, and Paul J. M. Havinga. 2017. Generic Online Animal Activity Recognition on Collar Tags. In Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International

Symposium on Wearable Computers - UbiComp '17. ACM Press, New York, NY, 597–606. https://doi.org/10.1145/3123024.3124407
[16] Jacob W. Kamminga, Lara M. Janßen, Nirvana Meratnia, and Paul J. M. Havinga. 2019. Horsing Around—A Dataset Comprising Horse Movement. Data 4, 4 (9 2019), 131. https://doi.org/10.3390/data4040131
[17] Jacob Wilhelm Kamminga, Duc V. Le, Jan Pieter Meijers, Helena Bisby, Nirvana Meratnia, and Paul J.M. Havinga. 2018. Robust Sensor-Orientation-Independent Feature Selection for Animal Activity Recognition on Collar Tags. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies IMWUT 2, 1 (2018), 1–27. https://doi.org/10.1145/3191747
[18] Jacob W. Kamminga, Nirvana Meratnia, and Paul J.M. Havinga. 2019. Dataset: Horse movement data and analysis of its potential for activity recognition. In DATA 2019 - Proceedings of the 2nd ACM Workshop on Data Acquisition To Analysis, Part of SenSys 2019. ACM, New York, 22–25. https://doi.org/10.1145/3359427.3361908
[19] S B Kotsiantis. 2007. Supervised Machine Learning : A Review of Classification Techniques. Informatica, An International Journal of Computing and Informatics 3176, 31 (2007), 249–268.
[20] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y. Ng. 2009. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09. ACM, Montreal, 1–8. https://doi.org/10.1145/1553374.1553453
[21] Honglak Lee, Y Largman, Peter Pham, and AY Ng. 2009. Unsupervised feature learning for audio classification using convolutional deep belief networks. In Proceedings of the 22nd International Conference on Neural Information Processing Systems. ACM, Vancouver, British Columbia, Canada, 1096–1104.
[22] Frederic Li, Kimiaki Shirahama, Muhammad Adeel Nisar, Lukas Koping, and Marcin Grzegorzek. 2018. Comparison of feature learning methods for human activity recognition using wearable sensors. Sensors (Switzerland) 18, 2 (2018), 1–22. https://doi.org/10.3390/s18020679
[23] Yongmou Li, Dianxi Shi, Bo Ding, and Dongbo Liu. 2014. Unsupervised Feature Learning for Human Activity Recognition Using Smartphone Sensors. Expert Systems with Applications 41, 14 (2014), 6067–6074. https://doi.org/10.1016/j.eswa.2014.04.037
[24] Kees Lokhorst. 2018. Smart Dairy Farming. Hogeschool van Hall Larenstein, Leeuwarden. 106 pages.
[25] MATLAB. 2018. version 9.5.0.944444 (R2018b). The MathWorks Inc., Natick, Massachusetts.
[26] Ran Nathan. 2008. An emerging movement ecology paradigm. Proceedings of the National Academy of Sciences of the United States of America 105, 49 (2008), 19050–19051. https://doi.org/10.1073/pnas.0808918105
[27] Julie K Petersen. 2001. Understanding Technologies Surveillance Spy Devices, Their Origins and Applications. CRC Press, New York, New York, USA.
[28] Thomas Plötz, Nils Y. Hammerla, and Patrick Olivier. 2011. Feature learning for activity recognition in ubiquitous computing. In IJCAI International Joint Conference on Artificial Intelligence. ACM, Barcelona, 1729–1734. https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-290
[29] Simone Romano. 2016. A general measure of data-set imbalance. Cross Validated. https://stats.stackexchange.com/q/239982
[30] Mohammad Sadegh, Anh Nguyen, Margaret Kosmala, Ali Alexandra Swanson, Meredith S Palmer, Mohammed Sadegh Norouzzadeh, Anh Nguyen, Margaret Kosmala, Ali Alexandra Swanson, Meredith S Palmer, Craig Packer, and Jeff Clune. 2018. Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. Proceedings of the National Academy of Sciences 115, 25 (2018), E5716–E5725. https://doi.org/10.1073/pnas.1719367115
[31] Jonathon Shlens. 2014. A Tutorial on Principal Component Analysis. CoRR 1404.1100 (2014), 13. https://doi.org/10.1.1.115.3503
[32] Muhammad Shoaib, Stephan Bosch, Ozlem Incel, Hans Scholten, and Paul Havinga. 2015. A Survey of Online Activity Recognition Using Mobile Phones. Sensors 15, 1 (2015), 2059–2085. https://doi.org/10.3390/s150102059
[33] Ian F. Spellerberg and Peter J. Fedor. 2003. A tribute to Claude-Shannon (1916-2001) and a plea for more rigorous use of species richness, species diversity and the 'Shannon-Wiener' Index. Global Ecology and Biogeography 12, 3 (2003), 177–179. https://doi.org/10.1046/j.1466-822X.2003.00015.x
[34] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. 2017. Revisiting Unreasonable Effectiveness of Data in Deep Learning Era. Proceedings of the IEEE International Conference on Computer Vision 2017-Octob (2017), 843–852. https://doi.org/10.1109/ICCV.2017.97
[35] Michael A. Tabak, Mohammad S. Norouzzadeh, David W. Wolfson, Steven J. Sweeney, Kurt C. Vercauteren, Nathan P. Snow, Joseph M. Halseth, Paul A. Di Salvo, Jesse S. Lewis, Michael D. White, Ben Teton, James C. Beasley, Peter E. Schlichting, Raoul K. Boughton, Bethany Wight, Eric S. Newkirk, Jacob S. Ivan, Eric A. Odell, Ryan K. Brook, Paul M. Lukacs, Anna K. Moeller, Elizabeth G. Mandeville, Jeff Clune, Ryan S. Miller, Steven J. Sweeney, Kurt C. Vercauteren, Nathan P. Snow, Paul A Di, Salvo Jesse, S Lewis Michael, D White Ben, James C. Beasley, Peter E. Schlichting, Raoul K. Boughton, Bethany Wight, Eric S. Newkirk, Jacob S. Ivan, Eric A. Odell, Ryan K. Brook, Paul M. Lukacs, Anna K. Moeller, Elizabeth G. Mandeville, Jeff Clune, Ryan S. Miller, and Ryan S. Miller. 2018. Machine learning to classify animal species in camera trap images: Applications in ecology. Methods in Ecology and Evolution 2019, September 2018 (2018), 585–590. https://doi.org/10.1111/2041-210X.13120
[36] Inertia Technology. 2017. ProMove mini. online. http://inertia-technology.com/
[37] Christopher C. Wilmers, Barry Nickel, Caleb M. Bryce, Justine A. Smith, Rachel E. Wheat, Veronica Yovovich, and M. Hebblewhite. 2015. The golden age of bio-logging: How animal-borne sensors are advancing the frontiers of ecology. Ecology 96, 7 (2015), 1741–1753. https://doi.org/10.1890/14-1401.1
[38] Mi Zhang and Alexander a Sawchuk. 2011. A feature selection-based framework for human activity recognition using wearable multimodal sensors. In Proceedings of the 6th International Conference on Body Area Networks. ACM, Beijing, 92–98. https://doi.org/10.4108/icst.bodynets.2011.247018

A FEATURE EXTRACTION METHODS

A.1 Engineered representations

Table 2 shows the used summary statistics. The features can be categorized as time and frequency domain features.

Table 2: Summary statistics that were calculated for each window of data

Domain	Feature	Description
Time	Maximum	Maximum value
	Minimum	Minimum value
	Mean	Average value
	Standard deviation	Measure of dispersion
	Median	Median value
	25^{th} percentile	The value below which 25 % of the observations are found
	75^{th} percentile	The value below which 75 % of the observations are found
	Mean low pass filtered signal	Mean value of DC components
	Mean rectified high pass filtered signal	Mean value of rectified AC components
	Skewness of the signal	The degree of asymmetry of the signal distribution
	Kurtosis	The degree of 'peakedness' of the signal distribution
	Zero crossing rate	Number of zero crossings per second
Frequency	Principal frequency	Frequency component that has the greatest magnitude
	Spectral energy	The sum of the squared discrete FFT component magnitudes
	Frequency entropy	Measure of the distribution of frequency components
	Frequency magnitudes	Magnitude of first six components of FFT analysis

A.2 Deep net of Sparse Auto Encoders

We implemented the SAE class from the Matlab Deep Learning Toolbox [25]. The number of neurons in the first and second layers was 150 and 100, respectively. The number of hidden neurons was initially determined by a grid search over the training data. It was fixed during the experiments to keep the parameter search space feasible. Both layers were trained with 100 epochs. The activation function σ was set to the logistic sigmoid function. All other parameters were found using a grid search during each experiment iteration. The sum of mean squared reconstruction errors from both layers was used as the selection criteria for the parameters. The regularization parameter was varied between $\lambda \in [0.1, \ldots, 2]\text{e}{-3}$ with steps of 0.5e−3. The sparsity parameter was varied between $\rho \in [0.05, \ldots, 0.4]$ with steps of 0.05. The sparsity coefficient was varied between $\beta \in [1, 3, 9]$. Because parameter optimization had to be performed for each experiment with a new subset of training data, the computational overhead for the grid-search with larger dataset sizes became problematic due to long run times. Therefore, for dataset sizes > 40 000 samples we sampled the dataset 4 times with 12 000 samples, of which 20 % was used as validation to compute the mean squared reconstruction error for each set of parameters.

A.3 Convolutional deep belief network

The CDBN was implemented by adopting the code that was available from [20] and [21] so that it could be used with our activity datasets. Parameter tuning was performed once using the unlabeled training data. The reconstruction error of both layers was used as a selection criterion – no labeled information was used for parameter tuning. The number of hidden bases in the first and second layer was varied between $[50, 125, 200]$. The sparsity regularization parameter was varied between $\beta \in [0.01, 0.05, 0.1]$. For both layers, the filter length was varied between $N_W \in [3, 5, 7]$, and the pooling ratio was varied between $C \in [2, 4, 6, 8]$. After the grid search, we fixed the parameters to the following settings. For the goat dataset, we used 150 and 50 bases in the first and second layers, respectively. The filter length N_W was set to 10 and 3 for the first and second layer, respectively. β was set to 0.05; For the horse dataset we used 100 and 150 bases in the first and second layer, respectively. The filter length N_W was set to 15 and 3 for the first and second layer, respectively. β was set to 0.1 and C was set to 2 for both datasets.

B EXPERIMENT METHODOLOGY

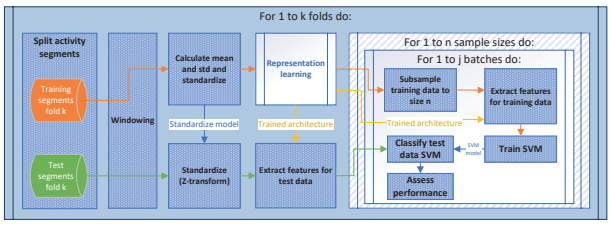

Figure 3: Overview of experiments where the size of the labeled data was varied. All available unlabeled data was used for representation learning.

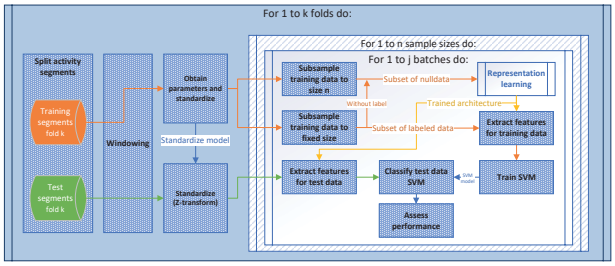

Figure 4: Overview of experiments where the size of the unlabeled data used for representation learning was varied. The size of the available labeled data used for training the SVM was fixed.

C DATASET DESCRIPTION

The imbalance in a dataset can be quantified using Shannon's diversity index and normalized [29, 33] as follows:

$$Balance = \frac{H}{\log k} = \frac{-\sum_{i=1}^{k}\left(\frac{c_i}{n}\log\frac{c_i}{n}\right)}{\log k} \quad (2)$$

, where H is the Shannon entropy, n is the total number of data samples, k the number of classes, and c_i the size of class i. A higher score denotes better balance. Overall, the class imbalances in our datasets described below were 0,94 versus 0,71.

C.1 Goat Movement Data

This dataset was collected at two farms over 5 days [13]. The dataset comprises data from 5 different goats that performed various activities. Two goats were from a different species than the other three. Thus the subjects are quite different from each other (mostly in size), and there is variability in the data. The ProMove-mini [36] sensor nodes from Inertia Technology were used, which contained a 3-axis accelerometer, gyroscope, and magnetometer and were sampled at 100 Hz. We used data from six sensors located at various positions and orientations around the goats' neck. We have studied the effect of sensor orientation in earlier work [17] and showed that robust AAR is possible with sensor-orientation-independent features from the neck. The sensor devices were always attached around the neck of the horses so that they could be worn without a saddle or halter. Furthermore, this location is often used in studies that monitor wildlife such as zebra [12] which increases the usability of our datasets for research related to other animals. The activities that were used in this research are listed in Table 3. The composition of the dataset is shown in Table 4. The observed goats mostly spend their time stationary or eating, and the dataset is imbalanced.

Table 3: Observed goat activities

Activity	Description
Stationary	Lying on the ground or standing still, occasionally moving the head or stepping very slowly.
Walking	The goat puts one foot down at the time. The pace of walking varies from very slowly to nearly trotting.
Trotting	The phase between walking and running. One front foot and its opposite hind foot come down at the same time. Trotting at different speeds but always 2 beat gait.
Running	One hind leg strikes the ground first, and then the other hind leg and one foreleg come down together, finally the other foreleg strikes the ground. This movement creates a three-beat rhythm.
Eating	Pulling fresh grass out of the ground, eating hay from a pile or twigs/grains on the ground.

Table 4: Composition of the goat dataset. Number of samples per subject

Subject / Activity	1	2	3	4	5	Total samples	Total fraction
Stationary	9879	10597	7275	3415	3199	34365	41%
Walking	4526	4737	2381	3339	1015	15998	19%
Trotting	100	526	120	154	145	1045	1%
Running	88	497	56	40	115	796	1%
Eating	12973	3811	8129	6781	152	31846	38%
Total	27566	20168	17961	13729	4626	84050	
Balance score	0,66	0,73	0,65	0,69	0,56	0,71	

C.2 Horse Movement Data

Movement data was collected at a riding stable over 7 days. The dataset comprises data from 6 individual horses that performed 13 different activities. Data was collected when the horses were being ridden and when they were able to move about the paddock freely. We used the Human Activity Monitor [11] sensor nodes from Gulf Coast Data Concepts, which contain a 3-axis accelerometer, gyroscope, and magnetometer. Identical to the goat dataset, the sensors were sampled at 100 Hz. A single sensor node was attached to the neck of the horses using a collar fabricated from hook and loop fastener. The activities that were observed during the day are listed in Table 6. The composition of the dataset is shown in Table 5. Because the horses were being ridden, this dataset is more balanced over the different types of gait than the goat dataset. This dataset has been made publicly available [14, 16, 18].

Table 5: Composition of the horse dataset. Number of samples per subject

Subject / Activity	1	2	3	4	5	6	Total samples	Total fraction
Standing	1750	1186	1244	347	341	245	5113	6%
Walking	11055	9642	5538	5239	4294	1677	37445	43%
Trotting	6423	7038	3402	3559	2673	1981	25076	29%
Running	1043	696	714	835	323	328	3939	4%
Eating	4331	5063	1951	1091	2496	1116	16048	18%
Total	24602	23625	12849	11071	10127	5347	87621	
Balance score	0,83	0,81	0,86	0,78	0,8	0,85	0,83	

Table 6: Observed horse activities

Activity	Description
Standing	Horse standing on 4 legs, no movement of head, standing still
Walking	The horse puts each foot down one at a time. Walking with and without rider on back.
Trotting	One front foot and its opposite hind foot come down at the same time, making a two-beat rhythm. Trotting at different speeds but always 2 beat gait. With and without rider on back.
Galloping	One hind leg strikes the ground first, and then the other hind leg and one foreleg come down together, finally the other foreleg strikes the ground. This movement creates a three-beat rhythm. With and without rider on back.
Eating	Head down in the grass, eating and slowly moving to get to new grass spots or head is up, chewing and eating food, usually eating hay or long grass.