Apply Machine Learning on Cattle Behavior Classification Using Accelerometer Data

Zhuqing Zhao

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Computer Engineering

Sook Shin Ha, Chair

Guoqiang Yu

Dong Sam Ha

March 30, 2022

Blacksburg, Virginia

Keywords: Machine Learning, Behavior classification, Accelerometer.

Apply Machine Learning on Cattle Behavior Classification Using Accelerometer Data

Zhuqing Zhao

(ABSTRACT)

We used a 50Hz sampling frequency to collect tri-axle acceleration from the cows. For the traditional Machine learning approach, we segmented the data to calculate features, selected the important features, and applied machine learning algorithms for classification. We compared the performance of various models and found a robust model with relatively low computation and high accuracy. For the deep learning approach, we designed an end-to-end trainable Convolutional Neural Networks (CNN) to predict activities for given segments, applied distillation, and quantization to reduce model size. In addition to the fixed window size approach, we used CNN to predict dense labels that each data point has an individual label, inspired by semantic segmentation. In this way, we could have a more precise measurement for the composition of activities. Summarily, physically monitoring the well-being of crowded animals is labor-intensive, so we proposed a solution for timely and efficient measuring of cattle's daily activities using wearable sensors and machine learning models.

Apply Machine Learning on Cattle Behavior Classification Using Accelerometer Data

Zhuqing Zhao

(GENERAL AUDIENCE ABSTRACT)

Animal agriculture has intensified over the past several decades, and animals are managed increasingly as large groups. This group-based management has significantly increased productivity. However, animals are often located remotely on large expanses of pasture, which makes continuous monitoring of daily activities to assess animal health and well-being labor-intensive and challenging [37]. Remote monitoring of animal activities with wireless sensor nodes integrated with machine learning algorithms is a promising solution. The machine learning models will predict the activities of given accelerometer segments, and the predicted result will be uploaded to the cloud. The challenges would be the limitation in power consumption and computation. To propose a precise measurement of individual cattle in the herd, we experimented with several different types of machine learning methods with different advantages and drawbacks in performance and efficiency.

Acknowledgments

I want to thank Dr. Sook Shin Ha for all the time and help related to this project and ideas. I really appreciated the time she spent so that I could refine my thoughts and better organize my future works. It is especially valuable when the motivation is simple kindness for a student's future.

I want to thank Dr. Dong Sam Ha for his advice and in-depth thought on general problem-solving skills. Discussing the concepts helped me organize my thought and clarify my knowledge. I enjoy working with him :). It is precious that our working environment is a delight and the communication is efficient.

I want to thank Dr.Guoqiang Yu for his sincere advice and help. Many of the advice is very helpful!

I also want to thank my teammate Abhishek, Morgan, and Ruizhe for their help with this project; my friends Saleh and Jinhua that provide help and advice; Zhengming, zhuzhu(guinea pig), and hengheng (guinea pig) for happiness.

Contents

List of Figures

List of Tables

List of Abbreviations

AP Average Precision

AUC Area Under the Curve

cGAN conditional Generative Adversarial Network

CNN Convolutional Neural Network

FN False-Negative

FP False-Positive

HGBDT Histogram-based Gradient Boosted Decision Trees

k-NN k-Nearest Neighbor

ML Machine Learning

PCA Principal Component Analysis

PR Precision-Recall

RBF Radial Basis Function

RF Random Forest

RFE Recursive Feature Elimination

ROC Receiver Operating Characteristic

SVM Support Vector Machine

Chapter 1

Introduction

This work presents an application of machine learning algorithms to identify cows' behaviors from 3-axis acceleration data. An accelerometer is mounted on a halter of a cow along with a camera to record videos for labeling. We collected acceleration data for 85 hours in total from 5 different cows on a real farm. We obtained a new set of 52 features based on the characteristic of activities in addition to commonly used statistical features. Then we eliminated 22 redundant and insignificant features using the Recursive Feature Elimination (RFE) with negligible impact to the balanced accuracy while reducing the training time. With the remaining 30 features, we applied four classification algorithms: Support Vector Machine (SVM), k-Nearest Neighbor (k-NN), Random Forest (RF), and Histogram-based Gradient Boosted Decision Trees (HGBDT). Among the four algorithms, HGBDT achieves the highest accuracy. The recall value of HGBDT is 84.37%for the 'standing', 99.45% for the 'grazing', 96.15% for the 'walking', 89.99% for the 'lying', and 99.11% for the 'ruminating'. This type of approach requires fewer data to train models, but the performance is highly dependent on the hand-crafted features.

In addition to the traditional machine learning approach, we also applied Deep Neural Network that is end-to-end trainable to solve the problem in a different aspect. Pattern recognition in computer vision could be adopted in our application because the image is a form of multi-dimension signals. The commonly used convolutional operations could be applied to transform the signals to desired feature maps with learned kernels so classifications and

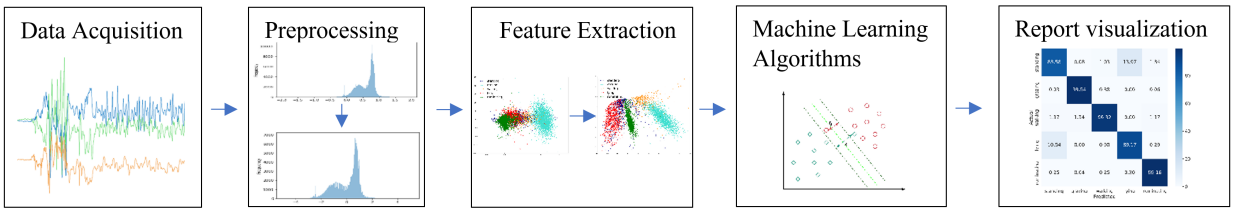


Figure 1.1: The workflow of the activity recognition, including data acquisition, preprocessing, feature extraction, machine learning algorithms, and report visualization.

other tasks could be performed. Thus, we used the Convolutional Neural Network to process a segment of raw motion data, extract the features of various activities, and predict the activity. Deep learning provides more flexibility but needs more data to train and usually requires more computation. To reduce model size and computation, we applied distilled learning and quantization. To address the fixed window size problem, we predict the activity composition (distribution) of the segment using U-Net inspired structure, so the calculated time duration is precise to individual data points.

Chapter 2

Review of Literature

2.1 Feature Extraction Based Classification

2.1.1 Human Activity Classification

In activity recognition, the idea is to use MEMS sensors for data collection and a Microcontroller for activity classification. Areas such as automated monitoring of subjects using wearable and off-body sensor-based devices on human activities recognition [7] have been explored in recent years. They used acceleration data with different window sizes to handcraft features as classification inputs. Some used SVM, kNN, Naive Bayes, Decision Trees, and Hidden Markova model, and others used Neural Networks: MLP, Convolutional Neural Network, Recurrent Neural Network, and others. The ideas behind are comparable to animal behavior classification: using acceleration to reveal the different characteristics of activities and classify behaviors based on the extracted variations. However, animal behavior classification is more challenging, because they behave more inconsistently. In nature, the animals have more interactions and more spread out random patterns.

2.1.2 Animal Behavior Classification

The essential concepts to measuring daily activities are behavior analysis and activity recognition. In behavior analysis, the behavior patterns reflect the satiety state relate to grazing [18] and ruminating [40], and physical movements relate to walking and resting.

Compared to the exploration of human activities recognition, the detailed analysis of animal behavior classifications is insufficient. There are studies related to classifying animal behaviors such as cattle and sheep with specific classification methodology such as decision, decision tree related ensemble method [16] , SVM [30]and others. [3, 39]

Some animal classification groups approach the problem from a different perspective by examining movement-related behavior and placing sensors on different position[6, 34]

: head, leg [36], ear-tag [35],and neck [25]. Various sampling frequencies, window sizes such as 3 seconds, 5 seconds, and 10 seconds [4] and different activities such as feeding[31], have been investigated in the cattle behavior classification. Among these researches, many of the groups chose 10s window size to compute features[2]. However, from our cattle behavior observations, different activities have different duration and are combined with others. Using a large window size might not be applicable in a real-world application. More studies related to window sizes and classifications of activities are needed. Additionally, there are various features selected for classifications but mainly focused on reflecting statistical information [37]. More variations and comparisons of feature selections are essential to avoid redundancy while maintaining the high accuracy of the model. Additionally, comparing the machine learning models in similar setups is significant for future references and designs.

2.2 Deep Learning approaches

In human activity recognition, both the discriminative and generative approaches have been explored. For classification purposes, there are many CNNs on Human Activity Recognition using the convolutional and pooling layers to extract translation invariant and hierarchical features from sensor data[9, 10, 11]. Another approach is finding the dependencies of sequential data using models such as Recurrent Neural Network, Long Short Term Memory, and Gated Recurrent Unit[12, 13, 23]. For generative model, many studies mainly used for dimensional reduction and feature representation purpose[32].

Two papers are closely related to our goal [42, 44]. To get rid of the fixed window size, the model could predict labels for individual data points, similar to instance segmentation where each pixel has a corresponding label. In [42] the author used FCN to predict dense labels for accelerometer readings. In [44], they replace the dense labels base models with U-Net. For this work, we designed three CNN networks: one for activity classification, one for U-Net inspired dense label prediction and the last a discriminative model for training conditional Generative Adversarial Network.

Chapter 3

Experiment Setup

3.1 Sensor Node and Camera

During the data acquisition process, we collected data using the WLR089U0 board and accelerometer BMA400. The sampling frequency was 50 Hz, ranging from $-2g$ to $+2g$ acceleration, where 1g is the acceleration due to gravity. The accelerometer has built-in digital signal filtering and bandpass filter from 12.5 Hz to 800 Hz. Additional signal processing and filtering are not necessary. We stored the acceleration data in the FIFO data structure, which waited for the coming accelerometer data to fill the array and next transferred the current data to files. Because of this strategy, the sampling frequency was not precisely 50 Hz. We used linear interpolation to synthesize dummy data padded to the delays to make result measurements match 50Hz and synchronize the acceleration with videos. The camera was mounted on the halter aside with an accelerometer to record videos for labeling. To record the nighttime activities and videos, we used an infrared camera that automatically switched on after sunset. Because the field is too large to monitor, tracking the individual cow is time-consuming and costly. We anchored the camera and accelerometer on the side of the halter to capture the cow's perception that we used to decide the activities for labeling.

3.2 Data Collection

We collected data on different days: July 27, September 3, November 4, and November 15, and on different cows: 6036, 6048, 7009, 8086, and 7E34. The final dataset is in a total of 85 hours. The acceleration data were recorded periodically at a data logger, the sensed data and videos were time-stamped, and missing or delay points are interpolated for synchronization.

3.3 Dataset Labeling

Then, we prepared the data for labeling based on the videos. Among the complex daily activities, there are five main activities composed of 92.92% of our labeled data, including standing, walking, lying, grazing, and ruminating. In addition to the majority of activities, there are twenty minorities including running, transitional activities, and cow interactions, which are 7.08% of our data. By training classifiers on the five major activities, we captured the majority of activities that cover the intake, resting, and active periods of the cows. Another factor we need to consider is that, for example, the cow would not keep grazing without walking to search for food or standing alone without interacting with other cows. The activities are inconsistent and mingled with other activities. To tackle this problem, we introduced the transitional activity labels so that the mixing activity patterns will have separate labels. For instance, when the cow hit each other with tills constantly or lick each other, these activities have their categories such as hitting and licking. Therefore, the transitional activities will not interfere with the general behavior pattern and confuse the machine learning models during the training process. On the contrary, we want the model to be robust enough to resist the sudden change in acceleration reading: specifically, if hitting happened only once in grazing activity, 5% hitting and 95% grazing in chosen segments,

we should still be able to classify the grazing activity. Thus, to classify the majority of activities correctly even with small disturbances, we adopted the following labeling strategies: if the activities, such as hitting, are intense that exceed the $\pm 2g$ accelerometer readings and consistent that more than 50% of the segments, we labeled them out as separate activities only when both criteria are met, otherwise the small activities are viewed as part of the main activities. Overall, we mainly focused on classifying the five major activities and got the clean state acceleration data for models to learn major activity patterns.

Chapter 4

Prepossessing

With the individual data collected on different days and cows, we normalized them to have zero mean and standard deviation of one and concatenated them to form the final dataset. Two-thirds of our segmented data is used for training and one-third for testing with the random seed of 42 in python.

4.1 Segmentation

After getting the clean state of data, we segmented the data into fixed window sizes to compute features. Many activity classifications solve the problems in this setting with a fixed window size such as 5s to 10 s; nevertheless, fixed window size has shortages we need to consider before proceeding. The activities will not have the same duration that matches the multiple of the segments. On the one hand, smaller window sizes capture the sudden change in activities and reduce the chances of mixing activities within an individual segment. For example in figure 4.1 On the other hand, larger window sizes minimize the percentage of noise and disturbances within the segments; as a result, calculated features could capture the majority of information and further improve classification accuracy and robustness to some degree. To determine the appropriate window sizes, we used the labeling tool to get the duration of each labeled activity and found the minimum possible window sizes. From our labeled data, the grazing activities usually take 4s to 24s, whereas ruminating takes 5s

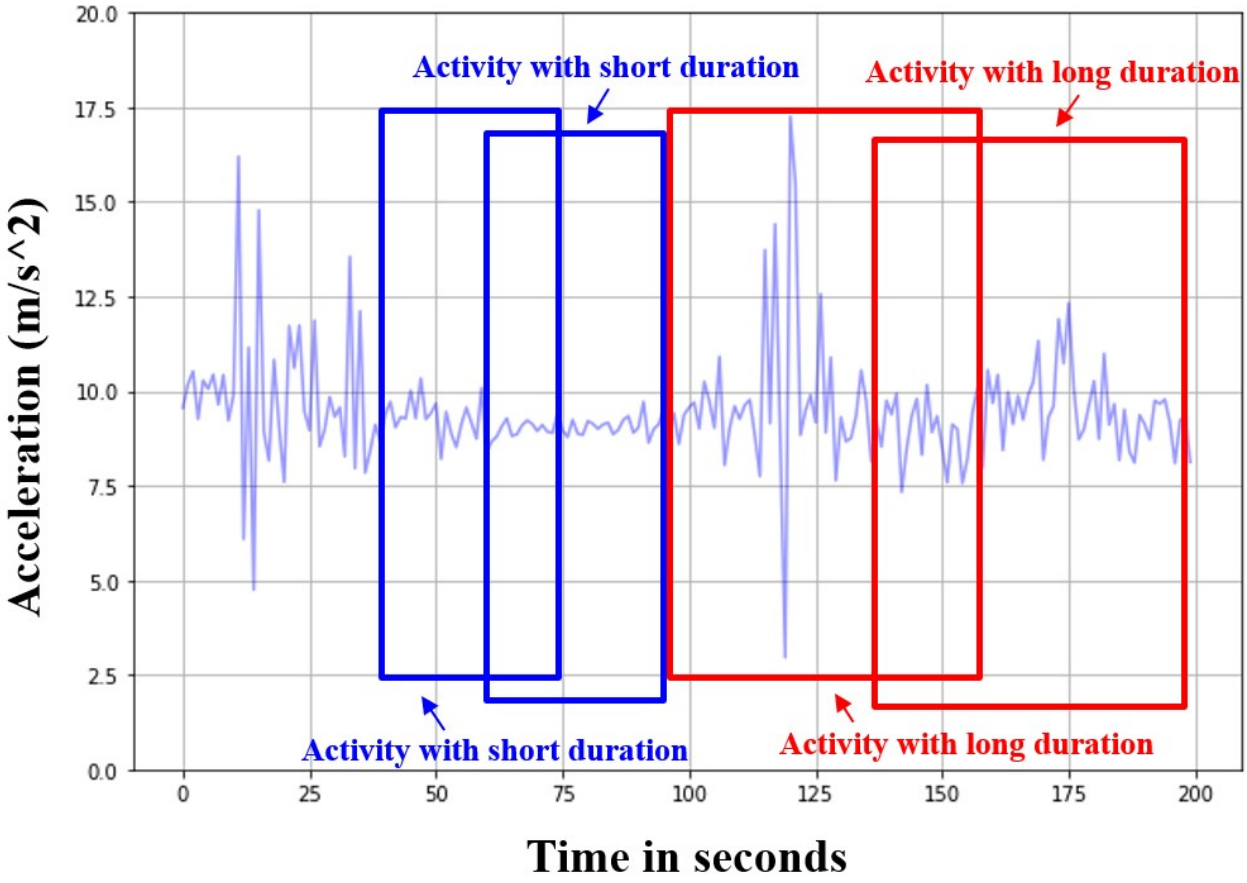


Figure 4.1: Activity with different duration.

to 14s, walking takes 4s to 15s, standing takes 1s to 16s, and lying takes 10s to 25s. We eventually chose 3s that were small enough to capture the majority of activities while the sample size was large enough to smooth out some of the noise and spikes due to hitting and other movements.

4.2 Class Imbalance

Another problem is the unequal instances for different classes. Similar to [2], grazing and ruminating are the majority classes, which are 36.56% and 25.36% in the five categories.

Lying, standing, and walking are the minority classes, which are 18.22%, 13.42%, and 6.46% in the five categories. Due to this class imbalance, the models are biased and in favor of the dominant class[41]. For example, grazing is dominant over walking, so there will be more misclassified grazing activities that contribute to the loss function during training so models are trained in favor of grazing activities because of the higher occurrence. To alleviate this problem, we trained our models using the sampling technique. Another alternative is giving weights to training instances based on their class. The weights to the loss function penalize more on the minority class, so each training instance in the minority class is more influential compared to the majority class. However, the given weight may push the decision boundary too much to introduce bias. To find the suitable ways, we compare the model performance of these two methods in Section 6.

Chapter 5

Feature Extraction and Visualization

5.1 Feature Extraction

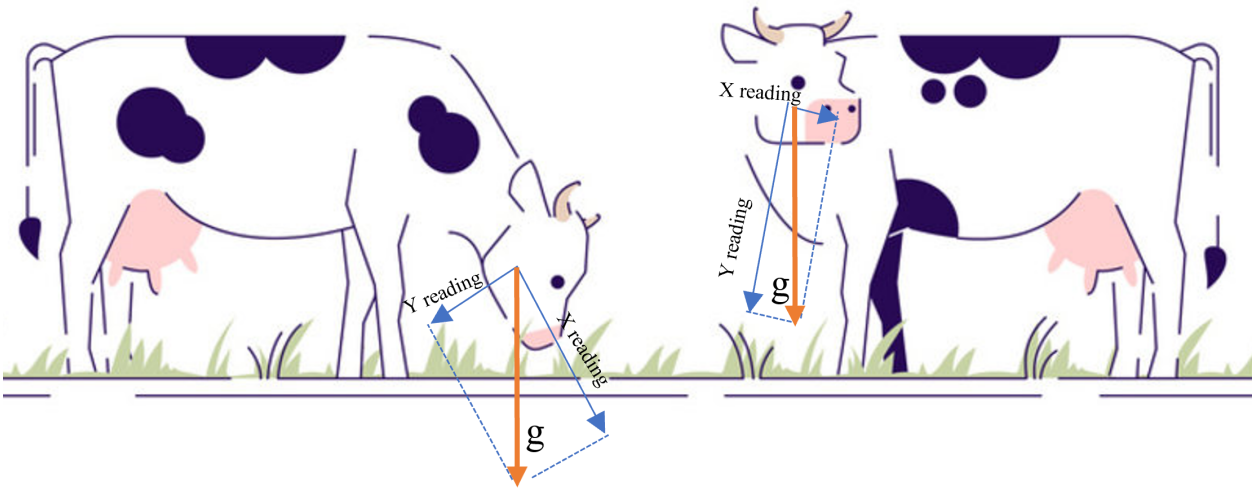


Figure 5.1: The illustration of how gravitational acceleration is distributed with a different head position in 'grazing' (left) and 'standing' (right) activities, causing variations in the x-axis and y-axis readings. Orange arrows denote the gravitational acceleration, and blue arrows are the decomposition of the gravity acceleration.

With the given segments of acceleration data, we cross-compared and analyzed accelerometer reading on different activities to find features that differentiate the activities using acceleration. The computed features capture the various activity patterns by extracting the head position and the body movement information.

Table 5.1: List of Features. For any given segment, Y is the ordered list, and n is the number of the data. P_i is the probability distribution of the given array. $\hat{x}_T(f)$ is the Fast Fourier Transform (FFT) of the data in each window, and N is the length of the FFT data.

FEATURES	EXPRESSIONS
MEDIAN	$\begin{cases} Y[\frac{n}{2}] & \text{IF } n \text{ IS EVEN} \\ \frac{Y[\frac{n-1}{2}]+Y[\frac{n+1}{2}]}{2} & \text{IF } n \text{ IS ODD} \end{cases}$
VARIANCE	$\frac{\sum(x_i-\bar{x}^2)}{n-1}$
ENTROPY	$-\sum_{i=1}^{n} P_i log(P_i)$
CROSS-CORRELATION	$(f \star g)[k] = \sum_n \overline{f[n]} g[n+k]$
SIGNAL-TO-NOISE RATIO	$\frac{\mu}{\sigma}$
MAGNITUDE	$\frac{1}{n}\sum\sqrt{x_i^2+y_i^2+z_i^2}$
SIGNAL MAGNITUDE AREA	$\frac{1}{n}\sum\lvert x_i\rvert$
[43]	
INTEGRAL	$\int_a^b f(x)dx \approx \frac{(b-a)}{2}(f(a)+f(b))$
MOVEMENT VARIATION	$\frac{1}{n}\sum\lvert x_{i+1}-x_i\rvert$
FREQUENCY DOMAIN	
PEAK FREQUENCY	$\frac{argmax(\hat{x}_T(f))*f_s}{N}$
PEAK FREQUENCY MAGNITUDE	$\frac{max(\hat{x}_T(f))*f_s}{N}$
POWER SPECTRAL DENSITY	$lim_{T\to\infty}\frac{1}{T}\lvert\hat{x}_T(f)\rvert^2$

The x-axis and y-axis readings correspond to the cow's head orientation as shown in Figure 5.1. When the cows move the head up and down, the x-axis and y-axis readings alter up and down. The z-axis reading provides the horizontal acceleration, measuring the left or right turns. In grazing activities the left cow in Figure 5.1, the cows lower their heads close to the ground, so the gravitational acceleration aligns with the x-axis direction causing the x-axis

reading around one g and the y-axis reading around zero. In other activities when the cows do not lower their heads, the gravitational acceleration is smaller on the x-axis and larger on the y-axis. Thus, the gravity force distributed in the tri-axial differentiate activities such as grazing and standing based on the head position. By learning the distribution of acceleration data, the machine learning models could classify some of the head position-related activities. Initially, we computed features to measure the relations and spread outs, including correlation between different acceleration axes, mean, percentiles, movement variation, min, max, integral of accelerations, and others as shown in Table 5.1. These features require a lot of computation and provide redundant information, so we evaluate the redundancy and select 30 among them.

Unlike the mean that extremely large or small values will fluctuate the calculation, the median is the middle value of a sorted list, so the outliers have little influence. Even though sorting needs more computation, the median could better represent the typical acceleration. Variance describes how data deviate from the expected values. A larger variance means more spread out in the data. Different from measuring the variation of value, entropy focuses more on the probability distribution. It measures the randomness of accelerations. If the acceleration clusters are more spread out like grazing or walking, entropy will be smaller; if the clusters are more compact such as lying and standing, entropy values will be larger. To visualize how the selected features affect the clustering of different activities, we use PCA on the three-axis median, three-axis variance, and three-axis entropy and plot the two principal components. From figure 3, grazing activities and ruminating activities are predominantly distinct, while walking, standing, and lying activities are mixed and spread out. They are more diverse: the cow could be walking while lowering its head, or rambling, or rushing. Moreover, the cow will not keep the same pose while standing. They are very active in turning or moving or interacting and lying activity and standing activity are similar. Therefore, only

median, variance, and entropy are not sufficient to separate standing, walking, and lying. Another piece of information contained in the accelerometer reading is the linear acceleration due to body movement. Because the acceleration reflects the energy intensity of the activities, we used movement features: magnitude, entropy, peak frequency, and Power spectral density to differentiate the still activities and active activities. The magnitude measures the intensity of total acceleration without direction. It combined tri-axial acceleration that will always contain the acceleration due to gravity, so the gravitational acceleration becomes a constant value and will have minimal effect on the variation of accelerometer readings. To measure the changes in acceleration, we applied Fast Fourier Transform to calculate the frequency features. The selected information was the peak frequency and the power spectral density over different frequency segments. From the frequency plot of accelerometer readings, most of the signal has a frequency below 10 Hz and peak frequency around zero, so the selected frequency range was from 0.07 Hz to 10 Hz. We found the peak frequency from 0.07 Hz to 2 Hz and 2 Hz to 5 Hz based on the belief that different activities had different peak frequencies. Additionally, to know how the power density spread over the frequency, we separate the 10 Hz frequency into three parts to find the average power spectral density in these segments. However, because the PSD value varies dramatically, we normalized the PSD array before finding the average value of each segment. The results of the PSD segment were relative to show which frequency segments are intense. Table 5.1 lists 12 groups of features considered for our work.

Since there is overlap between groups and acceleration information from 3-axis acceleration, some features contain redundant information. It is desirable to eliminate those to save time for training while maintaining classification accuracy. We adopted the RFE based on SVM in [20]. Redundant features are pruned recursively from the set of features. The features are ranked in each iteration based on the weight of the parameters in SVM and low ranking

Table 5.2: The 30 selected features computed using RFE from the expressions in Table 5.1.

FEATURES	AXIS
MEDIAN	X, Y
VARIANCE	X, Y, Z
ENTROPY	X, Y, Z
CROSS-CORRELATION	XY, YZ, XZ
SIGNAL TO NOISE RATIO	X, Y, Z
MAGNITUDE	ALL
SIGNAL MAGNITUDE AREA	X, Y, Z
INTEGRAL	X, Y, Z
MOVEMENT VARIATION	X, Y, Z
MAGNITUDE OF PEAK FREQUENCY	X, Y, Z

Table 5.3: Balanced accuracy versus number of selected features.

# FEATURES	SVM	K-NN	RF	HGBDT
20	90.74%	90.44%	92.22%	92.67%
30	91.39%	90.5%	92.4%	93.04%
52	91.89%	89.47%	92.42%	93.11%

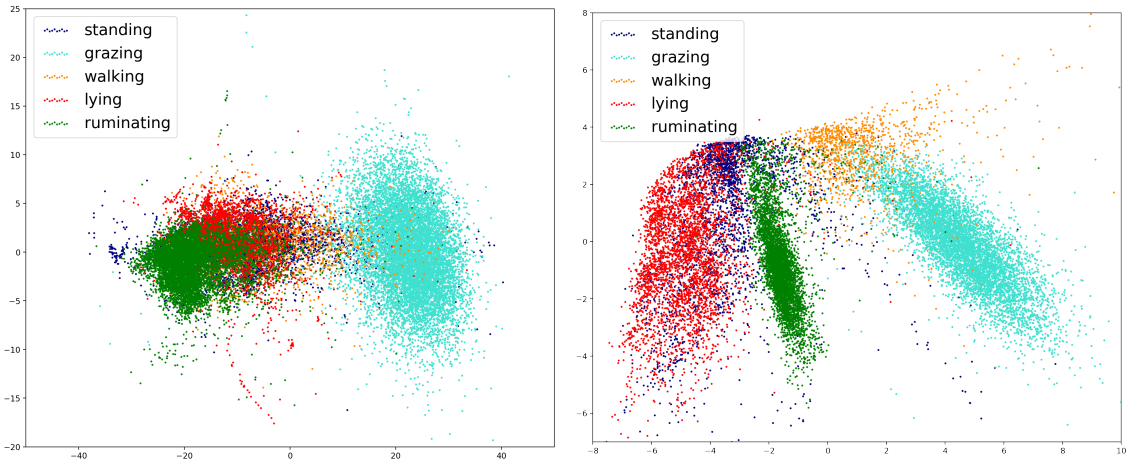
features are eliminated. With this setting, the features with complementary information have a higher weight than those with important but repeated information. Table 5.3 shows the balanced accuracy of four classification algorithms with three different numbers of the features. The classification algorithms considered are SVM, k-NN, RF, and HGBDT.

In general the balanced accuracy increase with the increase of selected features until convergence. One exception is k-NN, in which the accuracy is higher with a smaller number of features because it treats all features the same when calculating distance, and more features with insignificant information may confuse the model. The detailed settings for hyperparameters are described in Chapter 6.

The selected features are listed in Table 5.3, and most of them belong to the time domain.

5.2 PCA visualization

We standardized the selected 30 features for training and examined the impact of the selected feature set with PCA. Figure 5.2a shows the clusters of the five behaviors using the first and second principal components on the raw acceleration data. The 'grazing' and 'ruminating' activities are well separated, but the other three activities ('standing', 'walking', and 'lying') are mixed. Figure 5.2b shows the PCA result with the selected 30 features. The selected features promote the separation of different clusters. Note that 'walking' is separated now, but 'standing' and 'lying' are still mixed closely. Because the acceleration measured in the local coordinate closely resembles for 'lying' and 'standing', differentiation of the two activities is difficult without global reference coordinate or additional sensors. The limitation appears in the model performance in Table 7.1 as well.



(a) PCA visualization on the raw data, using the first and second principal components.

(b) PCA visualization on the selected features.

Figure 5.2: In the first subimage 'Grazing' and 'ruminating' activities are distinct, while the other activities are mixed. In second subimage 'Grazing', 'ruminating', and 'walking' are distinctive while 'standing' and 'lying' activities are mixed.

Chapter 6

Algorithms

To find an optimum model for the cows' behavior classification, we compared the classification performance of SVM, k-NN, RF, and HGBDT. We used the machine learning models and evaluation metrics from [33]. Since our dataset with selected features is imbalanced, we trained the algorithms using the sampling techniques and class weight, and fine-tune the hyperparameters. The sampling method is from [27]. We used balanced accuracy to evaluate the performance of models. As each class is represented by its recall regardless of the size, the balanced accuracy is helpful to spot possible predictive problems for rare and under-represented classes, specifically 'standing' and 'walking' activities in our application [17].

6.1 SVM

We applied SVM because it performs well in high dimensional space using kernel tricks for the complex decision surface. SVM is a classification method aiming to separate two data sets with the maximum distance between them. For linear classification, $y_i[wx_i + b] > 1$, $i = 1, ..., l$, the Lagrange form is the equation below:

$$l(w, b, a) = \frac{||w||^2}{2} - \sum_{i=1}^{l} \alpha_i \{y_i[wx_i + b] - 1\} \tag{6.1}$$

where $\alpha_i$is the Lagrange multiplier, $x_i \in R^n, i = 1, 2, ..., N$, and $y_i \in \{-1, 1\}$. The parameter w can be solved based on the condition $\sum_{i-1}^{l} \alpha_i y_i = 0, \alpha_i \geq 0, i = 1, ..., l$. Then, we substitute the w into the previous equation and rearrange the equation into the distance from and the optimum separation hyperplane (OSH) is: [19]

$$d(x) = \frac{b + x \sum_{i=1}^{l} \alpha_i y_i x_i}{|| \sum_{i=1}^{l} \alpha_i y_i x_i ||} \tag{6.2}$$

As the $|d|$ increases, we could obtain a decision boundary with a larger distance to the nearest training data to separate the two classes. We use the one-vs.-all method training multiple classifiers. It treats one activity as positive and the rest as negative when dealing with multiple activities. Then the prediction is based on the highest confidence score. Table 6.1 compares the balanced accuracy of SVM with different kernels and training methods. The RBF kernel performs slightly better than the polynomial with degree three and the Linear and far better than the Sigmoid. The Radial Basis Function (RBF) kernel and soft margin are adopted because the data may not be linearly separable. For example, in Figure 5.2b, although most of the data in the same activities cluster together, some data points are in the middle of other cluster groups. To address this problem, the SVM with RBF kernel uses weight coefficient combined with Gaussian basis function to learn the model from a higher dimensional space where the dataset is separable [38]. In this way, the SVM with RBF kernel could work with nonlinear classification problems.

6.2 k-NN

k-NNis an effective none parametric classification method that classifies the input based on the similarity of existing data. It predicts results without making strong assumptions about the dataset. [1]

Table 6.1: Balanced accuracy of SVM with different kernel tricks. SVM is trained with the random oversampling method or class weight using class weight

KERNEL	IMBALANCED	SAMPLING	CLASS WEIGHT
RBF	89.49	91.11	91.43
LINEAR	89.30	90.33	89.91
POLY=3	89.43	90.82	91.22
SIGMOID	76.83	72.58	69.03

Nearest-Neighbor classifies unlabeled observations by finding the most similar labeled examples based on distance measurement. K defines the number of neighbors near the query point. The most frequent label in the selected k ranges will be assigned to the unlabeled observation. The general idea is to use distance measurement to find the windows of data that are similar to the query with k defined window size. The variables do not need to be i.i.d. or linearity or linearly separable. Thus, we choose this algorithm to classify the problem from different aspects and to see how the model performs in our none linearly separable data set.

6.3 Random Forest

Random forest is an ensemble learning method that trains multiple decision trees and gets results based on the majority votes. Each tree is trained from bootstrap samples from the original data set. Then, at each node of each tree, the cut is based on the Gini impurity measure to split the set [8].

Because decision trees randomly select features for building trees, random forest is a non-parametric classification method as well. The bagging method ensures the trees learn the

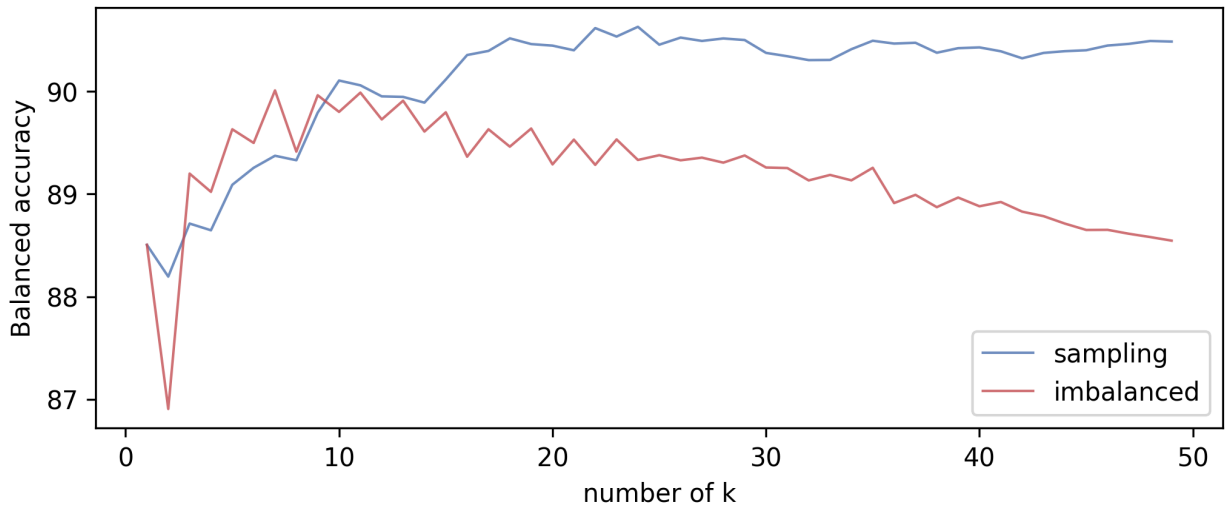


Figure 6.1: Balanced accuracy of k-NN versus k. The maximum accuracy is 90.63% under k=24.

data set with the same distribution. The goal is to maintain the strength while minimizing the correlation, so the trees learn the models independently and avoid making the same mistakes. After the trees are generated, the results are based on averaging the probabilistic prediction. To get the optimum performance, we test the max depth of the tree and the number of estimators. The max depth determines how complex the decision tree is while growing trees (Mtry). Higher depth could select more features and capture more complex decision surfaces but is susceptible to overfitting. A larger number of estimators enable the predictions to have more variation from multiple trees while keeping low bias (Ntree). As the number of estimator increase, the accuracy will converge [5].

To obtain the optimum performance, we tested the depth of the tree from 10 to 30 and the number of estimators from 50 to 600. A higher depth could select more features and capture more complex decision surfaces. As the number of estimators increases, the accuracy increases at the cost of higher computation complexity.

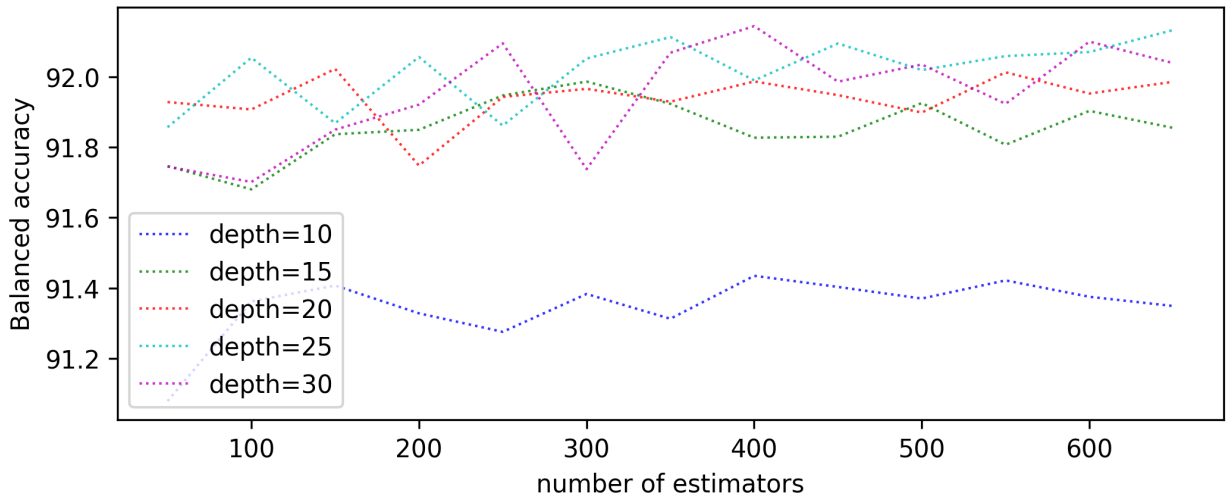


Figure 6.2: Balanced accuracy of random forest trained with class weight: number of estimators from 50 to 600, and the depth ranging from 10 to 30. We chose depth of 30 and 400 estimators with a balanced accuracy of 92.14%.

Figure 6.2 shows the balanced accuracy with a different number of depths and estimators trained using class weight. The balanced accuracy stabilizes to some degree for all depths after 200 estimators. Figure 6.3 shows the balanced accuracy of the random forest trained using random oversamples. The models trained with random oversamples perform slightly better than those trained with class weights. The optimum balanced accuracy trained with the class weight is 92.14% with the depth of 30 and 400 estimators. In contrast, the optimum balanced accuracy trained using random oversample is 92.60% with the depth of 25 and 300 estimators.

6.4 HGBDT

HGBDT is an ensemble method as well. The subsample of the training data is drawn at random without replacement to train the base learner. The minimum loss function is found

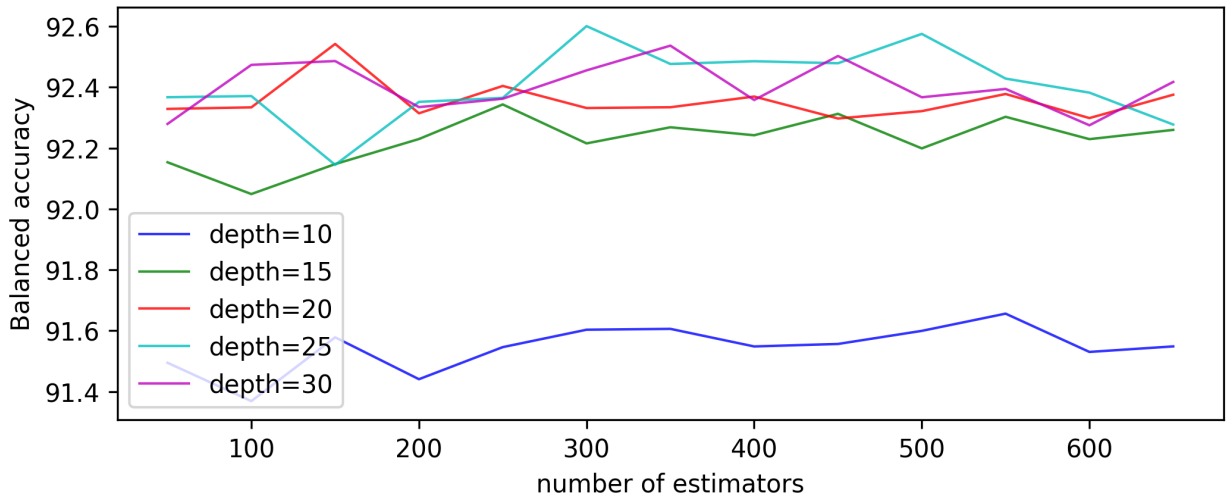


Figure 6.3: Balanced accuracy of random forest trained with random oversample: number of estimators from 50 to 600, and the depth ranging from 10 to 30. We chose depth of 25 and 300 estimators with a balanced accuracy of 92.60%.

through the gradient descent approach and prediction is based on the strong learner result from weak learners. Unlike the majority vote used by random forest, gradient boosting keeps learning from its predecessor in a stage-wise manner to minimize the loss function. To speed up the training process and reduce memory usage, we adopted the histogram-based method inspired by [26]. Instead of sorting the feature values, HGBDT partitions the input samples into integer-valued bins [33]. We tested HGBDT with $l2$ regularization and different numbers of bins ranging from 150 to 255. The balanced accuracy trained using the random oversample method is 93.19% with the maximum number of bins of 219, the $l2$ regularization, the maximum number of iterations of 300, and the shrinkage of 0.08. To compare the performance on the imbalanced dataset, we show the Receiver Operating Characteristic (ROC) and Precision-Recall (PR) curves.

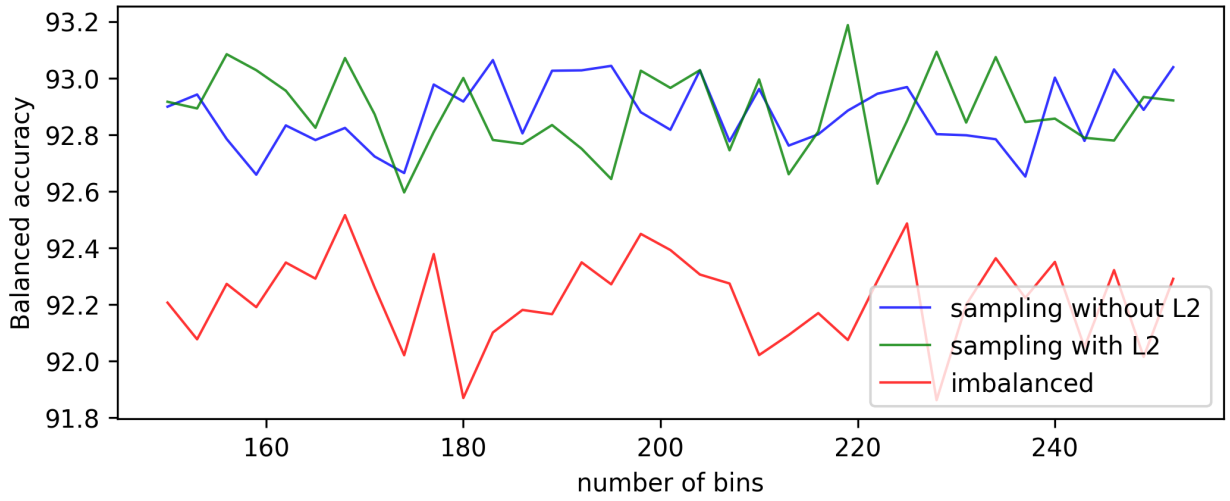


Figure 6.4: Balanced accuracy of HGBDT trained with random oversample without regularization, random oversample with $l2$ regularization, and original dataset.

Chapter 7

Evaluation

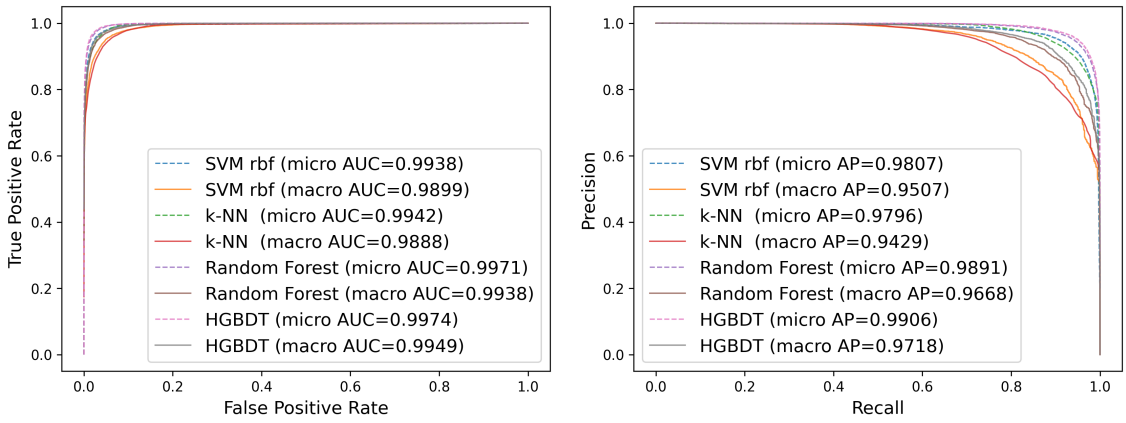
7.1 ROC Curve

The ROC curve provides information regarding how the number of correctly classified instances varies with the number of incorrectly classified negative ones [14]. The Area Under the Curve (AUC) score ranging from 0 to 1 is used for comparison. A high true positive rate and a low false-positive rate have AUC closer to 1. In Figure 7.1a, micro AUC is the average of AUC calculated from contributions aggregated of all classes. Because our dataset is imbalanced with 36.56% 'grazing' and 25.36% 'ruminating' and our models achieve high accuracy in these two classes. We also provide macro AUC as another measurement. Macro AUC is the arithmetic mean for each class, which reflects unbiased results by treating all classes equally.

7.2 PR Curve

Similar to the ROC curve, the PR curve is another way to evaluate the model performance; it is more useful for imbalanced classes by plotting the precision against the recall. To have high precision and recall, the model needs to reduce the false-positive (FP) and the false-negative (FN) values. Majority classes dominate the minority classes causing a bias in the models and hence are prone to higher FP or FN. Thus, the PR curve could show

the precision-recall trade-off and how the models deal with the class imbalance. Average Precision (AP) is similar to AUC to measure the performance of the models. In both Figure 7.1a and Figure 7.1b, random forest and HGBDT outperform the SVM and k-NN. When the classifiers in ensemble methods are diverse and uncorrelated, the majority of them are less likely to make the same mistakes. Therefore, the combination of individual hypotheses will have a smaller error rate [15]. To introduce diversity to the base learners, both random forest and HGBDT build trees with random sampling to improve the generalization performance.



(a) The ROC curve.

(b) The PR curve.

Figure 7.1: Performance evaluation of SVM, k-NN, random forest, and HGBDT

7.3 Overall Comparison

Table 7.1 compares the recall values of the four algorithms for five different behaviors. All the algorithms seem to have problem distinguishing between 'lying' and 'standing', because the acceleration of these two activities are similar where there isn't sufficient distinguishable characteristic to extract. HGBDT performs slightly better than random forest on the 'standing' class due to (i) our labeling method might have eliminated some noise, and the true decision boundary could be additive so that the resulting HGBDT model making a decision

Table 7.1: The recall of individual class with different models.

	STAND	GRAZE	WALK	LAY	RUMINATE
SVM	78.42	98.91	95.71	84.57	97.96
K-NN	77.48	98.53	95.36	83.31	97.62
RF	82.20	99.37	92.50	89.77	98.16
HGBDT	84.25	99.42	92.68	90.73	98.25

boundary more explicit for separation of 'lying' and 'standing' activities. It is also worth noticing that the recall value of walking activity dropped in RF and HGBDT. The possible reasons would be the walking activity have smaller duration. Indeed, the 3s windowsize not only capture walking activity, but also other activity such as grazing and standing as well. Thus, the computed features could reflect the grazing and standing activity as well and confuse the ensemble models. The RF and HGBDT could distinguish the standing and lying activity with complex decision surface but too much details may fail to capture the general trend which is reflecting on the decrease of recall values in 'walking'. Figure 7.2 shows the normalized confusion matrix with the recall of 84.25% for 'standing', 99.42% for 'grazing', 92.68% for 'walking', 90.73% for 'lying', and 98.25% for 'ruminating'.

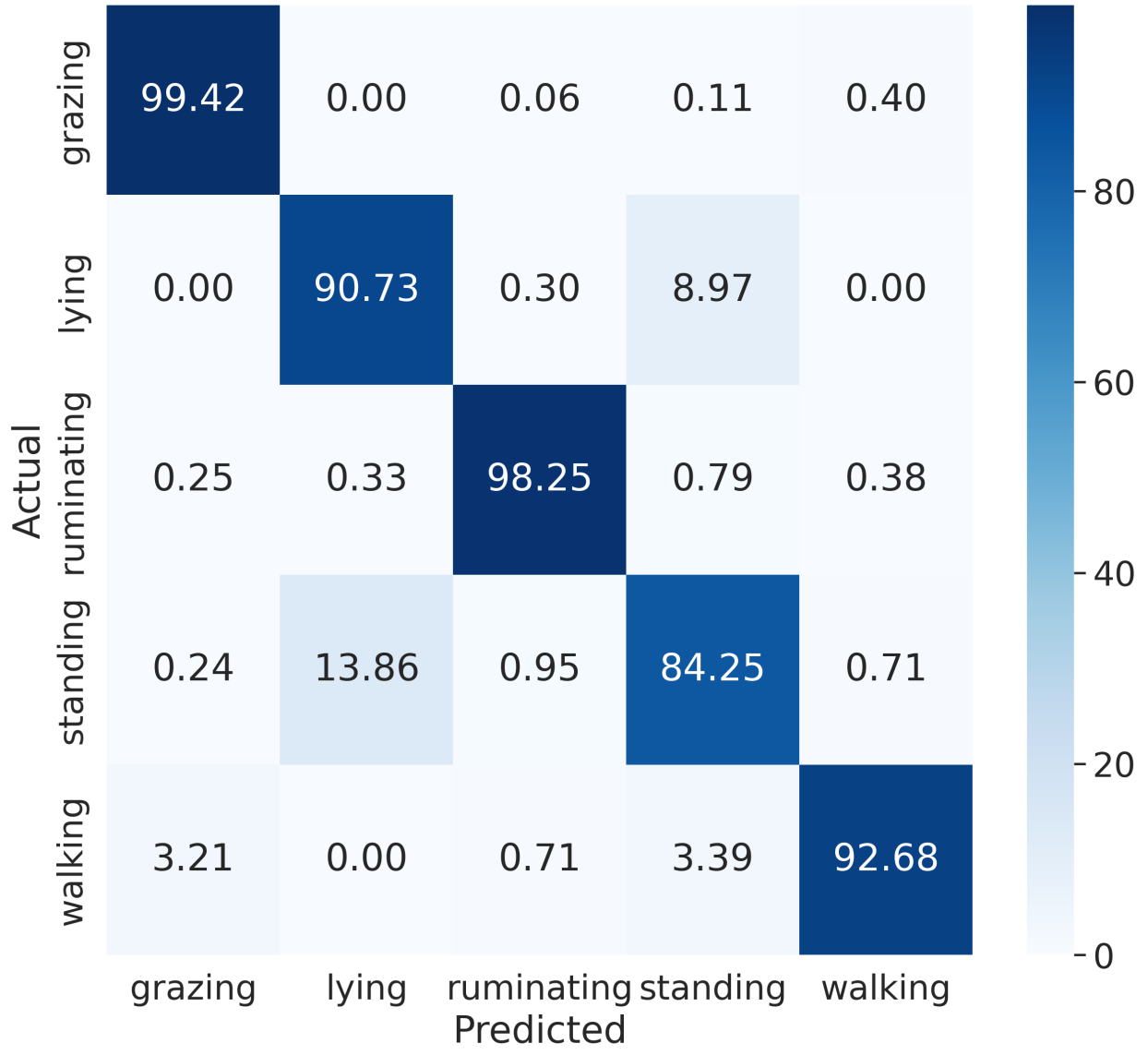


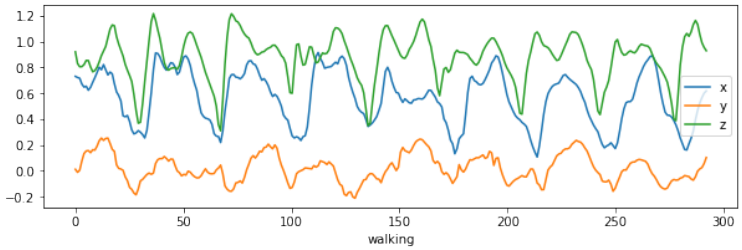
Figure 7.2: The confusion matrix of grazing, lying, ruminating, standing, and walking activities using histogram gradient boosted decision trees.

Chapter 8

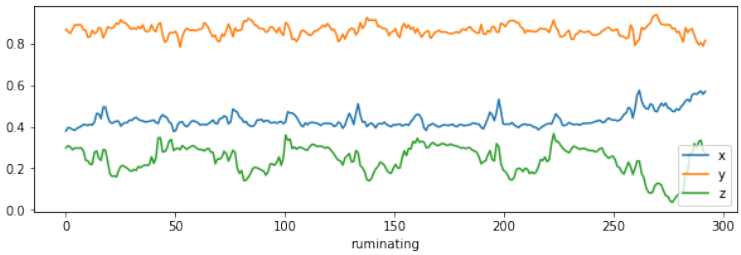
Deep learning

Since Artificial Neural Network (ANN) has gained more and more attention in recent years due to its ability to extract linear combinations of features and prediction power, we also consider applying Deep Learning to solve the problem in a different aspect. In the field of computer vision, the images are commonly interpreted as three channels with intensity values. Thus, the images could be viewed as multi-dimensional signals and many of the methods in computer visions such as Convolutional Neural Networks (CNN) could be adopted for analyzing and extracting the features in accelerometer data as well. For example in figure 8.1, there are clear patterns in the acceleration data that we could use to distinguish different classes.

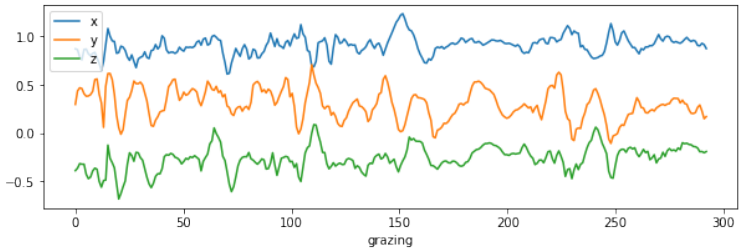
The main reason we did not apply computer vision on cow videos is that it is not realizable for our current capability for the following reasons: when there are multiple cows in the field, we need to 1. distinguish and track individual cows, 2. the cows may be very small from the images because of the large space which makes the detection task more difficult 3. perform images classifications on continues incoming videos need lots of computations. On the contrary, collar mounted sensor is a promising solution because no need to track individual cows, and fewer inputs for models so the model size and computation could be drastically reduced which makes edge computing possible.



(a) The walking acceleration.



(b) The ruminating acceleration.



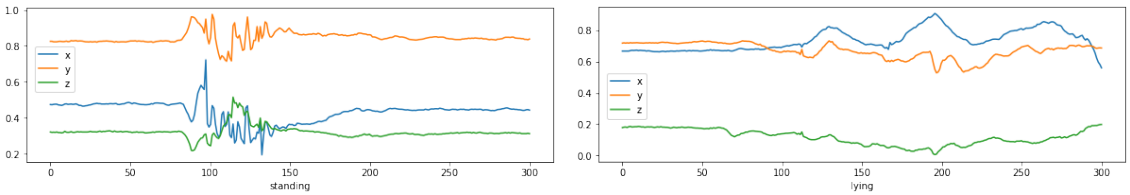
(c) The grazing acceleration.

Figure 8.1: The patterns in walking, ruminating and grazing activities

In the next two sections, we formulate the activity classification differently to predict a single label for a given segment or to predict the activity distribution of activities in a given segment. The first CNN is an improved version of the previous ML method, so no hand-crafted features are needed to transform the data to a separable space. The second CNN is aiming to address the shortages of fixed window-sized, so one segment could contain multiple activities and the models will determine the corresponding labels and positions for different data points, similar to instance segmentation where the boundaries between different instances are predicted as separated masks.

8.1 End to End Trainable Convolutional Neural Network

The normalized 3s segmented accelerometer data are used as inputs and the model will produce the activities for segments. During the designing process, we test both 1D convolution and 2D convolution. In the 1D setting, the acceleration data is scanned with a size 3 kernel and the 3-axis are inputted as channels. Therefore, the output would be a combination of 3-axis acceleration data. The filters are convoluted with the individual axis to extract patterns from them. This design is more efficient because fewer parameters and less computation are compared to the 2D case. However, one problem is that standing and lying activities have similar patterns in acceleration, for example in figure 8.2, so 1D CNN did poorly in distinguishing them. Figure 8.3 is a comparison in the early design stage of the performance of



(a) The standing acceleration. (b) The lying acceleration.

Figure 8.2: The comparison of standing and lying activities

1D and 2D CNN under similar settings (that feature maps obtained from previous layers are concatenated to later layers before reducing model size. The dataset only contains the samples collected before November 2021). In the 1D CNN standing have a recall value of 80.8% and lying have a recall value of 85.5%, whereas in 2D CNN, the recall value for standing increased to 85.5% and the recall value of lying increased to 86.8%. The walking activity in 2D CNN increased from 94.4% to 97.2%. It is worth noticing that the walking class has 10 times fewer samples than the grazing class, so the trained model performance varied and we cannot conclude whether the 2D CNN caused walking performance to increase. Then we

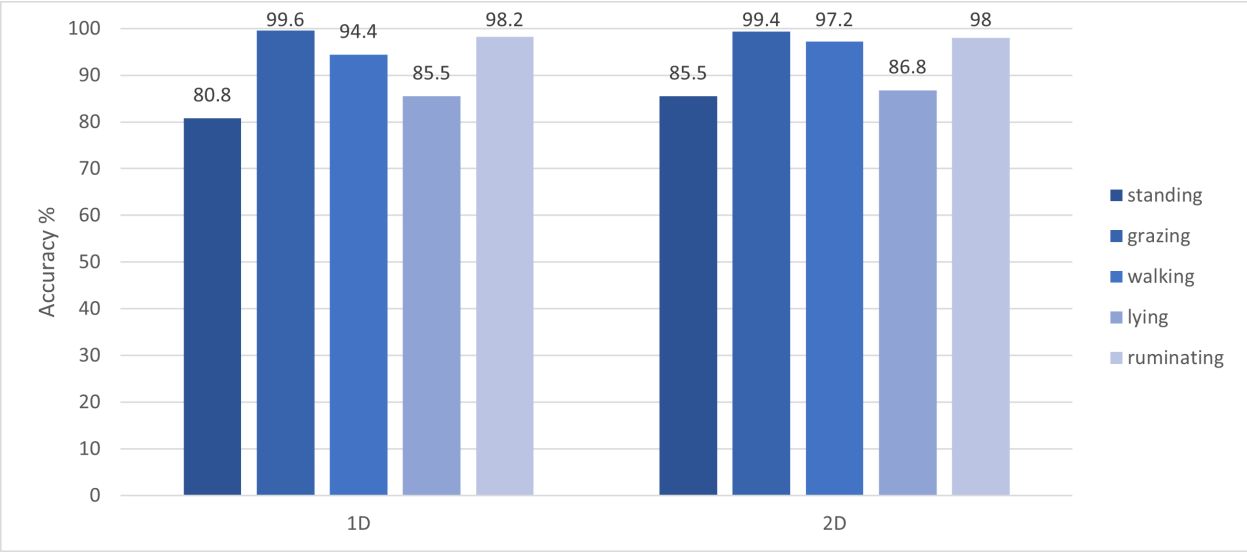


Figure 8.3: Comparison between 1D and 2D CNN.

used 2D CNN to extract the relationships between different axis as patterns to distinguish the standing and lying activities.

To design an efficient architecture, we first determined how many blocks of CNN are sufficient to extract and combine the features maps. The intuition is the receptive field should be large enough for all activities pattern and the depth of the network should be sufficient enough to learn the complex patterns. We picked 3*3 kernel with stride 2 and we found out three blocks are the ideal choice and additional blocks did not necessarily increase the performance. During the experiment process, we found that low-level features are also useful in the classification process, so we originally add skip connections to concatenated the feature maps from previous layers to later layers. The skip connections also serve as a shortcut to promote competition among shallow features and deeper features for the model's robustness. As a result, skip connections boosted accuracy, but it increase the network size drastically. We adopt the residual connection in ResNet to reduce size will keep the performance [22]. Because the response in lower layers could be noisy, we used a larger pooling size in max-

pooling to pull the same dimension results for addition. To reduce the channel-wise depth, we use the method in Network In Network [28], the 1*1 kernel to get the cross channel linear combination. After the three blocks of feature extraction, we fed the information to four layers of fully connected layers and predict the classification results. With this setting, we were able to reduce the number of trainable parameters from 186,259 to 69,715. The detailed architecture is in figure 8.4

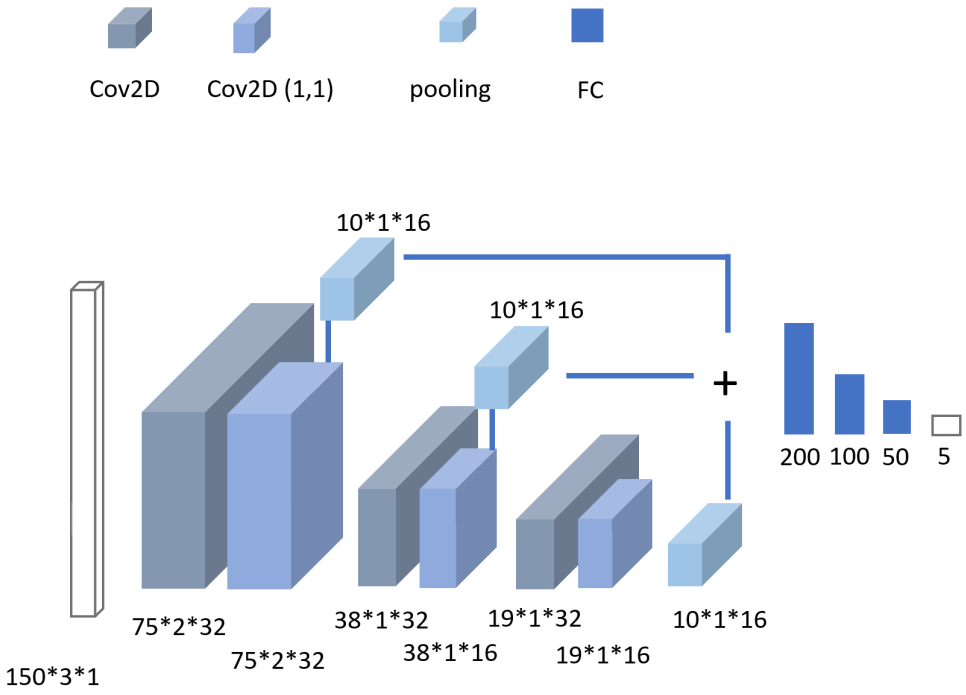


Figure 8.4: Simplified CNN.

For the training stage, we used random sampling to upsample the minority class to have the same number as the majority class. Then we used focal loss which will minimize the loss in easy classified instance[29]. It is a variation from cross-entropy loss. In the figure 8.5, by changing the γ, we could adjust the loss curve to give less loss to well-classified examples, so

only the uncertain instances contribute to the gradient. The equation we used is as follows:

$$FL(p_t) = -\alpha_t(1-p_t)^{\gamma} log(p_t) \tag{8.1}$$

We set the α to 0.6 and γ to 2.

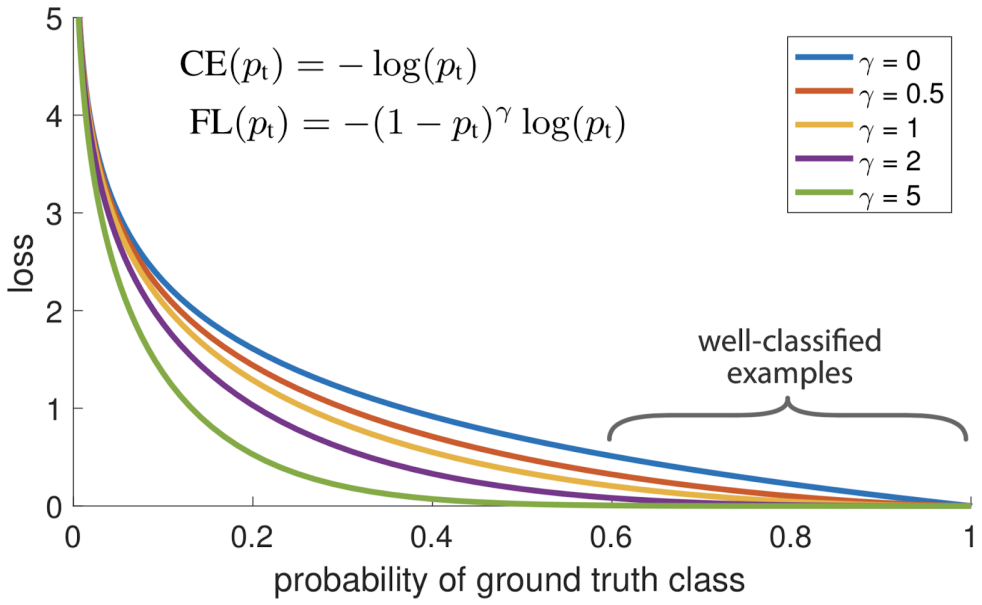


Figure 8.5: Focal Loss.

We used Adam optimizer and learning rate monitor to reduce the learning rate on the plateau for optimum performance. We used He Normalized initialization and PReLu as activation function[21]. The random initialization could faster and stabilize the training process. The PReLu is a variation of ReLu, so it's a piecewise activation as well. The advantage of PReLu over LeakyReLu and ReLue is when the input is below a certain threshold, the information remains and is multiplied by a learnable parameter α.

We used the precision and recall curve (PR) for performance measurement and computed the confusion matrix. The PR on the validation set is 98.84. The normalized confusion

matrix is in table 8.1.

Table 8.1: Normalized Confusion Matrix of Simplified CNN.

	grazing	lying	ruminating	standing	walking
grazing	99.3	0.0	0.2	0.1	0.5
lying	0.0	90.4	0.4	9.2	0.1
ruminating	0.3	0.4	98.5	0.6	0.3
standing	0.2	16.9	1.4	80.9	0.6
walking	2.9	0.0	0.7	1.2	95.1

To further reduce the size of the network, we implemented a preliminary distilled learning method to reduce the model size. The basic idea is the complex model (teacher model) will output a soft label for the simple model (student model) to learn the interrelationship between labels. Instead of learning the task directly, KL divergence is used as a loss function to train the student model that generates the labels to match the distribution of labels produced by the teacher model. The model in figure 8.4 is used as a complex teacher model to train a relatively simple student model with the number of filters reduced to half and 2D convolution layers replaced by separable convolution layers after the first convolution block. The architecture of the student model is in figure 8.6 In this way, the need for high

Table 8.2: Normalized Confusion Matrix of Student Model.

	grazing	lying	ruminating	standing	walking
grazing	99.0	0.0	0.2	0.2	0.7
lying	0.0	88.4	0.4	11.0	0.2
ruminating	0.1	0.4	98.5	0.6	0.4
standing	0.2	17.5	0.9	80.6	0.9
walking	2.5	0.0	0.5	2.2	94.8

dimension matrix multiplication is drastically reduced, where the convolutional operations are separated into pieces so the dimensions are switched from multiplication to addition. The student model is trained using KL divergence, alpha equal to 0.1, and temperature equal to

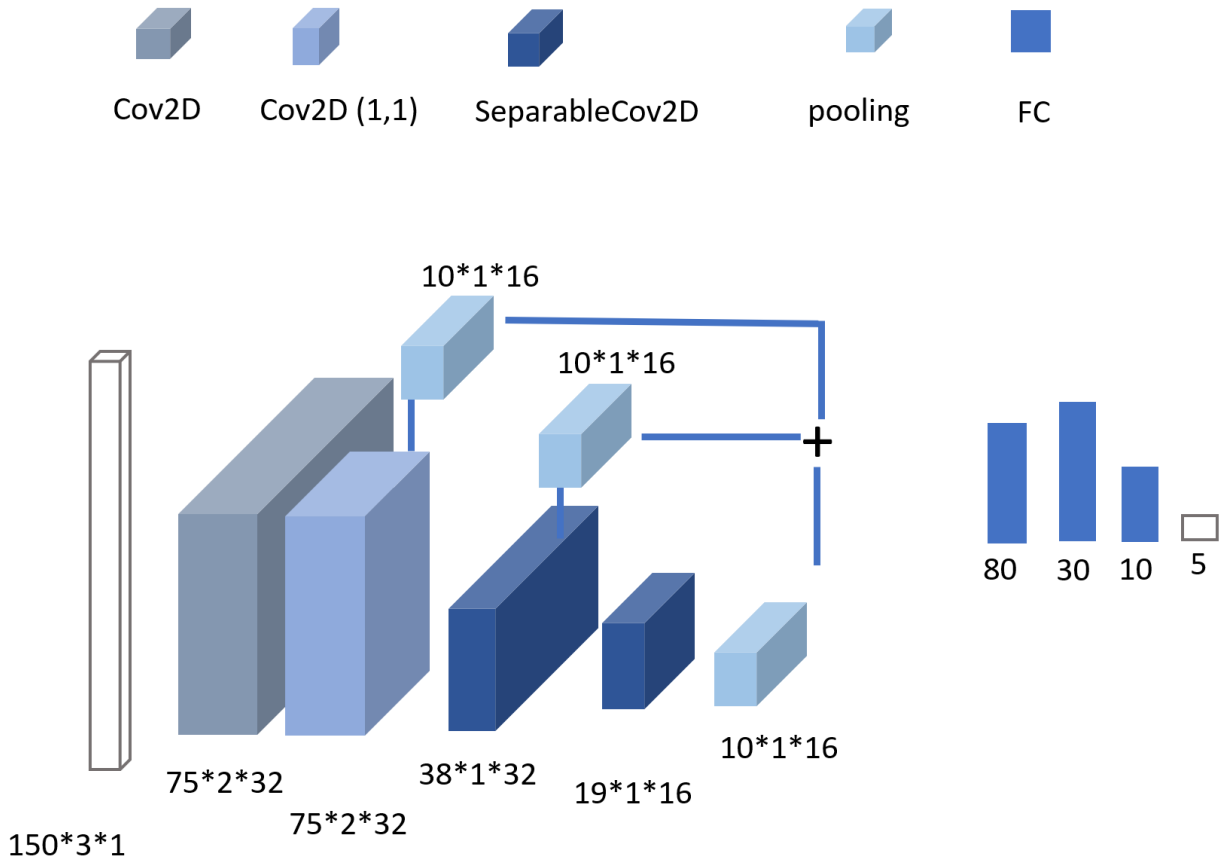


Figure 8.6: Simplified CNN.

1 because we do not have a large number of classes and the increasing temperature does not necessarily increase performance. Then, we were able to get a model with 9,763 trainable parameters. The student model could learn to match the performance of the teacher model without significant reduction in performance, as shown in table 8.2. (It is a basic experiment, more tuning and adjustment would be made in the future.)

To further reduce the stored model size and enable the model to run on MCU, we used quantization aware training to convert the parameter's data type to 8 bit. The final quantized model is reduced from 30 Mb to 0.0216Mb and is ready to be implemented and tested on a

Table 8.3: Normalized Confusion Matrix of Quantized Model.

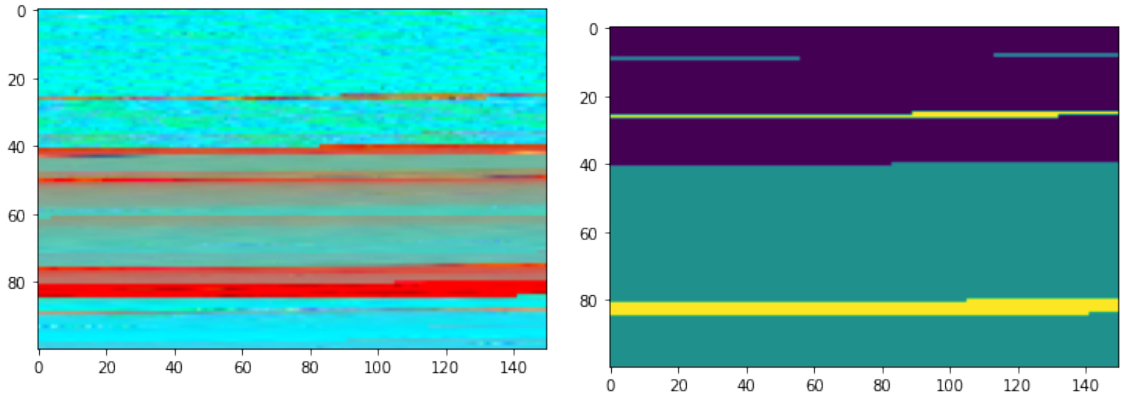
	grazing	lying	ruminating	standing	walking
grazing	99.0	0.0	0.2	0.1	0.7
lying	0.0	88.5	0.5	10.9	0.2
ruminating	0.2	0.4	98.3	0.8	0.3
standing	0.3	17.9	1.1	80.1	0.7
walking	1.8	0.0	0.3	1.7	96.3

MCU. The performance of the quantized model is similar to the student model and is shown in table 8.3

8.2 Instance Segmentation

Inspired by the instance segmentation in computer vision, we try to predict dense labels. Instead of predicting a single label for a fixed window size or using sliding windows, each data point will have a corresponding label. The logic behind this is in figure 8.7. If we plot the accelerometer data in RGB, the unique characteristic of each activity could be distinguished, so the CNN could learn the patterns to predict individual labels for every point and define the precise activity boundary.

We designed a U-Net-based structure. The feature maps from previous layers are concatenated to later layers so the detailed information could be used during transposed convolution. The detailed architecture is in figure 8.8. The input is an array with 150 accelerometer data points from three-axis within one channel, so the 3*3 kernel could extract the relationship between different axis, similar to the concept compared in figure 8.3. The output would be five stacks of masks corresponding to each activity. The convolution and max-pooling layers will extract the features from the accelerometer data and the transposed convolution will learn the kernel to expand the mask to predict labels for every data point. The convolutional



(a) The accelerometer data plot in RGB color.

(b) The mask of different activity.

Figure 8.7: The idea behind dense label prediction

layer with a 1*1 kernel is after every convolutional layer with a 3*3 kernel to learn the linear combination of cross-channel feature maps to reduce the dimension. In the last layer, a convolutional layer with a 1*1 kernel and five filters are used to predict the activity masks. The model is trained with Adam to optimize with a learning rate monitor to reduce the learning rate on the plateau. Because the walking activity is the minority and each training instance

Table 8.4: Normalized Confusion Matrix of Dense Label Predictions.

	grazing	lying	ruminating	standing	walking
grazing	99.1	0.0	0.2	0.2	0.5
lying	0.0	89.3	1.0	9.5	0.2
ruminating	0.4	0.7	97.7	0.7	0.5
standing	0.7	19.9	2.2	75.5	1.8
walking	4.5	0.1	1.3	2.9	91.3

may contain multiple activities, we could not use a random sampling method to upsample the walking activities. We used the focal loss to help address this problem, but it seems the recall value for minority classes still needs improvement in table 8.4.

Another problem related to dense label prediction is activities may be similar for a small segment. For example, when cows are walking, they may have various speeds, their head may

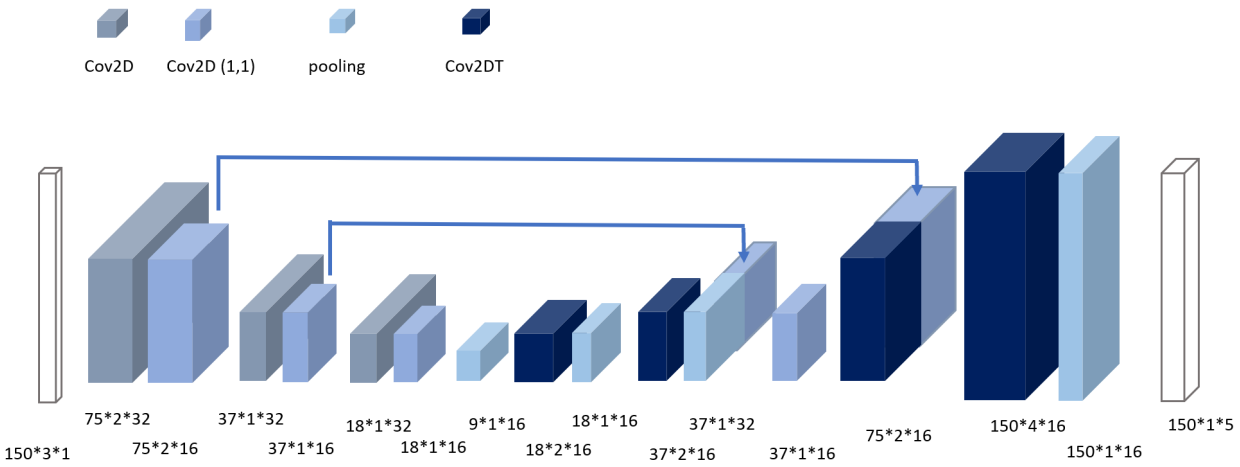
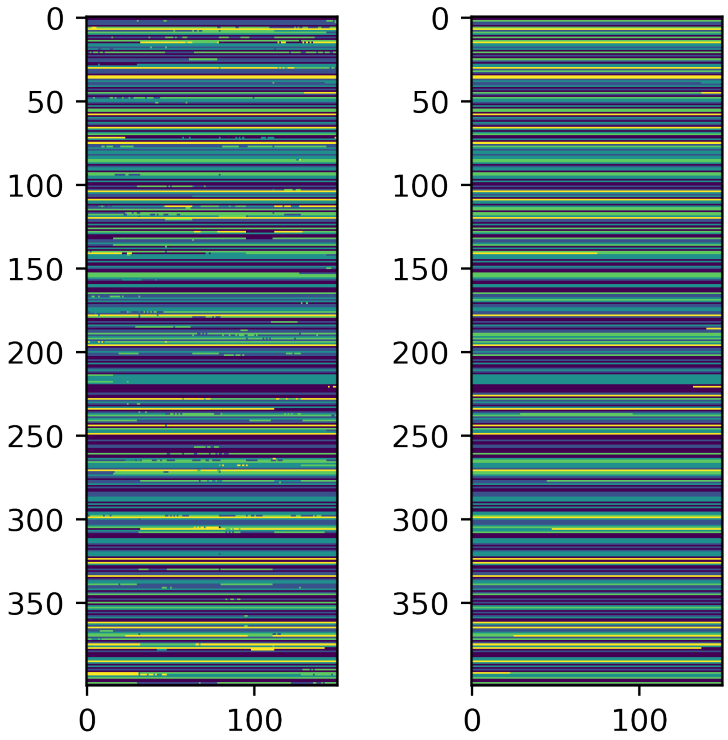


Figure 8.8: cnn.

move up and down. The time segment of walking activity with head down may look similar to grazing activities. Some lying activities are similar to standing activities as well. Therefore, the predicted mask are noisy as shown in figure 8.9 To force the network to learn the distribution of activities, we adopt the conditional Generative Adversarial Network (cGAN) structure to train the network [24]. The model in figure 8.8 is used as a generator and the model in figure 8.6 is used as a discriminator. The generator will receive the acceleration data and train to generate the mask that tricks the discriminator. The discriminator will receive the acceleration data with either the predicted mask or the ground truth mask to determine whether the mask is generated by the generator. In this training loop, the discriminator will provide feedback to help train the generator to produce the mask that is similar to ground truth. Thus, the discriminator will find the generated mask if the mask is too noisy, and the generator will try to reduce the noisy output. The drawback could be the generator is affected too much by the discriminator so it did not try to generate a mask based on acceleration data, but tried to generate a mask with different activity labels to trick the discriminator. To avoid this situation, we replace the l1 loss in custom loss function with



(a) The Predicted Mask. (b) The Ground Truth Mask.

Figure 8.9: The Ground Truth and Predicted Masks.

focal loss and added more weight on focal loss to the generator. We train the model using Adam optimizer and adjust the learning rate manually for every 20000 steps. Figure 8.10 is a mask generated by the cGAN and the preliminary results show some improvement in table 8.5. Another drawback of cGAN is the difficulty in training. The co-evolution loop could be stopped easily when one model stopped learning, while the task for the generator is harder and the discriminator could easily determine the generated models. How to efficiently use the feedback from the discriminator would need more work in the future.

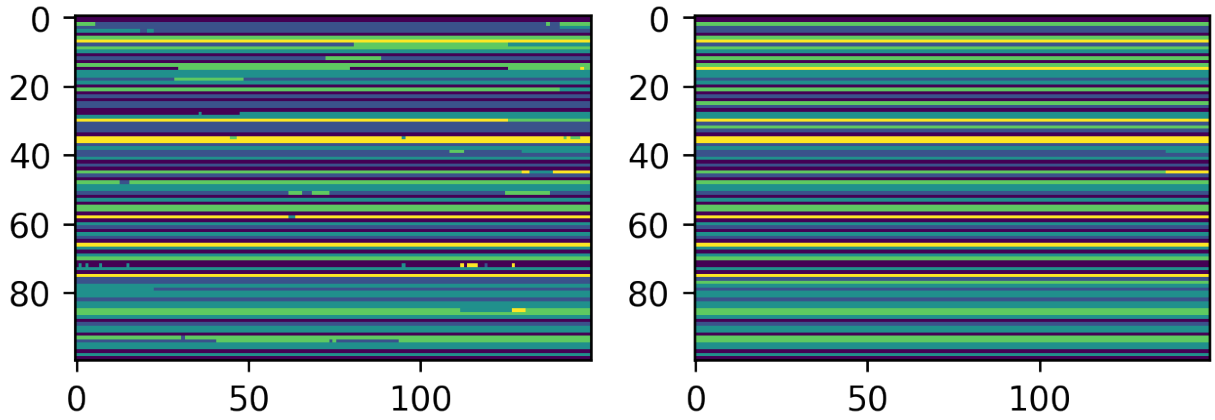


Figure 8.10: Generated Mask Train with cGAN and Ground Truth Mask.

Table 8.5: Normalized Confusion Matrix of cGAN Predictions.

	grazing	lying	ruminating	standing	walking
grazing	96.8	0.0	1.1	0.0	2.1
lying	0.0	80.6	0.0	19.4	0.0
ruminating	0.0	1.8	97.3	0.7	0.1
standing	0.0	3.8	0.0	91.8	4.3
walking	0.9	0.0	0.0	0.7	98.5

Chapter 9

Conclusions

We presented the performance of machine learning algorithms to classify cattle behaviors from the acceleration data. We collected the data for 85 hours in total in 4 days from four cows. The major activities considered for the classification are 'grazing', 'ruminating', 'lying', 'standing', and 'walking'.

We applied both the traditional machine learning approach and the deep learning approach. In the traditional machine learning approach, we extracted 52 features based on the characteristic of the activities and successfully eliminated 22 features to reduce the training time while the negligible impact on the balanced accuracy. With the remaining 30 features, we applied four classification algorithms, SVM, k-NN, RF, and HGBDT. Among the four algorithms, Among the four algorithms, HGBDT achieves the highest accuracy. In the deep learning approach, we implemented different types of CNNs to predict the activity labels or the activity masks. We applied the distilled learning and quantization to reduce the model size so the model could be implement on a MCU in the future. For the CNN that generate activity mask, we used cGAN structure to train the model and pushed the model to generate more accurate dense labels.

In the future, we will consider additional activities such as sleeping, which could be helpful to assess the health state of cows more accurately. Also, additional sensors might be helpful to distinguish between 'lying' and 'standing' activities. We will improve on the cGAN to maximize utility the feedback from discriminator.

Bibliography

[1] N. S. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992. ISSN 00031305. URL http://www.jstor.org/stable/2685209.

[2] Reza Arablouei, Lachlan Currie, Brano Kusy, Aaron Ingham, Paul L. Greenwood, and Greg Bishop-Hurley. In-situ classification of cattle behavior using accelerometry data. *Computers and Electronics in Agriculture*, 183:106045, 2021. ISSN 0168-1699. doi: https://doi.org/10.1016/j.compag.2021.106045. URL https://www.sciencedirect.com/science/article/pii/S0168169921000636.

[3] C. Arcidiacono, S.M.C. Porto, M. Mancino, and G. Cascone. Development of a threshold-based classifier for real-time recognition of cow feeding and standing behavioural activities from accelerometer data. *Computers and Electronics in Agriculture*, 134:124–134, 2017. ISSN 0168-1699. doi: https://doi.org/10.1016/j.compag.2017.01.021. URL https://www.sciencedirect.com/science/article/pii/S0168169916309917.

[4] Jamie Barwick, David William Lamb, Robin Dobos, Mitchell Welch, Derek Schneider, and Mark Trotter. Identifying sheep activity from tri-axial acceleration signals using a moving window classification model. *Remote Sensing*, 12(4), 2020. ISSN 2072-4292. doi: 10.3390/rs12040646. URL https://www.mdpi.com/2072-4292/12/4/646.

[5] Mariana Belgiu and Lucian Drăguţ. Random forest in remote sensing: A review of applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing*,

114:24–31, 2016. ISSN 0924-2716. doi: https://doi.org/10.1016/j.isprsjprs.2016.01.011. URL https://www.sciencedirect.com/science/article/pii/S0924271616000265.

[6] Said Benaissa, Frank A.M. Tuyttens, David Plets, Toon de Pessemier, Jens Trogh, Emmeric Tanghe, Luc Martens, Leen Vandaele, Annelies Van Nuffel, Wout Joseph, and Bart Sonck. On the use of on-cow accelerometers for the classification of behaviours in dairy barns. *Research in Veterinary Science*, 125:425–433, 2019. ISSN 0034-5288. doi: https://doi.org/10.1016/j.rvsc.2017.10.005. URL https://www.sciencedirect.com/science/article/pii/S003452881730423X.

[7] Sebastian D. Bersch, Djamel Azzi, Rinat Khusainov, Ifeyinwa E. Achumba, and Jana Ries. Sensor data acquisition and processing parameters for human activity classification. *Sensors*, 14(3):4239–4270, 2014. ISSN 1424-8220. doi: 10.3390/s140304239. URL https://www.mdpi.com/1424-8220/14/3/4239.

[8] Gérard Biau and Erwan Scornet. A random forest guided tour. *TEST*, 25, 11 2015. doi: 10.1007/s11749-016-0481-7.

[9] Daniel Castro, Steven Hickson, Vinay Bettadapura, Edison Thomaz, Gregory Abowd, Henrik Christensen, and Irfan Essa. Predicting daily activities from egocentric images using deep learning. *Proceedings. International Symposium on Wearable Computers*, 2015, 10 2015. doi: 10.1145/2802083.2808398.

[10] Konstantinos Charalampous and Antonios Gasteratos. On-line deep learning method for action recognition. *Pattern Analysis and Applications*, 19:337–354, 2014.

[11] Yuqing Chen and Yang Xue. A deep learning approach to human activity recognition based on single accelerometer. pages 1488–1492, 10 2015. doi: 10.1109/SMC.2015.263.

[12] Yuwen Chen, Kunhua Zhong, Ju Zhang, Qilong Sun, and Xueliang Zhao. Lstm networks for mobile human activity recognition. 01 2016. doi: 10.2991/icaita-16.2016.13.

[13] Edward Choi, Andy Schuetz, Walter Stewart, and J. Sun. Using recurrent neural network models for early detection of heart failure onset. *Journal of the American Medical Informatics Association*, 24:ocw112, 08 2016. doi: 10.1093/jamia/ocw112.

[14] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 233–240, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933832. doi: 10.1145/1143844.1143874. URL https://doi.org/10.1145/1143844.1143874.

[15] Thomas G. Dietterich. Ensemble methods in machine learning. In *MULTIPLE CLASSIFIER SYSTEMS, LBCS-1857*, pages 1–15. Springer, 2000.

[16] Ritaban Dutta, Daniel Smith, Richard Rawnsley, Greg Bishop-Hurley, James Hills, Greg Timms, and Dave Henry. Dynamic cattle behavioural classification using supervised ensemble classifiers. *Computers and Electronics in Agriculture*, 111:18–28, 2015. ISSN 0168-1699. doi: https://doi.org/10.1016/j.compag.2014.12.002. URL https://www.sciencedirect.com/science/article/pii/S0168169914003123.

[17] Margherita Grandini, Enrico Bagli, and Giorgio Visani. Metrics for multi-class classification: an overview, 2020.

[18] Paul Greenwood, Philip Valencia, Leslie Overs, David Paull, and Ian Purvis. New ways of measuring intake, efficiency and behaviour of grazing livestock. *Animal Production Science*, 54:1796–1804, 09 2014. doi: 10.1071/AN14409.

[19] Ergun Gumus, Niyazi Kilic, Ahmet Sertbas, and Osman N. Ucan. Evaluation of

face recognition techniques using pca, wavelets and svm. *Expert Systems with Applications*, 37(9):6404–6408, 2010. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2010.02.079. URL https://www.sciencedirect.com/science/article/pii/S0957417410001181.

[20] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, 01 2002. doi: 10.1023/A:1012487302797.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015. URL http://arxiv.org/abs/1502.01852.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

[23] Masaya Inoue, Sozo Inoue, and Takeshi Nishida. Deep recurrent neural network for mobile human activity recognition with high throughput. *Artificial Life and Robotics*, 23, 11 2016. doi: 10.1007/s10015-017-0422-x.

[24] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *CoRR*, abs/1611.07004, 2016. URL http://arxiv.org/abs/1611.07004.

[25] Jacob W. Kamminga, Duc V. Le, Jan Pieter Meijers, Helena Bisby, Nirvana Meratnia, and Paul J.M. Havinga. Robust sensor-orientation-independent feature selection for animal activity recognition on collar tags. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(1), March 2018. doi: 10.1145/3191747. URL https://doi.org/10.1145/3191747.

[26] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

[27] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017. URL http://jmlr.org/papers/v18/16-365.html.

[28] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. 12 2013.

[29] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *CoRR*, abs/1708.02002, 2017. URL http://arxiv.org/abs/1708.02002.

[30] Paula Martiskainen, Mikko Järvinen, Jukka-Pekka Skön, Jarkko Tiirikainen, Mikko Kolehmainen, and Jaakko Mononen. Cow behaviour pattern recognition using a three-dimensional accelerometer and support vector machines. *Applied Animal Behaviour Science*, 119(1):32–38, 2009. ISSN 0168-1591. doi: https://doi.org/10.1016/j.applanim.2009.03.005. URL https://www.sciencedirect.com/science/article/pii/S0168159109000951.

[31] Gabriele Mattachini, Elisabetta Riva, Francesca Perazzolo, Ezio Naldi, and Giorgio Provolo. Monitoring feeding behaviour of dairy cows using accelerometers. *Journal of Agricultural Engineering*, 47:54, 03 2016. doi: 10.4081/jae.2016.498.

[32] Henry Friday Nweke, Ying Wah Teh, Mohammed Ali Al-garadi, and Uzoma Rita Alo. Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges. *Expert Systems with Applications*, 105:233–261, 2018. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2018.03.056. URL https://www.sciencedirect.com/science/article/pii/S0957417418302136.

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

[34] A. Rahman, D.V. Smith, B. Little, A.B. Ingham, P.L. Greenwood, and G.J. Bishop-Hurley. Cattle behaviour classification from collar, halter, and ear tag sensors. *Information Processing in Agriculture*, 5(1):124–133, 2018. ISSN 2214-3173. doi: https://doi.org/10.1016/j.inpa.2017.10.001. URL https://www.sciencedirect.com/science/article/pii/S2214317317301099.

[35] S. Reiter, G. Sattlecker, L. Lidauer, F. Kickinger, M. Öhlschuster, W. Auer, V. Schweinzer, D. Klein-Jöbstl, M. Drillich, and M. Iwersen. Evaluation of an ear-tag-based accelerometer for monitoring rumination in dairy cows. *Journal of Dairy Science*, 101(4):3398–3411, 2018. ISSN 0022-0302. doi: https://doi.org/10.3168/jds.2017-12686. URL https://www.sciencedirect.com/science/article/pii/S0022030218300419.

[36] B. Robert, B.J. White, D.G. Renter, and R.L. Larson. Evaluation of three-dimensional accelerometers to monitor and classify behavior patterns in cattle. *Computers and Electronics in Agriculture*, 67(1):80–84, 2009. ISSN 0168-1699. doi: https://doi.org/10.1016/j.compag.2009.03.002. URL https://www.sciencedirect.com/science/article/pii/S0168169909000490.

[37] Bradley Robért, Brad White, David Renter, and Robert Larson. Determination of lying behavior patterns in healthy beef cattle by use of wireless accelerometers. *American journal of veterinary research*, 72:467–73, 04 2011. doi: 10.2460/ajvr.72.4.467.

[38] B. Schoelkopf, K. Sung, C. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik. Comparing support vector machines with gaussian kernels to radial basis function classifiers. Technical report, USA, 1996.

[39] Daniel Smith, Ashfaqur Rahman, Greg J. Bishop-Hurley, James Hills, Sumon Shahriar, David Henry, and Richard Rawnsley. Behavior classification of cows fitted with motion collars: Decomposing multi-class classification into a set of binary problems. *Computers and Electronics in Agriculture*, 131:40–50, 2016. ISSN 0168-1699. doi: https://doi.org/10.1016/j.compag.2016.10.006. URL https://www.sciencedirect.com/science/article/pii/S0168169916303180.

[40] M.L. Stangaferro, R. Wijma, L.S. Caixeta, M.A. Al-Abri, and J.O. Giordano. Use of rumination and activity monitoring for the identification of dairy cows with health disorders: Part i. metabolic and digestive disorders. *Journal of Dairy Science*, 99(9): 7395–7410, 2016. ISSN 0022-0302. doi: https://doi.org/10.3168/jds.2016-10907. URL https://www.sciencedirect.com/science/article/pii/S0022030216303940.

[41] Shoujin Wang, Wei Liu, Jia Wu, Longbing Cao, Qinxue Meng, and Paul J. Kennedy. Training deep neural networks on imbalanced data sets. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 4368–4374, 2016. doi: 10.1109/IJCNN.2016.7727770.

[42] Rui Yao, Guosheng Lin, Qinfeng Shi, and Damith Chinthana Ranasinghe. Efficient dense labeling of human activity sequences from wearables using fully convolutional networks. *CoRR*, abs/1702.06212, 2017. URL http://arxiv.org/abs/1702.06212.

[43] Mi Zhang and Alexander Sawchuk. A feature selection-based framework for human activity recognition using wearable multimodal sensors. 01 2011. doi: 10.4108/icst.bodynets.2011.247018.

[44] Yong Zhang, Yu Zhang, Zhao Zhang, Jie Bao, and Yunpeng Song. Human activity recognition based on time series analysis using u-net, 09 2018.