FACULTY OF SCIENCES

Internet of Animals: Foaling detection based on accelerometer data

Timo De Waele
Student number: 01402316

Supervisors: Prof. dr. ir. Wout Joseph, Prof. dr. ir. Eli De Poorter
Counsellors: Dr. ir. Margot Deruyck, Anniek Eerdekens, Ir. Jaron Fontaine

Master's dissertation submitted in order to obtain the academic degree of
Master of Science in de informatica

Academic year 2019-2020

GHENT UNIVERSITY

FACULTY OF SCIENCES

Internet of Animals: Foaling detection based on accelerometer data

Timo De Waele
Student number: 01402316

Supervisors: Prof. dr. ir. Wout Joseph, Prof. dr. ir. Eli De Poorter
Counsellors: Dr. ir. Margot Deruyck, Anniek Eerdekens, Ir. Jaron Fontaine

Master's dissertation submitted in order to obtain the academic degree of
Master of Science in de informatica

Academic year 2019-2020

GHENT UNIVERSITY

Het internet der dieren: veulendetectie aan de hand van accelerometer data

Timo De Waele

Supervisors: Prof. dr. ir. Wout Joseph, Prof. dr. ir. Eli De Poorter
Counsellors: Dr. ir. Margot Deruyck, Ir. Anniek Eerdekens, Ir. Jaron Fontaine

Masterproef voorgelegd voor het behalen van de graad: Master in de Informatica.

Academiejaar 2019-2020

Faculteit Wetenschappen
Universiteit Gent

Samenvatting

Er wordt veel mankracht gestoken in het observeren van zwangere merries om een goed verloop van de bevalling te verzekeren. Automatische observatie van de zwangere merries zou paardeneigenaars kunnen geruststellen. Dit onderzoek stelt een methode voor die veulendetectie kan uitvoeren aan de hand van accelerometer data. Een op een autoencoder gebaseerd anomalie detectie algoritme werd ontwikkeld dat het normale gedrag van de merrie kon onderscheiden van het gedrag dat de merrie vertoonde wanneer de bevalling werd ingezet. Verschillende autoencoder architecturen en andere verbeteringen zoals de discrete Fourier transformatie van de accelerometer data werden geëvalueerd om de performantie van het algoritme te verbeteren. Door een dynamische beslissingsmetriek die zijn beslissing of een merrie op het punt staat te bevallen of niet baseerd op bepaalde statistieken van elke merrie apart werden veelbelovende resultaten geboekt. Uiteindelijk werden alle bevalling correct herkend maar voor sommige merries werden valse gedetecteerd in de dagen voor de bevalling.

Keywords

Paarden, veulendetectie, gedragdherkenning, machine learning, autoencoder, anomalie detectie, accelerometer

Internet of Animals: Foaling detection based on accelerometer data

Timo De Waele

Supervisors: Prof. dr. ir. Wout Joseph, Prof. dr. ir. Eli De Poorter
Counsellors: Dr. ir. Margot Deruyck, Ir. Anniek Eerdekens, Ir. Jaron Fontaine

Master's dissertation submitted in order to obtain the academic degree of Master of Science in de informatica

Academic year 2019-2020

Faculty of Sciences
Ghent University

Summary

Lots of effort is put into the monitoring of pregnant mares to ensure a healthy delivery of the foal. Automatic monitoring of the pregnant mares and their unborn foals could bring horse owners peace of mind. In this research a method is proposed to perform foaling detection based on accelerometer data. An autoencoder based anomaly detection algorithm was developed that could distinguish the mare's normal behavior from the behavior shown when the mare entered labor. Different autoencoder architectures and other enhancements such as the discrete Fourier transform of the accelerometer values were evaluated to enhance the performance of the algorithm. By using a dynamic decision metric that based its decision if a foaling is about to take place or not on certain statistics of each mare individually promising results could be achieved. In the end all foalings got correctly detected but some mares still showed one or more false alarms in the days before parturition.

Keywords

Equines, foaling detection, behavior detection, machine learning, autoencoder, anomaly detection, accelerometer

Internet of Animals: Foaling detection based on accelerometer data

Timo De Waele

Supervisors: Prof. dr. ir. Wout Joseph (promotor), Prof. dr. ir. Eli De Poorter (promotor), Dr. ir. Margot Deruyck, Ir. Anniek Eerdekens, Ir. Jaron Fontaine

Abstract— **In this research data acquired from an accelerometer was used to develop a foaling detection algorithm. The proposed method made use of anomaly detection using an autoencoder to detect behaviors that indicated the start of labor. Several different configurations and parameters were evaluated to improve the performance of the algorithm.**

Keywords— **Equines, foaling detection, behavior detection, machine learning, autoencoder, anomaly detection, accelerometer**

I. Introduction

With over 16 million horses worldwide, the equine industry results in 1.6 million full time jobs and a total global revenue of more than 270 billion euros [1]. It is clear that a lot of money is involved in this growing sector and a major part of it is the breeding of top sport horses and hence the selling of their sperm and embryos, with a single straw of sperm costing up to €8,000 and embryo's being auctioned off for more than €50,000 [2][3]. Therefore, the breeding of new foals with a good heritage includes financial and emotional involvement of the breeders. Automatic monitoring of pregnant mares and their unborn foals can bring horse owners peace of mind.

Many methods to predict the time of parturition already exist, such as looking at the size of the udder and inspecting the amount and character of mammary secretion [4]. Although, this indication is not exact and is mainly based on intuition built upon previous experience which makes it a subjective decision. To improve these predictions many different technologies have been developed to predict and recognize the time of parturition, such as FoalGuard, Foalert and Birth Alert [5] [6] [7]. But these all made compromises on either horse comfort, accuracy or ease of use.

In this abstract an autoencoder based anomaly detection algorithm will be developed that could be deployed for foaling detection. Several configurations and parameters of the proposed model will be evaluated to improve the performance of algorithm.

II. Methodology

A. Data collection procedure

The data acquisition was done in collaboration with the Ghent University clinic of large animal reproduction. During the 2019 foaling season 15 mares that were stabled there for observation during their pre-foaling period were fitted with a triaxial Axivity AX3 accelerometer (Axivity Ltd, Newcastle, United Kingdom). The sensors were attached to the halter in the orientation shown in figure 1. By attaching the device to the halter worn by all mares stabled at the clinic, the impact on the comfort of the mare was minimized.

Fig. 1. Direction of each axis in respect to the horse [8]

B. The Datasets

For each dataset the accelerations on all three axes were captured at 50 Hz with a range of -8 g to +8 g. An overview of the size of the dataset per mare is shown in figure 2. Because of the high sampling rate each dataset grows quickly to large proportions. To reduce the computational load for handling the amount of data each dataset was reduced to a 1 Hz sampling rate. This was done by taking the average of each group of 50 continuous samples. The individual behaviors of each mare were still identifiable but the computational load was drastically reduced.

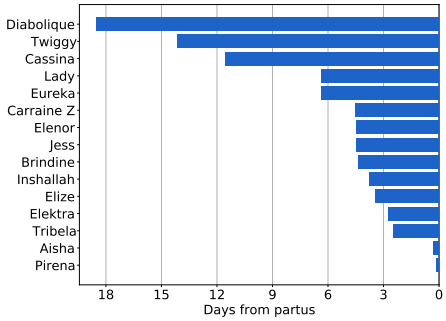

Fig. 2. Total time of movement data in days before partus for the participating horses.

III. Anomaly detection model

A. Overview

An autoencoder based anomaly detection algorithm was used as a basis for the foaling detection system. The main benefit of this approach was that it could be trained unsupervised. This

was necessary due to the limited amount of foaling events resulting in a heavily unbalanced dataset. The idea behind using an autoencoder to perform anomaly detection is to train the autoencoder on regular data only. This makes it overfit on reconstructing regular data making it perform worse on data that significantly differs from its training set. Because pregnant mares often show signs of restlessness and symptoms of colic when they enter stage one of parturition this idea could be used for detecting the start of foaling since this behavior is significantly different from the mares normal behavior [9].

B. Architecture

The architecture of the autoencoder consisted of two convolutional layers for both the encoder and the decoder part of the network. By using convolutional layers the network could perform automatic feature extraction and learn certain features during training. Next to this architecture two other architectures were evaluated as well, one that consists of recurrent layers and another one that is a combination of both recurrent and convolutional layers. In figure 3 a visualization of the three different types of autoencoder architecture is given.
The input of the autoencoder was set at a fixed number of samples and thus a fixed timeframe, these were obtained from the acquired data via a sliding window approach. Table I lists an overview of the different tweakable parameters used in this study.

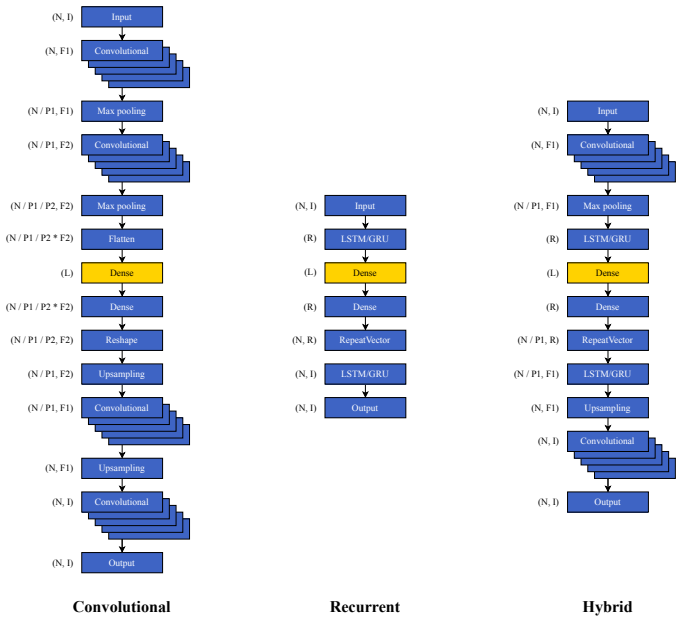

Fig. 3. The three used autoencoder architectures. The annotations indicate the layer output size with a description of each parameter given in table I

IV. Results

Out of the 15 datasets that were captured, 11 contained more than 3 days of pre-foaling data. These 11 mares were divided into a training and an evaluation dataset, the training set containing data of 6 mares, the evualtion set containing the data of the other 5 mares. All of the networks were trained using the parameters shown in table II unless specified otherwise.

Parameter	Description
N	Number of input samples
M	Sliding window stride length
F1, F2, ...	# of convolutional filters
P1, P2, ...	Size of max pooling window
L	# of dimensions of the latent space
R	# of units of the recurrent network
SR	Sampling rate of the network input, in Hz

TABLE I
AUTOENCODER HYPERPARAMETERS

Parameter	Value
Sampling rate	1 Hz
Input window length	1800 (30 minutes)
Stride length	900 (15 minutes)
Number of convolutional filters	64-32
Pooling size	1 (no pooling)-10
Activation function	ReLu
Batch size	32
Epochs	100
Loss function	MSE
Optimizer	Adam

TABLE II
TRAINING PARAMETERS

A. Influence of the architecture

To evaluate the influence the precise architecture had on the performance of the autoencoder several experiments were performed. Several configurations for each type of autoencoder were tried out, varying the input sizes, number of filters, base architecture, sampling rate, etc. But in the end this only resulted in differences in the absolute values of the reconstruction errors. Since the decision if a certain window is anomalous or not depends only on the shape of the reconstruction error signal and not the absolute value the different configurations made no difference on the anomaly detection performance of the autoencoder. In figure 4 a an example of the reconstruction error signal leading up to parturition for a random mare between all three architectures is given, none of the three resulted in a peak close to parturition.

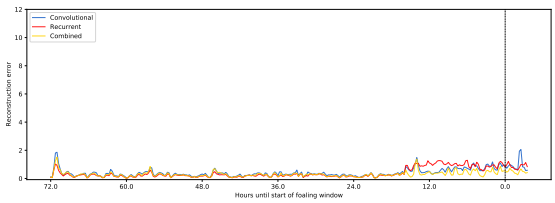

Fig. 4. Comparison of reconstruction errors for the 3 different architectures

B. Method of standardization

One of the most influential parts of the proposed method was the way the data was standardized. Two different methods of standardization were evaluated. First, data was standardized per mare to reduce the influence of halter placement and mare size on the network. Second, each input window was standardized separately so the autoencoder input has a mean of 0 and a standard deviation of 1. This facilitates the autoencoder 's learning to reconstruct its inputs.
When comparing the two methods, as shown in figure 5, it can be seen that while standardizing per mare a large peak in the reconstruction error appears at parturition. However, when standardizing per input window this peak completely dissappears and the entire signal is almost completely flat. With standardization per input window the near-parturition data and the regular data become completely indistinguishable to the network, standardizing the data per mare is thus necessary to make the proposed algorithm function correctly.

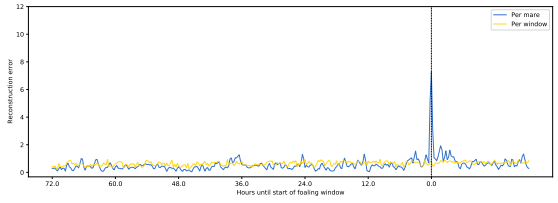

Fig. 5. Comparison of reconstruction errors for both methods of normalization

C. Discrete Fourier transform

By using the acceleration values as input to the autoencoder the network becomes sensitive to the orientation of the accelerometer. The network could get confused if the sensor suddenly shifts or gets mounted upside down as the baseline of the data is now different to what it has seen during training. A way to alleviate the influence is to transform the input windows from the time to the frequency domain by applying the discrete Fourier transform [10]. In the frequency domain the data becomes much less dependent on the exact orientation of the sensor as it now consists of the frequencies that are part of the signal and not the absolute acceleration values.
In figure 6 an example is presented of the reconstruction errors for both accelerometer values as an input and the DFT of these values as an input. The regular model shows no clear peak close to parturition but the one that makes use of the DFT does. However, to further evaluate the influence of the DFT on the performance of the proposed method more data is required.

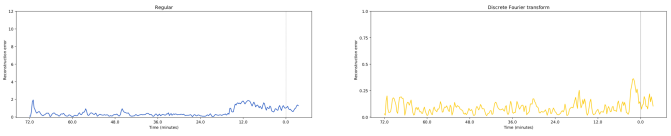

Fig. 6. Reconstruction errors for an autoencoder trained with acceleration values as input (left) and one that was trained with the DFT of the acceleration values as its input (right)

D. Other performed experiments

In addition to these three experiments several others were performed as well. These include trying out a custom loss function during training, applying transfer learning to update the model with specific knowledge of each mare, using the latent space of the autoencoder to do predictions, etc. All of these experiments showed no visible improvement to the reconstruction error signal that was used for anomaly detection.

E. Making a decision

The final step of the anomaly detection algorithm proposed in this study is making a decision based on the reconstruction errors of the autoencoder. This can be done in several ways but the method proposed is just setting a threshold. If the reconstruction error for a certain window goes above this set threshold an alarm is triggered. This threshold can be set in many different ways, being either statically were the threshold is the same for each mare, or dynamically were the threshold is different for each mare. A dynamically set threshold is preferred as the height of the peaks and the baseline of the reconstruction error signal can differ significantly between mares.
To set this threshold each mare should first go trough an analysis phase were the values of the reconstruction errors get analyzed to decide the value of the threshold. In table III the performance of a number of these thresholds, based on statistics of the reconstruction errors during the analysis phase, are presented. The best results in terms of true positives and false positives were obtained by using a threshold based of the mean plus a fixed number.
Out of the 11 mares, 11 foalings were succesfully recognized and 7 mares resulted in one or more false alarms in the three days leading up to parturition. With more data and tweaking of the number of standard deviations to add to the mean the last method of deciding a threshold could also prove as succesful. The benefit of adding a number of standard deviations instead of a fixed value is that it automatically adjusts the threshold to the variability of the reconstruction error signal of each mare.

Method	TP	FP	FN
$\mathbf{max}$	8	9	3
$\mathbf{max + 1}$	6	5	5
$\mathbf{mean + 1}$	11	7	0
$\mathbf{mean + 1.5}$	10	5	1
$\mathbf{mean + 3\sigma}$	11	9	0
$\mathbf{mean + 5\sigma}$	10	7	1

TABLE III
OVERVIEW OF THE NUMBER OF CORRECT PREDICTIONS/TRUE POSITIVES (TP), FALSE ALARMS/FALSE POSTIVES (FP) AND UNDETECTED FOALINGS/FALSE NEGATIVES (FN) FOR A DYNAMICALLY CHOSEN THRESHOLD

V. CONCLUSION

In this research an algorithm is proposed to detect foalings from accelerometer data based on an anomaly detection model using an autoencoder. By training the autoencoder on regular

behavior a metric can be used to decide if a given input is showing behavior common to mares entering labor or not based on the reconstruction error. By making this metric dynamically adjust to each mare specifically 11 out of the 11 foalings that were used for evaluation got succesfully detected.
Out of these 11 mares there were still some that triggered one or more false alarms leading up to parturition. Several methods were proposed and evaluated to reduce the amount of false positives but due to a lack of data no conclusion could be made. Future work should include the acquisition of new datasets to further evaluate these proposed improvements. Studies about the influence of different sensor locations on the performance of the proposed algorithm should be conducted as well.

REFERENCES

[1] P. Cross, "Global horse statistics internal 02 2019", Feb. 2019.

[2] Jan. 2020. [Online]. Available: https://www.t-online.de/sport/id_75116260/totilas-mitbesitzer-paul-schockemoehle-senkt-preise-fuer-wunderhengst-samen.html.

[3] Jan. 2020. [Online]. Available: https://www.flandersfoalauction.be/nl/nieuws/Grand-finale-Flanders-Foal-Auction-sluit-af-met-20108-euro-gemiddeld.

[4] P. M. McCue and R. Ferris, "Parturition, dystocia and foal survival: A retrospective study of 1047 births", *Equine Veterinary Journal*, no. 44, pp. 22–25, 2012.

[5] 2007. [Online]. Available: http://www.foalguard.com.

[6] 2019. [Online]. Available: https://foalert.com.

[7] L. A Bate, D. Hurnik, and J. G. Crossley, "Benefits of using a photoelectric alert system for swine farrowing operations'", *Can. J. Anim. Sci.*, vol. 71, pp. 909–911, 1991.

[8] [Online]. Available: https://www.premierequine.co.uk/plain-padded-horse-head-collar-c2x21459520.

[9] T. S. Mair *et al.*, *Equine Medicine, Surgery and Reproduction*, 2nd edition. Edinburgh: Elsevier, 2013.

[10] A. Roxburgh, "On computing the discrete fourier transform", Dec. 2013.

Lay summary

In this study an algorithm was designed that triggers an alarm when a mare was about to give birth. This system made use of data about the movements of the horse. To acquire this data a sensor that could detect these movements, called an accelerometer, was attached to the halter the mare was wearing. There were two ways this type of system could be developed, the first is to manually go trough to the data and look for specific signs that indicate the start of foaling. Then based on the findings of this step a computer program could be written that would detect these signs and trigger an alarm. Not only would it be very time consuming to go trough the data manually as this consisted of many millions of data points, it would also result in an immensely complex computer program as these signs could depend on hundres or even thousands of variables. Because developing such a program would be virtually impossible a second approach was used for this study, machine learning. In machine learning a computer can, based on complex mathematical formulas and algorithms, learn to solve complex problems on its own, without human intervention. The human only needs to define the space the computer can search trough to look for a solution, after which the computer can train to solve the problem on its own by letting it look at lots of examples of already solved problems. When this training phase is done the computer can be given unsolved problems and solve these on its own by using the knowledge learned during training.

To apply machine learning for developing a foaling detector the idea was to let the computer learn what regular horse behavior looks like. Once it had learned this, an input where the mare was showing abnormal behavior, such as rolling, flank watching, etc. when entering labor, would confuse the computer would get confused as it doesn't know anything about this type of behavior. By measuring this confusion an alarm could be triggered once this confusion goes above a certain threshold. Several different methods were evaluated during this study to improve the capabilities of the computer to distinguish regular from irregulal behavior, such as transforming the data about the movements of the mare or using different types of mathematical formulas to let the computer learn what regular behavior looks like.

In the end the computer could correctly detect all of the 11 foalings that it was given to evaluate the performance. While this may seem as a perfect

result, the proposed approach also resulted in 7 mares giving false alarms, which could lead to alarm fatigue where people don't take an alarm serious as it has a high chance of being a false alarm. Because of this future research should mainly focus on getting the amount of false alarms as low as possible.

Contents

Chapter 1

Introduction

1.1 Problem description

With over 16 million horses worldwide, the equine industry results in 1.6 million full time jobs and a total global revenue of more than 270 billion euros [1]. It is clear that a lot of money is involved in this growing sector and a major part of it is the breeding of top sport horses and hence the selling of their sperm and embryos, with a single straw of sperm costing up to €8,000 and embryo's being auctioned off for more than €50,000 [2][3]. Therefore, the breeding of new foals with a good heritage includes financial and emotional involvement of the breeders. Automatic monitoring of pregnant mares and their unborn foals can bring horse owners peace of mind.

More than 10% of pregnant mares suffer from dystocia during foaling, which can be recognized by a prolonged or failure of progression of the first or second stages of parturation [4][5]. In most of the cases, dystocia occurs due to a malposture of the foal in the uterus or birth canal [6]. This event requires early detection to prevent the foal from dying of asphyxiation. If stage 2 of the labor takes more than 40 minutes, the percentage of foal mortality increases to 20% [4]. The gestation duration of horses can be highly variable, ranging from 300 days up to more than 360 days, and is dependent on many factors such as period of insemination, sex of the foal, breed and hereditary factors [7][8]. Therefore, lots of effort goes into the 24/7 monitoring of pregnant mares. For example, at the Ghent University veterinary clinic, teams of veterinarians and veterinary medicine students monitor the stabled pregnant mares 24/7 and assist with foalings[9].

By looking at several features of the pregnant mare, like the vulva laxity, vulvar discharges, relaxation of the pelvic ligaments, but especially the size of the udder and the amount and character of mammary secretion, observers can get a strong indication of when the parturation is about to take

place [5]. Although, this indication is not exact and is mainly based on intuition built upon previous experience which makes it a subjective decision. Therefore, a lot of research, which will be discussed in the next section, is done in developing automatic foaling detection systems resulting in different technologies.

1.2 Related research

Most research in the field of not only foaling detection but also calving and farrowing detection is focused on two aspects of the pregnant animal as predictors, namely the temperature and/or the behavior of the animal.

Research has shown that there is a significant decrease in body temperature in both mares and cows the day before parturition [10] [11] [12]. To use this as a predictor for parturition continuous body temperature monitoring is required, this can either be done manually by using a rectal or tympanic infrared thermometer or by reading out an implantable microchip transponder [13]. But this approach is time consuming since human interventions are required for every reading. Using a telemetric gastrointestinal pill could result in more frequent measurements with wireless transmission to a base station, thus alleviating the need for a human intervention [14]. However, by requiring a sensor belt to house the receiving and transmitting equipment, this system imposes a burden on the horse's comfort, with excessive wear of the surcingle possibly contributing to rubbing on the mare. Because of the drawbacks of both methods, despite body temperature being a good predictor for the detection of foaling, it is hard to implement in practice.

Another feature that is shown to be useful for the prediction of parturition is the behavior of the animal. With small activity trackers that incorporate wireless transmission capabilities and extensive battery life becoming more and more prevalent and affordable, this has the potential to be a practical approach for foaling detection. Research has shown that a significant difference in behavior can be observed in the period leading up to to parturition for horses but for cows and pigs as well [15] [16] [17] [18] [19]. The difference in behavior is not as well-defined as the change in body temperature and thus further analysis is required to build a predictor out of it. The total locomotor activity as well as the frequency and total duration of standing, lying, eating and other well-defined behaviors like tail raising, flank watching, urinating et cetera, are found to be useful features in a predictive model. By using these features as input to a machine learning model, researchers have been able to develop a calving detector with good performance [20]. An quick summary of the related research is given in table 1.1.

Based on this and other research, some foaling detectors have been put into production and are used in the field today. A short overview of some of these systems will be given in the next section.

Study	**Animal**	**Features**
Body temperature and behaviour of mares during the last two weeks of pregnancy (Shaw et al., 1988)	Horse	Body temperature, frequency of shown behaviors
Body temperature fluctuations in the periparturient horse mare (Cross et al., 1992)	Horse	Body temperature
Methods and on-farm devices to predict calving time in cattle (Saint-Dizier & Chastant-Maillard, 2015)	Cow	Body temperature, tail raising, lying bouts, clinical signs
Detection of the time of foaling by accelerometer technique in horses (Equus caballus)—a pilot study (Aurich et al., 2018)	Horse	Increase in activity
Monitoring of total locomotor activity in mares during the prepartum and postpartum period (Bazzano et al., 2015)	Mare	Increase in activity
Predicting farrowing based on accelerometer data (Hietaoja et al., 2013)	Pig	Increase in activity
Predicting farrowing of sows housed in crates and pens using accelerometers and CUSUM charts (Hietaoja et al., 2016)	Pig	Increase in activity
Prediction of parturition in Holstein dairy cattle using electronic data loggers (Bas et al., 2015)	Cow	Number of steps, lying bouts, lying time, standing time
Machine-learning-based calving prediction from activity, lying, and ruminating behaviors in dairy cattle (Bewley et al., 2017)	Cow	Number of steps, lying bouts, lying time, standing time
Internet of Animals: Foaling detection based on accelerometer data (De Waele, 2020)	Horse	Acceleration values of the mare's head

Table 1.1: Overview of related research

1.3 Current technology

The different foaling alert systems can be broadly categorized into 3 different categories, namely systems that work by using sensors placed externally on the mare, systems that use a device in or around the vagina/vulva of the mare and external monitoring tools.

External sensor based systems

Several foaling detection systems work by using accelerometers and/or a gyroscopes to determine if the mare is in a lateral recumbent position. The main benefits of these systems are the fact that no surgical intervention is required for placement and usage is not limited to stabled mares. Drawbacks are that they are prone to false positives, e.g. if the mare uses a lateral recumbent position to rest or sleep, that could lead to alarm fatigue, as well as false negatives during dystocia in the initial parturition phase or when the mare is not laterally recumbent during foaling.

Two examples of this type of system are FoalGuard and Birth Alarm, FoalGuard, depicted in figure 1.1, works by using an accelerometer attached to the halter to determine the position of the mare which, because of the small form factor, only has a small negative impact on the comfort of the mare [21][22]. Birth Alarm, shown in figure 1.2, uses a gyroscope attached to a surcingle, it manages to obtain a lower occurence of false alarms by waiting a couple of minutes to see if the mare stays down to filter out occurences where the mare is sleeping. This results in a delayed alarm trigger but reduces the number of false positives and thus the risk of alarm fatigue. However, by attaching the gyroscope on top of the surcingle and thereby restricting the mare's freedom to roll over it, the horse's comfort is penalized.

Safemate Foalalarm, shown in figure 1.3, implements a different approach by using a sensor that senses perspiration to detect the start of foaling, this could however lead to false positives on warm days [23].

Internally placed systems

The Foalert system, shown in figure 1.4, uses two magnets that get sutured to either side of the mares vulva. The alarm gets triggered when the two magnets separate by the foal being pushed out of the mare, the alarm gets triggered [27]. The false positive rate is low because of the physical interaction of the foal being required to trigger the alarm, but this also has a negative effect on the number of false negatives in the case of dystocia in the early stages of parturation. Surturation requires veterinary assistance to mount and uninstall the device and also adversely affects the wearing

Figure 1.1: FoalGuard [24]

Figure 1.2: Birth Alarm [25]

Figure 1.3: Safemate Foalalarm [26]

comfort of the mare.

Another type of foaling detection system in this category is Birth Alert,

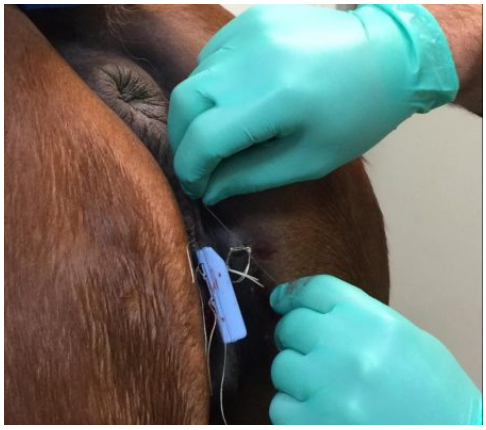

Figure 1.4: Foalert [29]

displayed in figure 1.5 [28]. This system consists of 2 parts, a light sensitive sensor, which gets placed inside of the mares vagina, and a microphone, once the light sensor gets pushed out of the vagina and starts detecting light again it will make a distinctive sound, that when detected by the microphone will result in an alarm. No veterinary assistance is required for installation, but it is prone to false positives when the sensor falls out on its own and it does suffer from the same false negative rate in case of dystocia as Foalert.

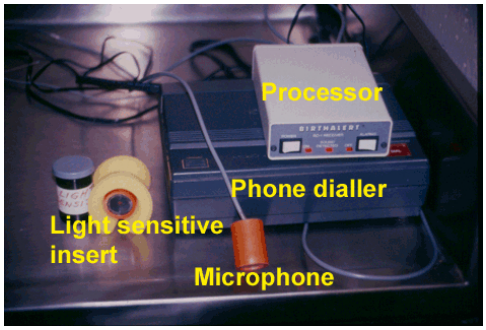

Figure 1.5: Birth Alert [30]

External monitoring tools

The last category of tools function without any placement of sensors on the mare but by using external cameras or microphones, and therefore they do not impede the comfort of the mare. The EquiView360 uses a camera placed in the stable and implements machine vision algorithms to track the behavior of the horse [31]. While it is primarily used for colic detection, it

could be modified for foaling detection use, but this is not yet tested.

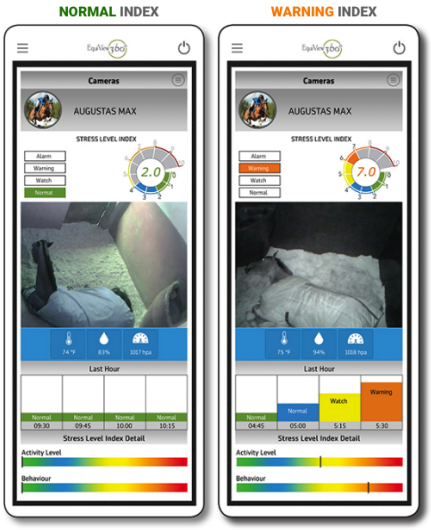

Figure 1.6: EquiView360 [32]

1.4 Followed approach

The goal for this thesis is to research the possibility to implement a foaling detection system based on accelerometer data that performs equally as good or better on both comfort and accuracy than current technologies. To achieve this, small accelerometers attached to the mare will be used to collect data about the behavior of the pregnant mare. This data will be passed trough our foaling detection machine learning algorithm that will trigger an alarm if a foaling is about to occur.

In this study, the anomaly detection subfield of machine learning will be explored for approximating the time of the partus. The rest of this thesis is organised as follows. In chapter 2 a description will be given of the data acquisition method and the followed approach for the exploring and cleaning of data, after which the proposed machine learning algorithm to tackle the problem will be further explained. Chapter 3 lists the conducted experiments and the obtained results. In chapter 4 these results will be further discussed and finally a conclusion will be drawn in chapter 5.

Chapter 2

Methodology

2.1 Data

For this study data from 15 expecting mares stabled at the Ghent University clinic of large animal reproduction was collected, from May 2019 to August 2019. The length of each dataset ranges from three hours to over two weeks prepartus. A complete overview of the size of the dataset per mare is given in figure 2.1. Out of these 15 mares, 13 entered labor between 10PM and 6AM, the other 2 gave birth around noon. One of these mares, Tribela, gave birth to a twin of foals, a rare occurence [33].

2.1.1 Procedure

The Axivity AX3 triaxial accelerometers (Axivity Ltd, Newcastle, United Kingdom), depicted in figure 2.2, were used for data collection. These were chosen for their compact size, broad range of configurability, robustness and extensive battery life. A full overview of the specs is given in table 2.1.

At the start of the measurements, each mare was equipped with two sensors, one placed on top of the withers on a surcingle and one attached on top of the halter, as shown in figure 2.3. There were however some issues with the sensor on surcingle. First it would slide down when the mare was active resulting in the sensor changing location which made it hard to use this data in practice. Another, more severe issue, was that due to the sliding the surcingle would rub against the mares withers and induce rubbing wounds, therefore it was chosen to not use the surcingle for data collection as the welfare of the mare was of the uttermost importance during this study.

The sensor on top of the halter was attached with ducttape to fix it firmly in place, after which a couple layers of cohesive bandage was wrapped around it to make sure no ducttape was rubbing against the mares head. The sensors were placed in the same orientation each time, with the logo facing down

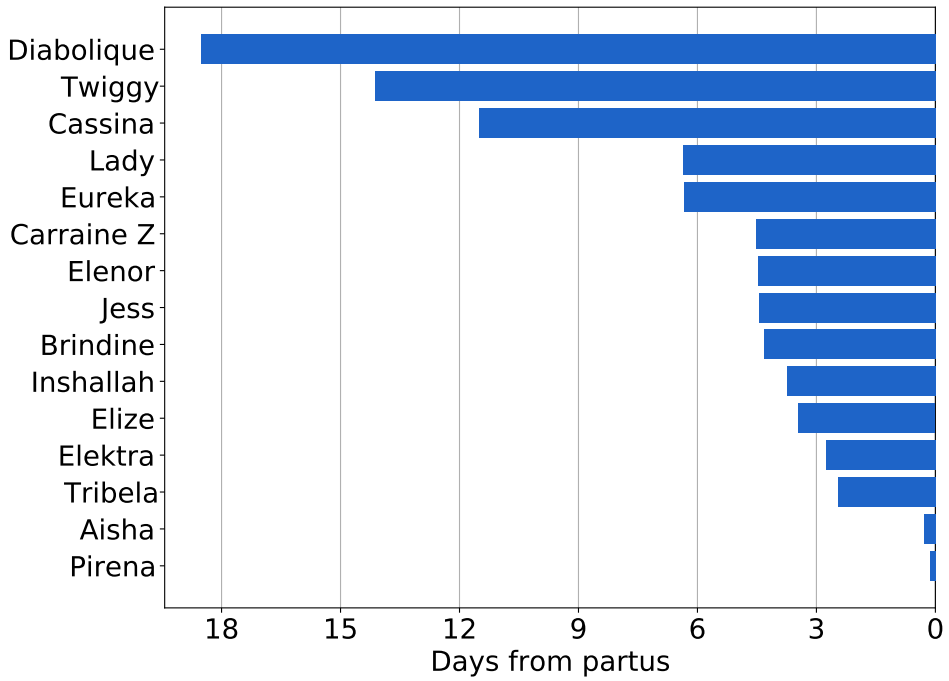

Figure 2.1: Total time of movement data in days before partus for the participating horses.

Figure 2.2: Axivity AX3 Triaxial Accelerometer [34]

and the USB port pointing to the right side of the mare, resulting in the axis orientation that is shown in figure 2.4. For one mare the sensor was placed upside down for some days but this was later fixed during the data cleaning step by flipping the x and z axis. Total time of movement data in days before partus for the participating horses. As for the recording parameters, a measurement range between -8 g and 8 g was used with a sampling frequency of 50 Hz as it gave a wide range of resampling possibilities while

still having ample battery life.

If the sensor was mounted, it remained on the mare for at least two days before being removed for downloading the data and checking whether it still worked correctly. For example, one sensor stopped recording after a few hours due to a defective battery, but this was during the first recording period so only two days of data were lost. The timestamps of the gaps created by removing the sensor were stored in a text file so that they could be fixed during the data preprocessing phase. Six stables at the veterinary clinic were equipped with CCTV cameras, the videos from these were downloaded and used for analyzing behavior leading up to the partus as well as to check for anomalies such as a removed halter, so a gap in the data could be recorded.

The raw data for each continuous dataset was then saved to a csv file, containing per sample the timestamp and the x, y and z acceleration values. Alongside this raw data a metadata file was stored per horse containing the name of the mare, the timestamps of when each recording was started and stopped and the moment the amniotic sac burst according to the observing students.

Figure 2.3: Placement of the accelerometers

2.1.2 Data cleaning

The first step of the data cleaning process was to trim each dataset to only contain the data of when the sensor was attached to the mare. This was

Figure 2.4: Direction of each axis in respect to the horse [35]

Parameter	**Value**
Dimensions	23 x 32.5 x 7.6 mm
Weight	11g
Moisture ingress	IPx8 1.5m for 1hr
Dust ingress	IP6x
Memory	512 MB Flash non-volatile
Acceleration Sample Rate	12.5 - 3200Hz Configurable
Battery Life	30 days @ 12.5Hz, 14 days @ 100Hz
Accuracy Range	± 2/4/8/16g Configurable
Accuracy Resolution	upto 13 bit

Table 2.1: Axivity AX3 Triaxal Accelerometer specifications

done by loading the csv containing the raw data via the python framework Pandas and only keeping the valid sections by removing the sections were the sensor was not yet on the mare. The trimmed data was then saved to disk. As a next step, these datasets were concatenated so that a single dataset was obtained for each horse containing all the data, which was then stored as a csv file. These datasets do contain gaps in the data from when the sensor was taken of for checking if it was still functioning correctly and the intermediate downloading of the data.

Because of the 50 Hz sampling rate each dataset grows quickly to large proportions. One day of logged data, for example, results in over four million datapoints. There are almost no existing visualization tools that can handle this amount of data easily. Therefore, a custom visualization tool was introduced using the D3.js platform to check raw data on errors or anomalies. The idea behind the visualizer was to resample the data at different zoom levels so only a limited number of datapoints is shown at each time while keeping the ability to zoom. By resampling the data during a preprocessing step and not during the runtime of the visualisation the overhead of zooming is reduced to only the time it takes to load and display the new datapoints. Two parameters for this type of visualization that need to be chosen is how many datapoints will be displayed at one given time as well as the number of regions we want to zoom into, these were set at 10.000 points and 10 regions of zoom for this thesis. In figure 1 a screenshot of the application is shown, the alternating white and gray sections each indicate a region on which the user can click to zoom in. To reduce the data from

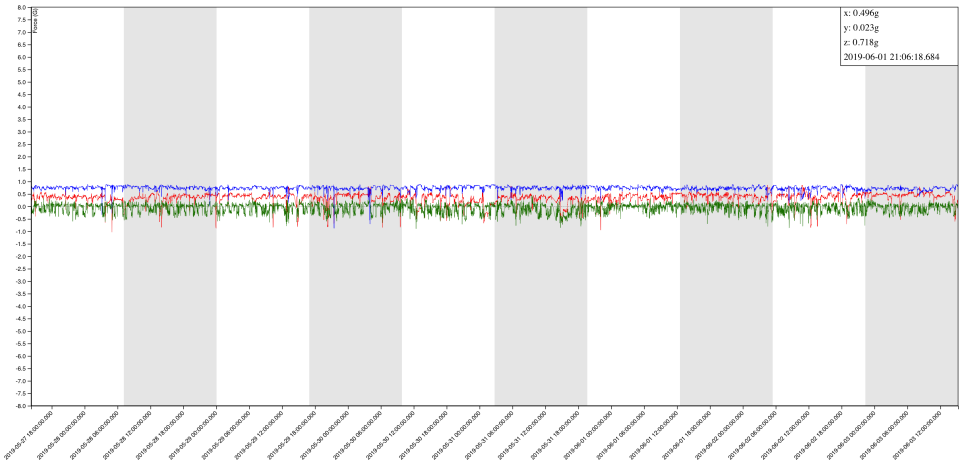

Figure 2.5: Screenshot of the visualisation application

millions of datapoints to only 10.000 points the timeframe of the data series was divided into 10.000 different sections of which the average was taken as the value of each section. This was done recursively for each level of zoom until the last level contained less than 10.000 points. Each sampled section was then stored to disk for use in the visualization.

By checking the data of each mare using the visualizer a couple of errors were found and manually corrected, for example, for one mare the sensor was attached upside down for a couple of days, thus flipping the orientation of the y and z axis. Another issue that was quickly spotted using the

visualization tool was that for another mare the timezone of the sensor was set incorrectly, resulting in a 6 hour difference between the actual time of a sample and the recorded timestamp, both of these issues had to be fixed up manually by loading and correcting the relevant dataset.

2.1.3 Data exploration

In this section an overview will be given of some of the most common behaviors seen in the hour leading up to foaling. First, the data was dowsampled from 50 Hz to 1 Hz, at this sampling rate the different behaviors were still easily distinguishable but the amount of processing power required to query and visualize the datasets was drastically reduced. Video footage of the hour before foaling was available for 10 out of the 15 mares, this footage was then used to detect the different types of recurrent behavior shown leading up to partus.

The first noticeable behaviour detected is pacing, i.e. the mare walks restlessly in circles in the stable. Figure 2.6 displays the accelerometer pattern of this behavior together with normal activity of the same mare for comparison. The most notable difference between the data of the normal behavior versus pacing around is the frequency of the peaks and valleys in the y axis signal, which are much more frequent when the mare is pacing around. This could be attributed to the fact that the mare is walking around and thus accelerating forward with each step.

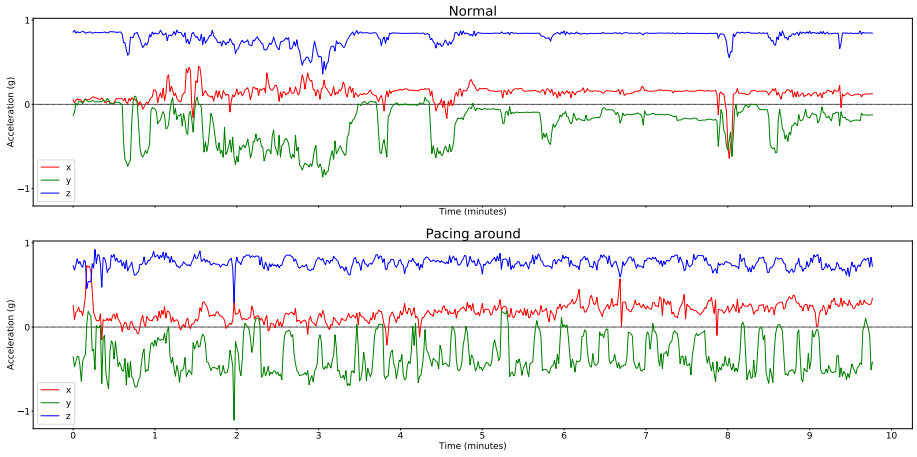

Figure 2.6: Comparison of normal behaviour vs. pacing around

The second observed behavior is headshaking together with flank watching where the mare would turn her head backwards to watch her side, often

combined with shaking her head upwards and/or sideways. This is shown as an accelerometer trace which is compared to normal behavior for the same mare in figure 2.7. The first indicated region is the mare shaking her head, which can be recognized by the high volatility of all three axis. The second region indicates flank watching, since for this action the mare needs to tilt her head sideways this results in a change of orientation of both the y and z axis.

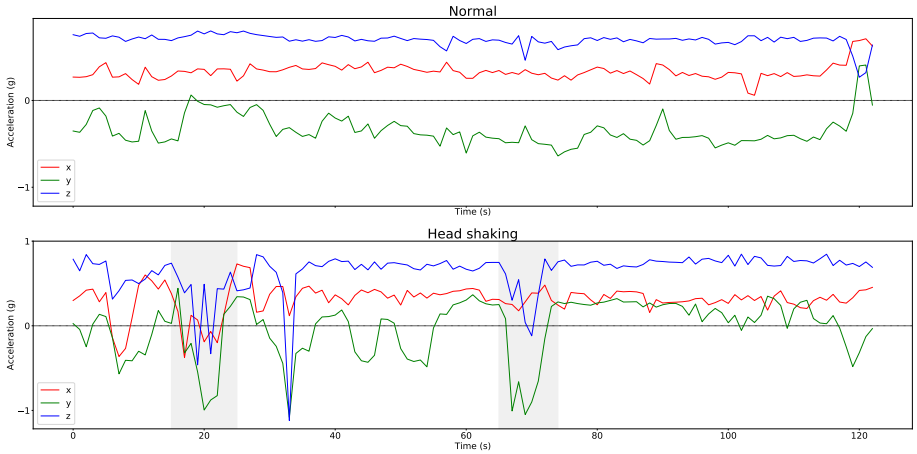

Figure 2.7: Comparison of normal behaviour vs. head shaking (first region) and flank watching (second region)

Out of the 10 mares for which video footage was available, 9 gave birth while lying down. Horses have two ways of lying down, being either sternally or laterally recumbent, in the first case the horse is lying on its sternum with its head still held up, in the latter case the horse is lying completely flat on its side with its head on the ground. In figure 2.8 the accelerometer data for both of these positions is shown, the darker sections indicate where the mare was lying laterally recumbent, the lighter section indicates a sternally recumbent position. Lateral recumbency is easily detectable since the z axis becomes negative as the mare puts her head down, the small peaks in the z value can be attributed to the mare lifting her head occasionally. Sternally recumbency on the other hand can be recognized by the fact that the values of the three axis are more distant from eachother as well as that the x axis now has the highest positive value and the y axis is now closer to zero. The signal while lying down is also less noisy than when the mare was standing upright since the mare mostly takes this position to rest in, thus result-

ing in less and smaller head movements. This behaviour is what is used in a couple of commercially available foaling detection systems as mentioned before, though most of them have the downside of producing many false positives as this is also one of the natural positions horses use to rest or sleep.

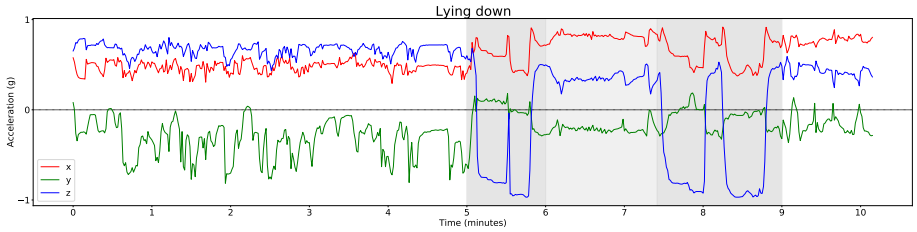

Figure 2.8: Accelerometer data of a mare lying in a sternally and laterally recumbent position

The final behavior that is often seen before foaling and that is easily detectable in the accelerometer data is rolling, as shown in figure 2.9. This motion can easily be distinguished in the data since reversal of all three axis is taking place, as the head of the mare moves upside down while performing a rolling motion. Most horses however will roll over when they are sweating or itchy so this behaviour is also not unique to the period leading up to foaling.

None of these behaviors on its own are good predictors for a foaling predic-

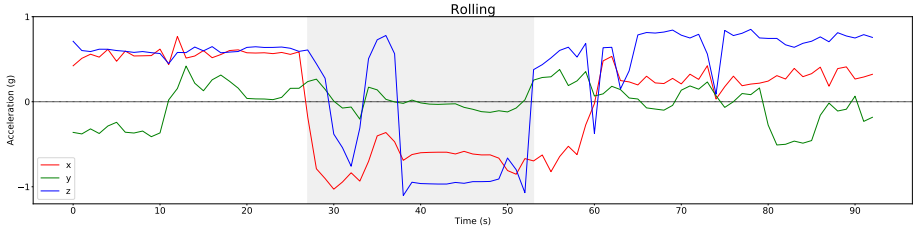

Figure 2.9: Accelerometer data of a mare performing a rolling motion

tion algorithm, combining them however and doing predictions based on the frequency and variability of each behavior and transitions between behaviors could potentially be used. Research has already shown that the detection of most of these behaviours based solely on accelerometer data is possible, unfortunately this was done by attaching the sensors to the forelegs of the

mare so the result of this research could not be used directly in this thesis [36]. Due to the lack of video recordings required to label data, the option of designing an algorithm for detecting the shown behavior was chosen not to further investigate.

2.2 Model

This section will describe the model used to tackle the problem of foaling detection based on the gathered accelerometer data. In deciding the approach that will be taken for this thesis, two problems had to be taken into consideration. First, the lack of video footage, this was only available for 10 out of the 15 observed mares and even for the ones that had footage it was only for a limited amount of time and not the entire period the mare was wearing the accelerometer. Because of this, the chosen approach can not make use of labeled behaviors since we lack the ground truth data to label and train a classifier model for these behaviors. The second issue to consider was the time we had as an indication for parturition. Students watching the mares wrote down the time of amniotic sac rupture, although this was more of a rough approximation than an accurate value since the rupture was mostly noticed after it had already happened. Because of this it would be hard to train a model that was just a classifier or a regression model since we have no precise ground truth to label and train the model with, the variability in the time that was noted could result into the model getting confused during training, the chosen approach thus had to be able to handle the uncertainity in its prediction variable. As a result of these two issues it was opted to go for an anomaly detection approach that was based on a model that could be trained unsupervised.

The main benefit of an unsupervised anomaly detection model is that it can be trained entirely without any labeled anomalous instances, which is preferable in this case since there is a large class inbalance as we only have one foaling event per mare lasting about 15 minutes but a couple of days worth of normal data per mare. The idea is to train the model to recognize normal data after which it could be used to detect samples that are significantly different to its training set. Because pregnant mares often show signs of restlessness and symptoms of colic when they enter stage 1 of parturition this idea could be used for detecting the start of parturition since this behavior is significantly different from the mares normal behavior [4]. There are several different methods available for performing unsupervised anomaly detection, such as principal component analysis and isolation forests but for this thesis it was opted to use a deep learning approach based on an

autoencoder neural network model architecture. The benefit of this type of model is the large amount of configurability such as the type of layers and the number of layers as well as for the final decision metric making it easy to adapt to almost every possible scenario.

The architecture that will be used in this research consists of two parts, the autoencoder that will transform the original input, and a second model or metric that will decide if a certain input is anomalous or not based on the output of the autoencoder. An autoencoder on its own consists of two parts, an encoder and a decoder network, the encoder transforms the input into a latent representation, the decoder then takes this latent representation and uses it to reconstruct the input. A visualization of this type of architecture is given in 2.10. This class of models can be used for anomaly detection by training the model to take regular data as an input and reconstruct it, the idea is that the model will be overfitted on normal data and will fail to correctly reconstruct the data when it gets anomalous data as an input as this will differ significantly from the normal data it was trained on to reconstruct. In the context of foaling detection this means that the network gets trained with data from mares that are still a couple of days away from parturition, so the network can learn to recognize and reconstruct the regular behavior of a mare. If the network then gets passed data from a mare close to foaling as input, this will contain a combination of behaviors that the network probably has not seen during training. This will result in a worse reconstruction of the input data the closer the mare gets to foaling as she is behaving increasingly abnormal. The possibility of using the reconstruction error of this autoencoder to detect when a mare is about to enter labor will be further researched in this thesis.

2.2.1 Model input data

The complete dataset of each mare firstly got resampled to a lower sampling rate than the original 50 Hz to reduce the computational load. This was done by taking the average of each group of continuous samples. For example, if the final sampling rate was 1 Hz, the average was taken over 50 samples. Whilst doing this the gaps in the data that occured due to the removal of the sensors on certain moments could be part of a continuous subset of samples, so special care was taken to make sure these were not present in the final datasets. This was done by leaving out the last $N - 1$ samples before each gap, where N stands for the size of the window over which was averaged. For a final sampling rate of 1 Hz, N is 1 so the last $N - 1 = 49$ samples before the start of each gap were left out. In this study, different sampling rates were evaluated as will be described in chapter 3.

The next step is to split of a part of each dataset where the mare is showing

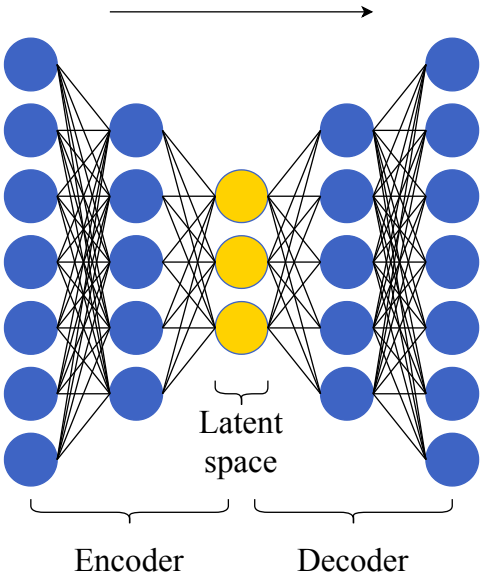

Figure 2.10: Visualization of an autoencoder architecture

regular behavior for use during training of the autoencoder. The threshold used for making the decision between regular behavior that will be used for training and non training data is also a hyperparameter that will be explained further on in this thesis.

The final step in transforming the datasets into input for the autoencoder was dividing each dataset into different smaller input subsets of equal length corresponding to the input size of the network. To do this the sliding window method was used where a subset of length N is taken followed by sliding the "window" forward by M steps, this is illustrated in figure 2.11. The choice for using a sliding window approach instead of just dividing the dataset in equally sized parts without stride was made so that every captured behavior was fully contained into at least one of the subsets. Otherwise a certain behavior could be partly in one subset and partly in the following subset so that no input subset contained the full behavior. The two parameters i.e., the length of the window and the stride length are again hyperparameters than can and will be changed during experiments.

2.2.2 Model architecture

There are many different types of autoencoder architectures, such as sparse autoencoders, contractive autoencoders and variational autoencoders, however for this thesis a regular autoencoder will be developed. The general shape of an autoencoder is one of an hourglass, where the in- and output

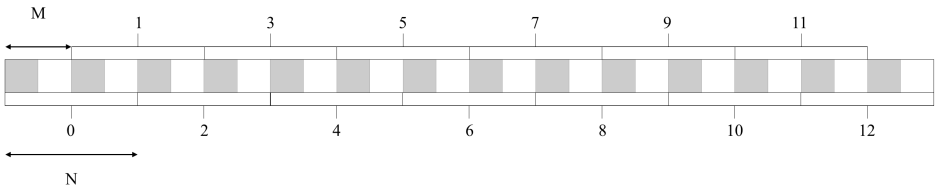

Figure 2.11: Sliding window method

layers contain a larger number of nodes than the internal nodes representing the latent representation. The simpelest form of an autoencoder is one of a simple feedforward, non-recurrent neural network, but this is not a requirement as the encoder and decoder layer can both take the shape of a broad variety of neural networks, containing not only dense layers but also convolutional and recurrent layers. In this thesis three different architectures will be evaluated, the first one was build up by using a convolutional network for the encoder and decoder, the second one is build up by using a recurrent network such as the long short-term memory network, and the final autoencoder architecture that was evaluated is a hybrid between these two, consisting of both convolutional and recurrent layers. Figure 2.12 shows an overview of these three architectures. All of the parameters for each type of layer, combined with the sliding window stride length and sampling rate of the input form the hyperparameters of the autoencoder network. A description for each of these parameters is given in table 2.2.

Parameter	**Description**
N	Number of input samples
M	Sliding window stride length
F1, F2, ...	# of convolutional filters
P1, P2, ...	Size of max pooling window
L	# of dimensions of the latent space
R	# of units of the recurrent network
SR	Sampling rate of the network input, in Hz

Table 2.2: Autoencoder hyperparameters

The idea behind the convolutional network is to allow the network to perform automatic feature extraction since the acquired data was not sufficient to support labeling. Due to the lack of video footage a seperate classifier to detect behaviors such as standing, walking, rolling... could not be trained. By training convolutional filters on the input data the network can train to recognize certain features on its own. The second type of architecture

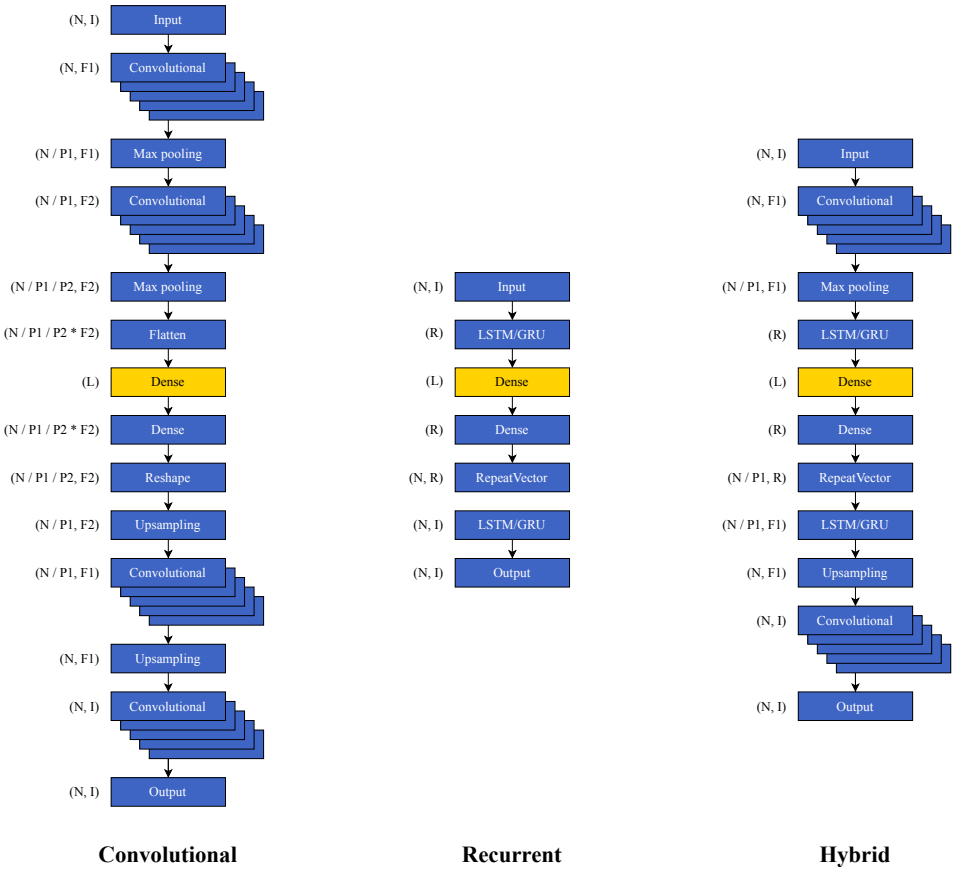

Figure 2.12: The three used autoencoder architectures. The annotations indicate the layer output size with a description of each parameter given in table 2.2

makes use of recurrent network layers instead of convolutional layers. In this thesis two types of recurrent layers were used and compared against eachother i.e., the long short-term memory network and the gated recurrent unit network [37] [38]. Recurrent networks have shown outstanding performance in everything that has to do with sequences of data, being textual or time series [39]. The core concept behind these two network types is the implementation of a memory state inside of the network. When feeding a sequence through the network, certain parts of the sequence can be recalled or forgotten and this memory eventually becomes the output of the network. The thought behind using this for modelling the behavior of the mare is that the network can keep track of certain behavioral aspects to use these for representing the entire sequence in the latent space. Since a recurrent network takes sequences as an input, a layer that repeats its input data a certain number of times was needed in the decoder to go from the scalar represent-

ation of the latent space back to a sequence of samples. The final approach that was evaluated was a combined approach with both convolutional and recurrent layers, first the input gets passed to a convolutional network to perform automatic feature extraction, these self-learned features then get sent to the recurrent network to transform it into a latent embedding.

Once the autoencoders are trained they can be used to perform anomaly detection by obtaining the stream of reconstruction errors for each mare. This is calculated as the mean squared error (MSE) between the input and output of the autoencoder, the formula for MSE is given in equation 2.1 [40].

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2 \tag{2.1}$$

Where N stands for the number of samples in each window, Y is the original input window and $\hat{Y}$ is the reconstructed input as given by the autoencoder.

This stream of reconstruction errors for each input window to the autoencoder can then be used to determine if a mare is about foal or not. The method of determining this will be explained in further detail in the following section.

2.3 Making a decision

An example of a stream of reconstruction errors for a given period and mare is given in figure 2.13. This stream is taken from an input of 100 windows of each 30 minutes with a stride length of 15 minutes and a sampling rate of 1 Hz. This means that if the first sample window started at 00:00 and ended at 00:30 then the second window would start at 00:15 and end at 00:45. Based on this stream of reconstruction errors a decision should be made on when the mare is about to foal, however due to the sliding window approach this can only be done based on each window. The precision of the prediction is thus dependent on the stride length and sample length of the inputs to the autoencoder. For example, if the stride length is set to 15 minutes then a prediction can only be made each 15 minutes based on the previous 30 minutes of data.

The idea is that the stream of reconstruction errors will change when the mare is getting closer to foaling, as she will start showing behaviors or combinations of behaviors the autoencoder does not know how to reconstruct as it is unseen and different data, thus increasing the reconstruction error. Based on this difference in reconstruction errors a decision then has to be made if the mare is close to parturition or not. This can be done in a number

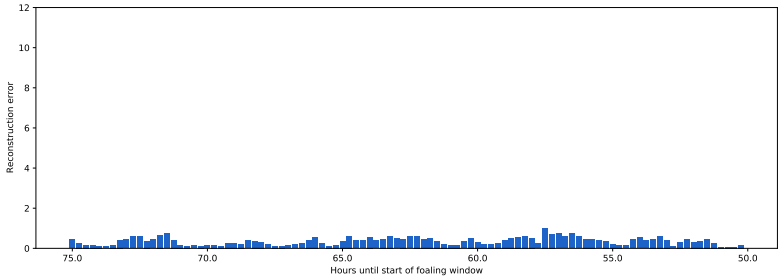

Figure 2.13: Example of reconstruction errors

of ways which will be explained in further detail in this section. In figure 2.14 an example is given of how the reconstruction errors change leading up to parturition for a given trained autoencoder. In this chart the last window is the window where foaling occured, the reconstruction error of the autoencoder becomes visibly larger leading up to parturition, it was well below 2 for the entire period but saw a gradual increase from about 2 to 3 hours before parturition and had a clear spike in the hour before parturition. For this case setting a fixed threshold that will trigger an alarm once the reconstruction loss goes above it could be a viable approach. The problem with using this approach is that sometimes the autoencoder will fail on reconstructing its input when the mare is not close to foaling, as shown in 2.15. If a single fixed threshold was used in this situation it would result in a false positive. There are two ways of fixing this problem, make the autoencoder better reconstruct the behaviors shown when the mare is not close to foaling. In some cases this is not possible however because some mares will show similar behaviors during a normal situation as when entering labor. This could lead to a false negative and an undetected birth. A second way of reducing the amount of false positives and false negatives would be to use a different metric of deciding if a mare is about to foal.

One of the other metrics that could be used is using a seasonal-trend decomposition to split the signal of reconstruction errors into its seasonal, trend and residual components [41]. In this decomposition the trend is the global increasing or decreasing value of the underlying signal, the seasonal component is the repeating signal of a given frequency included in the signal and the residual is the noise in the signal that cannot be explained by either the trend or the seasonal component, for the mares a seasonal frequency of 24 hours was taken to filter out the moments where the mares were restlessy waiting for food or were being walked by the observers of the veterinary clinic which was both done at a set time each day. An example of such a decomposition for the average norm of the acceleration vector over five

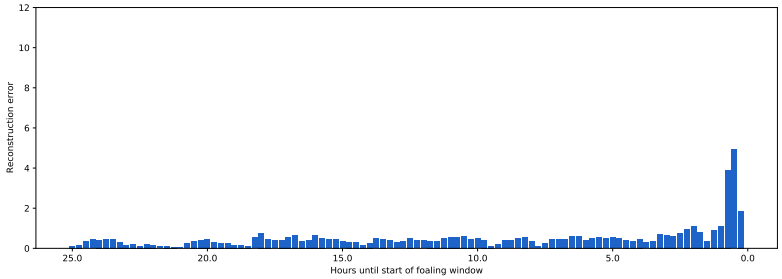

Figure 2.14: Reconstruction errors before foaling

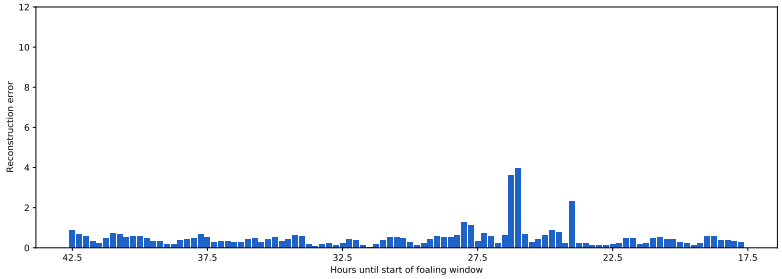

Figure 2.15: Reconstruction errors during normal behavior

minute windows of one of the mares is given in figure 2.16.

When this method gets applied to the stream of reconstruction errors a clear jump in the trend becomes visible leading up to parturition for most mares, an example of this is given in figure 2.17. This does show hope for using this signal as a predictor for parturition, however this suffers from the same drawback as the fixed threshold approach for certain trained autoencoders, as a jump in the trend of the decomposition could occur when the mare was still a couple of hours or days away from parturition, thus again resulting in false positives, as shown in figure 2.18. This is might be something that could be worked around with by using some heuristics such as the angle of the incline and the trend before the incline, some of these heuristics will be experimented with further on in this thesis.

There are lots of other methods for performing anomaly detection based on the reconstructions of the autoencoder, such as training a different classifier or regression model on the stream of reconstruction errors or applying a clustering algorithm to the latent representation, since the representation

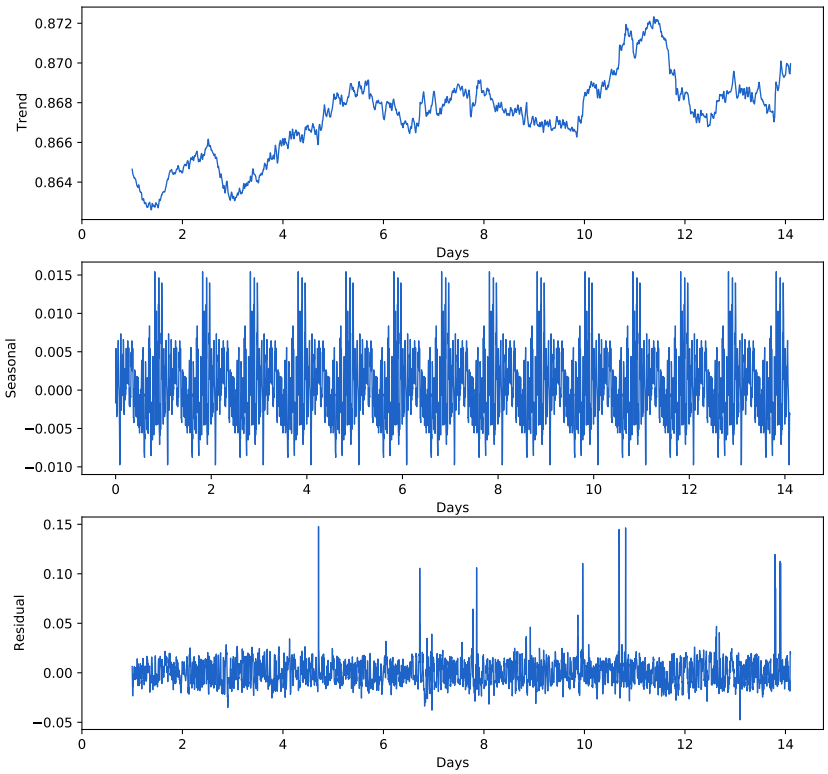

Figure 2.16: Example of seasonal-trend decomposition for a certain two week period

of the behaviors close to parturition could be significantly different to the ones far away from parturition. These methods however will not be studied further in detail for this thesis. In the following chapter the influence of the several hyperparameters of the autoencoders as well as the difference in performance for the three types of autoencoder architecture will be evaluated, the threshold and seasonal-trend decomposition metrics will be used during these experiments for the evaluation of each approach.

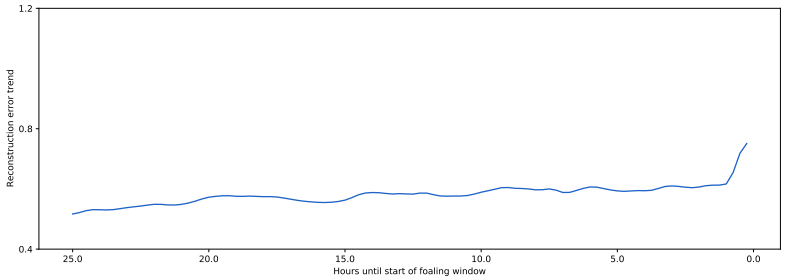

Figure 2.17: Trend of reconstruction errors before foaling

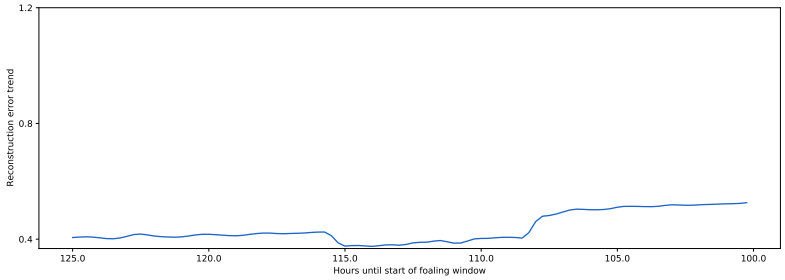

Figure 2.18: Trend of reconstruction errors during normal behavior

Chapter 3

Results

For all of the experiments, except for the leave-one-out crossvalidation and transfer learning, the dataset of 15 mares was split up into a training set and a validation set [1]. The training set consisted of 6 mares that had more than 3 days of data before foaling. The 3 days threshold was chosen as the splitting point between training data for the autoencoder where the mare was showing her regular behavior and the data close to foaling where the mare could start showing irregular behavior leading up to parturition, as illustrated in figure 3.1. The full datasets of the other 9 mares were used to evaluate the performance of the autoencoder on unseen mares during training as well as in the experimental fase. For all of the experiments except for the one involving the different methods of standardization the datasets were standardized per mare, for each individual mare the mean and standard deviation was calculated for all three axes and used to standardize them according to equation 3.1 [42]. This was done to alleviate influences in magnitude and orientation of the acceleration vectors due to differences in size of the mare and placement of the sensors on each mare. The sampling rate of the input data was fixed at 1 Hz to reduce computational load during training as well as speed up the data loading. At this rate all of the different behaviors were still distuingishable. The performance of the model on different sampling rates will still be evaluated in this chapter however.

$$\begin{aligned} x' &= \frac{x - \mu_x}{\sigma_x} \\ y' &= \frac{y - \mu_y}{\sigma_y} \\ z' &= \frac{z - \mu_z}{\sigma_z} \end{aligned} \tag{3.1}$$

[1] Due to the COVID-19 pandemic it was no longer possible to obtain an independent test set of data samples

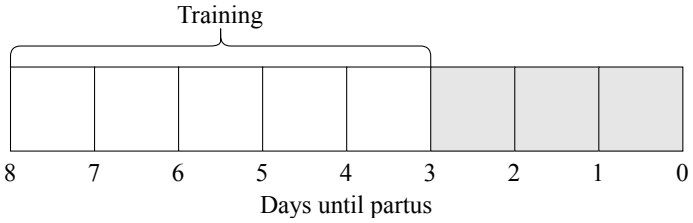

Figure 3.1: Section of data intended for training the autoencoder

During the first phase of this thesis the models were trained and evaluated using the HPC infrastructure of Ghent University [2]. This was equipped with a cluster containing a number of powerful Intel Xeon CPUs and Nvidia Tesla V100 GPUs. However due to the small size of the tested models and the added steps of transfering data between the HPC cluster and a local machine to train the models and perform visual analysis it was opted to use Google Colab [3] for training and evaluation of the models. Colab is a free to use platform were users can upload and edit jupyter python notebooks to run on virtual instances hosted by Google equipped with powerful GPUs to speed up the training of neural networks build in Tensorflow, specifications of this platform are given in table 3.1. The choice of programming language and software packages for this thesis were the defaults that the Google Colab runtime provided, these were Python 3.6.9 as a programming language together with the Tensorflow 2.2.0 and Keras 2.3.0-tf packages as deep learning library. All of the models were trained for 100 epochs with the adam optimizer and a starting learning rate of 0.001, mean squared error was used as loss function. If the training loss did not decrease during 10 epochs, the learning rate was reduced by a factor of 0.2, the kernel sizes for all convolutional layers was set fixed at 30 samples. The batch size was set at 32 input windows, in total there were 4144 training input windows and 5178 validation samples. The validation set consisted of 5 mares that had 3 days or more of prefoaling data and 4 that contained less than 3 days of prefoaling data which could thus not be used for training.

CPU	2 vCPU @ 2.2GHz
GPU	Nvidia Tesla K80, T4, P100
RAM	13GB

Table 3.1: Google Colab specifications

[2] https://www.ugent.be/hpc/en
[3] https://colab.research.google.com

3.1 Autoencoder architectures

3.1.1 Convolutional autoencoder

The first experiment performed was the influence of the number of convolutional layers on the reconstructive capabilities of the autoencoder. Therefore two models were compared against eachother, one with only one convolutional layer in both the encoder and decoder and one containing two convolutional layers in both encoder and decoder. A schematic of both networks is given in figure 3.2. After 100 epochs of training the first network with only one convolutional layer converged at a training loss of 0.36 and a validation loss of 0.44, the second network with two convolutional layers converged at a training loss of 0.37 and a validation loss of 0.45. Thus adding an extra convolutional layer to both the encoder and decoder did not seem to make a significant difference. This is further confirmed if the stream of reconstruction losses is plotted for a mare that had no clear increase in reconstruction error close to parturition, there is no clear improvement or difference between both autoencoders as can be seen in figure 3.3.

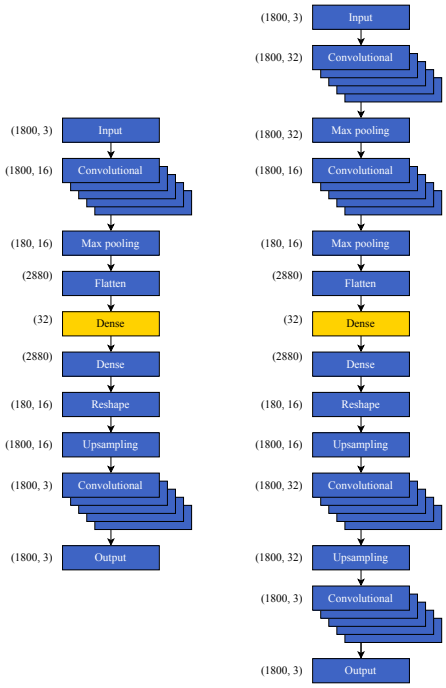

Figure 3.2: One layer autoencoder versus two layer autoencoder

Since the number of layers did not make a significant difference on the reconstruction performance of the autoencoder, the number of filters used

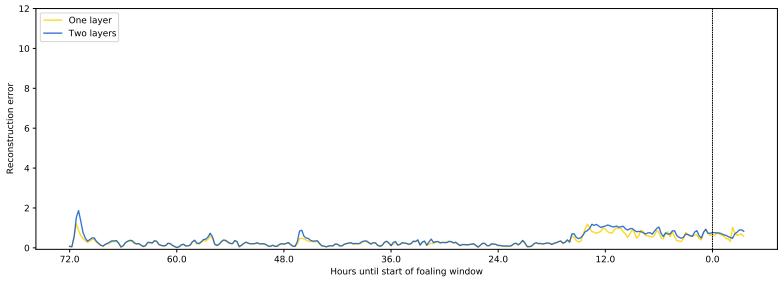

Figure 3.3: Reconstruction errors of both autoencoders

during convolutions was also evaluated. For this experiment the one layer convolutional autoencoder was used with three different numbers of filters: 16, 32 and 64 filters. The training results are given in table 3.2, there are some small differences between the losses but these do not affect the overall reconstructive abilities of the autoencoder as can be seen in the reconstruction error signal in figure 3.4.

# of filters	Training loss	Validation loss
16	0.36	0.44
32	0.36	0.46
64	0.35	0.46

Table 3.2: Training results for different numbers of filters

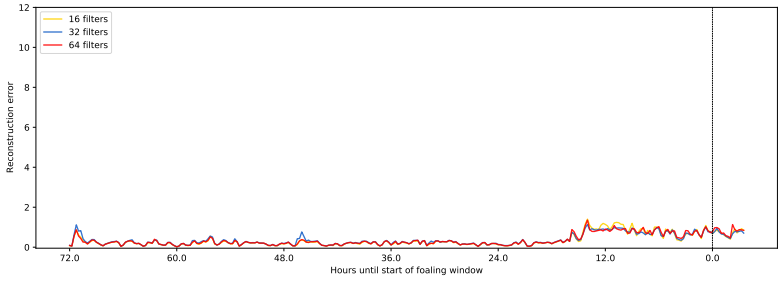

Figure 3.4: Comparison of reconstruction errors for all 3 configurations

3.1.2 Recurrent autoencoder

For the recurrent autoencoder four different configurations were evaluated. Both the influence of the number of recurrent features and the type of re-

current layer on the autoencoders performance was evaluated. The two layer types being the long short-term memory layer (LSTM) and the gated recurrent unit layer (GRU). The main difference between these two layers is that the LSTM has the output gate separate to its hidden state gate, whilst for the GRU, the hidden state is the same as its output gate. This reduces the amount of connections in the layer, making it potentially faster to train than an LSTM while keeping similar performance [43]. Due to the increased amount of time per epoch to train a recurrent network these networks were only trained for 50 instead of 100 epochs. Training results for all four configurations can be found in table 3.3. This shows that the gated recurrent unit scores slightly better for reconstructing its input. Adding more recurrent features also has a positive effect on the reconstruction loss of the autoencoder as the network can remember more information from the past. For the LSTM this does however inflict a penalty on the training time of the network. For the gated recurrent unit doubling the amount of features in its hidden state does not influence its training time significantly. This can probably be attributed to differences in implementation between the two layers. Figure 3.5 shows the plotted reconstruction error signal for a mare that does not have a clear increase in loss when nearing parturition. No useful difference, in terms of foaling prediction capabilities, between the four implementations can be seen, as the only difference is a higher average reconstruction error for the LSTM with 32 hidden features as this one scored significantly lower after training. There is also no significant difference in the general shape of the reconstruction error signal between the recurrent and the convolutional autoencoder, both will show similar performance when being used for foaling prediction.

Type	**# of features**	**Training loss**	**Validation loss**	**s/epoch**
LSTM	**32**	0.84	0.85	18s
LSTM	**64**	0.72	0.69	21s
GRU	**32**	0.70	0.68	20s
GRU	**64**	0.69	0.67	20s

Table 3.3: Training results for the recurrent autoencoder

3.1.3 Combined autoencoder

The final autoencoder architecture evaluated in this thesis is a combined approach with both a convolutional and a recurrent layer. The first layer of this autencoder is a convolutional layer to perform automatic feature extraction. This gets followed by a max pooling layer to reduce the input along the time axis and only keep the most relevant extracted features of the input sequence. Finally it gets passed into the recurrent layer to transform it from

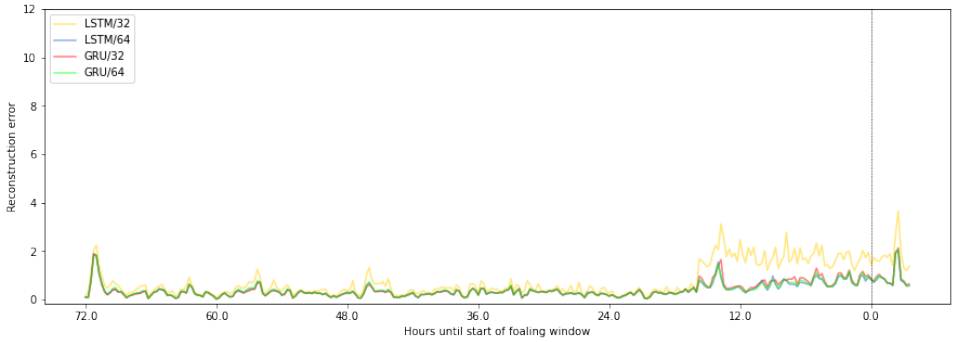

Figure 3.5: Comparison of reconstruction errors for all 4 configurations

a time series into the latent space. A combination of the best tested settings from previous experiments was used for both layer types. This being 16 convolutional filters and a gated recurrent unit with 64 hidden features. Since the size of the input to the recurrent layer was reduced by a factor of ten due to the max pooling the training time was also reduced to 4 seconds per epoch. Because of this the network was again trained for the full 100 epochs. The training resulted in a training loss of 0.50 and a validation loss of 0.51. This was better than a recurrent only architecture but worse than the convolutional only model. In figure 3.6 a final comparison between all three architectures is is given. No architecture managed to create a peak in reconstruction errors before parturition for this mare. The differences in performance thus can only be attributed to a difference in ability to reconstruct the original input. There are no fundamental differences in the reconstruction errors that make one better for foaling prediction than the other. Because of this it was opted to only use the convolutional only autoencoder in the following experiments due to its significantly faster training time.

3.2 Sliding window parameters

Two of the most important hyperparameters for the predictive model developed in this thesis are the parameters for the sliding window, the size and stride length of the window. The stride length affects the number of training samples that can be obtained and the frequency at which a prediction about the state of the mare can be made. The size of the window has the possibility to influence the predictive capabilities of the model. Certain behaviors or sequences of behaviors could only be differentiable by the model with larger windows or potentially even smaller windows since there will be less other behaviors to add noise. Three different window sizes and according stride lengths will be evaluated in this section, all three were eval-

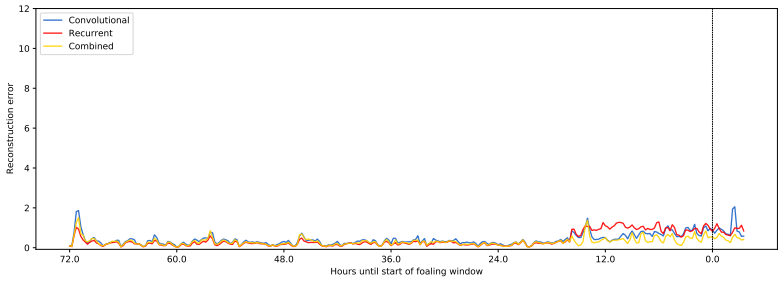

Figure 3.6: Comparison of reconstruction errors for the 3 different architectures

Size	Stride	Training samples	Validation samples
15 minutes	7.5 minutes	8313	10383
30 minutes	15 minutes	4144	5178
60 minutes	15 minutes	4110	5144

Table 3.4: Configuration for the sliding window parameters

uated with the two layer convolutional autoencoder architecture with a first layer filter count of 32 and a second layer filter count of 16. An overview of the tested parameters and respective training and validation dataset sizes is given in table 3.4. The training results of this experiment are given in table 3.5. The losses get significantly larger when the window size increases as the network has to compress longer input sequences into the same number of latent dimensions. In figure 3.7 the reconstruction errors are plotted for one of the mares. Again no significant differences can be seen in the abilities of the models to predict foaling. To evaluate the influence of the number of training samples on the reconstruction errors of the autoencoder a second experiment was done with a fixed window size of 30 minutes and 3 different stride lengths to vary the amount of training and validation samples. An overview of the number of samples per stride length can be found in table 3.6, with an overview of the training results given in table 3.7. The results in this table show that the validation loss goes down when the stride length goes down, as the network has had more training samples to learn from and thus could better learn to generalize. In figure 3.7 the reconstruction losses according to each evaluated stride length are plotted. It is clear that the stride length and thus the amount of training samples has no significant influence on the shape of the reconstruction error signal and as a result thereof it does not improve the false positive or false negative rate. A lower stride length could still be beneficially since the lower this value is the more frequent predictions about the state of the mare can be done.

Size	Training loss	Validation loss
15 minutes	0.33	0.34
30 minutes	0.36	0.44
60 minutes	0.42	0.54

Table 3.5: Training results for different sliding window sizes

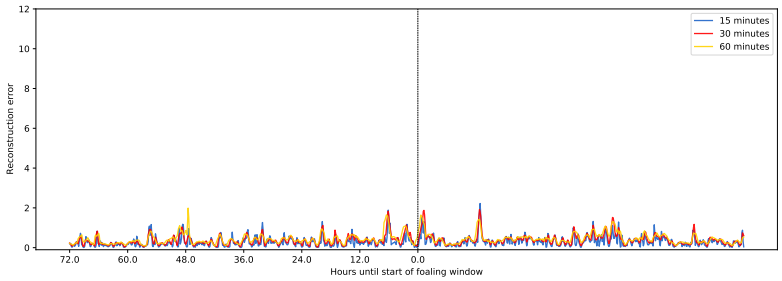

Figure 3.7: Comparison of reconstruction errors for 3 different window sizes

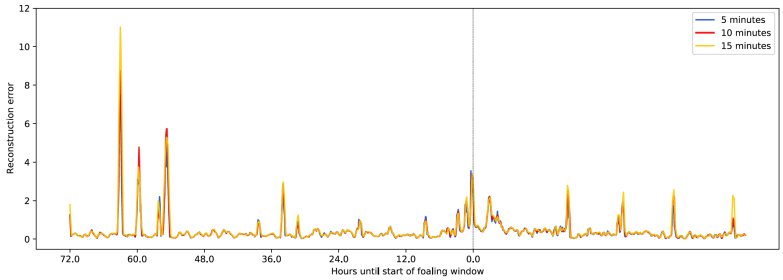

Figure 3.8: Comparison of reconstruction errors for 3 different stride lengths

3.3 Sampling rate

To reduce the computational load and data loading times it was opted to resample the data that was captured at 50 Hz down to 1 Hz by taking the mean for each group of 50 samples. However, a sampling rate of 5 Hz was

Stride	Training samples	Validation samples
5 minutes	12416	15520
10 minutes	6213	7764
15 minutes	4144	5178

Table 3.6: Number of samples for a given stride length

Stride	Training loss	Validation loss
5 minutes	0.38	0.40
10 minutes	0.37	0.41
15 minutes	0.37	0.44

Table 3.7: Training results for 3 different stride lengths

also trained to evaluate the influence of the sampling rate on the predictive capabilities of the autoencoder. This resulted in a training loss of 0.47 and a validation loss of 0.55. This is significantly higher than the losses for 1 Hz, which were 0.37 and 0.44 respectively. This is because the autoencoder has to encode much more information in the same size latent space. In the plotted reconstruction errors in figure 3.9 a small difference can be seen at about 10 hours before foaling. At 5 Hz a small peak is visible while for 1 Hz this peak is not present. During the period just before foaling no significant differences can be seen. For the architecture used in this thesis a sampling rate of 1 Hz will thus give similar performance for foaling prediction compared to higher sampling rates.

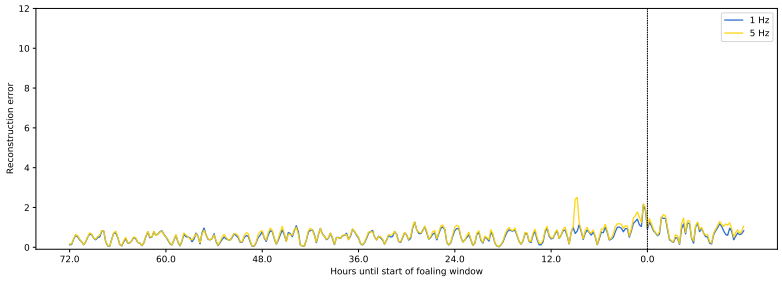

Figure 3.9: Comparison between a 1 Hz and a 5 Hz sampling rate

3.4 Standardization

One of the most influential aspects on the performance of the autoencoder was which method of standardization was used. The input data was standardized to place the acceleration values of each mare within the same range as this could vary widely with the placement of the sensor and the size of the mare. For this thesis two different methods of standardization were used and evaluated, being euther standardizing the data per input window or standardizing per mare. In figure 3.10 a plot is given of an input window that had a low reconstruction error for a network trained with per mare normalization. It can be seen that the differences of the acceleration values

between both normalization methods are fairly small in this case. When looked at the same plot but for a window that had a high reconstruction error the difference in acceleration values between the two becomes much larger. Because the data looks so different between the two methods it could be that the way of normalizing the data could influence the performance of the network. To test this hypothesis the network was trained using both approaches, the losses for the network trained with per window normalized data were significantly higher at a training loss of 0.64 and a validation loss of 0.73, while per mare normalization was at 0.36 and 0.44. This could be explained by the fact that the input data is smoothed out when normalized per window thus making it harder for the network to learn and reconstruct specific behaviors. In figure 3.12 the reconstruction errors signal is plotted for both methods. When normalized per window almost no difference can be seen in the signal for the entire three days before foaling. It can be concluded that normalizing the data per mare instead of per window is prefered and even necessary for this model to work for foaling prediction.

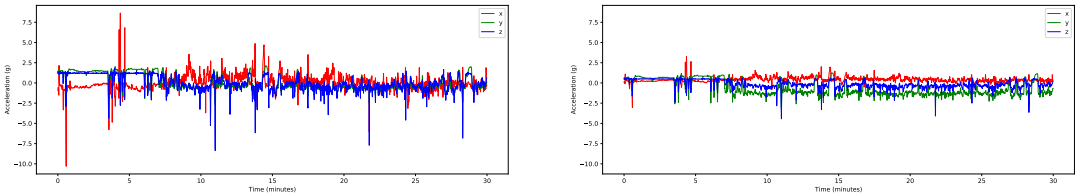

Figure 3.10: Comparison between per window normalization (left) and per mare normalization (right) for a low reconstruction error window

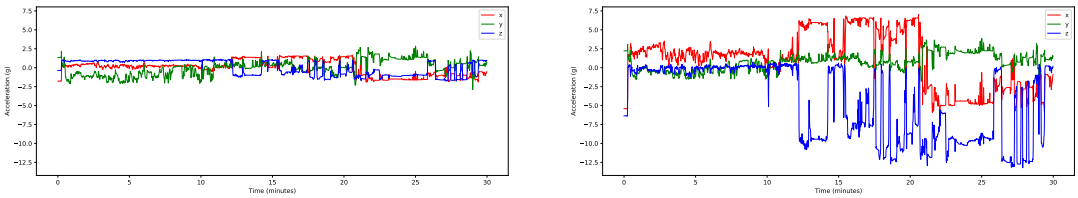

Figure 3.11: Comparison between per window normalization (left) and per mare normalization (right) for a high reconstruction error window

3.5 Discrete Fourier transform

Whilst the model so far shows promising results, with 13 out of the 15 mares showing a peak in the reconstruction error close to parturition there is still

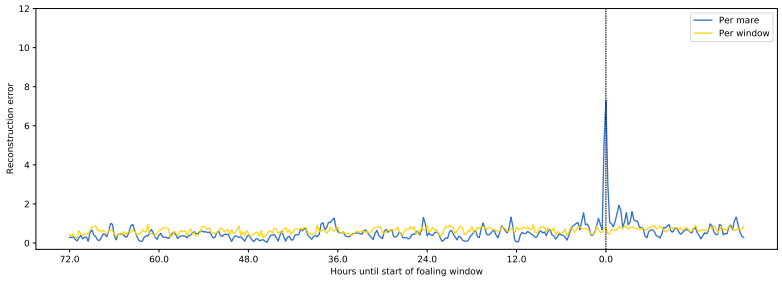

Figure 3.12: Comparison of reconstruction errors for both methods of normalization

room for improvement. Of the 15 mares, 10 showed peaks in reconstruction errors when they were still a few hours or even days away from parturition, which could lead to a large number of false positives. Apart from the high number of potential false positives there also were two mares that showed no change in reconstruction errors when the parturition approached. One of the approaches tried to improve the performance of the model was to pass the discrete Fourier transformation (DFT) of the input to the autoencoder instead of the acceleration values. The discrete fourier transformation transforms its input from the time domain to the frequency domain [44] i.e., it will show how strongly each frequency is present in the acceleration signal of the mares. To compute this transformation the fast Fourier transform algorithm was used, implemented in the scientific computing library Numpy [45]. This transformation was computed for each input window sampled at 10 Hz. This higher sampling rate was necessary as due to the Shannon-Nyquist sampling theorem [46] the discrete Fourier transform can only transform the input to a range of 0 to half the sampling frequency to the frequency domain.

An example of such a transformation for both the input window containing parturition and a regular window is given in figure 3.13. Only the frequencies between 0 and 0.2 Hz are displayed since outside this range there was little to no variation. From this visualization it is immediatly clear that most of the activity is confined to the lower frequencies between 0 and 0.05 Hz. The Fourier transform of the window containing the foaling shows that there is a lot more activity and variability in these lower frequency regions when the mare is entering labor than during a regular period.

Because of this difference between the result of the transformation on the two windows it was chosen to test the model with the Fourier transform of the acquired data as its input. This showed promising results, with 15 out of the 15 mares now showing a visible increase in its reconstruction errors

when nearing parturition, an increase of 2 mares in comparison to using just the acceleration values as input. When looking at the number of mares that showed a spike in reconstruction errors when still some time away from parturition there was also an improvement with only 9 out of 15 mares showing such a spike compared to the 10 out of 15 for the regular approach. An example of a mare which had no clear spike before parturition with the regular input data but that did have an increase in its reconstruction errors when using the Fourier transform is given in figure 3.14. Note the difference in the range of the y-axis, for the Fourier transform the absolute error values were significantly smaller in magnitude.

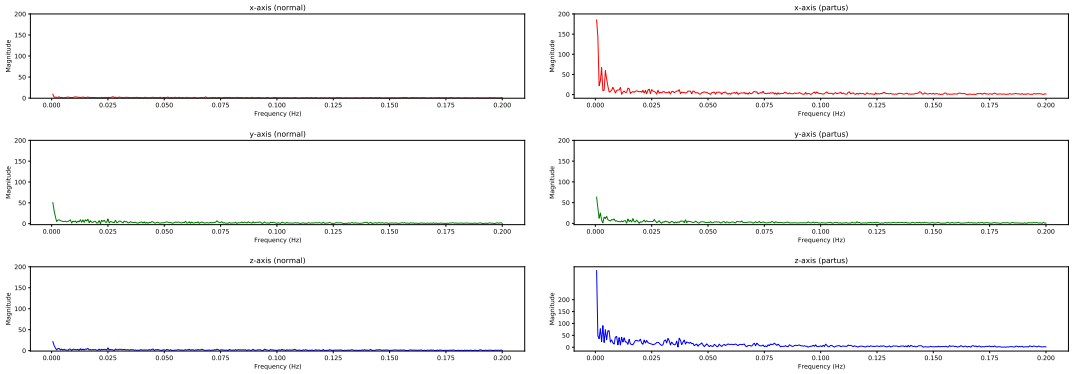

Figure 3.13: Example of the discrete Fourier transform for two sample windows, the left one containing regular behavior and the right one containing the behavior shown around foaling

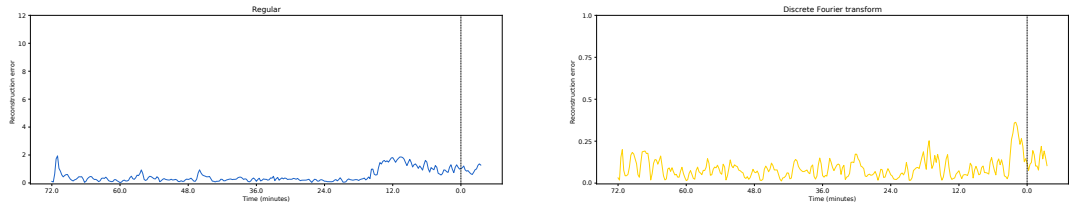

Figure 3.14: Reconstruction errors for an autoencoder trained with acceleration values as input (left) and one that was trained with the DFT of the acceleration values as its input (right)

3.6 Custom loss function

A second idea that was tried out for improving the performance of the autoencoder was implementing a custom loss function. The idea that was implemented by the custom loss function is that the reconstruction error of the autoencoder should be low during regular behavior and high when nearing parturition. To achieve this a weight function was added to the loss function of the model during training that would reward the network when it achieved a higher reconstruction error on the data close to foaling. This weight function took the shape of a logistic sigmoid curve that was given by equation 3.2 [47].

$$w = \frac{W + 1}{1 + e^{(-S \cdot (t - \frac{T}{2})}} - W \tag{3.2}$$

Where W is the minimum weight value, S is the steepness of the sigmoid curve, t is the time in minutes until parturition as indicated by the rupture of the amniotic sac and T is the time before parturition in minutes from where the assigned weight starts to deviate from 1.

To use this during training every training sample that was closer to parturition than the value of T was annotated with its time to foaling in minutes. All of the other samples as well as the entire validation dataset were annotated with the value of T as its time to foaling feature to keep the input shape constant. For this approach the training datasets were not capped at 3 days before partus since the entire dataset is now required during training. The training loss function used for training the autoencoder is given in equation 3.3.

$$loss = w \cdot MSE \tag{3.3}$$

Where w is the weight as given in equation 3.2. By capping the annotated time until foaling for training samples at T the weight factor function takes on the shape as plotted in figure 3.15.

Because of the reward for a higher reconstruction error close to parturition the network now had the tendency to severly overfit on its training data and just focus on the parts of the data that would reward it. To combat this behavior L1 regularization with a regularization factor of 0.01 was used on all trainable layers. For evaluation the network was trained several times with different combinations of the parameters for the sigmoid weight function. In the end the settings used for evaluating the custom loss function were the ones that were used in figure 3.15. This custom function resulted in significantly higher reconstruction errors on peaks while the baseline was kept the same as shown in figure 3.16. Since by including the three days before parturition into the training dataset the model now was trained in a more supervised manner. Because of this only the validation dataset will be

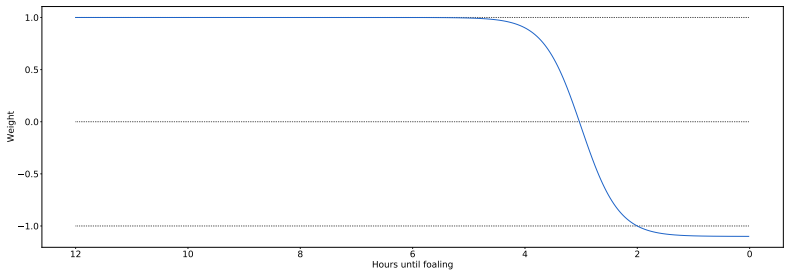

Figure 3.15: Plot of the weight factor for the custom loss function with $T = 360$, $W = 1.1$ and $S = 0.05$

looked at for evaluation. Out of the 9 mares used for validation 8 showed increases in the reconstruction errors when nearing parturition. As for peaks that were not near parturition not much has changed as well, with 5 out of 9 mares showing peaks in reconstruction errors during regular behavior. So while this does not improve the overal performance of the model, it does have the potential to achieve better performance by acquiring more data to make the network learn to better differentiate between near-foaling behavior and regular behavior.

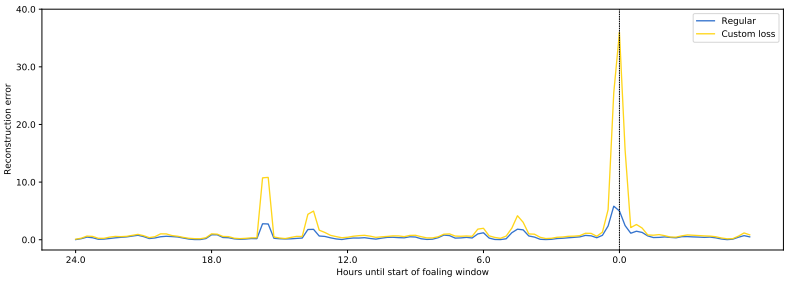

Figure 3.16: Comparison of reconstruction errors between the regular loss function (MSE) and a custom implementation favoring higher losses near parturition

3.7 Latent representation

Until now only the reconstruction error signal was used to assess the performance of the autoencoder on the foaling prediction problem. The distribution of input windows in the latent space of the autoencoder does however

also have the potential to be a good predictor. It could be that the autoencoder transforms its input into the latent space in such a way that outlier detection or clustering on these representations could be a good predictor for parturition.

The first experiment conducted to evaluate this hypothesis was rdone by reducing the number of latent dimensions to just two dimensions. By doing this the distribution of the latent representations could be inspected visually. But before this could be done the difference between reconstruction behavior for the network with 32 and 2 latent dimensions had to be evaluated to make sure that there was no significant penalty for reducing the number of dimensions. The training and validation loss of the autoencoder with two latent dimensions was significantly higher than the one with 32 latent dimensions at 0.70 and 0.78 respectively. This was expected since the autoencoder had to embed more information in a smaller space. In figure 3.17 a visual comparison is given between both autoencoders. Both implementations show peaks at the same time, albeit higher peaks for the two dimensional implementation. It can thus be concluded that reducing the latent space to only two dimensions will not significantly reduce the performance of the autoencoder for foaling prediction as this is mainly dependent on the shape of the reconstruction error signal and not the absolute values.

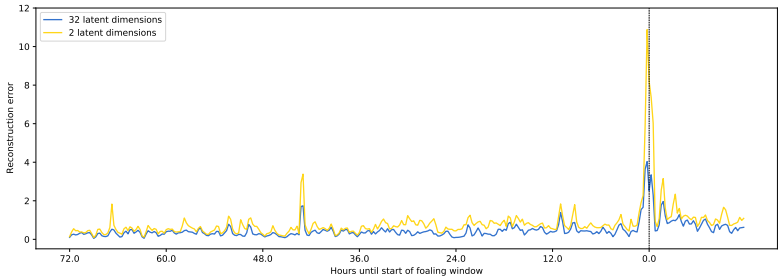

Figure 3.17: Comparison of reconstruction errors between an autoencoder with 2 and one with 32 latent dimensions

In figure 3.18 a scatterplot of the latent representations in these two dimensions is given as an example. All of the mares showed the same pattern where the representations for the windows close to foaling are randomly distributed troughout the latent space. This random distribution of the input windows in the latent space also occured in a three dimensional latent space. It can thus be concluded that performing a clustering algorithm on the latent space would not be a viable method of performing foaling detection for an

autoencoder approach as there is no noticeable difference between the representations of windows close to foaling and windows far away from foaling.

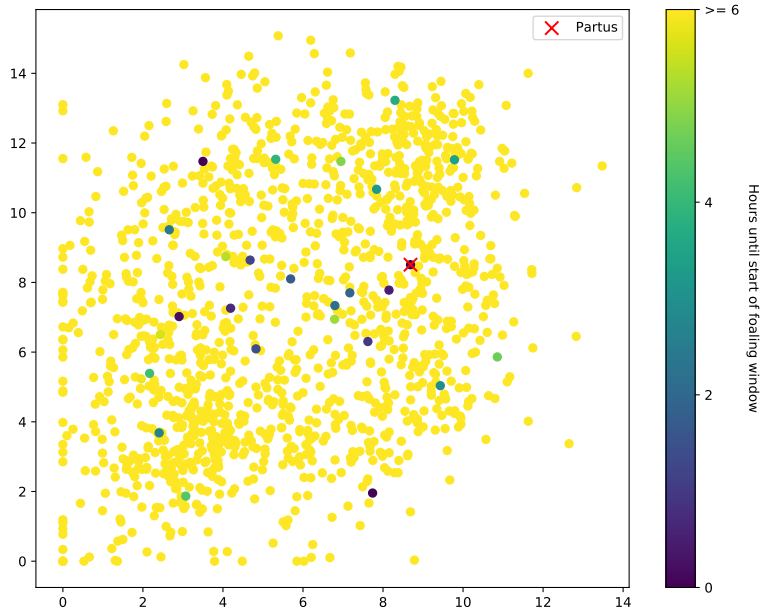

Figure 3.18: Example scatterplot of the latent representations for a random mare

3.8 Leave-one-out cross-validation

For all of the previous experiments in this section a fixed training and validation set of 6 and 9 mares respectively was used for each trained model. To proof that the proposed model did not depend on the specific training set that was used but could generalize well a Leave-one-out cross-validation (LOOCV) was performed. The first step in the LOOCV experiment was to filter the original datasets to only contain the mares that had more than 3 days of pre-foaling data available because each mare will now be part of a training dataset and thus had to be cut at the three day before parturition point. In the end 11 out of the 15 mares had more than three days of pre-foaling data and were available for use during LOOCV. For each of these 11 mares a seperate autoencoder was trained using the dataset of this mare for

validation and the other 10 datasets for training. In figure 3.19 a comparison between the reconstruction errors for a regular trained autoencoder and an LOOCV trained autoencoder is given. For this visualization the data of a mare that was part of the training set of the regularly trained autoencoder was used to see if there was a significant difference in reconstruction errors if the autoencoder has seen the mare's behavior during training or not. By visually inspecting the difference in reconstruction errors signals for both models it can be observed that there are only small differences in the absolute values. The peaks and valleys of the signal are still the same shape and take place at the same time. This observation can be made for all 11 mares. Because of this it can be concluded that the network sufficiently generalizes and no overfitting to the training data took place.

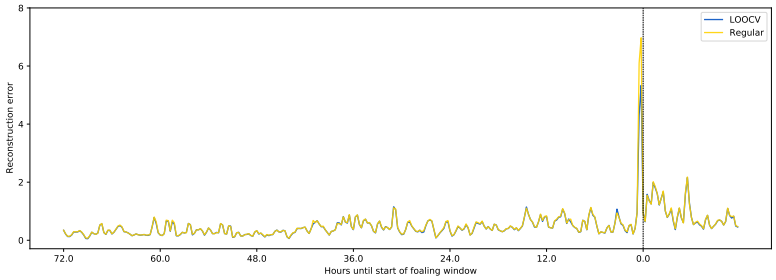

Figure 3.19: Comparison of reconstruction errors between regular training and leave-one-out cross-validation training

3.9 Transfer learning

While the approach used up this point shows good results in terms of predicting parturition there still were a large number of false positives. Out of the 11 mares that had more than three days of pre-parturition data available, 5 showed a clear spike in reconstruction errors in the three days leading up to parturition but still more than a couple of hours away from partus. This could potentially result in a false positive depending on the metric used to decide when parturition is about to take place. To try and combat this, transfer learning was tried out and evaluated. This was already an idea to implement from the start of this research. The idea behind using transfer learning for foaling prediction was to first train a general base model from a dataset of regular behavior from a couple of unspecifiied mares. This would allow the autoencoder to learn how to reconstruct general horse behavior. If the model then got deployed in a real world setting it would first go trough a setup phase. During this phase it would continuously apply

transfer learning in an online fashion and update its knowdlegde for the mare it was observing. After a couple of days of transfer learning the model should have learned the intrinsics of the observed mare after which it would switch over to inference mode. Because the model is now better fitted to this specific mare it could be better in differentiating between regular and abnormal behavior.

To test out this hypothesis in the setting of this research the regular model was used as a base model on which, for each of the five validation mares, transfer learning was then applied. First only the data up to three days before parturition of the validation mares was kept for use during training. Once the training data was obtained the weights of base model got loaded for each of the five mares after which the model got trained for an additional 25 epochs on data of a single mare. The result of this experiment for one mare that showed false positives with the regular model is shown in figure 3.20. Again there are high peaks in the reconstruction errors visible long before parturition is about to take place, transfer learning the entire model did not result in a potential lower false positive rate.

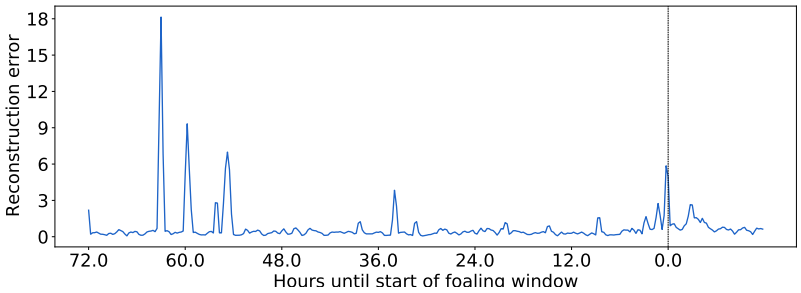

Figure 3.20: Example of the reconstruction errors after transfer learning the entire model for an additional 25 epochs

In transfer learning an already trained model it is common practice to freeze certain layers during the transfer learning phase [48]. In most cases the earlier layers in the network get frozen for transfer learning, and only the later layers actually get updated. This is done because earlier layers mostly extract more general features while later layers are trained to extract more specific features in regards of the training data. In the case of the autoencoder used in this thesis it was chosen to freeze the first and last convolutional layer and leave the other layers unlocked during transfer learning. Doing this however resulted in almost no visible difference when looking at the reconstruction errors, as shown in figure 3.21. In both graphs there is almost no difference between the reconstruction error signals. With

the current settings and available data transfer learning did make no usable difference, even when locking certain layers for training.

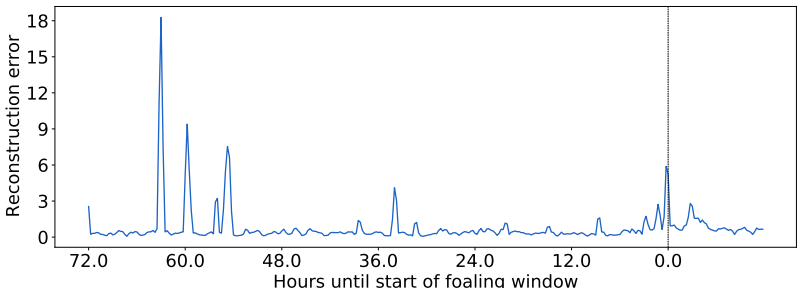

Figure 3.21: Example of the reconstruction errors after transfer learning only a part of the model for an additional 25 epochs

3.10 Filtering

Up until this point the used method to decide which part of the data contains regular behavior for use during training is purely a time based one. At the moment this threshold was set at three days before parturition. It could however be that the mare already showed some behavior that was similar to the behavior shown when nearing parturition during this period. One possible cause is that the mares are more nervous since they were not stabled at their home location. To combat this potential problem the influence of applying filtering on the training datasets was evaluated.

The filtering method that was evaluated consisted of filtering out all training windows containing over 10% of datapoints that were more than a certain number of standard deviations away from the mean. Since the data was normalized per mare it consequently had a mean of 0 and a standard deviation of 1. Each window consisted of 1800 samples so a window would be filtered out if over 180 samples, combined over all three axes, had an absolute value higher than a certain set threshold. Two different standard deviation thresholds were evaluated, i.e. 3 and 5. When filtering out all windows containing more than 10% of samples that have an absolute value higher than 3 about 25% of training windows get filtered out. When filtering with a threshold of 5 only 5% of windows get removed from the training dataset. In figure 3.22 a visual comparison is given for three different autoencoders that were trained with these three different filtering approaches. All three graphs look almost identically with only subtle differences between the sizes and shapes of the peaks. Because of this it can be concluded that filter-

ing out windows with high variability during training does not significantly influence the reconstructive power of the proposed method.

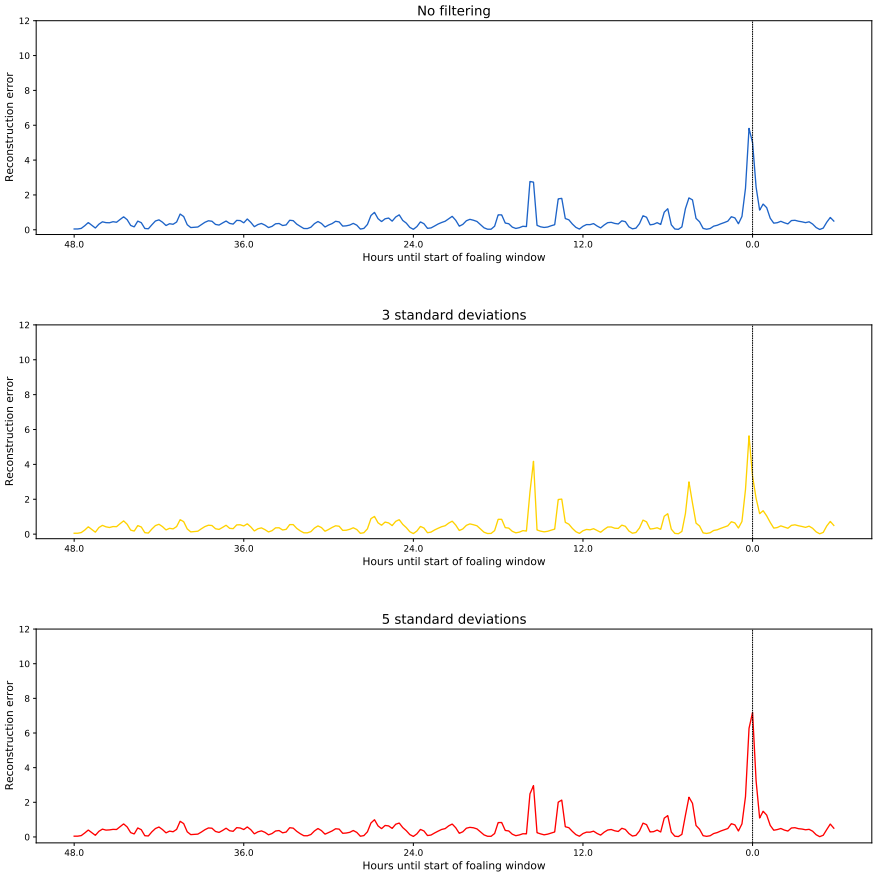

Figure 3.22: Comparison of the reconstruction errors between the three different filtering methods

3.11 Decision metric

Lots of different methods to improve the predictive performance of the autoencoder were evaluated in the previous sections, but the goal of this thesis is to develop a foaling prediction algorithm. To accomplish this a final decision metric should be evaluated. This metric will decide, based on the reconstruction errors of the autoencoder, when parturition is about to take place. Two different approaches for this metric will be evaluated, deciding based on the raw reconstruction errors signal or deciding based on the

trend signal of the seasonal-trend decomposition. To make the comparison between the different proposed methods as fair and realistic as possible only the eleven mares that had more than three days of pre-parturition data available were used during this evaluation. This was done to get an accurate representation of the number of false positives during these three days pre-foaling. For all mares only these three days were used to decide if a false positive was present in the data or not. The threshold for being a correctly predicted peak or a false positive was set at three hours before parturition. If the model resulted in a trigger up to three hours before foaling it was labeled as a correct prediction, if it triggered at an earlier moment this was labeled as a false positive.

3.11.1 Reconstruction errors based threshold

The first method that was looked at was deciding if parturition is nearing based on just the reconstruction error signal. The decision that parturition is about to take place was made when this signal went above a certain set threshold. There were two ways of setting the value for this threshold, picking the same threshold for all mares or choosing one for each mare individually. The results from choosing a static threshold for all mares are given in table 3.8. It can be seen that the lower this threshold the higher the number of correctly recognized foaling events as the threshold will be reached for lower reconstruction errors, this makes this approach more sensitive. Because of this increase in sensitivity when the threshold gets lowered the number of false positives also grows. At a value of 1.5 for the threshold 7 out of the 11 mares triggered one or more false alarms.

Threshold	**TP**	**FP**	**FN**
5.0	3	2	8
3.0	7	3	4
2.0	9	3	2
1.5	10	7	1

Table 3.8: Overview of the number of correct predictions/true positives (TP), false alarms/false postives (FP) and undetected foalings/false negatives (FN) for a statically chosen threshold

A way of improving the performance would be to set a threshold for each mare individually instead of a global threshold. This individual value would be decided by analyzing the reconstruction errors from each mare some time before parturition to get a baseline of what to expect from each mare in terms of reconstruction errors. In the setting of this thesis all data up to three days before parturition was used for this analysis. Three different methods of selecting the threshold based on this data were evaluated. The first one was

setting the threshold at the maximum value, or maximum plus a fixed value, encountered during the analysis phase. The second one was calculating the mean of the reconstruction errors during the analysis phase and setting the threshold at this mean plus a certain value. Because some mares showed a lot more variability in their reconstruction error signal than others the final method that was evaluated was again taking the mean but now adding a number of standard deviations to it to set the threshold. Making the added value depend on the standard deviation of each mare should account for the differences in variability between each mare. The results for using a dynamic threshold are presented in table 3.9. Again the same pattern is visible, the lower the value of the threshold the higher the number of true positives but also the higher the number of false positives. Noticeable is that while the number of true positives is the same for both the fixed addition to the mean and the addition based on the standard deviation this is not the case for the number of false positives. On the limited dataset available standard deviation based addition performs significantly worse. This is however an issue that could be fixed by further finetuning the amount of standard devations away the threshold is set at on more data.

Method	**TP**	**FP**	**FN**
max	8	9	3
max + 1	6	5	5
mean + 1	11	7	0
mean + 1.5	10	5	1
mean + 3σ	11	9	0
mean + 5σ	10	7	1

Table 3.9: Overview of the number of correct predictions/true positives (TP), false alarms/false postives (FP) and undetected foalings/false negatives (FN) for a dynamically chosen threshold

3.11.2 Seasonal-trend composition based threshold

The second proposal for a metric that decides when parturition is about to take place is not based on the stream of reconstruction errors but is based on the trend component of the seasonal-trend decomposition of this stream. The benefit of this decomposition is that it singles out the seasonal component of the signal. This could influence the performance of the proposed approach since some mares will become restless and show increased activity around feeding time, which occured mostly at the same time in the animal clinic. This could confuse the model when not taken into account. To calculate this decomposition the *seasonal_decompose* method from the stats-

models library was used. [4] The problem with using this trend signal for deciding the time of parturition is that the baseline of this trend could differ a lot between mares, making it harder to decide purely on a fixed threshold. To place every mare on the same baseline and make it easier to decide on a threshold the difference between each subsequent value is taken and used then this was used as a signal. An example of the trend of a mare and these differences between subsequent values is shown in figure 3.23. From this it is clear that using the differences between subsequent values for thresholding is a better choice than just using the trend signal. With the differences it is easy to only trigger on rising edges in the trend signal and not on falling edges. In table 3.10 the results of using this approach are given, for this experiment only two statically chosen thresholds were evaluated. It can be seen that, for the limited dataset of 11 mares, this approach performs similar to using a dynamic threshold on the reconstruction error signal.

Threshold	**TP**	**FP**	**FN**
0.05	7	2	4
0.025	10	5	1

Table 3.10: Overview of the number of correct predictions/true positives (TP), false alarms/false postives (FP) and undetected foalings/false negatives (FN) for a statically chosen threshold on the differences between subsequent values of the trend from the seasonal-trend decomposition

[4] `https://www.statsmodels.org/stable/generated/statsmodels.tsa.seasonal.seasonal_decompose.html`

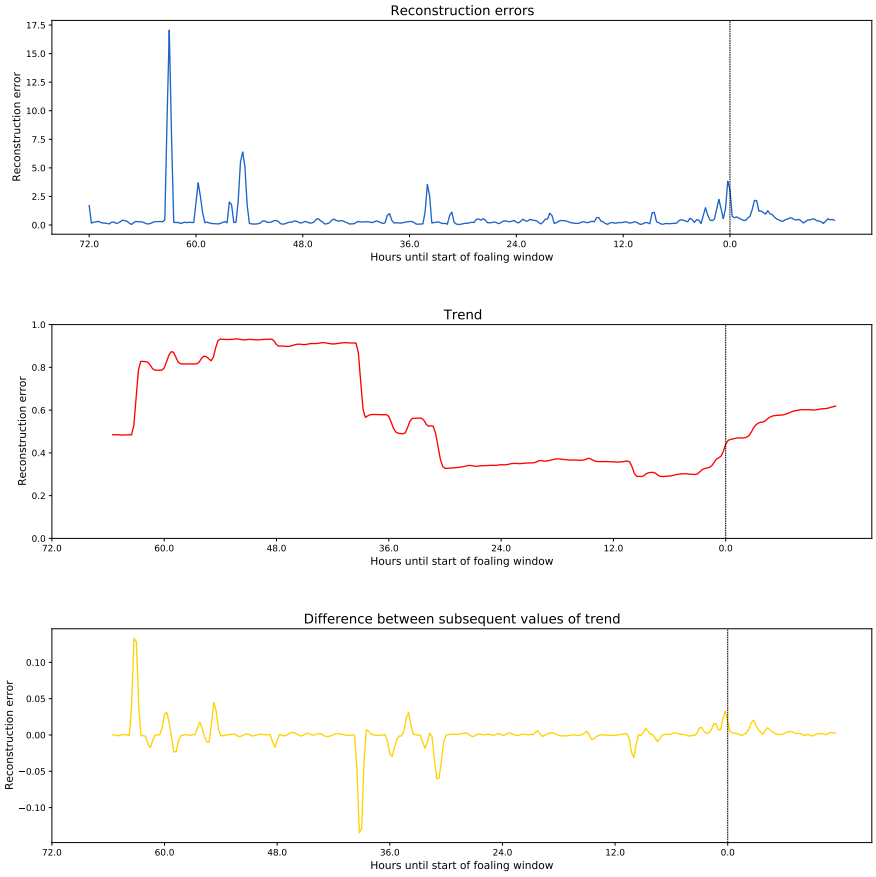

Figure 3.23: Example of the trend component of a seasonal-trend decomposition together with differences between subsequent values of this trend signal

Chapter 4

Discussion

The goal of this study was to develop a machine learning model that could, based on accelerometer data from the mare's head, recognize when a mare is entering labor. To achieve this anomaly detection by using a deep learning approach based on an autoencoder architecture was used. The benefit of this approach is the wide range of configurability from tuning the autoencoder 's underlying architecture to defining the used decision metric, as well as the fact that no data had to be labeled, a task normally very labor intensive. In the previous chapter several of these possible configurations were evaluated. In table 4.1 a summary of the findings of these experiments is given. Due to the limited amount of acquired data some of these experiments had no conclusive result since more data was required to be able to correctly evaluate these ideas.

In the end the proposed method should make use of a convolutional autoencoder to generate a stream of reconstruction errors. A convolutional only approach is preferred since these train significantly faster than when using a recurrent network. By using the lowest possible sampling rate that still obtains the necessary accuracy the data footprint and computational load is kept at a minimum. By changing the stride length of the sliding window method for obtaining input data the frequency at which predictions are made can be influenced. The lower the stride length the higher the frequency, but this also increases the chance of repeating false positives. So a balance between both should be found. To alleviate the influence of a misplaced sensor or a shifted halter the discrete Fourier transform of the acceleration values can be used to make the predictions of the model more robust against these types of changes. With enough training data it could prove beneficially to use a custom training loss function that makes the autoencoder learn to perform worse on windows showing near-parturition behavior.

Description	Significant influence	Conclusion
Architecture	No	2 layer convolutional only autoencoder
Sliding window	No	30 minutes window size, stride length as low as possible to improve frequency at which predictions can be made
Sampling rate	Potentially	More data is required to evaluate, only slight differences visible for the current dataset
Standardization	Yes	Standardizing per mare
DFT	Yes	Makes the algorithm less prone to errors in sensor placement, more data is required to evaluate
Custom loss	Potentially	With more data the network could better learn the intrinsics of the behavior close to parturition
Latent represenation	No	Using the reconstruction errors for prediction, not the latent representation
LOOCV	No	The performance of the autoencoder does not depend on the training set and generalizes well
Transfer learning	No	Since the model generalizes well transfer learning did not improve its performance
Filtering	No	Filtering out high variability input windows does not improve performance
Decision metric	Yes	A dynamic threshold based on the mean and standard deviation

Table 4.1: Overview of the outcome of the performed experiments

The autoencoder is only one part of the proposed approach, based on the output of this autoencoder a decision has to be made on if the mare is about to enter labor or not. To do this the reconstruction error of the autoencoder

on the input window is calculated using the mean squared error formula. Based on this reconstruction error a decision is then made by comparing this value against a threshold, if the value is above the threshold a foaling alarm is triggered. For deciding what this threshold should be the mare should first go trough an analysis phase where the reconstruction errors are calculated for a couple of days, the mean of these values during this phase will then be used for deciding the threshold. Because some mares show a lot more variability in the values of the reconstruction error over time, the standard deviation of the errors during the analysis phase gets calculated as well. When this analysis phase is completed the threshold for triggering an alarm is set at the mean of the analysis phase plus a number of standard deviations. By changing the amount of standard deviations that get added the sensitivity of the decision can be influenced.

Chapter 5

Conclusion and Future work

The aim of this study was to design a machine learning algorithm that warns in time that a pregnant mare is entering labor. By only using a small accelerometer attached to a halter the impact on the comfort of the mare is kept at a minimum. Anomaly detection based on the reconstruction error of an autoencoder neural network was proposed as a model to recognize the start of labor based on the accelerometer data.

For the training and evaluation of the proposed method data of 15 pregnant mares was captured at the Ghent University veterinary clinic. Of these 15 mares 11 had more than 3 days of pre-parturition data and were used for evaluation. The impact of different architectures, training tactics and decision metrics were evaluated and in the end all foalings of the 11 mares used for evaluation were correctly recognized. This came however at the cost of many false alarms, in the best case 7 out of the 11 mares suffered from one or more false alarms in the three days leading up to parturition. Due to the wide configurability of several of the different parts of the proposed method this could be reduced to more acceptable numbers but more data is required to explore the different possibilities.

In figure 5.1 an example of how to apply this system in practice is given. The proposed method consists of 3 or 4 different parts: a microcontroller with an accelerometer and transmitting antenna placed on the mare's halter, a server with a receiving antenna that runs the model, and a device that can communicate with the server that will be used to warn an observer of a potential foaling. Optionally a camera can be added to the system so that the observer can decide based on video footage if the triggered alarm is a false alarm or not. The microcontroller on the mare continuously gathers accelerometer data and transmits this to the server, this transmission can be done using a wirelless communication protocol that best fits the situation at hand, being either BLE, LoRaWAN, Wi-Fi et cetera. The server then

takes in this data and feeds it to the autoencoder which will calculate the reconstruction error on each input window. If the error is above the set threshold the server will send an alarm to the observer, this can be either via a smartphone notification or via a seperate dedicated device. To make sure the alarm is not a false alarm and parturition is really about to take place the observer can first check the videofeed when he receives an alarm. Because the symptoms of labor are very similar to the symptoms of colic this system to be used for the detection of health related behavior.

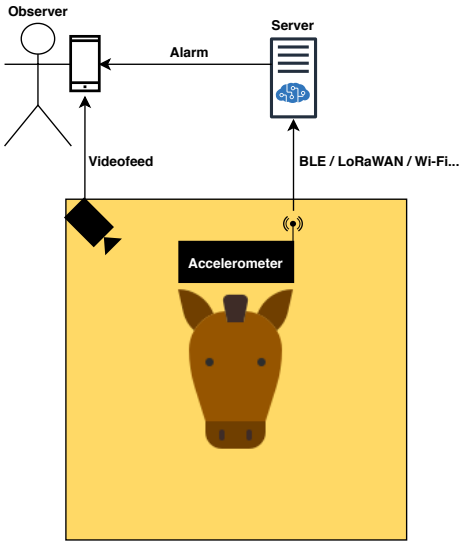

Figure 5.1: Example of how to deploy the model in a real world scenario

The main goal of future work should be to acquire more data. More data of a head mounted accelerometer gives the possibility to further explore and evaluate several of the proposed enhancements of this study such as the Fourier transform or custom loss function during training. Different places to attach the sensor should also be evaluated, such as the legs or tail. Instead of using the raw accelerometer data as an input a seperate model could be trained that classifies the observed behavior after which anomaly detection could be performed on combinations of these detected behaviors. Since the symptoms of the first stage of labor and the sympoms of colic are very similar the possibility to deploy this system as a colic detector could be explored as well.

Bibliography

[1] P. Cross, 'Global horse statistics internal 02 2019', Feb. 2019.

[2] Jan. 2020. [Online]. Available: https://www.t-online.de/sport/id_75116260/totilas-mitbesitzer-paul-schockemoehle-senkt-preise-fuer-wunderhengst-samen.html.

[3] Jan. 2020. [Online]. Available: https://www.flandersfoalauction.be/nl/nieuws/Grand-finale-Flanders-Foal-Auction-sluit-af-met-20108-euro-gemiddeld.

[4] T. S. Mair *et al.*, *Equine Medicine, Surgery and Reproduction*, 2nd edition. Edinburgh: Elsevier, 2013.

[5] P. M. McCue and R. Ferris, 'Parturition, dystocia and foal survival: A retrospective study of 1047 births', *Equine Veterinary Journal*, no. 44, pp. 22–25, 2012.

[6] O. J. Ginther and D. Williams, 'On-the-farm incidence and nature of equine dystocias', *Journal of Equine Veterinary Science*, no. 16, pp. 159–164, 1996.

[7] L. Heck, M. Clauss and M. Sánchez-Villagra, 'Gestation length variation in domesticated horses and its relation to breed and body size diversity', *Mammalian Biology - Zeitschrift für Säugetierkunde*, vol. 84, pp. 44–51, Jan. 2017.

[8] P. D. Rossdale and R. V. Short, 'The time of foaling of thoroughbred mares', *J. Reprod. Fertil.*, vol. 13, pp. 341–343, 1967.

[9] Jan. 2020. [Online]. Available: https://www.ugent.be/di/vvb/en/services/clinic-reproduction-ghd.htm.

[10] E. B. Shaw, K. A. Houpt and D. F. Holmes, 'Body temperature and behaviour of mares during the last two weeks of pregnancy body temperature and behaviour of mares 1 during the last two weeks of pregnancy body temperature and behaviour of mares during the last two weeks of pregnancy', *Equine Veterinary Journal*, vol. 20, pp. 199–202, 1988.

[11] D. T. Cross, W. R. Threlfall and R. C. Kline, 'Body temperature fluctuations in the periparturient horse mare', *Theriogenology*, vol. 37, pp. 1041–1048, 1992.

[12] M. Saint-Dizier and S. Chastant-Maillard, 'Methods and on-farm devices to predict calving time in cattle', *The Veterinary Journal*, vol. 205, pp. 349–356, 2015.

[13] S. D. Goodwin, 'Comparison of body temperatures of goats, horses, and sheep measured with a tympanic infrared thermometer, an implantable microchip transponder, and a rectal thermometer', *Journal of the American Association for Laboratory Animal Science*, vol. 37, no. 3, pp. 51–55, 1998.

[14] E.-L. J. Verdegaal, C. Delesalle, C. G. Caraguel, L. E. Folwell, T. J. McWhorter, G. S. Howarth and S. H. Franklin, 'Evaluation of a telemetric gastrointestinal pill for continuous monitoring of gastrointestinal temperature in horses at rest and during exercise', *American journal of veterinary research*, vol. 78, no. 7, pp. 778–784, 2017.

[15] C. Hartmann, L. Lidauer, J. Aurich, C. Aurich and C. Nagel, 'Detection of the time of foaling by accelerometer technique in horses (equus caballus)—a pilot study', *Reproduction in domestic animals*, vol. 53, pp. 1279–1286, 2018.

[16] C. Giannetto, M. Bazzano, S. Marafioti, C. Bertolucci and G. Piccione, 'Monitoring of total locomotor activity in mares during the prepartum and postpartum period', *Journal of Veterinary Behavior*, vol. 10, pp. 427–432, 2015.

[17] M. Pastell, J. Hietaoja, J. Yun, J. Tiusanen and A. Valros, 'Predicting farrowing based on accelerometer data', in *Precision Livestock Farming 13*, D. Berckmans and J. Vandermeulen, Eds., 2013, pp. 819–824.

[18] M. Pastell, J. Hietaoja, J. Yun, J. Tiusanen and A. Valros, 'Predicting farrowing of sows housed in crates and pens using accelerometers and cusum charts', *Computers and Electronics in Agriculture*, vol. 127, pp. 197–203, 2016.

[19] M. Titler, M. G. Maquivar, S. Bas, P. J. Rajala-Schultz, E. Gordon, K. McCullough, P. Federico and G. M. Schuenemann, 'Prediction of parturition in holstein dairy cattle using electronic data loggers', *Journal of Dairy Science*, vol. 98, no. 8, pp. 5304–5312, 2015.

[20] M. R. Borchers, Y. M. Chang, K. L. Proudfoot, B. A. Wadsworth, A. E. Stone and J. M. Bewley, 'Machine-learning-based calving prediction from activity, lying, and ruminating behaviors in dairy cattle', *Journal of Dairy Science*, vol. 100, no. 7, pp. 5664–5674, 2017.

[21] 2007. [Online]. Available: `http://www.foalguard.com`.

[22] 2020. [Online]. Available: https://birthalarm.com.

[23] 2020. [Online]. Available: https://www.facebook.com/SafemateFoalalarm/.

[24] 2007. [Online]. Available: http://www.foalguard.com/products.htm.

[25] 2016. [Online]. Available: https://www.gallaghereurope.com/nl_nl_ge/birth-alarm.

[26] [Online]. Available: https://baltichorse.eu/product/foal-alarm/.

[27] 2019. [Online]. Available: https://foalert.com.

[28] L. A Bate, D. Hurnik and J. G. Crossley, 'Benefits of using a photo-electric alert system for swine farrowing operations'', *Can. J. Anim. Sci.*, vol. 71, pp. 909–911, 1991.

[29] Feb. 2017. [Online]. Available: https://www.littletonequine.com/faq/foalalertversescaslicks/.

[30] [Online]. Available: http://people.upei.ca/bate/html/birth_alert_system.html.

[31] [Online]. Available: http://www.equiview360.com.

[32] [Online]. Available: http://www.equiview360.com/index.php?page=3.

[33] O. Ginther, 'Twinning in mares: A review of recent studies', *Journal of Equine Veterinary Science*, vol. 2, no. 4, pp. 127–135, 1982.

[34] [Online]. Available: https://axivity.com/product/ax3.

[35] [Online]. Available: https://www.premierequine.co.uk/plain-padded-horse-head-collar-c2x21459520.

[36] A. Eerdekens, 'Automatic detection of abnormal behaviour of equines', Master's thesis, Ghent University, 2019.

[37] S. Hochreiter and J. Schmidhuber, 'Long short-term memory', *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[38] J. Chung, C. Gulcehre, K. Cho and Y. Bengio, 'Empirical evaluation of gated recurrent neural networks on sequence modeling', *arXiv preprint arXiv:1412.3555*, 2014.

[39] I. Sutskever, O. Vinyals and Q. V. Le, 'Sequence to sequence learning with neural networks', in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[40] 'Mean squared error', in *Encyclopedia of Machine Learning*, C. Sammut and G. I. Webb, Eds. Boston, MA: Springer US, 2010, pp. 653–653, ISBN: 978-0-387-30164-8. DOI: 10.1007/978-0-387-30164-8_528. [Online]. Available: https://doi.org/10.1007/978-0-387-30164-8_528.

[41] R. B. Cleveland, W. S. Cleveland, J. E. McRae and I. Terpenning, 'Stl: A seasonal-trend decomposition', *Journal of official statistics*, vol. 6, no. 1, pp. 3–73, 1990.

[42] 'Score normalization', in *Encyclopedia of Biometrics*, S. Z. Li and A. Jain, Eds. Boston, MA: Springer US, 2009, pp. 1134–1135, ISBN: 978-0-387-73003-5. DOI: 10.1007/978-0-387-73003-5_767. [Online]. Available: https://doi.org/10.1007/978-0-387-73003-5_767.

[43] R. Fu, Z. Zhang and L. Li, 'Using lstm and gru neural network methods for traffic flow prediction', in *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 2016, pp. 324–328. DOI: 10.1109/YAC.2016.7804912.

[44] A. Roxburgh, 'On computing the discrete fourier transform', Dec. 2013.

[45] H. J. Nussbaumer, 'The fast fourier transform', in *Fast Fourier Transform and Convolution Algorithms*, Springer, 1981, pp. 80–111.

[46] A. J. Jerri, 'The shannon sampling theorem—its various extensions and applications: A tutorial review', *Proceedings of the IEEE*, vol. 65, no. 11, pp. 1565–1596, 1977.

[47] P. Verhulst, 'Notice sur la loi que la population poursuit dans son accroissement in: Correspondance mathématique et physique, vol. 10', 1838.

[48] S. Akçay, M. E. Kundegorski, M. Devereux and T. P. Breckon, 'Transfer learning using convolutional neural networks for object classification within x-ray baggage security imagery', in *2016 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2016, pp. 1057–1061.