IT/sensor-based, easy-to-use framework for decision-making and advisory support for small dairy farms in Alpine space region.

Thorsten Hehn, 23.12.2021

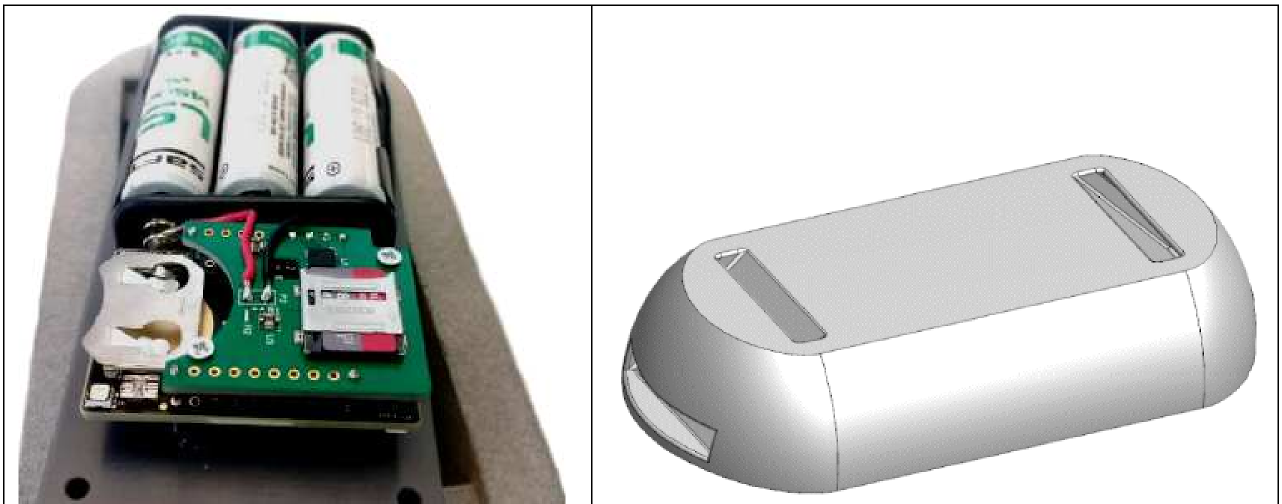
1 Introduction

This report summarizes work package T1 (WP T1) named “Development and piloting of sensor-based monitoring and decision-making system for family and other typical types of alpine dairy farms” which was under the responsibility of Hahn-Schickard. The WP T1 outcomes were the basis of the other activities in the SESAM project, especially the development of the second level algorithm and the application for the farmers (“Alibaba”).

This report is structured as follows: First, the small pilot is described which had the goal to gather data for developing the first level algorithm. Then, the main outcomes of the big pilot are highlighted. The last paragraph summarizes the results of the first level algorithm.

2 Small pilot

The small pilot had the goal to gather data for developing the first level algorithm. Therefore, ten cows on six farms were equipped with sensors developed in the previous project KuhBa. These sensors contain an accelerometer and a microphone and are surrounded by a waterproof (IP67) housing (see Figure 1). The data is stored on an SD card, which was sent to Hahn-Schickard by the farmers after the defined data acquisition period was over.



(a) **(b)**

Figure 1: Sensor used for the small pilot, (a) electronics and (b) housing

For the small pilot stage, the seven cow activities walking, eating, ruminating (laying), ruminating (standing), laying, drinking, standing were defined. For the labelling procedure, a commercial surveillance system consisting of two cameras and one recording station was installed in the farms. The recorded videos were saved into an external hard drive. A PC software was developed to simplify the labelling process (see Figure 2).

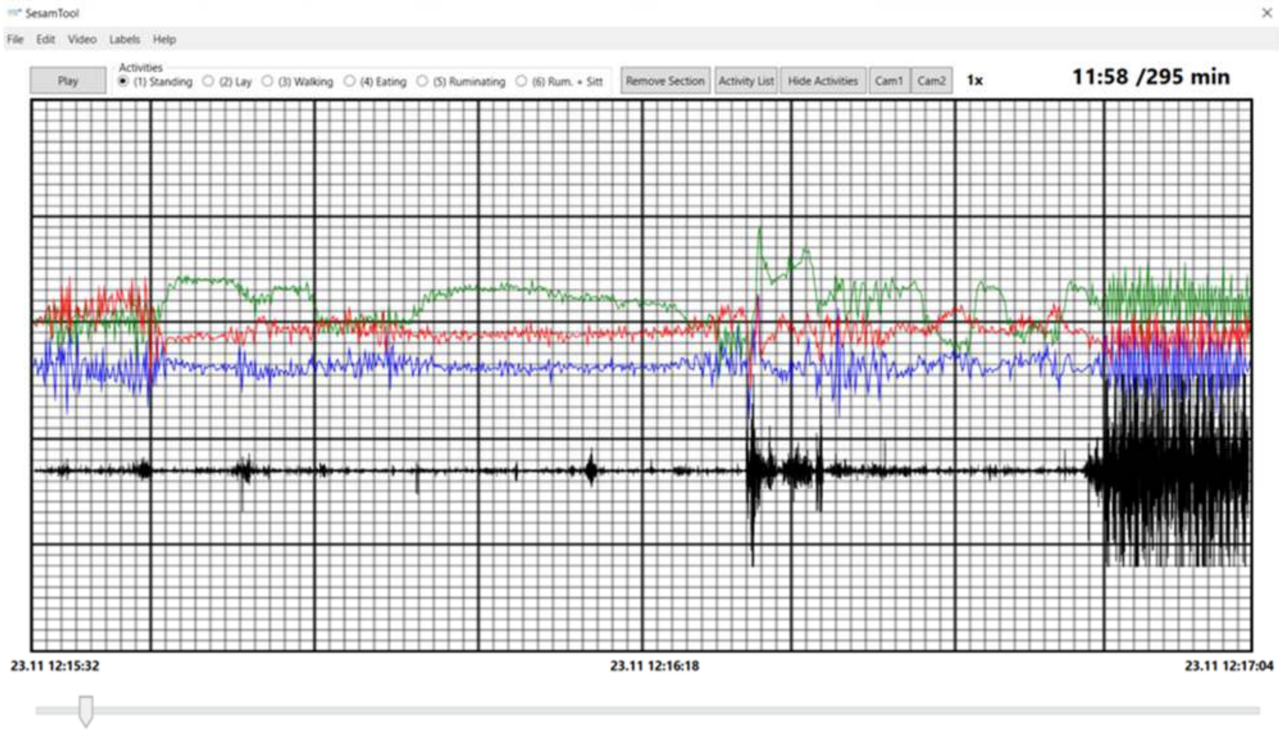


Figure 2: PC software for activity labelling written in C#

The PC software allows seeing both camera angles in two different windows. The acceleration and microphone data is visible in the main window. The cows were marked with a number (i.e. from 1 to 10, see Figure 3). The user can select in the software which cow is visible on camera and select an activity. For each activity, the user defines a time period where an activity is visible on camera. Later on, the labelling results are mapped to the sensor raw data of the same period. In order to get the correct timestamps, it is mandatory to synchronize the camera footage and the sensor raw data.

Figure 3: Video footage showing numbered cows, used for labelling

It has shown during the labelling procedure that the visibility of the cows was sometimes problematic. Often, the cow number could not be seen or the cow was off the screen. In total six barns were equipped with ten sensors each. In the end, over 30 days of video footage with data was available and had to be processed. The most problematic aspect of labelling and classification was the underrepresentation of certain activities. Standing and laying, for example, was ten times more prevalent than walking. This causes issues during the training phase, as the classification models are favouring the overrepresented activities, which reduces the classification accuracy. To overcome this problem, a sampling method was used to increase the amount of data points for the underrepresented activities.

3 Big pilot

During the big pilot, approx. 30 gateways were distributed to the partner farms (one gateway per farm), and a total of approx. 1220 cows were equipped with sensors. The main goal of the big pilot stage was to further improve the first level algorithm. This has been done by instantaneous observation of the cows in the barns and saving the start and end time of each activity in a separate file. Afterwards, the raw data of the corresponding time period has been remotely extracted. Finally, an automated script reads in the two data sources (labelling file and raw data) and feeds it into the first level (machine learning) algorithm which provides the classified activities.

3.1 Communication concept

The communication concept of the SESAM infrastructure is shown in Figure 4. The sensors attached to the cows are sending the raw (i.e. unprocessed) acceleration data via the wireless technology Bluetooth Low Energy to the basestations. Depending on the size of the barn, one to three basestations are required to ensure a reliable connection quality to all sensors. The basestations are relay stations forwarding the raw acceleration data to the barn gateway via a WiFi network. In case the cow is out of range (e. g. because the cow is outside the barn), an integrated flash memory in the sensors serves as intermediate storage for a maximum of roughly 20 hours of data. As soon as the sensor is able to establish a connection to the basestation again, the stored data is automatically transmitted. More details are explained in section 3.2.2.

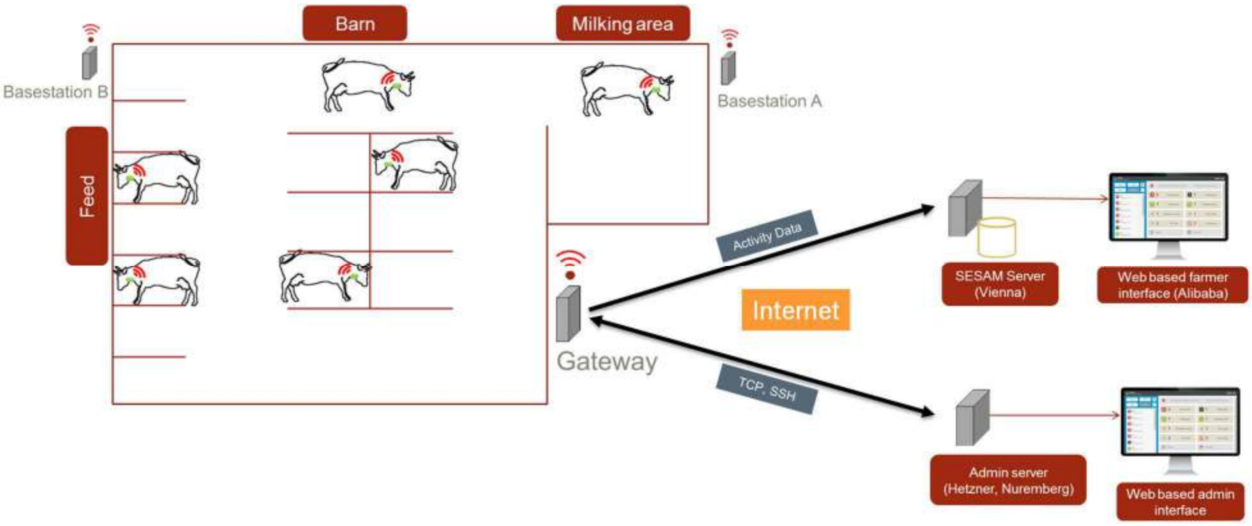


Figure 4: Communication concept of the SESAM infrastructure

On the gateway, the first level algorithm processes the raw acceleration data into activities, which are transferred to the central SESAM server in Vienna over internet. On the SESAM server, a second level algorithm is computing high level data providing information about the health status of the cows, that can be visualized in the web based farmer application called “Alibaba”. For maintenance and monitoring purposes, the gateway also sends the raw acceleration data to the maintenance (admin) server located in Nuremberg. Several applications running on the maintenance server allow authorized personnel to remotely monitor and configure the gateways, the basestations and the sensors.

3.2 Sensor

The sensors used in the big pilot are an optimized version of the sensors used in the small pilot (see section 2). The sensor printed circuit board (PCB) is surrounded and protected by a plastics housing and attached to the neck of the cows by a collar.

3.2.1 Hardware

Figure 5 shows the PCB of the sensor electronics, whereas the main components are labelled. The 3-axis MEMS accelerometer is the sensor element providing the spatial movement of the cow. The selected device ADXL362 from Analog Devices consumes only a few microamperes. Since accelerations above 4 g are very rare, the sensor is configured to a range between -4 g and +4 g. The flash memory has a storage capacity of 4 MB of data which is sufficient to store roughly 20 hours of accelerometer data in case the sensor is out of reach and this cannot transmit its captured data instantaneously to the basestation. The Blueooth chip is a device from the BGM13S series of Silicon Labs containing a radio module and a microcontroller, which consumes very low energy. The three 3.6 V 2.45 Ah mignon batteries ensure a lifetime of several years without recharging or replacing.

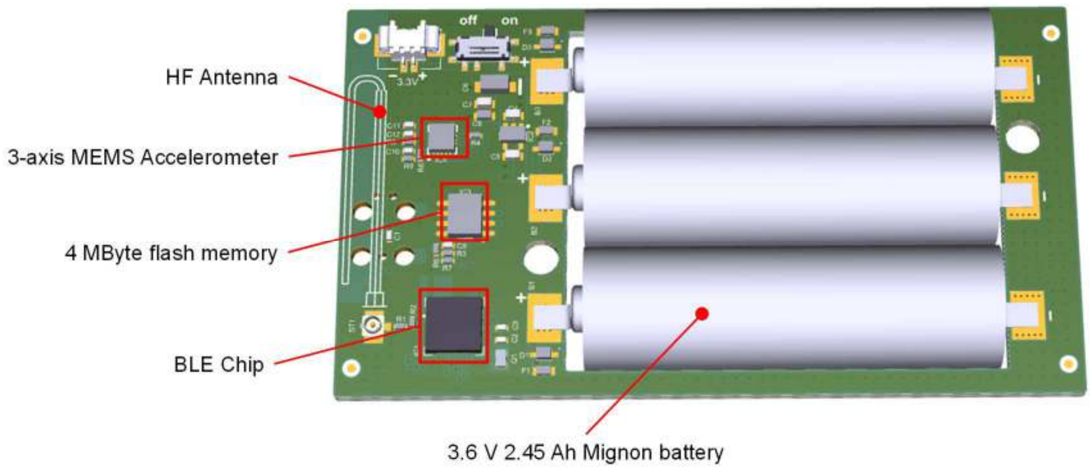
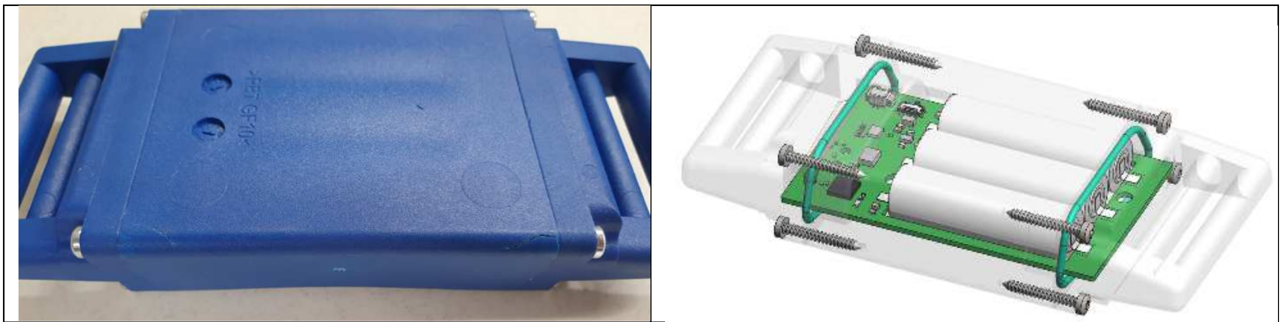


Figure 5: Printed circuit board of the sensor electronics

For the harsh environment in the barn, a robust housing for the sensor PCB has been designed by Hahn-Schickard and fabricated by Aruso-Plast Ruoff GmbH in Schwäbisch Hall using plastic injection molding (see Figure 6). The housing consists of the middle part and two caps, which are fixed by screws. An NFC tag containing a unique identification code is glued inside the housing. Approximately 2000 sensors have been assembled by Dorazil GmbH in Berlin. Therefore, Hahn-Schickard developed software and hardware tools that could by used by Dorazil for assembly and test of the sensor devices.
During production of the sensor devices in advance of starting the big pilot, many housings cracked while tightening the screws. Since Dorazil was not able to solve the problem on its own, a Hahn-Schickard mechanical expert travelled to Berlin in order to commonly investigate the problem. Finally, the problem could be solved by warming the plastic material to 40°C before tightening the screws.



(a) **(b)**

Figure 6: Sensor used for the big pilot, (a) photograph and (b) transparent view from Solidworks with printed circuit board and screws

3.2.2 Firmware

The source code of the microcontroller firmware can be downloaded from the Hahn-Schickard Github repository. The firmware has been written in C and developed with the Simplicity Studio development environment provided by Silicon Labs. During initialization, a watchdog timer starts, causing the microcontroller to restart in case of failure. The main loop waits for the accelerometer interrupt which is triggered in case enough accelerometer data has accumulated. The accelerometer samples data with roughly 12.5 Hz. According to the threshold configured in the accelerometer, an interrupt occurs roughly every 2-3 seconds. Then the data is readout from the accelerometer and written into the flash memory.

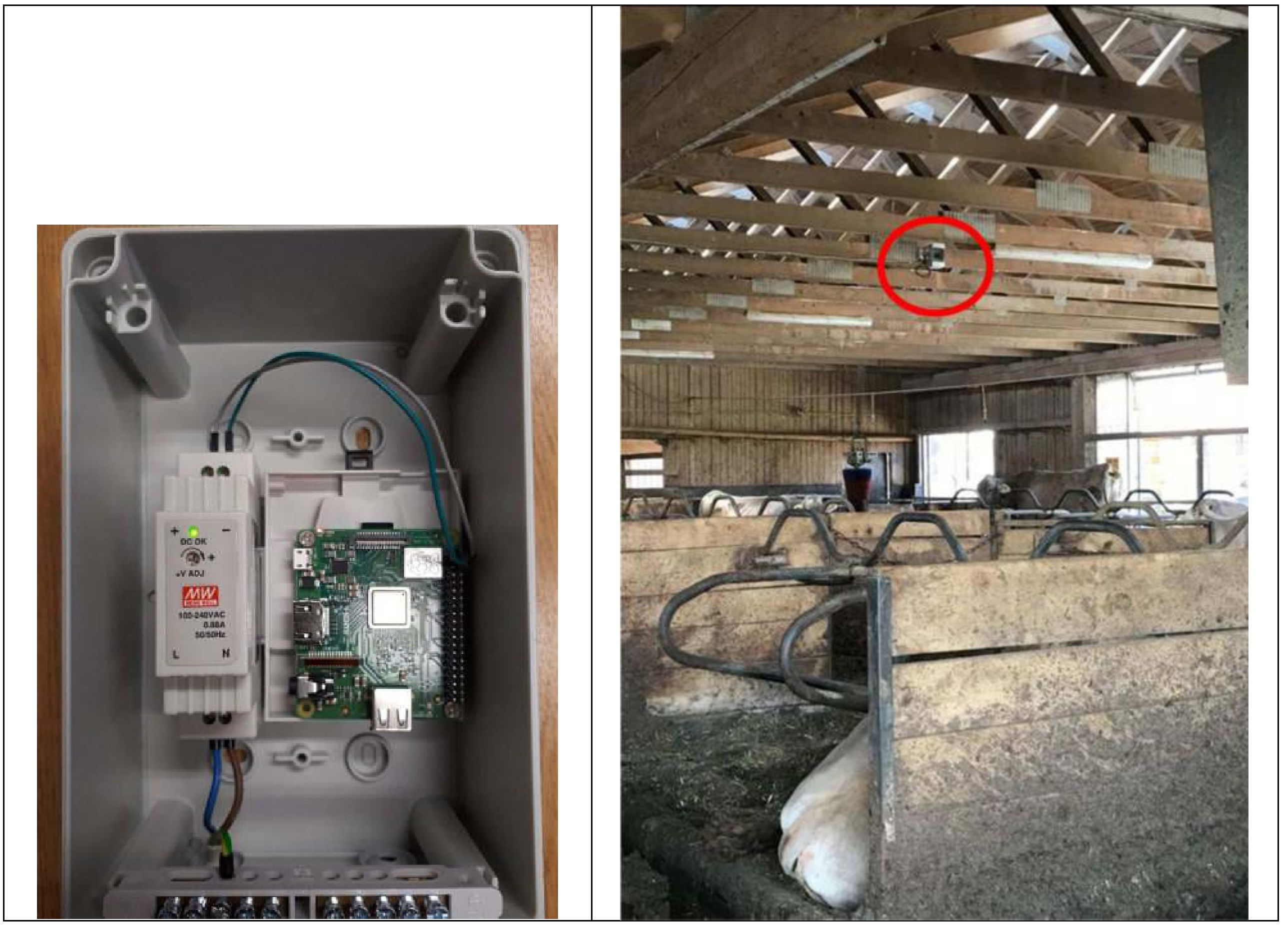
Although in principal several devices could connect at the same time to the basestation, a time multiplexing scheme is used to synchronize the sensors. For each sensor, a certain time slot with a length of 20 seconds is assigned which can be used to send raw data to the basestation. After 20 seconds, the next sensor connects to the basestation and sends data, and so on, until all sensors have been handled. A timer triggers transmission of the sensor measurements. By default, the timer is set to 1000 seconds, that means that every 1000 seconds, all the registered sensors sequentially transmit their stored data to the basestation. Hence, a maximum of 50 sensors (1000 seconds per measurement cycle divided by 20 seconds per device) can be polled during one measurement cycle.
Due to this time multiplexing scheme, it can take some time for the stored data to be completely transmitted in case the cow has been outside the barn for a long time. For example, it takes approx. 83 minutes to transmit the data of 8 h stored in the sensor flash memory.

3.3 Basestation

The basestations act as a relay node between the sensors and the gateway. Up to 20 basestations can be placed in a barn. The basestations use the WiFi network created by the gateway hotspot to connect and transmit data to the gateway.

3.3.1 Hardware

As can be seen in Figure 7(a), the basestation consists of the Raspberry Pi 3B+ and a power supply converting the 230 V mains power into 5 V required by the Raspberry Pi. This single board computer was selected as it has integrated Bluetooth 4.2 and WiFi modules as well as a sufficiently powerful processor at small costs. Figure 7(b) shows the basestation attached to a wooden beam inside the barn.



(a)	(b)

Figure 7: The basestation (a) with open case and (b) attached in the barn of M. Brauchle in Leutkirch.

3.3.2 Software

The Raspberry Pi uses Arch Linux as operating system. The main advantage is that is has a very new Linux kernel. However, it should also work with another Linux operating system. The Arch Linux enables services like *system-timesyncd* for clock synchronization.

The *basestation* program consists of several software threads. The main thread does some Bluetooth and other initialization routines and then creates a thread to manage the TCP/IP connection to the gateway. It also tries to reconnect if the connection is lost. Then the main thread creates a second thread that will scan for Bluetooth advertisements. If it receives an advertisement, it checks whether it stems from a SESAM sensor. If this is the case, it forwards an adjusted package with the advertisement information to the gateway. Then the main loop goes into a while loop in which it receives commands from the gateway. If the gateway issues a connection command, the basestation creates a new thread to establish a connection to a sensor and receive measurement data. After this is done, the thread function returns and the thread ends.

3.4 Gateway

The main task of the gateway is to send commands to the basestations, store the received data to the solid state drive (SSD), process the sensor data using the first level algorithm and send the results to the SESAM server.

3.4.1 Hardware

Figure 8(a) shows the interior of the gateway. The main component is the Jetson Nano from nVidia. This single board computer was selected due to its performance and its relatively low price. Although the graphics processing unit (GPU) acceleration is currently not used, it could be exploited if required for further algorithm versions. A large SSD with 128 GB or more in order to store the sensor raw data and the processed activity data is connected to the Jetson Nano via USB. The power supply converts the 230 V mains voltage into 5 V required by the Jetson Nano. Since the internal WiFi module is prone to instable behaviour, the USB/Ethernet adapter generates a second Ethernet port to connect the external WiFi access point/hotspot. The access point is creating a WiFi network to which the basestations can connect to. The first (native) Ethernet port can be used to connect to the internet directly by LAN cable or an LTE modem. Additionally, a WiFi USB stick is available if it’s desired to establish internet access via an existing barn WiFi.

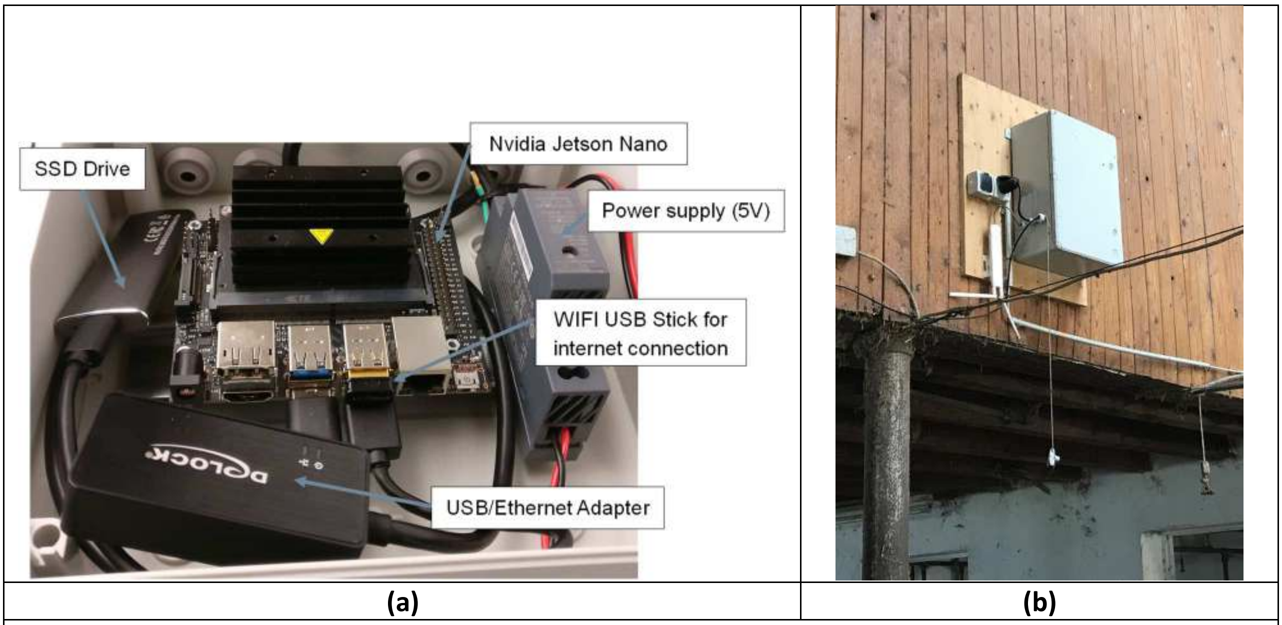


Figure 8: The gateway (a) with open case and (b) attached in the barn of M. Brauchle in Leutkirch.

3.4.2 Software

The lower block in Figure 9 depicts the software architecture of the gateway. The program *Gateway Base* decides which basestation should connect to a particular sensor, receives the data from the basestations and transmits the data to other programs (*SesamClassification* and *Logstat*) for post-processing. The *Logstat* service receives the raw sensor data from the *Gateway Base* program and saves the data into a binary raw file on the SSD. The saved sensor data can be processed manually later on, if necessary. The first level algorithm running in the *SesamClassification* service computes the activities based on the raw sensor data. First, some features are calculated from the acceleration signal and then a classification model is used to get the cow activities. See section 4 for more details. The results are written into a SQL database on the SSD. Finally, the *SesamConnector* service reads out the SQL database every 5 minutes and sends the classification results (cow activities) including time stamps to the SESAM server.
During the big pilot, it has been observed that the quite complex software architecture in the gateway is prone to losing data on its way to the SESAM server. Whereas for some gateways the activity distribution in the Alibaba application looked sensible, for some other gateways there was constantly a gap of about 10-20% of the day. Despite intensive investigations this issue could only partially be solved.

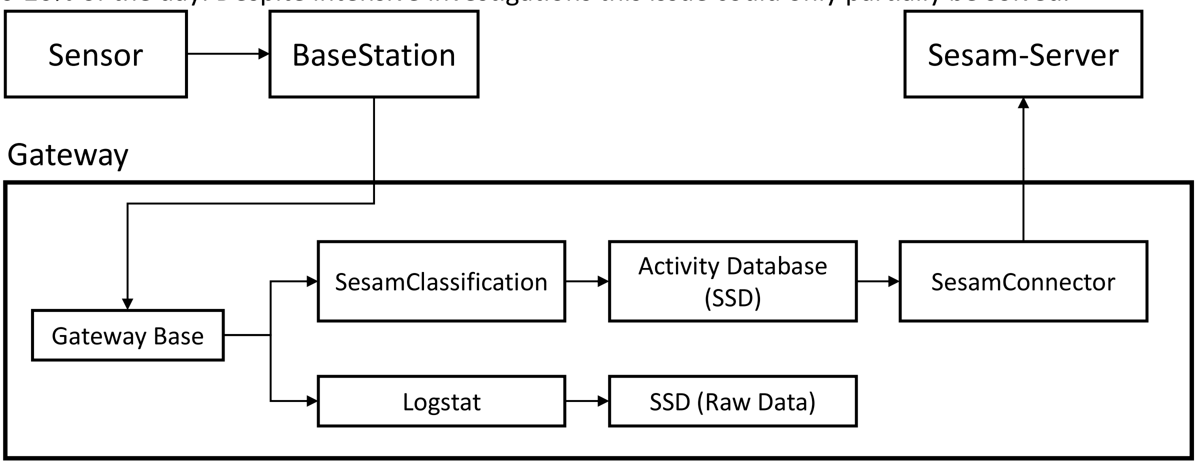


Figure 9: Architecture of the gateway software (only the most important software blocks are shown in the lower box)

3.5 Software update

Since the programming interface on the sensor PCB cannot be accessed after the sensor assembly, a firmware update using this way is not possible. In principle, the sensor firmware allows for an update over-the-air, but this feature has not been tested yet. Hence, the sensor firmware is considered “as is”.
In contrast, the basestation and gateway software can easily be updated via internet, or, in case of an internet connection failure, the SD card can be replaced with another one containing a newer software image stored on the Nextcloud server. In principle, this SD card replacement can be done by the farmer himself or another technically skilled person on the farm.

3.6 Maintenance server

The status of the sensors, the basestations and the gateways can be monitored by a web application (CGI script) running an the maintenance server (also called “Hetzner server” or “admin server”) located in Nuremberg. On the homepage, the status of all known gateways are listed, i. e. if they are currently connected or disconnected. More detailed information about installed software versions, temperature, SSD status, connected basestations, etc. can be accessed by clicking on the corresponding link in the column “Details”. Figure 10 shows a screenshot of the maintenance server homepage.
Clicking on the link of a certain gateway name in the column “hostname” leads to another website showing the active sensors. From there, the raw acceleration signals for the x, y and z axis can be monitored for each sensor. An example graph is depicted in Figure 11.

Status Sesam Gateways

Gateway IP	Gateway Status	Last Handshake	Hostname	SQL Query	Details
10.0.0.2	Connected	1 min 34 s	GW2	Activity Data	Show
10.0.0.10	Connected	1 min 24 s	GW10	Activity Data	Show
10.0.0.11	Connected	57 s	GW11	Activity Data	Show
10.0.0.14	Connected	32 s	GW14	Activity Data	Show
10.0.0.19	Connected	1 min 48 s	GW19	Activity Data	Show
10.0.0.20	Connected	56 s	GW20	Activity Data	Show
10.0.0.21	Connected	36 s	GW21	Activity Data	Show
10.0.0.22	Connected	2 s	GW22	Activity Data	Show
10.0.0.23	Connected	1 min 16 s	GW23	Activity Data	Show
10.0.0.24	Connected	6 s	GW24	Activity Data	Show
10.0.0.25	Connected	1 min 52 s	GW25	Activity Data	Show
10.0.0.26	Connected	10 s	GW26	Activity Data	Show
10.0.0.27	Connected	26 s	GW27	Activity Data	Show
10.0.0.29	Connected	1 min 52 s	GW29	Activity Data	Show

Figure 10: Web interface on the maintenance server showing a status overview of the gateways

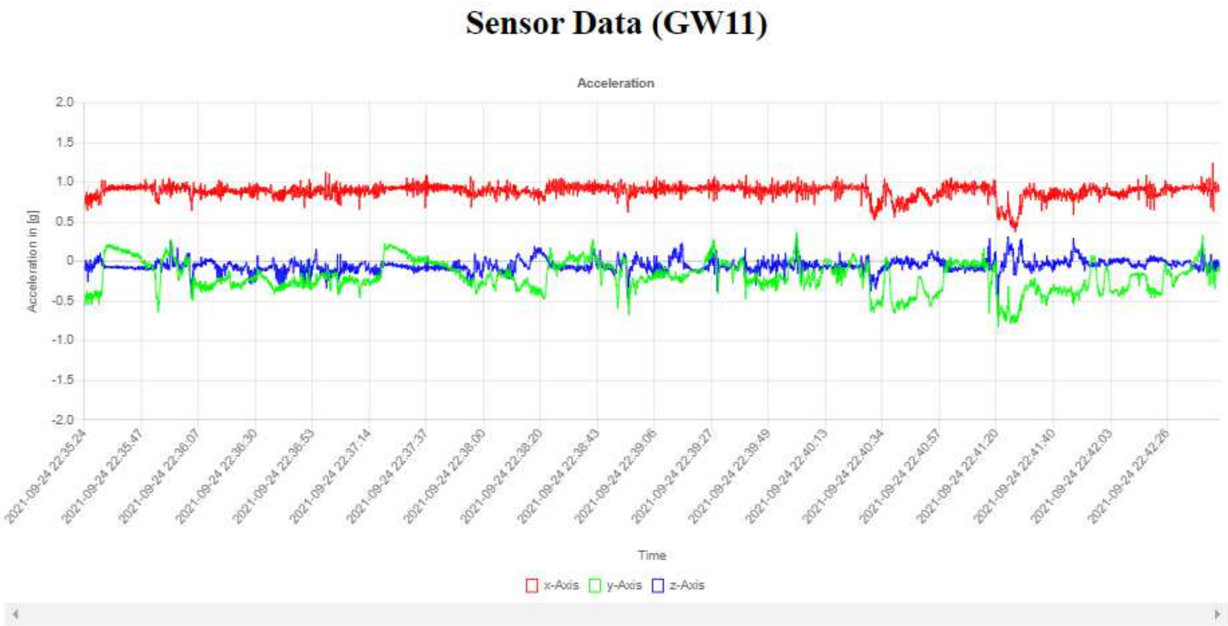


Figure 11: Raw data of the accelerometer (one separate line for each axis)

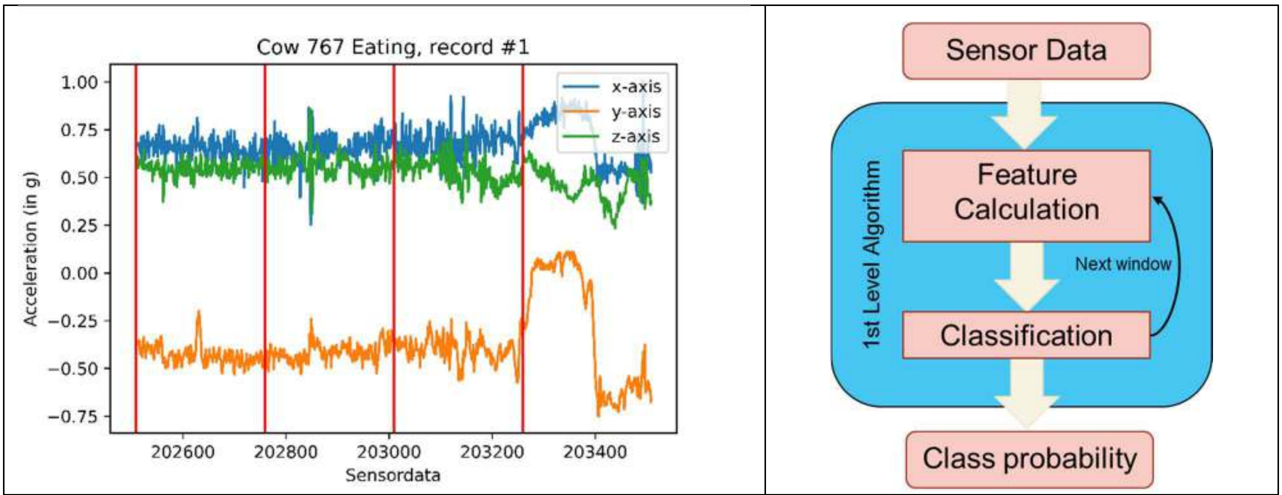
4 First level algorithm

The first level algorithm calculates the activities of the cows from the raw acceleration data.

4.1 Theory

As input for the classification algorithm data of an accelerometer is used, measuring the acceleration along the X, Y and Z-Axis. Additionally the derivate and the integral of each axis was calculated. Based on this data a featureset consisting out of 345 features was created. These features consist out of time-domain features, such as different statistical measures like the mean and the standard deviation, but also frequency-domain features to capture activities with repetitive nature like e.g. walking. These features were calculated over a window size of 250 samples, which corresponds to a window size of 20 seconds with a sampling frequency of 12.5Hz of the accelerometer (see Figure 12(a)). To prevent overfitting a smaller set of features was selected, consisting out of 40 different features, which were determined to be the most influential ones.

The resulting features were used as input for a decision tree ensemble consisting out of multiple decision trees. Each decision tree calculates the probability for a given class (Eating, Laying, Ruminating, Standing, Walking). These probabilities get averaged across the trees and the output of the model will be the class that yields the highest probability. This way we get a predicted class for every 250 samples of accelerometer data. Figure 12(b) depicts the flow of the first level algorithm.



(a) **(b)**

Figure 12: (a) Windows containing 250 samples, (b) flowchart of the first level algorithm

4.2 Specification of activity accuracy

On 02.03.2020, the specifications for the classification accuracy were revisited and priorities for the classes were defined together with LKV BW and CAA (see Table 1). As a result, the two classes *Ruminating* and *RuminatingAndLay* were combined into one ruminating class and the minimum classification accuracy for the classes *Laying* and *Standing* was lowered to > 85%. The average accuracy of all classes is 88%.

Table 1: Specification of the activity accuracy, as defined on 02.03.2020

Priority	Class	Specification
1	Walking	> 90%
2	Ruminating	> 90%
3	Eating	> 90%
4	Laying	> 85%
5	Standing	> 85%

4.3 Continuous improvement by new labelling

The first level algorithm had been developed based on the data from the small pilot. As shown in Figure 13, the results were quite promising: Each activity was predicted with an accuracy of minimum 83%. Surprisingly, the results with the first dataset captured after the start of the big pilot in July 2020 were much worse than shown in Figure 14. Hence, the first level algorithm has been optimized in seven steps from November 2020 until July 2021 using newly labelled data from different farms. Whereas some of the datasets have been used for training, others have been used as independent testset.

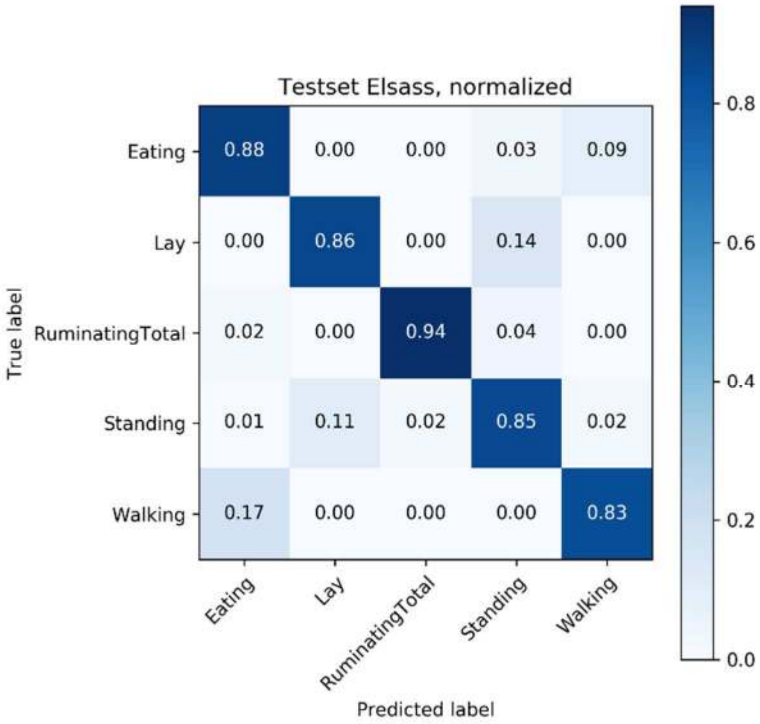


Figure 13: Training with data from Bavaria/Austria, test on data from Alsace (small pilot)

The final result of the improvement steps has been reported on 16.07.2021. Table 2 and Table 3 reference the labelled data used for traning and as testset. Figure 14 shows the final results.

Table 2: Labelled data used for training

Labelling person	GW no.	Date
Morgane Hoenen	GW26	03.11.2020
Morgane Hoenen	GW26	06.11.2020
Rebekka Kromer	GW44	22.10.2020
Rebekka Kromer	GW44	30.09.2020
Morgane Hoenen	GW26	02.12.2020
Rebekka Kromer	GW44	14.12.2020
Rebekka Kromer	GW44	31.12.2020
Rebekka Kromer	GW44	08.02.2021
Rebekka Kromer	GW44	19.05.2021
Toni Maast	GW19	01.07.2021
Nejc Valcl	GW22	09.07.2021

Table 3: Labelled data used as testset

Labelling person	GW no.	Date
Anita Ule	GW50	22.01.2021
Morgane Hoenen	GW17	30.09.2020
Anita Ule	GW50	24.02.2021
Nejc Valcl	GW50	24.02.2021

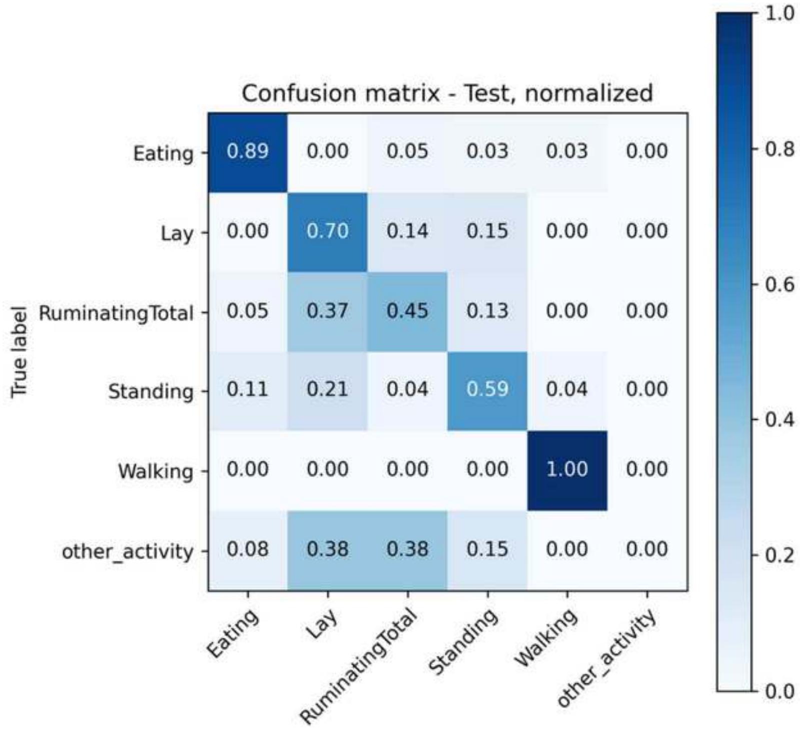


Figure 14: Results reported on 16.07.2021

It becomes evident that the good results from the small pilot shown in Figure 13 have not been achieved in the big pilot. Especially the problems of the Ruminating class persist (accuracy of only 45%). With an average accuracy of 73%, the specification of 88% has not been met. On the other hand, the most positive aspect is that the priority 1 specification for the walking class has been met. Also, the priority 3 specification for eating has almost been met. These results implicate that the current model gives decent results, but it has to be further improved in the future.