論文 / 著書情報
Article / Book Information

題目(和文)	
Title(English)	Data Augmentation for Time Series Data in Edge AI Devices for Cattle Behavior Estimation
著者(和文)	LI CHAO
Author(English)	LI CHAO
出典(和文)	学位:博士(学術), 学位授与機関:東京工業大学, 報告番号:甲第12470号, 授与年月日:2023年3月26日, 学位の種別:課程博士, 審査員:伊藤 浩之,岡田 健一,德田 崇,白根 篤史,吉村 奈津江 ,COSENTINO Sarah
Citation(English)	Degree:Doctor (Academic), Conferring organization: Tokyo Institute of Technology, Report number:甲第12470号, Conferred date:2023/3/26, Degree Type:Course doctor, Examiner:,,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Data Augmentation for Time Series Data in Edge AI Devices for Cattle Behavior Estimation

by

Chao Li

A Ph.D. dissertation submitted in partial fulfilment

of the requirements for the degree of

Doctor of Philosophy

in the

Department of Electrical and Electronic Engineering

in the

School of Engineering

of

Tokyo Institute of Technology

Supervised by

Assoc. Prof. Hiroyuki Ito

Spring 2023

To my family,

Acknowledgement

First, I would like to thank Assoc. Prof. Hiroyuki Ito for giving me this wonderful opportunity to be a part of Ito Laboratory, and Tokyo Institute of Technology. I would like to express my sincere thanks to him, for his valuable guidance till now starting from the day of enrollment. Even with his very busy schedule, he always gives me great help and support in my research. I am thankful for his advice, comments, and suggestions on every issue. For sure, without his guidance this work could not be done. In short, I appreciate his trust in me.

I am deeply grateful to Prof. Kenichi Okada, Prof. Takashi Tokuda, Assoc. Prof. Atsushi Shirane, Assoc. Prof. Natsue Yoshimura, Assoc. Prof. Sarah Cosentino, for their valuable time for examining my thesis.

I am also grateful to Assoc. Prof. Ludovico Minati and Assist. Prof. Korkut Kaan Tokgoz for their help and support. Their advice on every issue in my research is very valuable and I always tried to follow them. They gave me detailed and instructive suggestions for developing my research and writing my thesis, which helped me to clarify and express my ideas efficiently.

I would like to especially thank Prof. Kazuya Masu for his encouragement and consent when I applied for the Ito laboratory. I am glad that he was the president of Tokyo Institute of Technology from the year I entered.

I also, am thankful to other faculty in the laboratory, that is, Prof. Noboru Ishihara, Prof. Katsuyuki Machida, Prof. Shiro Dosho, Assist. Prof. Sangyeop Lee, Assist. Prof. Parthojit

Chakraborty, Assist. Prof. Aravind Tharayil Narayanan, for their guidance and suggestions on my research work during the group meetings. Their advice and explanations are always very helpful.

I, also, would like to thank all other members of the cow research group, that is, Masamoto Fukawa, Jim Bartels, Aran Hagihara, Ikumi Rachi, Kazuki Maari.

I would like to thank my parents and relatives for their silent support in China for my study in Japan. Their support gives me the motivation to persist in research.

Abstract

Internet of Things (IoT) technologies depend on the ability to automatically interpret various sensor data sources, which usually produce time series. For example, these may represent the motion, such as acceleration, of a person, animal, vehicle, or other machine during normal daily life or activity. To perform useful tasks, IoT devices need to be able to classify the recorded time series according to some set of behaviors or situations. Since IoT devices must operate in the real world, many challenges need to be addressed. One challenge is that collecting data for training artificial intelligence systems, such as neural networks, is often very difficult and expensive, because it requires extensive observation and labeling of the individual behaviors. Another challenge is that, typically, the behaviors are very unbalanced: some behaviors are more frequent than others, but all of them need to be classified with high accuracy.

The aim of the thesis is to propose new ways of solving these two problems through the paradigm of data augmentation, which is still not widely explored from a time series perspective. Essentially, this involves generating a large amount of new synthetic data, starting from a limited number of measurements and combining the original data with appropriate hypotheses about the features that should be maintained and those that can be changed for augmentation. Throughout the thesis, different approaches are presented and compared in a systematic way. Some of these methods are based on specific characteristics of the

system under consideration, while others are more abstract. The complexity of the mathematics also varies.

To address the matters indicated above, this thesis focuses on cases of automatic classification of cattle behaviors, such as feeding, resting and walking. The resulting data are difficult to label and have an imbalanced distribution, therefore, are appropriate as examples to deal with. In the target application, it is important to accurately quantify the proportions of these behaviors because subtle changes can be a strong indicator of disease development that would otherwise be invisible to farmers. This analytical insight enables a form of precise livestock management, which can help improve economic efficiency and reduce environmental emissions, for example by limiting the number of livestock that are wasted. As with many other applications of the IoT, inference needs to take place at the edge, aiming for maximum accuracy under limited resources. This paper shows that data augmentation methods allow training convolutional neural network (CNN) models to achieve extremely high accuracy that would otherwise be very difficult to attain.

First, based on the specific characteristics of the monitored system, a paradigm in which the rotation of the sensor around the cow's neck is simulated is used as a means of generating new data for augmentation. An initial dataset of 2 cows is used. The results show that from a baseline starting at 77%, an overall accuracy of 98% can be obtained on a large dataset from two cows, which is heavily imbalanced over the 5 behaviors under consideration.

Second, a more realistic scenario is considered wherein the amount of data is approximately 10 times smaller, but gathered from 6 cows. In order to maintain a high level of accuracy, several augmentation methods are deployed. These are based on assumptions specific to the system under consideration, i.e., cow behavior. In addition to sensor rotation,

they include reversing time series, recombining time series between segments, and performing other time-domain operations. On this basis, the accuracy of 94% is obtained from an 83% starting point.

Third, a more principled approach is developed and applied. Starting from the observation that some of the previous augmentation methods retain only linear features of the signal, an implementation using a Fourier surrogates method is created. The method is combined with a sampling technique that is well suited to address data imbalances, while the surrogates avoid data duplication. Unlike the methods considered in the previous chapters, this approach is not designed for a specific system, because it is not based on an operation that only has meaning for a particular scenario, such as rotation. It could therefore, in principle, be considered for other applications in time series classification. The highest recognition rate obtained is 96%, and even more important is the fact that the most problematic behaviors show improved accuracy. In addition, since the method is based on fundamental principles of signal theory, it is possible to enquire about which ones are specific aspects of a time series' content that lead to successful data augmentation. Through a series of data reduction steps, it is found that autocorrelation, mean and variance convey most of the information about cow behavior. As a result, it seems that this approach may be also applicable to other systems. This is tentatively confirmed in additional analyses of three datasets coming from different applications and sensor time series. Furthermore, with a better understanding of the correlated signal features, it will become possible to design more specific hardware to extract and analyze these features.

This thesis demonstrates a coherent journey through data augmentation methods, starting with a simple implementation based on system-specific features and ending with an abstract approach that could perhaps be widely used and helps to identify relevant signal features. The results obtained have been made possible by this progression of approaches, because each result inspired an insight for a new, broader approach. The method finally

proposed is built up in this form, and would not have been possible to guess at the start. In addition, the multidisciplinary aspect of the research provided the practical basis for understanding the importance of the rare behaviors. This provided motivation for seeking better accuracy in all behaviors, which is different from the usual focus on overall performance. Specifically, the results given in the published papers open up a variety of applications of edge inference on the IoT that would otherwise be difficult to implement because sufficient accuracy is very difficult to obtain with realistic dataset sizes. Additional work on very different dataset types that are representative of other IoT applications provided an initial confirmation of the potentially broader usefulness of the methods. Future work should study more deeply the properties of the proposed methods in combination with other neural network types, as well as the design of specially-designed hardware architectures based on time series features determined to be relevant. As for the specific applications under consideration, the results obtained confidently achieve automated livestock monitoring with high accuracy levels.

Contents

List of Figures

List of Tables

Chapter 1

Introduction: Edge AI, time series and real world challenges

1.1 From IoT to Edge AI and time series

According to the general understanding of the term, the Internet of Things (IoT) promises to solve a wide range of contemporary societal problems by improving the connectivity of devices, cloud servers, and people. Essentially, it aims to enable various forms of distributed intelligence that allow higher quality, more reliable and sustainable services while minimizing resource expenses. Recent reviews argue that this should be intended in the broadest way, as IoT applications aim to reduce waste of energy, land, and materials [1-3]. Because of the always increasing focus on intelligence, it has also been proposed to rename the field as Artificial Intelligence of Things (AIoT), to put more emphasis on the artificial intelligence aspect [4,5].

For example, one application area is the smart city, where the optimization of large services such as waste collection and transportation has a big impact on quality of life and

sustainability [6]. Other application areas encompass the industrial IoT, which focuses on equipment maintenance, and agriculture and farming IoT [7], which is another promising domain [8]. In my opinion, it is so because, after all, agriculture and farming have always been two fundamental activities for human survival. According to accepted international data, their scale and density make the corresponding emissions not less important than those of urban living and industrial manufacturing [9]. In fact, I would like to point out that the field of IoT in agriculture and farming is itself very broad, as it encompasses a wide variety of use cases, control targets and data sources. As discussed in a recent review, some representative cases are about the monitoring of land plots, logistics of raw materials, semi-finished and finished products [10]. In agriculture and farming, for example, the IoT supports new ways of managing the resources and planning [11,12]. This thesis treats cattle farming as an application scenario, although the technology being developed, as described below, appears to be applicable in a broader sense, too.

It is widely understood that one of the main aspects of the modern idea of the IoT is that, even though a large overall amount of information is collected every day (maybe on the order of gigabytes), very little data (on the order of 10 bytes) is received from each node at a time [13,14]. In this sense, for this approach to work, the data acquisition process must be extremely data efficient, so that very small packets of data can contain as much meaningful information as possible for a given application. On the other hand, IoT sensor devices operate under complex conditions of high uncertainty and noise, and, in my opinion, this is clearly true when considering applications such as monitoring the behavior of free-roaming animals. For these reasons, one of the main requirements of successful IoT projects is that a significant amount of data compression should be performed on the sensor devices themselves [1-4]. This is because issues such as cost, data transmission limitations, and power availability in general make it difficult sending raw time series to cloud servers for remote processing [12,13].

It is for this reason that IoT and AI are linked to form the field now known as edge AI, which is closely similar to the idea of AIoT. According to recent reviews, in many IoT applications, sensor nodes should have a degree of intelligence that is at least sufficient to significantly reduce the data to be sent: for example, after categorizing a time series that is several kilobytes long into a set of possible behaviors, it should be compressed into a few bytes [15-17] . It is also clear that there are many sensor types that generate by themselves small amounts of data, that do not need to be reduced by means of classification: some examples are energy meters, solar irradiation sensors, air quality sensors, and so on [18]. However, I think it is reasonable to say that a growing number of IoT applications need to perform significant data reduction very near the sensors, and the emerging way to achieve this is undoubtedly Edge AI [19-21].

It is generally understood that the IoT, as the name implies, is the interface between the real, physical, social and digital worlds, and belongs to the field sometimes referred to as cyber-physical systems [22]. By definition, the real world is dynamic, which means that all the variables describing it change over time. This means that all sensors produce some kind of time series, ultimately. As mentioned earlier, some of these time series are sampled slowly enough that each point can be sent directly to the cloud for remote processing, but many of them need to be analyzed locally. In the application considered in this thesis, where cattle movement is monitored using accelerometers, I will show that acceptable sampling rates produce a data stream that is several orders of magnitude higher than what, for example, a low-power wide-area (LPWA) IoT infrastructure can handle. In fact, there are many examples of such situations, such as monitoring human behavior, or the vibration of a vehicle or machine. One might also think of completely different parameters that can change too quickly over time to be sent as raw data, such as water pressure in a pipe, current in a transformer, etc. To sum up, it is widely agreed that a large part of Edge AI is about dealing with time series [23]. In practice, this usually means picking a window on a given time

series and classifying the corresponding state of the system under consideration according to a set of possible behaviors (usually small, meaning, less than ten). Then, only the label corresponding to this behavior and the associated timestamp need to be transmitted remotely [13,18,21,24,25]. This thesis is about techniques to improve the accuracy of the classification process, which is eventually intended in the way to be as abstract as possible and, therefore, potentially useful also in other scenarios.

As one can read in any book on this topic, it is possible to quantify many characteristics of any time series, such as mean and variance, or more complex aspects reflecting frequency content or irregularity over time [26]. Interestingly, I think it can be said that because compute power, energy and cost are important defining factors in the field of edge AI, the usual approaches for time series classification that are found in many fields are not always the best solution. On the one hand, the amount of processing required to compute representative statistical parameters is usually not small. On the other hand, it is usually necessary to make assumptions about the characteristics of the particular system to be monitored to select one feature or another, but a totally generic solution would be preferred, so that the system can be adapted easily. One consequence of these facts is that, typically, edge AI of time series data is more and more often achieved by simply feeding the time-windowed data to a classifier, which is usually a deep neural network, such as a convolutional neural network (CNN) [24,27]. Contemporary papers state that this shifts the complex and time-consuming task of identifying and extracting relevant features to the training process itself, allowing developers to quickly explore various network sizes and configurations without having to rethink how the feature vectors are created each time [28-32]. However, I think it can be said that the process of training such networks is by no means trivial. That is the key problem, as will be explained below. The purpose of this thesis is to provide some techniques and algorithms to help with the related issues, that will be listed in the next subsection.

1.2 The real world vs. network training: conflicting requirements

I think that anyone knowledgeable about training even the simplest AI models can explain what the desirable features of a good dataset are: large size, homogeneous, high-quality behavioral labels, and a balanced distribution of the behaviors to be identified. It is also well-known that acquiring a dataset having such features is very difficult, time-consuming, and sometimes close to impossible [33,34].

First, it is generally known that acquiring the data itself can be a time-consuming operation. Since training an AI model requires access to entire (usually raw) time series, it is necessary to organize a data acquisition session in which each system to be monitored (e.g., a cow) has a recording system installed with a sufficiently deep memory and a sufficiently long battery life. This step is not trivial: it implies that the training data for the AI models cannot usually be simply acquired using the same IoT wireless link provided for the final application. This is because the required data stream is much higher in terms of the amount of data to send per minute or hour (for example, WiFi or Bluetooth may be needed instead of LPWA). As one can imagine, this step results in considerable human, technical, and logistical costs, including installing the recording devices, checking and operating them, and retrieving the data. These issues were highlighted in a recent survey of this area [35]. Moreover, it is also clear that the real world in which Edge AI and IoT systems must operate is far from homogeneous: not every street in a city has the same characteristics, not every farm has the same environment for raising animals, and not every animal has the same behavior because of different strains, sexes, ages and temperaments. Therefore, I think it should be clear that homogeneity is very hard to attain when it comes to acquiring data in the field. Instead, in order to be useful, these systems have to deal with considerable and

unpredictable variability on a daily basis. If they can't handle it reliably, performance declines and a technology becomes a kind of curiosity rather than an enabler of society [13,36-38].

Just as it can be said it is pointless to own a large car if one does not have access to gasoline, the same is true for time series datasets. The usefulness or not of the datasets used to train edge AI devices depends not only on the quantity and quality of the sensor data itself, but also on the associated labels. It is clear that, without labels, data is nearly useless. In relatively rare cases, unsupervised learning can be used, or self-supervised techniques can alleviate the problem. In most cases, however, reputable human operators need to patiently tag a set of possible behaviors on each time point. This operation is not only unappealing to humans, but also very time-consuming and error-prone unless done very carefully by experts, often requiring the involvement and cross-checking by multiple operators. As discussed in several recent works, the result is that labeling a dataset can be as costly as acquiring it, if not even more; therefore, many automatic and semi-automatic labeling (also called annotation) approaches have been proposed [39-41]. In the case of cattle behavior, it is a very important operation for a trained operator to guess the behavior of cattle by their movements and postures in video frames; any mistake will lower the final system performance. It is clear that human experts are necessary and always preferable for creating a good quality training dataset, however, at the same time, it is not feasible to employ expert human operators to classify cow behavior in general farming application situations.

Last, but not less important, there is the data imbalance problem. From a mathematical point of view, nearly all training algorithms compute an overall accuracy: this is a valid and unbiased classification performance measure, but only in the case when the amount of data for each behavior is matched. It is evident from everyday experience that such a situation is a rather special case and, on the contrary, it is normal to have different occurrence rates. For example, we spend much more time sitting at our desks and walking around than we

do eating or washing up, and this is true for animals as well as other systems such as machines. However, less frequent behaviors are often not unimportant in terms of the impact of classification performance on the usefulness of edge AI systems. This raises a dilemma. Should we adapt the performance metrics used in training to this situation in some way? Or should one acquire more data than needed and then selectively retain only some data to balance out the behaviors? A solution is needed that allows all the data to enter the training process while ensuring that those behaviors that occur the least are not neglected [42-45].

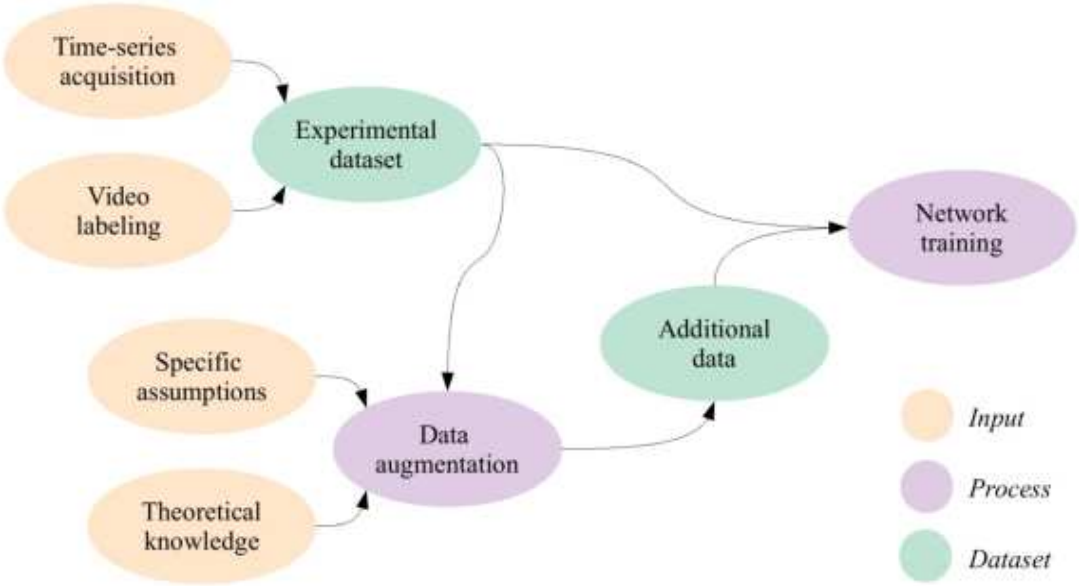


Figure 1.1: Conceptual diagram of data augmentation in the context of cattle behavior time series.

1.3 The promise of data augmentation

To be clear, the cost and effort of collecting and labeling datasets can be significantly higher than the cost of processing them automatically. Then, data augmentation represents an attempt to trade computational power for human effort. As can be seen in **Fig. 1.1**, the

idea is to post-process the data in an appropriate manner after labeling. Doing so, the time series that actually go into the training process are much more numerous, representative and balanced compared to the time series initially obtained from the recordings. This approach actually originated in the field of computer vision, where it is well known that the generalization performance of neural networks can be a serious problem. To help model training, computer vision scientists have proposed a series of artificial operations that are considered "reasonable" according to human observers, because they do not change the meaning of an input image (e.g., a road sign) while significantly altering its perceptual-level features. Some of these operations are linear and trivial, such as changing brightness, contrast, or size. However, in most cases, the operations are not trivial and include rotating, warping and distorting the image as much as possible while staying within the bounds of what a human observer would consider as a recognizable, complete image. In effect, the goal is to expand the boundaries of the classifier so as to maximize its ability to respond correctly to unknown inputs. For example, consider a system designed to recognize "stop", "give way" and "one-way" signs. If only the original images are fed into the training process, one immediately realizes that even small changes irrelevant to a human observer can lead to unpredictable classification errors with potentially catastrophic consequences – one should think of self-driving cars. On the other hand, the situation is more reassuring if each image is presented numerous times after suitable manipulations and network performance is confirmed based on this variability [46-49].

By reading up to this point, a reader might already imagine that the success of data augmentation depends heavily on the validity of the underlying assumptions. To what extent are specific operations that alter the data allowed to be applied? Which operations truly represent the real world? These two seemingly simple questions contain many dilemmas, because one must take care not to corrupt the labeling of the data. When considering road signs, it may be necessary to rotate them by about 30 degrees. However, if one observes

that including a larger rotation will ultimately have a beneficial effect on overall performance, should one do so, keeping in mind that this will likely not be encountered in a real-world application? Similar arguments apply to warpage and twist, for example. At this point, it is clear that not only is the data acquisition process fraught with compromises and uncertain decisions, but in fact, data augmentation adds more assumptions. This may sound concerning, however, there is now an extensive literature, outlined below, confirming that its overall effectiveness is absolutely positive. As a result, it is integrated into the training process of many image processing neural networks and, in fact, it is quite difficult to train these neural networks without its help [46-49].

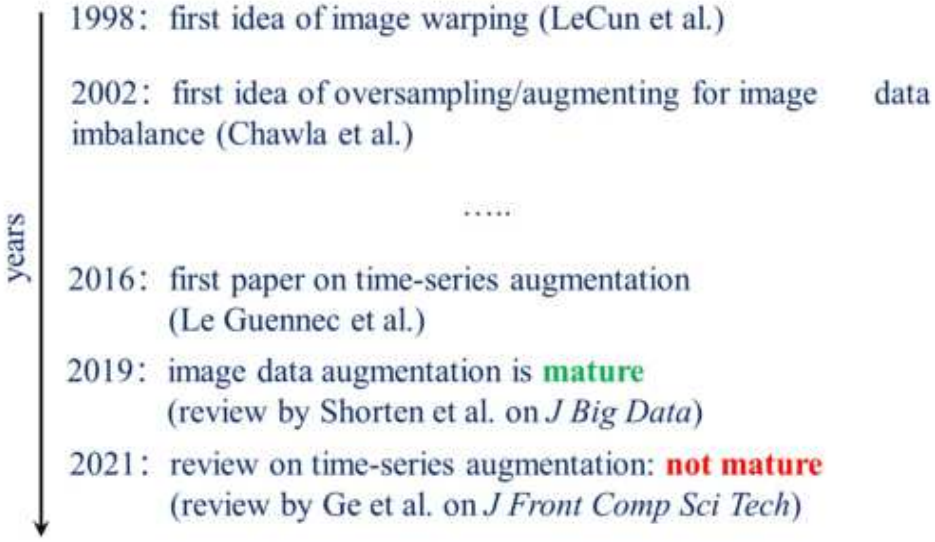


Figure 1.2: Simplified time-line on the development of data augmentation (see text for reference citations).

However, as mentioned above, most IoT applications involve time series, not images. This not only means a much smaller amount of data to start from, but also a much smaller number of possible operations for augmentation, since the number of data dimensions

equals 1 instead of 2. Not only, but also choosing what is an acceptable criterion becomes more complex, as human perceptual plausibility no longer applies, since our visual system can be used as a reference for images, but not signals. We do not know, how a reasonable or bad signal "looks like". Data augmentation has so far been used much less in the field of time series classification than image processing, though the number of publications addressing specific applications is increasing rather quickly [50-52].

To provide further historical perspective, a time-line putting side-by-side the development of data augmentation for images and time series is presented in **Fig. 1.2**. In brief, one can see that the idea of data augmentation for images was introduced in the year 1998 by LeCun et al. [53] as regards warping. A few years later, in the year 2002, it was also introduced by Chawla et al. [54] for oversampling with the aim of reducing class imbalance. About 20 years later, in the year 2019 a review by Shorten et al. [55] said that the field was essentially mature. By comparison, time series data augmentation was introduced around the year 2016 by Le Guennec et al. [56], and a recent review in the year 2021 by Ge et al. [57] indicates that the field is still not mature. The development of data augmentation for time series is several years delayed compared to image processing.

I think this is quite a practical problem because most sensors used in IoT applications generate time series data. Therefore, improving the accuracy of time series data classification when performed by edge AI devices is very important for the society. That is the purpose of this thesis.

1.4 The rationale for and contribution of this work

After having read up to this point, it should be clear that data augmentation is really not an exact science. On the contrary, it involves many assumptions and compromises that draw a "gray area". Particularly, one aspect is how much the transformations applied to the

data should reflect an understanding of the system under consideration, rather than general issues about signal theory and related topics. Some recent reviews in this area clearly demonstrate this complex situation, as they show that there is no universally accepted "right" or "wrong" choice. In fact, improvements in accuracy are often seen as a benchmark for measuring the effectiveness of data augmentation methods [42,51]. Inevitably, this paper will follow this approach as well, to some extent. However, as will become clear in the next sections, unlike most existing work in this area, I will follow a structured approach, gradually shifting from a very system-specific implementation to a very abstract one.

In the current application, the behavior of cattle needs to be classified based on tri-axial accelerometer data obtained from sensors embedded in the collar, as visible in **Fig. 1.3** [58,59]. In fact, the path outlined in the next few chapters starts with a physical problem that is not in itself related to data augmentation. Rather, the problem is that the collar can rotate around the cow's neck, so the system should be sufficiently insensitive to these random rotations around the sensor axis parallel to the neck [60]. In the process of adding data to solve this invariance problem, I realized that the imbalance in the data set was a serious impediment to network training. Then, these two ideas were combined to get the notion that, in effect, random rotations can be used not only as a method to make classifiers less sensitive to collar rotations. They can also be useful as a means of generating artificial new data to counteract the initial behavioral imbalance. Early results obtained with this approach are presented in Chapter 3 of this thesis. Although the concept was new and valuable at the time, it is clear in hindsight that it is based on system-specific considerations. Such rotations are not meaningful in the other cases where the time series do not correspond to different axes of the same sensor, or where the sensor itself cannot be rotated.

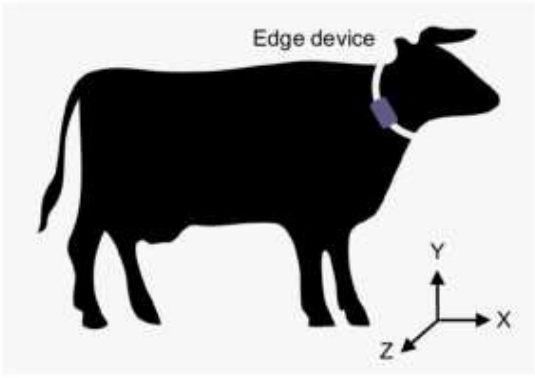
As is often the case in IoT system development, the development of hardware and software, as well as the data collection for training in edge AI systems, took place in parallel. By the time the first study was nearing its conclusion, more cows had been covered by the

new data, but the amount of labeled data for each one was smaller. This created a new challenge, which is very typical for this type of system: the total amount of data used for training was small, especially compared to the variability between individual recording sessions. Initial attempts to improve performance using random rotation methods were not completely satisfactory. At this point, since the work had been explicitly focused on data augmentation, I included a number of additional operations based on assumptions that are considered reasonable and that do not destroy the underlying labels and behavior of the data [61]. These include, for example, playing back the time series in reverse, and reassembling windows by cutting and merging window segments. While consistent with the latest techniques in computer vision, these operations are based on "common sense" considerations only. They are accepted because of their beneficial effects on accuracy and ease of computation. In many senses, they represent a middle ground. They are not as system-specific as random rotations; however, they lack a strong theoretical basis. However, as is often the case in this field, they can be accepted based on the important performance improvements obtained on smaller data sets. Therefore, results for this sort of middle ground are given in Chapter 4.

In the post hoc analysis of these results, one aspect stood out. In fact, the principle of time reversal may be seen as problematic, because it is almost impossible for the actual cow behavior to be reversed in time. Nevertheless, there are clear accuracy benefits to be gained by including this operation among the transformations for data augmentation. From a theoretical point of view, it can be seen as altering the nonlinear structure in the signal, which means that only linear features are preserved. This is because the order of events is changed. A definition of linear and nonlinear features is introduced in Chapter 5. In turn, this is exactly the operation performed during the generation of the so-called surrogate data using methods such as phase scrambling. The focus of the investigation therefore shifted to exploring the use of surrogate data as a basis for data augmentation, particularly as a means

of avoiding data duplication when resampling multiple times on the same time series to counteract behavioral imbalances. This approach could achieve higher performance than previously recorded. At the same time, it makes no assumptions about specific operations that may or may not be reasonable for a given system, and is based on basic properties of the Fourier transform.

(a)



(b)

Figure 1.3: Position of the monitoring device. (a) Edge device worn on cow's neck (b) Schematic diagram of the coordinate frame of the three-axis acceleration sensor with X-axis (longitudinal), Y-axis (vertical) and Z-axis (horizontal)

Therefore, it is clearly a superior approach compared to the previous methods [62]. Furthermore, since the surrogate data generation methods specify which features of the initial experimental recordings should be retained and which should not, at this point it is possible to gain more insight into the practical aspects of the signal content that support high classification accuracy. It is clear that it is the mean and variance and autocorrelation that dominate, while the cross-correlation and nonlinear aspects are less important. By this point, the data augmentation had evolved into a completely different situation compared to the beginning of this work. That is, from a technique based on operations that are reasonable according to common sense understanding, it became a fully principled approach achieving not only higher performance, but also higher abstraction from the specific features of the initial application. In addition, a new way of addressing the imbalance problem was introduced, consisting of a window-based sampling, wherein one or more windows are preselected from each time series snippet. The number of windows is chosen with the purpose of making the amount of data for each behavior approximately equal. As explained below, by performing this operation in an online way, integrated together with training, the usual issues of data duplication and data loss could be solved. Thus, by combining surrogates with sampling, Chapter 5 represents the culmination of the journey.

As explained in the conclusion, although this work is about data processing technology and related algorithms, its impact at the system and application level is far-reaching. The results given unlock an application that would perhaps otherwise not be feasible based on the amount of data realistically available. Similarly to the situation for image recognition, this allows a small CNN to reach otherwise unimaginable levels of accuracy without more demanding signal decomposition, enabling Edge AI-based low-power, small-scale inference. In the next chapter, the specific problem of cattle behavior classification and the associated challenges are described in more detail.

1.5 Overview of the thesis

The organization of the thesis is shown in **Fig. 1.4**, and consists of the following chapters.

Chapter 1 "*Introduction: Edge AI, time series and real world challenges*" presents the general situation and need for data augmentation as a means of addressing the issues of small dataset size and dataset imbalance. It introduces the idea that a large amount of new synthetic data can be obtained starting from a limited number of measurements together with appropriate hypotheses regarding which data features should be retained unchanged, and which ones can be changed, possibly randomly, for generating additional data.

Chapter 2 "*From livestock management to neural networks: the need for data augmentation*" summarizes in deeply detail the requirements for automated animal behavior classification, such as feeding, resting and walking. The issues of small dataset size and imbalance are considered in terms of their impact on the available classifiers, followed by an overview of the state of the art in time series data augmentation.

Chapter 3 "*Random rotation-based data augmentation*" presents a first data augmentation method, which is based on the specific characteristics of the monitored system. Namely, a paradigm is introduced in which the sensor rotation around the cow's neck is simulated, and is used as a means of generating new data for augmentation. The results show that from a baseline starting at 77%, an overall accuracy of 98% can be obtained on a large dataset from 2 cows, which was heavily imbalanced over 5 behaviors under consideration.

Chapter 4 "*Data Augmentation based on combining multiple empirical methods*" considers a more realistic scenario wherein the amount of data was approximately 10 times smaller but gathered from 6 cows. In order to maintain a high level of accuracy, several augmentation techniques are deployed. These are based on some assumptions specific to

the system under consideration, i.e., cow behavior; in addition to sensor rotation, they include reversing time series, recombining time series between segments, and performing other time-domain operations. On this basis, the overall accuracy of 92% was obtained from an 83% starting point.

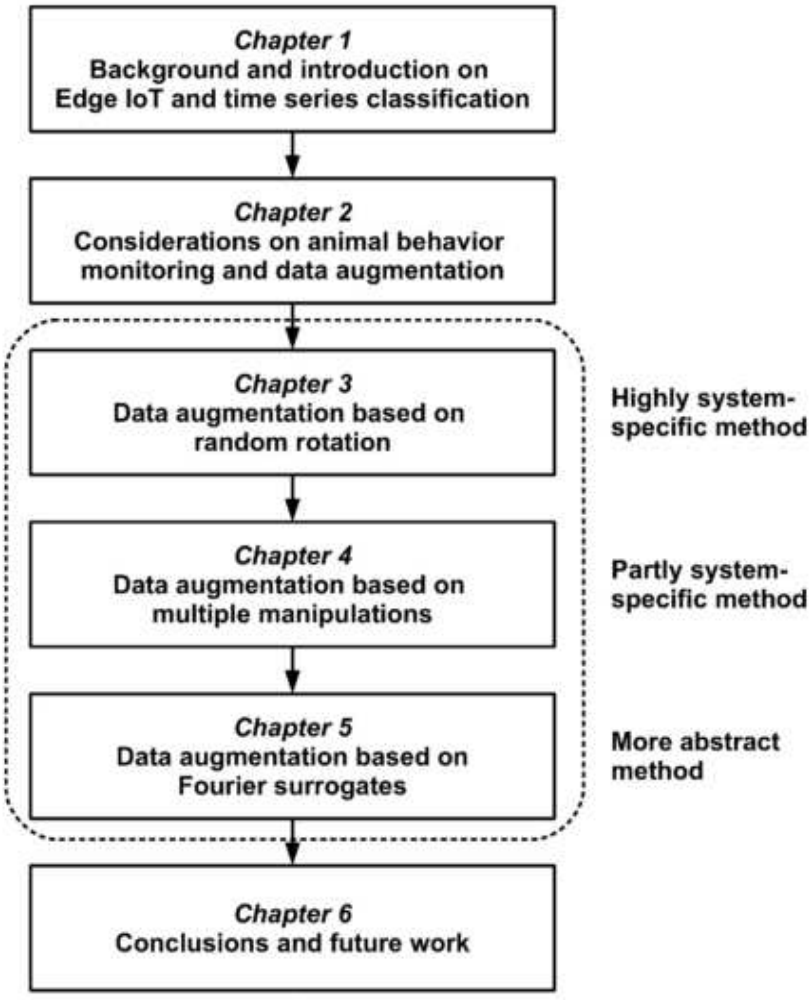


Figure 1.4: Organization of this thesis.

Chapter 5 "*Improving abstraction by combining Fourier surrogates and sampling schemes*" develops and applies a more principled approach. Starting from the observation

that several of the previous augmentation approaches retain only linear features of the signal, an implementation using a Fourier surrogates-based method is proposed. The method is combined with a sampling technique that is well suited to address data imbalance, while the surrogates avoid data duplication. Unlike the methods considered in the previous chapters, this approach is essentially completely independent of the system under consideration and, therefore, could in principle be considered also for other applications. The higher accuracy is 96%, and even more important is the fact that the most problematic behavior shows improved accuracy. Since the method is based on general signal theory, it is possible to study specific aspects of the time series, leading to successful data augmentation. Through a series of data reduction steps, I found that autocorrelation, mean, and variance convey most of the information about cow behavior. This approach can easily be applied to other systems. This is confirmed by analyzing three additional datasets. Using them, it is always found that combining sampling and surrogates gives an improvement in classification performance. With a better understanding of the correlated signal features, it becomes possible to design more specific hardware to extract and analyze these features.

Chapter 6, "*Conclusion and Future Work,*" summarizes these findings and describes the merits of the proposed method and the avenues for future investigation to enhance the impact of the technology, especially, as regards future extension to other network types and other applications. In short, this study demonstrates a coherent journey through data augmentation methods. It starts with a simple implementation based on system-specific features and ends with a more abstract approach that can be widely used and helps to identify relevant signal features. As for the specific applications under consideration, the results obtained confidently achieve automated livestock monitoring with high accuracy levels.

1.6 Bibliography

[1] J. H. Nord, A. Koohang, and J. Paliszkiewicz, “The internet of things: Review and theoretical framework,” Expert Systems with Applications, vol. 133, pp. 97–108, 2019.

[2] S. S. Goel, A. Goel, M. Kumar, and G. Molto, “A review of internet of things: qualifying technologies and boundless horizon,” Journal of Reliable Intelligent Environments, vol. 7, no. 1, pp. 23–33, 2021.

[3] J. Chin, V. Callaghan, and S. B. Allouch, “The internet-of-things: Reflections on the past, present and future from a user-centered and smart environment perspective,” Journal of Ambient Intelligence and Smart Environments, vol. 11, no. 1, pp. 45–69, 2019.

[4] Y. Lin et al., "Artificial Intelligence of Things Wearable System for Cardiac Disease Detection," 2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS), 2019, pp. 67-70.

[5] W. C. -C. Chu, C. Shih, W. -Y. Chou, S. I. Ahamed and P. -A. Hsiung, "Artificial Intelligence of Things in Sports Science: Weight Training as an Example," in Computer, vol. 52, no. 11, pp. 52-61, Nov. 2019.

[6] H. Samih, “Smart cities and internet of things,” Journal of InformationTechnology Case and Application Research, vol. 21, no. 1, pp. 3–12, 2019.

[7] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, “The industrial internet of things (iiot): An analysis framework,” Computers in industry, vol. 101, pp. 1-12,2018.

[8] R. Gómez -Chabla, K. Real-Avilés, C. Morán, P. Grijalva, and T. Recalde, “Iot applications in agriculture: A systematic literature review,” in 2nd International conference on ICTs in agronomy and environment. Springer, 2019, pp. 68–76.

[9] F. N. Tubiello, M. Salvatore, R. D. C´ondor Golec, A. Ferrara, S. Rossi, R. Biancalani, S. Federici, H. Jacobs, and A. Flammini, "Agriculture, forestry and other land use emissions by sources and removals by sinks," Rome, Italy, 2014.

[10] V. K. Quy, N. V. Hau, D. V. Anh, N. M. Quy, N. T. Ban, S. Lanza, G. Randazzo, and A. Muzirafuti, "Iot-enabled smart agriculture: Architecture, applications, and challenges," Applied Sciences, vol. 12, no. 7, p. 3396, 2022.

[11] J. Xu, B. Gu, and G. Tian, "Review of agricultural iot technology," Artificial Intelligence in Agriculture, 2022.

[12] S. Ratnaparkhi, S. Khan, C. Arya, S. Khapre, P. Singh, M. Diwakar, and A. Shankar, "Smart agriculture sensors in iot: A review," Materials Today: Proceedings, 2020.

[13] S. Greengard, The internet of things. MIT press, 2021.

[14] S. Khare and M. Totaro, "Big Data in IoT," 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2019, pp. 1-7.

[15] E. Bertino and S. Banerjee, "Artificial intelligence at the edge," arXiv preprint arXiv:2012.05410, 2020.

[16] W. Su, L. Li, F. Liu, M. He, and X. Liang, "Ai on the edge: a comprehensive review," Artificial Intelligence Review, pp. 1–59, 2022.

[17] T. Sipola, J. Alatalo, T. Kokkonen and M. Rantonen, "Artificial Intelligence in the IoT Era: A Review of Edge AI Hardware and Software," 2022 31st Conference of Open Innovations Association (FRUCT), 2022, pp. 320-331.

[18] C. P. Filho, E. Marques Jr, V. Chang, L. Dos Santos, F. Bernardini, P. F. Pires, L. Ochi, and F. C. Delicato, "A systematic literature review on distributed machine learning in edge computing," Sensors, vol. 22, no. 7, p. 2665, 2022.

[19] L. Pioli, C. F. Dorneles, D. D. de Macedo, and M. A. Dantas, “An overview of data reduction solutions at the edge of iot systems: a systematic mapping of the literature,” Computing, pp. 1–23, 2022.

[20] J. D. A. Correa, A. S. R. Pinto, and C. Montez, “Lossy data compression for iot sensors: A review,” Internet of Things, p. 100516, 2022.

[21] M. Merenda, C. Porcaro, and D. Iero, “Edge machine learning for AI-enabled IoT devices: A review,” Sensors, vol. 20, no. 9, p. 2533, 2020.

[22] I. Singh, D. Centea, and M. Elbestawi, “Iot, iiot and cyber-physical systems integration in the sept learning factory,” Procedia manufacturing, vol. 31, pp. 116–122, 2019.

[23] A. A. Cook, G. Mısırlı and Z. Fan, "Anomaly Detection for IoT Time series Data: A Survey," in IEEE Internet of Things Journal, vol. 7, no. 7, pp. 6481-6494, July 2020.

[24] F. Shi, L. Yan, X. Zhao, and R. Xian-Ke, “Machine learning-based time series data analysis in edge-cloud-assisted oil industrial iot system,” Mobile Information Systems, vol. 2022, 2022.

[25] J. Chen and X. Ran, “Deep learning with edge computing: A review,” Proceedings of the IEEE, vol. 107, no. 8, pp. 1655–1674, 2019.

[26] J. D. Hamilton, Time series analysis. Princeton university press, 2020.

[27] E. Oyekanlu, "Predictive edge computing for time series of industrial IoT and large scale critical infrastructure based on open-source software analytic of big data," 2017 IEEE International Conference on Big Data (Big Data), 2017, pp. 1663-1669.

[28] L. Sadouk, “Cnn approaches for time series classification,” Time Series Analysis-Data, Methods, and Applications, pp. 1–23, 2018.

[29] B. Zhao, H. Lu, S. Chen, J. Liu, and D. Wu, “Convolutional neural networks for time series classification,” Journal of Systems Engineering and Electronics, vol. 28, no. 1, pp. 162–169, 2017.

[30] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," Data mining and knowledge discovery, vol. 33, no. 4, pp. 917–963, 2019.

[31] J. Faouzi, "Time series classification: A review of algorithms and implementations,"Machine Learning (Emerging Trends and Applications), 2022.

[32] G. A. Susto, A. Cenedese, and M. Terzi, "Time series classification methods: Review and applications to power systems data," Big data application in power systems, pp. 179–220, 2018.

[33] Z. Omary and F. Mtenzi, "Machine learning approach to identifying the dataset threshold for the performance estimators in supervised learning," International Journal for Infonomics (IJI), vol. 3, no. 3, pp. 314–325, 2010.

[34] M. Romero, Y. Interian, T. Solberg, and G. Valdes, "Training deep learning models with small datasets," 2019.

[35] Y. Roh, G. Heo and S. E. Whang, "A Survey on Data Collection for Machine Learning: A Big Data - AI Integration Perspective," in IEEE Transactions on Knowledge and Data Engineering, vol. 33, no. 4, pp. 1328-1347, 1 April 2021.

[36] S. Whang and J.-G. Lee, "Data collection and quality challenges for deep learning," 2020.

[37] Q. Wang, A. Farahat, C. Gupta, and S. Zheng, "Deep time series models for scarce data," Neurocomputing, vol. 456, pp. 504–518, 2021.

[38] Y. Roh, G. Heo, and S. E. Whang, "A survey on data collection for machine learning: a big data-ai integration perspective," IEEE Transactions on Knowledge and Data Engineering, vol. 33, no. 4, pp. 1328–1347, 2019.

[39] R. Cowie, C. Cox, J.-C. Martin, A. Batliner, D. Heylen, and K. Karpouzis, "Issues in data labelling," in Emotion-oriented systems. Springer, 2011, pp. 213–241.

[40] T. Fredriksson, D. I. Mattos, J. Bosch, and H. H. Olsson, "Data labeling: an empirical investigation into industrial challenges and mitigation strategies," in Proceeings of International Conference on Product-Focused Software Process Improvement. Springer, 2020, pp. 202−216.

[41] T. Lyons and I. P. Arribas, "Labelling as an unsupervised learning problem," arXiv preprint arXiv:1805.03911, 2018.

[42] H. Kaur, H. S. Pannu, and A. K. Malhi, "A systematic review on imbalanced data challenges in machine learning: Applications and solutions," ACM Computing Surveys (CSUR), vol. 52, no. 4, pp. 1–36, 2019.

[43] Krawczyk, "Learning from imbalanced data: open challenges and future directions," Progress in Artificial Intelligence, vol. 5, no. 4, pp. 221–232, 2016.

[44] H. He and E. A. Garcia, "Learning from imbalanced data," IEEE Transactions on knowledge and data engineering, vol. 21, no. 9, pp. 1263–1284, 2009.

[45] N. Moniz, P. Branco, and L. Torgo, "Resampling strategies for imbalanced time series," in Proceeings of IEEE International Conference on Data Science and Advanced Analytics (DSAA), 2016, pp. 282–291.

[46] J. Schmidhuber, "Deep learning in neural networks: An overview," Neural networks, vol. 61, pp. 85–117, 2015.

[47] A. Krizhevsky, "Learning multiple layers of features from tiny images,"2009.

[48] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of IEEE conference on computer vision and pattern recognition, pp.770–778, 2016.

[49] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[50] J. Zhang et al., "Data augmentation and dense-lstm for human activity recognition using wifi signal," IEEE Internet of Things Journal, vol. 8, no. 6, pp.4628– 4641, 2020.

[51] B. K. Iwana and S. Uchida, "An empirical survey of data augmentation for time series classification with neural networks," PloS One, vol. 16, no. 7, p. e0254841, 2021.

[52] M. Kim and C. Y. Jeong, "Label-preserving data augmentation for mobile sensor data," Multidimensional Systems and Signal Processing, vol. 32, pp.115–129, 2021.

[53] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998

[54] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique", Journal of Artificial Intelligence Research, vol. 16, pp. 321-357, 2002.

[55] C. Shorten, T.M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning", J Big Data, vol. 6, no. 60, 2019.

[56] A. Le Guennec, S. Malinowski, and R. Tavenard, "Data augmentation for time series classification using convolutional neural networks," in Proc. ECML/PKDD Workshop Adv. Analytics Learn. Temporal Data, 2016.

[57] Y. Ge, X. Xu, S. Yang, Q. Zhou, and F. Shen, "Survey on sequence data augmentation", J. Front. Comput. Sci. Technol.," vol. 15, no. 7, pp. 1207-1219, 2021.

[58] L. Schmeling et al., "Training and validating a machine learning model for the sensor-based monitoring of lying behavior in dairy cows on pasture and in the barn," Animals, vol. 11, no. 9, p. 2660, 2021.

[59] J. Bartels et al., "Tinycownet: Memory-and power-minimized rnns implementable on tiny edge devices for lifelong cow behavior distribution estimation," IEEE Access, 2022.

[60] C. Li, K. K. Tokgoz, N. Saito, A. Okumura, K. Toda, H. Matsushima, T. Ohashi, K. Takeda, H. Ito, "A Data Aug-mentation Method for Cow Behavior Estimation Systems Using 3-Axis Acceleration Data and Neural Network Technology," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E105-A, no. 4, 2021.

[61] C. Li, K. K. Tokgoz, M. Fukawa, J. Bartels, T. Ohashi, K. Takeda, H. Ito, "Data Augmentation for Inertial Sensor Data in CNNs for Cattle Behavior Classification," IEEE Sensors Letters, vol. 5, no. 11, pp. 1-4, 2021.

[62] C. Li, L. Minati, K. K. Tokgoz, M. Fukawa, J. Bartels, S. A, K. Tekeda, H. Ito, "Integrated Data Augmentation for Sensor Time series in Cattle Behavior Recognition: Roles of Sampling, Balancing and Fourier Surrogates," IEEE Sensors Journal, vol. 22, no. 24, pp. 24230-24241, 2022.

Chapter 2

From livestock management to neural networks: the need for data augmentation

2.1 The need for precision livestock management

A fundamental issue in the modernization of society and human lifestyle revolves around animal welfare. From the outside, such a statement may seem surprising, given that there is a wide variation in perceptions about the importance of animal welfare across societies and geographical locations. In general, the topic remains linked to an impression of something more important for individual feeling rather than practical common good [1,2]. This viewpoint, however, is incorrect because actually the issue is that, along with greater and greater urbanization, it has become possible to enclose animals that are used as food sources (either directly, e.g., for meat, or indirectly, e.g., for milk and eggs) into environments that are completely artificial, or otherwise completely human-controlled. While this allows reducing costs and increasing productivity, it causes a deep deterioration in the living conditions of livestock. It affects not only their subjective wellbeing but also many factors

that determine biological sustainability. The latter is the key aspect, and a clear example is the emergence of the superbug crisis. This point was put forward -among others- by Dr. Bernard Rollin, an animal science professor at Colorado State University and has triggered substantial debate in both specialized circles and society at large. The bottom line is that an even more intensive agricultural development in the future is not sustainable, but every effort needs to be put to increase sustainability, particularly via better animal welfare [3,4].

It is a well-known fact in economic geography studies that livestock farming is essential for human survival, and in fact is the main component of many societies worldwide, estimated to be about half of the overall agricultural GDP. Remarkably but also worryingly, the amount of livestock products per person has doubled over the last 40 years [5]. According to recent data from the United Nations, cattle farming accounts for about 10% of the overall emissions of carbon dioxide [6]. This is, however, not the only problem, because greater and greater concentration of animals in poor health conditions also increases other environmental emissions. It leads to other problems such as the eutrophication of fresh water, and the over-nutrition of soils [7].

It is necessary to understand that the solution is improving the animal welfare. This is also because it can reduce the wastage of beef and milk, which is unavoidable when disease occurs. The situation seems all the more urgent when considering that a further doubling of the demand for livestock products is expected by 2050 [8].

One possible solution to help with this potentially dramatic situation is the usage of precision livestock farming (PLF), which represents the convergence of new technologies and a new concept. The new technologies consist of edge AI, miniaturized sensors and wireless equipment able to transmit health- and behavior-related data at long distance for extended periods of time. In this aspect, it is an IoT application. The new concept is that it is essential to consider each and every single animal as an individual. This approach is actually nothing new, as ancient farmers used to know each animal by name, and also would

be able to point out its parents and offspring. However, with ever increasing density and reliance on automatic equipment and streamlined processing, the process has been dehumanized to the point that livestock are often considered a cargo without identity, like would be the case for agricultural produce such as apple or carrots. It turns out that this is not only unethical, but also dangerous. That is since most diseases are not visible at herd level, until they have reached the level of being extremely severe. But then a dilemma arises: given that it is economically unfeasible to scale down herd size to the levels of decades ago, how to solve the problem? One cannot individually take care of so many animals. For this purpose, therefore PLF is conceived as a means to attempt doing automatically, at a large scale and in a consistent manner, what ancient farmers did. The purposes of PLF, actually, are several and diverse, and encompass detecting subtle signs of disease affecting individual animals, to charting group-level parameters such as the average weight and so on. PLF is, in fact, often integrated with other highly automated systems for feeding and milking, and the combination of data that is obtained can be very useful to quantify and optimize the performance of a farm [9-13].

PLF, therefore, indirectly promises to reduce the environmental impact of farming, because a healthy herd is a more efficient means of converting cow feed into meat and milk, and because healthy cows emit less carbon dioxide and methane. Other benefits include reducing disease, such as mastitis, which lead to widespread usage of antibiotics, resistance, and a vicious circle of ever-increasing medication. However, to date, the availability of PLF in the global farming market remains quite low [9-13]. Recent studies indicate that the present adoption is limited by factors that are usually associated with products that are not completely mature. It is said that the technical complexity of installation and usage is often excessive, and the initial investment costs are rather larger, but most of all, the accuracy is not always high enough [9,14]. The present thesis intends to provide a useful increase in behavior classification performance while decreasing as much as possible the system cost,

starting from limiting the costs associated with data collection and labeling, therefore lowering the barriers to PLF adoption.

2.2 The importance of cattle behavior monitoring

Based on the consideration that cattle farming has the biggest impact on greenhouse gas emissions from farming worldwide, the motivation is strong to focus PLF deployment towards this direction [6]. Moreover, cows are animals whose behavior is a good indicator of their health status. Due to their large size, they are easy to fit with recording instruments without causing annoyance or altering their behavior: these are big advantages that should be taken as an opportunity [15-17].

As said above, cattle behavior is both a valuable and an easily observable indicator of livestock well-being and health status. Both of them influence, in several ways, the amount and the quantity of end-products that can be acquired from a herd. Therefore, monitoring cattle behavior is highly desirable for understanding the actual situation of livestock welfare, developing effective decision support systems for managing foraging land and large farms, and so on. That would lead to generally improving the yield and quality of beef and dairy products while reducing the environmental impact. At the same time, as introduced above, it is quite impractical to monitor livestock behavior personally, especially in large herds, at all times. Identifying and keeping track of all individuals in the herd and accurately determining their health statuses, would require a very large amount of effort for direct observation, and high skill. Not only that, but this type of observation is also prone to error since it depends heavily on individual expertise. This is even more a problem when considering the large number of human operators that would be required. Automatic behavior monitoring systems, which are a subset of PLF, attempt to address these limitations by recognizing and quantifying the frequency, duration and transitions of a predetermined set of behaviors

through automatic measurements. As in the present thesis, these are often tri-axial accelerometer time series, but other data sources are also possible, including high-precision satellite location monitoring and sound recording. In any case, regardless of the specific type of data, due to the diversity of behaviors, inter-individual variability, and presence of various external noise sources, drawing inferences from these time series is not trivial. That is, in a sense, what distinguishes farming IoT from industrial IoT: animals are not machines, and they have temperaments and unique features, that need to be considered carefully.

For all the reasons above, a substantial amount of literature has been accumulating under the purpose of relating measures of individual cow behavior with the level of welfare and the general performance of farm management [15-17]. As mentioned, cows show several behaviors, including walking, resting, grazing, and rumination, some of which in several different postures. Changes in all of the overall prevalence and switching between these behaviors is highly diagnostic. For example, a reputable study from Spain has shown that deteriorating health conditions, such as mastitis, ketosis and lameness, are associated with variations in behaviors, which include changes in rumination time [18]. Changes in feeding behavior due to lameness were also found in other studies and, in general, multiple disease types are associated with reduced appetite, which seems intuitively reasonable [19].

Before considering in greater detail the methods for cattle behavior monitoring, it is necessary to clarify in more detail what is the relationship between animal welfare and behavior monitoring in particular. First of all, while the term animal welfare may sound intuitive, it should be properly defined. Because the topic is so broad, there is not one universal definition, however, a common one is according to the so-called five domains model. It says that animal welfare requires positive situations in five domains: nutrition, environment, health, behavior, and mental state. Naturally, there are important elements of biology and biochemistry in all these domains, but should be noted that actually behavior has a central importance. Behavior is one of the domains in itself, in addition, it is immediately related

to mental state and to nutrition, since feeding in itself is one of the key behaviors. According to a recent systematic review based on this model of animal welfare, the largest proportion of sensor applications for animal welfare, namely 90%, is within the 'Behavior' Domain; in addition, about two thirds of them are about monitoring location and motion [20]. This viewpoint is closely reflected in one of the fundamental papers describing the concepts and purpose of the overall research project of which the work undertaken in this thesis is part of [21]. In that paper, it is explained that animal welfare is a global trend, is closely reflected in the five freedoms, that are related to the domains indicated above, and is an important generator of value because high-welfare products have many advantages in quality and marketing. It is also explained that cow behavior and posture provide much information about welfare, in line with the idea that location and motion are essential, as explained in the previous paper cited above. One aspect on which there is emphasis is grazing, because feeding is an essential behavior, however, freely-grazing cows are different to monitor with barn-based systems due to the larger distances.

A detailed explanation of animal welfare and the use of sensors in general is the topic for animal science. However, to get a better picture of the usefulness of this work, it is useful to consider some papers in the order of publication year. Already in the year 2017, one review paper indicated that smart sensing and computing for animal welfare is a very broad and developing field. It explained that many different sensor technologies are possible, and one key seems to be the wireless technology to gather information from them and built specific networks [22]. In the same year, another review of precision livestock farming techniques put forward the importance of biosensors, such as microfluidic analyzers, together with other techniques such as sound and image analysis, movement analysis, sweat and salivary sensing, blood-based diagnosis, and others. Already in this paper, it was said that the movement and behavior of farm animals can provide information about activity and

well-being, therefore, cattle monitoring system and tracking dairy cow behavior are important [23]. One year later, another review paper indicated that sensor-based systems for precision livestock farming were proliferating, and that many sensor types were possibly suitable including movement, temperature, heart rate, sound and even electronic nose. An interesting statement was that in the farming of low-value animals like fish, poultry or pig, systems based on fixed-sensors like cameras would be commonplace, whereas for higher value animals like cattle, wearable technologies dominate the market [24]. Altogether, it seems that these early papers revealed a rich field with many ideas and possibilities, already going in the direction of wearable sensors, and considering accelerometers or other movement sensors as one possibility.

Going forward to year 2020, a definitely clearer focus on accelerometers could be seen. One specific review paper on the animal welfare assessment using accelerometers explained that accelerometers can be used for welfare assessment based on the principles of the welfare quality assessment protocol, to monitor behaviors like moving, resting, feeding and drinking in cows and pigs. This work provided big comparison tables showing that many systems are available [25]. One year later, another review paper considered in particular the applications in rangelands, coming to the conclusion that real-time tracking and accelerometer monitoring are useful to remotely detect livestock disease and grazing distribution and quality [26]. The approach and conclusion of this paper are in good agreement with the paper cited above describing the overall research project of which the work undertaken in this thesis is part of [21]. These conclusions were reiterated and expanded in a more recent paper, from last year, which concluded that precision livestock management in the rangeland can improve sustainable meat production, again with reference to the five freedoms, through continuous monitoring. In particular, ear or collar mounted accelerometers can detect behavior changes related to the quality of grazing, disease and stress [27]. In my opinion, the papers considered up to this point show a transition from a first phase, until 5-6 years

ago, where many sensor types were being actively considered, to a more recent phase, where accelerometers together with GPS tracking seem to be the predominant approach. It seems that this transition may have happened because accelerometer monitoring meets several criteria. It is usable in the rangeland, where cameras, sound and smell analyzers are much harder to implement. It is relatively inexpensive because existing commercial sensors can be used easily, and it is simple because it does not require invasive operations like blood sampling, which are not well tolerated by the animals [21, 25-27]. Then, the even more recent works that I survey next provide a clearer picture of the usefulness of these sensors and the need for data augmentation, that is the topic of this thesis.

One paper published in the year 2021 provided a systematic overview focusing specifically on externally validated and commercially available precision livestock farming technologies for sensor-based welfare assessment in cattle. Interestingly, this analysis concluded that the highest validation rate is for accelerometer-based systems, about 30%, much higher than other sensors such as cameras and chemical analyzers. It was found that some behaviors such as resting and ruminating are easier to classify than others, and said that higher accuracy and better validation are necessary to increase usefulness [28]. It seems that, by this point, the focus on accelerometers are the sensor of choice is quite clear. Another more recent paper provided additional confirmation of the importance of accelerometers attached to ears, jaws, noses, collars and legs in cattle and sheep to classify their behaviors. This paper provided some more information about the specific changes expected. The studies that it reviewed indicated that feeding is a key indicator of animal welfare, because infection with pathogens causes a very clear and visible reduction in feeding time. On the other hand, the duration of rumination is an indicator of forage quality, because longer time spent ruminating often means poor-quality grass with less nutritious content. Furthermore, reduced activity in general, that is more time resting, usually accompanies an increase in body temperature caused by inflammation to fight infection, such as mastitis [29]. These ideas are

key to the paper describing overall research project of which the work undertaken in this thesis is part of, particularly regarding the importance of precision grazing and measuring its quality in the rangeland situation [21]. The final two papers that I consider here confirm the situation and make an explicit link to the need for data augmentation, that is further described below. In a recent systematic analysis of the literature in this field, it is said that wearable accelerometer sensors are the most promising way of capturing livestock behaviour, however, there are issues with how the data is analyzed that limit usefulness. There are two problems. One is that the models are usually less accurate in classifying the less frequently observed behaviors. The other is that many models show poor generalization and overfitting, limiting real usefulness. This paper advocates using suitable preprocessing to improve performance on the behaviors being studied [30]. As I will explain below, these are exactly the issues that this thesis addresses by means of data augmentation. These issues are repeated almost exactly in another even more recent systematic review of the literature, that goes even further in saying that monitoring cattle behavior is a fundamental requirement for sustainable development and quality control in cattle farming, and collar-mounted accelerometers are the most common and a highly convenient sensor. However, class imbalance in the training datasets is a problem limiting the behavior classification accuracy [31]. Again, as I explain below, this is exactly one of the key issues addressed by this thesis.

Taken altogether, these papers show that over the recent years, precision livestock farming has moved from considering many sensor types to specifically focusing on accelerometers, in particular collar mounted accelerometers, and many independent papers agree on this. However, there remain issues with model training and data analysis which limit the accuracy and therefore the usefulness. These are the issues addressed in this thesis. Since all these papers explain that behavior monitoring is important, even essential, for animal welfare in cattle farming, it follows that improving the accuracy will positively impact animal welfare in this situation.

2.3 The existing automatic methods

As said above, automatic methods for behavior monitoring and estimation have been developed for several species of farmed animals, and commercial products exist. For instance, the gait of sheep has been comprehensively studied by means of tri-axial accelerometers, addressing the issue of automatic detection of subtle lameness [32]. On the other hand, considering the management of other farmed animals, many works have focused on behavioral modeling at herd level, for example in goats, reindeer as well as in dairy cows [33-35].

For the sake of completeness, it should be considered that, even though the present thesis focuses on sensor-based methods, another approach that has been experimented with is based on image recognition. This entails analyzing automatically the video streams coming from one or more cameras picturing the field or barn, then, tracking each specific cow. From that, one attempts to quantify select aspects of its behavior, such as the frequency of each behavior and amount of movement. At first, image-based methods seem to be convenient because they do not require attaching a sensor to each animal. However, in practice, there is a growing consensus that they are not the preferred approach for the future. This because not only image processing is computationally very demanding, for example requiring high-power graphical processing units (GPUs), but also the reliability of this approach is questionable. There seem to be three primary reasons for this. Firstly, the complex shape and coloring of each cow make its automatic identification and extraction from each video frame relatively difficult, particularly when cows may overlap each other in a video frame. This means that large classifiers requiring powerful computers have to be applied. Secondly, when the area is quite large, such as for freely grazing cows, it is quite difficult to ensure coverage by the cameras at all times. Thirdly, the behavioral identification based on videos is not a trivial operation to perform, because many things must be decided regarding what parts of the body are considered, and so on. By comparison, tri-axial acceleration recordings

seem to be considerably more informative due to the direct nature of the kinematic measurements, which are taken from a physically attached sensor [36-43]. Probably as a consequence of these issues, to date, the market in this area is prevalently dominated by systems measuring activity amount changes, based on accelerometers [44-49]. This thesis is therefore concerned with time series analyses, with a relevance primarily for sensor data. However, insofar as time series can be extracted from image data, such as by tracking movement, the present methods may also be applicable to such camera-based systems, after all.

Within this realm, representative examples of existing systems include apparatuses to classify the behaviors of sow, horse, sheep and cattle using neck, ear and jaw-mounted accelerometers. A comprehensive review is beyond the scope of this thesis, but it should be considered that such systems generally classify up to 3 behaviors (although there are exceptions), and their power consumption is practically not in line with long-term monitoring requirements [33-35,50-55]. Some possible exceptions are systems based initially on a decision tree and later on a RNN, that were proposed by our laboratory [56,57].

To date, a range of classification models has been considered, including decision trees and support vector machines, however, aiming to classify relatively small sets of behaviors [58,59]. As is known, these algorithms perform well when handling small-scale recordings and few features, however, their relatively simple nature implies that the ability to handle large variability is limited. Moreover, these basic methods seem not to scale well beyond 3 behaviors, according to the studies cited above, in line with expectations for the known performance differences between diverse classifier types [60-63].

Unlike these explicit models, convolutional neural networks represent all information implicitly under the form of large numbers of weights and they do not require features to be extracted prior to entering the data. Their performance can be outstanding but, as a consequence of the fact that the number of free parameters is much larger, their training poses heavy requirements on the amount of data. Returning to the issues presented in Chapter 1,

this requirement is difficult to meet in the case of many, or even most, Edge AI applications for reasons related to the expensive and demanding nature of the data acquisition process. The present thesis aims to alleviate this situation.

2.4 The impact of data augmentation

As written, the success of neural network-based classifiers across diverse fields rests primarily on the availability of large amounts of high-quality data [64]. Accordingly, data quantity is always associated with model robustness and generalization performance [65]. The process of training suitable classifiers, in fact, encounters some practical issues that tend to be everywhere in edge-based sensing applications of the internet of things, namely, limited dataset size and dataset imbalance [66-68]. As previously mentioned, the first arises because of the human and financial costs of data gathering. In the present case, long-term video monitoring requires dedicated personnel for footage acquisition and subsequent behavioral labeling, in addition to the practical difficulties of filming while attempting not to disturb the natural cattle behavior. The second problem, dataset imbalance, arises because a balanced prevalence across behaviors is rare. On the contrary, animals and humans tend to exhibit some activity patterns more frequently than others because their daily functioning requires this, for example, because of biological reasons. Similar situations are encountered when considering wearable systems for livestock activity monitoring [69], human behavior classification [70], gesture recognition [71], as well as room occupancy and activity detection [72].

One of the possible ways of addressing these situations is by applying data augmentation techniques, that is, performing suitable algorithmic manipulations that enlarge the amount of data entered into the training process, but without overburdening the data acquisition. Essentially, augmenting data can be considered as a preprocessing step that exploits

prior information regarding the expected invariant features of the time series concerning certain transforms, such as sensor axis rotations, rescalings, and temporal manipulations. When appropriately applied, data augmentation generates synthetic patterns that expand the classifier's decision boundaries [73], improving network generalization performance at very limited cost.

This is the point where data augmentation promises to improve the performance of automatic methods for cattle monitoring. The majority of existing systems are based on really simple classifiers, which only support a small number of behaviors, however, are relatively easy to train on small amounts of data. This considerably cuts the performance of PLF, since it is not only the quantification of the predominant behaviors but also the careful observation of the less dominant ones that can impact deeply the amount of information which may be extracted. In this sense, the usefulness or practical impact of a system can be seen as increasing drastically when raising the number of behaviors from 3 to 4 and 5 behaviors [15-19]. That, however, almost surely requires using neural networks, thus increasing the amount of data needed. Enabling this transition while sustaining high accuracy is the impact that data augmentation can have, and its pathway to social relevance is drawn as a diagram in **Fig. 2.1**. This graphics shows that the societal relevance of data augmentation is gradually increased by considering its impact on the device performance, which makes more efficient farming possible, finally improving sustainability.

To be clear, according to the discussion above, the expected impact of data augmentation is going to be realized eventually because it will make more accurate and more detailed animal behavioral monitoring possible. In turn, this will make higher the quality and potential of PLF, enabling it to fulfil, in addition to higher productivity, also and especially its societal purposes of improving animal welfare and sustainability. However, it is very difficult to quantify exactly how much an increase in classification accuracy will result in better animal welfare and sustainability. To do that, some deep quantitative understanding of the

economics and logistics would be needed, and is not easily available. Nevertheless, I think that it is possible to make the point that the impact is expected to be substantial, based on considering the most recent reviews of the current situation of PLF. On the one hand, several recent reviews of PLF point out and confirm that it has a clear potential to improve animal welfare and sustainability, therefore, this link of the chain is probably strong [74-76]. On the other hand, there are also several recent papers that criticize the current versions of PLF and point out its several flaws, showing how the existing limitations can translate actually into threats for animal welfare [77,78]. Many different issues are pointed out, and for example have to do with farming procedures, loss of farmer skill, and economic aspects (such as more pressure to do intensive farming). Of the issues that are raised, at least two of the major ones are related in some way to PLF accuracy: one is the threats deriving from equipment malfunction (which could be caused by many factors including misrecognizing some behaviors), and the other is the threats related from the lack of validation. It seems that many PLF systems today have a poor level of validation, therefore, the accuracy in the final application is not sure. In this thesis, an external validation is not conducted, however, the purpose is to substantially increase the behavior classification accuracy, therefore, this point is expected to be indirectly improved. Especially, the authors of the review in Ref. [77] explicitly state that there are many possible causes of low accuracy of classification algorithms, and these have mainly to do with the training data, such as overfitting. Because data augmentation exactly has the purpose of improving the quality of the training process, by making the most out of the data available, it seems likely to provide an important positive contribution to this problem.

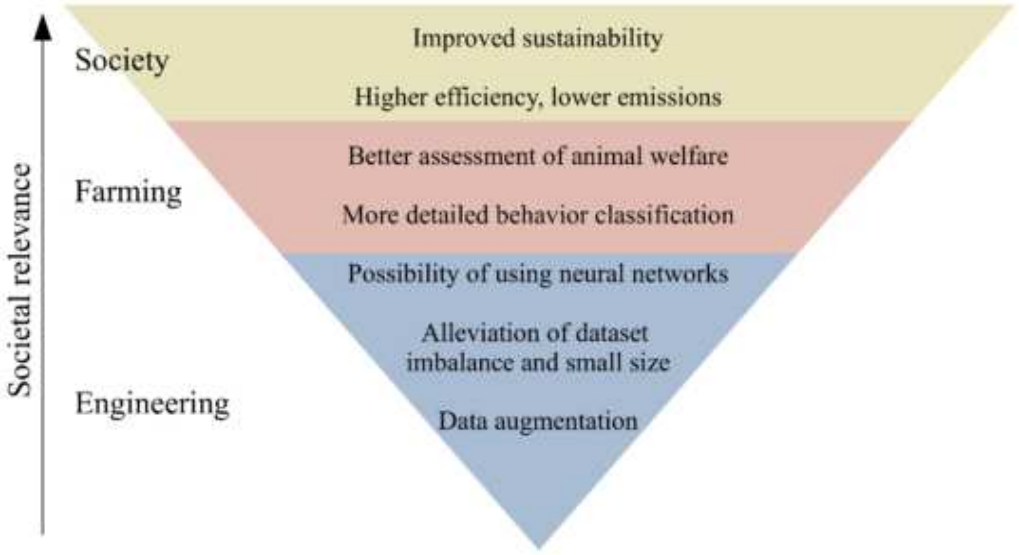


Figure 2.1: Concept diagram of the pyramid from the specific technology to the society impact.

2.5 The state of the art for time series augmentation

In the existing research on time series-based deep learning, many augmentation approaches are based on random transformations originally inspired by image data augmentation. Examples include scaling (global magnitude changes), window slicing (equivalent to image cropping), magnitude warping (modulating signal magnitude by a smooth curve), rotating (flipping for univariate cases; rotating for multivariate cases), and jittering (e.g., adding Gaussian noise). Such augmentations have been widely used for various data sources and applications on time series classification tasks. Three recent reviews, namely, Iwana et al. [79], Wen et al. [80] and Ge et al. [81], offer a comprehensive survey of this

field. For instance, Le Guennec et al. [82] proposed a method of window slicing and warping to generate new samples, by randomly slicing the original time series and speeding up/slowing down the extracted small-size slices. Um et al. [83] applied a variety of data augmentation methods, including permutating, cropping, rotating, scaling, jittering, time-warping, and magnitude-warping methods to wearable sensor data, with focusing on the application of CNNs to Parkinson's disease monitoring. Frequency warping is also a prominent approach for augmenting time series data, although it is mostly used in audio and speech recognition and similar.

The method proposed in this work attempts, particularly in the final formulation, to be more principled. As explained in Chapter 5, it involves extracting a fixed-length snippet from each segment of the time series, wherein a segment is defined as the time interval between two behavior transitions. Clearly, as such transitions are irregular, some segments are longer than others, however, a fixed snippet is extracted in each case, starting from a random time. Within the framework of Iwana et al [79], this resembles an implementation of the slicing method. According to Wen et al. [80], this method, while not explicitly mentioned, would be an instance of cropping. However, it should be underlined that neither cropping nor slicing, in themselves, provide the homogeneous snippet length that I have implemented.

Another key aspect of our proposed sampling is that, as clarified below, it is performed online, that is, fully integrated with the training process. While online data augmentation is commonplace in computer vision applications, it remains almost unexplored in the context of time series analysis, as confirmed by three recent reviews [79-81]. In agreement with the results reported below, the present work shows that even a quite simple method such as segment sampling can have a significant impact on performance when implemented in an online form, that is, during training rather than run only once beforehand.

To be complete, it should also be mentioned that an alternate approach to random transformations is pattern mixing, which combines multiple samples of intraclass data to generate new ones. An application example is the one put forward by Takahashi et al. [84], wherein sound recording are summed together. A problem with this approach is that out-of-phase overlap can occur, and nonperiodic time series data may lead to malformed patterns. Therefore, it is not suited well at all for situations such as animal behavior recognition.

As regards the issue of dataset imbalance, broadly put, three approaches are possible: undersampling, that is, artificially reducing the prevalence of the most frequent behaviors, oversampling, that is, increasing the prevalence of the least frequent behaviors either by repetitive presentation or by interpolation, and the generation of new data based on some rules. Well-known reviews of the prevalent approaches can be found in Kaur et al. [85], Patel et al. [86] and Tanha et al. [87], with additional considerations about the impact on the learning process given by Krawczyk [88] and He et al. [89].

In other words, an integrated approach to dataset imbalance mitigation appears to attenuate the drawback of learning the boundaries between differently-represented classes. It ensures that eventually, the majority of the available input data variance is still made available to the training process. It appears noteworthy that, even in the specific survey on resampling provided by Moniz et al. [90], these aspects are not considered. This seems to mean that the focus remains on precalculated dataset adjustments, and that the beneficial impact of integrated preprocessing remains largely to be clarified. An exception is the work of Cao et al. [91], which, however, is different from the present one: it relies on oversampling by interpolation between neighbors in the feature space instead of sampling in the time domain as in the present case.

2.6 Bibliography

[1] D. M. Broom, "Animal welfare: an aspect of care, sustainability, and food quality required by the public," Journal of veterinary medical education, vol. 37, no. 1, pp. 83–88, 2010.

[2] J. J. McGlone, "Farm animal welfare in the context of other society issues: toward sustainable systems," Livestock production science, vol. 72, no. 1-2, pp. 75–81, 2001.

[3] B. E. Rollin, "Toxicology and new social ethics for animals," Toxicologic pathology, vol. 31, no. 1 suppl, pp. 128–131, 2003.

[4] B. E. Rollin, "Animal rights as a mainstream phenomenon," Animals, vol. 1, no. 1, pp. 102–115, 2011.

[5] T. Raney et al., ''The state of food and agriculture: Livestock in the balance,'' Food and Agriculture Organization of the United Nations: Rome, Italy, 2009.

[6] P. J. Gerber, H. Steinfeld, B. Henderson, A. Mottet, C. Opio, J. Dijkman, A. Falcucci, and G. Tempio, Tackling Climate Change Through Livestock: A Global Assessment of Emissions and Mitigation Opportunities. Food and Agriculture Organization of the United Nations (FAO), 2013.

[7] J. Martinez, P. Dabert, S. Barrington, and C. Burton, ''Livestock waste treatment systems for environmental quality, food safety, and sustainability,'' Bioresource Technol., vol. 100, no. 22, pp. 5527–5536, Nov. 2009.

[8] M. M. Rojas-Downing, A. P. Nejadhashemi, T. Harrigan, and S. A. Woznicki, ''Climate change and livestock: Impacts, adaptation, and mitigation,'' Climate Risk Manage., vol. 16, pp. 145–163, Jan. 2017.

[9] T. M. Banhazi, H. Lehr, J. Black, H. Crabtree, P. Schofield, M. Tscharke, and D. Berckmans, "Precision livestock farming: an international review of scientific and

commercial aspects," International Journal of Agricultural and Biological Engineering, vol. 5, no. 3, pp. 1–9, 2012.

[10] D. Berckmans, "General introduction to precision livestock farming," Animal Frontiers, vol. 7, no. 1, pp. 6–11, 2017.

[11] A. Monteiro, S. Santos, and P. Gonc ̧alves, "Precision agriculture for crop and livestock farming—brief review," Animals, vol. 11, no. 8, p. 2345, 2021.

[12] N. Hostiou, J. Fagon, S. Chauvat, A. Turlot, F. Kling-Eveillard, X. Boivin, and C. Allain, "Impact of precision livestock farming on work and human-animal interactions on dairy farms. a review," Biotechnologie, Agronomie, Soci ́et ́e et Environnement/Biotechnology, Agronomy, Society and Environment, vol. 21, no. 4, pp. 268–275, 2017.

[13] T. Mottram, "Animal board invited review: precision livestock farming for dairy cows with a focus on oestrus detection," Animal, vol. 10, no. 10, pp. 1575–1584, 2016.

[14] M. O. Vaintrub, H. Levit, M. Chincarini, I. Fusaro, M. Giammarco, and G. Vignola, "Precision livestock farming, automats and new technologies: possible applications in extensive dairy sheep farming," Animal, vol. 15, no. 3, p. 100143, 2021.

[15] I. Dittrich, M. Gertz, and J. Krieter, "Alterations in sick dairy cows' daily behavioural patterns," Heliyon, vol. 5, no. 11, p. e02902, 2019.

[16] R. Lardy, M. M. Mialon, N. Wagner, Y. Gaudron, B. Meunier, K. H. Sloth, D. Ledoux, M. Silberberg, A. d. B. des Roches, Q. Ruin et al., "Understanding anomalies in animal behaviour: data on cow activity in relation to health and welfare," Animal-Open Space, vol. 1, no. 1, p.100004, 2022.

[17] M. B. Jensen and K. L. Proudfoot, "Effect of group size and health status on behavior and feed intake of multiparous dairy cows in early lactation," Journal of dairy science, vol. 100, no. 12, pp. 9759–9768, 2017.

[18] P. Llonch, E. Mainau, I. R. Ipharraguerre, F. Bargo, G. Tedó, M. Blanch, and X. Manteca, "Chicken or the Egg: The reciprocal association between feeding behavior and animal welfare and their impact on productivity in dairy cows," Frontiers in Veterinary Science, vol. 5, no. DEC, 2018.

[19] S. Hassall, W. Ward, and R. Murray, "Effects of lameness on the behaviour of cows during the summer." The Veterinary record, vol. 132, no. 23, pp. 578–580, 1993.

[20] E. Fogarty, D. Swain, G. Cronin, and M. Trotter, "A systematic review of the potential uses of on-animal sensors to monitor the welfare of sheep evaluated using the five domains model as a framework," Animal Welfare, vol. 28, no. 4, pp. 407–420, 2019.

[21] H. Ito, K. K. Tokgoz, T. Ohashi, and K.-i. Takeda, "Monitoring Technology Development for Ultra-precision Grazing," The Journal of the Institute of Electronics, Information and Communication Engineers, vol. 104, no. 6, pp. 544–551, 2021.

[22] A. Jukan, X. Masip-Bruin, and N. Amla, "Smart computing and sensing technologies for animal welfare: A systematic review," ACM Computing Surveys (CSUR), vol. 50, no. 1, pp. 1–27, 2017.

[23] S. Neethirajan, "Recent advances in wearable sensors for animal health management," Sensing and Bio-Sensing Research, vol. 12, pp. 15–29, 2017.

[24] I. Halachmi, M. Guarino, J. Bewley, and M. Pastell, "Smart animal agriculture: application of real-time sensors to improve animal well-being and production," Annual review of animal biosciences, vol. 7, pp. 403–425, 2019.

[25] J. M. Chapa, K. Maschat, M. Iwersen, J. Baumgartner, and M. Drillich, "Accelerometer systems as tools for health and welfare assessment in cattle and pigs–a review," Behavioural Processes, vol. 181, p. 104262, 2020.

[26] D. W. Bailey, M. G. Trotter, C. Tobin, and M. G. Thomas, "Opportunities to apply precision livestock management on rangelands," Frontiers in Sustainable Food Systems, vol. 5, p. 611915, 2021.

[27] C. T. Tobin, D. W. Bailey, M. B. Stephenson, M. G. Trotter, C. W. Knight, and A. M. Faist, "Opportunities to monitor animal welfare using the five freedoms with precision livestock management on rangelands," Frontiers in Animal Science, p. 93, 2022.

[28] A. H. Stygar, Y. G´omez, G. V. Berteselli, E. Dalla Costa, E. Canali, J. K. Niemi, P. Llonch, and M. Pastell, "A systematic review on commercially available and validated sensor technologies for welfare assessment of dairy cattle," Frontiers in veterinary science, vol. 8, p. 634338, 2021.

[29] B. Fan, R. Bryant, and A. Greer, "Behavioral fingerprinting: Acceleration sensors for identifying changes in livestock health," J, vol. 5, no. 4,pp. 435–454, 2022.

[30] L. Riaboff, L. Shalloo, A. F. Smeaton, S. Couvreur, A. Madouasse, and M. T. Keane, "Predicting livestock behaviour using accelerometers: A systematic review of processing techniques for ruminant behaviour prediction from raw accelerometer data," Computers and Electronics in Agriculture, vol. 192, p. 106610, 2022.

[31] A. da Silva Santos, V. W. C. de Medeiros, and G. E. Gonc¸alves, "Monitoring and classification of cattle behavior: A survey," Smart Agricultural Technology, p. 100091, 2022.

[32] J. Barwick, D. Lamb, R. Dobos, D. Schneider, M. Welch, and M. Trotter, "Predicting lameness in sheep activity using tri-axial acceleration signals," Animals, vol. 8, no. 1, p. 12, 2018.

[33] M. Moreau, S. Siebert, A. Buerkert, and E. Schlecht, "Use of a tri-axial accelerometer for automated recording and classification of goats' grazing behaviour," Appl. Anim. Behav. Sci., vol. 119, no. 3–4, pp. 158–170, Jul. 2009.

[34] G. Body, R. B. Weladji, and Ø. Holand, "The recursive model as a new approach to validate and monitor activity sensors," Behav. Ecol. Sociobiol., vol. 66, no. 11, pp. 1531–1541, 2012.

[35] D. N. Ledgerwood, C. Winckler, and C. B. Tucker, "Evaluation of data loggers, sampling intervals, and editing techniques for measuring the lying behavior of dairy cattle," J. Dairy Sci., vol. 93, no. 11, pp. 5129–5139, Nov. 2010.

[36] B. Shao and H. Xin, "A real-time computer vision assessment and control of thermal comfort for group-housed pigs," Comput. Electron. Agric., vol. 62, no. 1, pp. 15–21, 2008.

[37] P. Ahrendt, T. Gregersen, and H. Karstoft, "Development of a real-time computer vision system for tracking loose-housed pigs," Comput. Electron. Agric., vol. 76, no. 2, pp. 169–174, 2011.

[38] H. H. Kristensen and C. Cornou, "Automatic detection of deviations in activity levels in groups of broiler chickens–a pilot study," Biosyst. Eng., vol. 109, no. 4, pp. 369–376, 2011.

[39] X. P. Burgos-Artizzu, P. Dollár, D. Lin, D. J. Anderson, and P. Perona, "Social behavior recognition in continuous video," in 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 1322–1329.

[40] S. M. C. Porto, C. Arcidiacono, U. Anguzza, and G. Cascone, "A computer vision-based system for the automatic detection of lying behaviour of dairy cows in free-stall barns," Biosyst. Eng., vol. 115, no. 2, pp. 184–194, 2013.

[41] Z. Wang, S. A. Mirbozorgi, and M. Ghovanloo, "Towards a Kinect-based behavior recognition and analysis system for small animals," in 2015 IEEE Biomedical Circuits and Systems Conference (BioCAS), 2015, pp. 1–4.

[42] F. Lao, T. Brown-Brandl, J. P. Stinn, K. Liu, G. Teng, and H. Xin, "Automatic recognition of lactating sow behaviors through depth image processing," Comput. Electron. Agric., vol. 125, pp. 56–62, 2016.

[43] S. Kumar and S. K. Singh, "Cattle Recognition: A New Frontier in Visual Animal Biometrics Research," Proc. Natl. Acad. Sci. India Sect. A Phys. Sci., pp. 1–20, 2019.

[44] M. Schwager, D. M. Anderson, Z. Butler, and D. Rus, "Robust classification of animal tracking data," Comput. Electron. Agric., vol. 56, no. 1, pp. 46–59, 2007.

[45] J. P. Bikker et al., "Evaluation of an ear-attached movement sensor to record cow feeding behavior and activity," J. Dairy Sci., vol. 97, no. 5, pp. 2974–2979, 2014.

[46] E. S. Nadimi, H. T. Søgaard, and T. Bak, "ZigBee-based wireless sensor networks for classifying the behaviour of a herd of animals using classification trees," Biosyst. Eng., vol. 100, no. 2, pp. 167–176, 2008.

[47] S. Kuankid, T. Rattanawong, and A. Aurasopon, "Classification of the cattle's behaviors by using accelerometer data with simple behavioral technique," in Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific, 2014, pp. 1–4.

[48] T. M. Hill et al., "Evaluation of an ear-attached movement sensor to record rumination, eating, and activity behaviors in 1-month-old calves," Prof. Anim. Sci., vol. 33, no. 6, pp. 743–747, 2017.

[49] J. Barwick, D. W. Lamb, R. Dobos, M. Welch, and M. Trotter, "Categorising sheep activity using a tri-axial accelerometer," Comput. Electron. Agric., vol. 145, pp. 289–297, 2018.

[50] V. M. Suresh, R. Sidhu, P. Karkare, A. Patil, Z. Lei, and A. Basu, ''Powering the IoT through embedded machine learning and Lora,'' in Proc. IEEE 4th World Forum Internet Things (WF-IoT), Feb. 2018, pp. 349–354.

[51] G. F. Marchioro, C. Cornou, A. R. Kristensen, and J. Madsen, "Sows' activity classification device using acceleration data—A resource constrained approach,'' Comput. Electron. Agricult., vol. 77, no. 1, pp. 110–117, Jun. 2011..

[52] D. Gutierrez-Galan, J. P. Dominguez-Morales, E. Cerezuela-Escudero, A. Rios-Navarro, R. Tapiador-Morales, M. Rivas-Perez, M. Dominguez-Morales, A. Jimenez-Fernandez, and A. Linares-Barranco, ''Embedded neural network for real-

time animal behavior classification,'' Neurocomputing, vol. 272, pp. 17–26, Jan. 2018.

[53] J. P. Dominguez-Morales, L. Duran-Lopez, D. Gutierrez-Galan, A. Rios-Navarro, A. Linares-Barranco, and A. Jimenez-Fernandez, "Wildlife monitoring on the edge: A performance evaluation of embedded neural networks on microcontrollers for animal behavior classification,'' Sensors, vol. 21, no. 9, p. 2975, Apr. 2021.

[54] S. P. le Roux, R. Wolhuter, and T. Niesler, ''Energy-aware feature and model selection for onboard behavior classification in low-power animal borne sensor applications,'' IEEE Sensors J., vol. 19, no. 7, pp. 2722–2734, Apr. 2019.

[55] R. Arablouei, L. Currie, B. Kusy, A. Ingham, P. L. Greenwood, and G. Bishop-Hurley, ''In-situ classification of cattle behavior using accelerometry data,'' Comput. Electron. Agricult., vol. 183, Apr. 2021, Art. no. 106045.

[56] J. Bartels, K. K. Tokgoz, M. Fukawa, S. Otsubo, L. Chao, I. Rachi, K. Takeda, and H. Ito, ''A 216 μW, 87% accurate cow behavior classifying decision tree on FPGA with interpolated Arctan2,'' in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), May 2021, pp. 1–5.

[57] J. Bartels et al., "Tinycownet: Memory-and power-minimized rnns implementable on tiny edge devices for lifelong cow behavior distribution estimation," IEEE Access, 2022.

[58] B. Charbuty and A. Abdulazeez, "Classification based on decision tree algorithm for machine learning," Journal of Applied Science and Technology Trends, vol. 2, no. 01, pp. 20-28, 2021.

[59] G. Mountrakis, J. Im, and C. Ogole, "Support vector machines in remote sensing: A review," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 66, no. 3, pp. 247-259, 2011.

[60] S. E. Jozdani, B. A. Johnson, and D. Chen, "Comparing deep neural networks, ensemble classifiers, and support vector machine algorithms for object-based urban land use/land cover classification," Remote Sensing, vol. 11, no. 14, p. 1713, 2019.

[61] E. Raczko and B. Zagajewski, "Comparison of support vector machine, random forest and neural network classifiers for tree species classification on airborne hyperspectral apex images," European Journal of Remote Sensing, vol. 50, no. 1, pp. 144–154, 2017

[62] S. Sakr, R. Elshawi, A. M. Ahmed, W. T. Qureshi, C. A. Brawner, S. J. Keteyian, M. J. Blaha, and M. H. Al-Mallah, "Comparison of machine learning techniques to predict all-cause mortality using fitness data: the henry ford exercise testing (fit) project," BMC medical informatics and decision making, vol. 17, no. 1, pp. 1–15, 2017.

[63] C. Hung, W. Chen, P. Lai, C. Lin and C. Lee, "Comparing deep neural network and other machine learning algorithms for stroke prediction in a large-scale population-based electronic medical claims database," 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2017, pp. 3110-3113.

[64] J. Schmidhuber, "Deep learning in neural networks: An overview," Neural networks, vol. 61, pp. 85–117, 2015.

[65] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," nature, vol. 521, no. 7553, pp. 436–444, 2015.

[66] Y. Li, H. Hu, and G. Zhou, "Using data augmentation in continuous authentication on smartphones," IEEE Internet Things J., vol. 6, no. 1, pp. 628–640, 2018.

[67] J. Zhang et al., "Data augmentation and dense-lstm for human activity recognition using wifi signal," IEEE Internet Things J., vol. 8, no. 6, pp. 4628– 4641, 2020.

[68] E. S. Jeon, A. Som, A. Shukla, K. Hasanaj, M. P. Buman, and P. Turaga, "Role of data augmentation strategies in knowledge distillation for wearable sensor data," IEEE Internet Things J., 2021.

[69] L. Schmeling et al., "Training and validating a machine learning model for the sensor-based monitoring of lying behavior in dairy cows on pasture and in the barn," Anim., vol. 11, no. 9, p. 2660, 2021.

[70] K. Chen, D. Zhang, L. Yao, B. Guo, Z. Yu, and Y. Liu, "Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities," ACM Comput. Surv., vol. 54, no. 4, pp. 1–40, 2021.

[71] Y. Song, L. Morency, and R. Davis, "Distribution-sensitive learning for imbalanced datasets," in Proc. 10th IEEE Int. Conf. Workshops Autom. Face Gesture Recognit., pp. 1–6, 2013.

[72] N. S. Sani, I. I. S. Shamsuddin, S. Sahran, A. Rahman, and E. N. Muzaffar, "Redefining selection of features and classification algorithms for room occupancy detection," Int. J. Adv. Sci., Eng. Inf. Technol., vol. 8, no. 4-2, pp. 1486–1493, 2018.

[73] B. K. Iwana and S. Uchida, "An empirical survey of data augmentation for time series classification with neural networks," PloS One, vol. 16, no. 7, p. e0254841, 2021.

[74] Y. Gómez, A. H. Stygar, I. J. Boumans, E. A. Bokkers, L. J. Pedersen, J. K. Niemi, M. Pastell, X. Manteca, and P. Llonch, "A systematic review on validated precision livestock farming technologies for pig production and its potential to assess animal welfare," Frontiers in Veterinary Science, vol. 8, p. 660565, 2021.

[75] C. Aquilani, A. Confessore, R. Bozzi, F. Sirtori, and C. Pugliese, "Precision livestock farming technologies in pasture-based livestock systems," Animal, vol. 16, no. 1, p. 100429, 2022.

[76] S. Morrone, C. Dimauro, F. Gambella, and M. G. Cappai, "Industry 4.0 and precision livestock farming (plf): An up to date overview across animal productions," Sensors, vol. 22, no. 12, p. 4319, 2022.

[77] F. A. M. Tuyttens, C. F. M. Molento, and S. Benaissa, "Twelve threats of precision livestock farming (PLF) for animal welfare," Frontiers in Veterinary Science, p. 727.

[78] J. Schillings, R. Bennett, and D. C. Rose, "Exploring the potential of precision livestock farming technologies to help address farm animal welfare," Frontiers in Animal Science, p. 13, 2021.

[79] B. K. Iwana and S. Uchida, "An empirical survey of data augmentation for time series classification with neural networks," PloS One, vol. 16, no. 7, p. e0254841, 2021.

[80] Q. Wen et al., "Time series data augmentation for deep learning: A survey," in Proc. Int. Jt. Conf. Artif. Intell., pp. 4653–4660, 2021.

[81] Y. Ge, X. Xu, S. Yang, Q. Zhou, and F. Shen, "Survey on sequence data augmentation", J. Front. Comput. Sci. Technol.," vol. 15, no. 7, pp. 1207-1219, 2021.

[82] A. Le Guennec, S. Malinowski, and R. Tavenard, "Data augmentation for time series classification using convolutional neural networks," in Proc. ECML/PKDD Workshop Adv. Analytics Learn. Temporal Data, 2016.

[83] T. T. Um et al., "Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks," in Proc. 19th ACM Int. Conf. Multimodal Interact., 2017, pp. 216–220.

[84] N. Takahashi, M. Gygli, B. Pfister, and L. van Gool, "Deep convolutional neural networks and data augmentation for acoustic event detection," in Proc. Annu. Conf. Int. Speech Commu. Assoc., 2016, pp. 1982–2986, arXiv:1604.07160.

[85] H. Kaur, H. S. Pannu, and A.K. Malhi, "A systematic review on imbalanced data challenges in machine learning: Applications and solutions," ACM Comput. Surv., vol. 52, no. 4, pp. 1-36, 2019.

[86] H. Patel, D. Singh Rajput, G. Thippa Reddy, C. Iwendi, A. Kashif Bashir, and O. Jo, "A review on classification of imbalanced data for wireless sensor networks," Int. J. Distrib. Sens. Networks, vol. 16, no. 4, p. 1550147720916404, 2020

[87] J. Tanha, Y. Abdi, N. Samadi, N. Razzaghi, and M. Asadpour, "Boosting methods for multi-class imbalanced data classification: an experimental review," J. Big Data, vol. 7, no. 1, pp. 1–47, 2020.

[88] B. Krawczyk, "Learning from imbalanced data: open challenges and future directions," Prog. Artif. Intell., vol. 5, no. 4, pp. 221–232, 2016.

[89] H. He and E. A. Garcia, "Learning from imbalanced data," IEEE Trans. Knowl. Data Eng., vol. 21, no. 9, pp. 1263–1284, 2009.

[90] N. Moniz, P. Branco, and L. Torgo, "Resampling strategies for imbalanced time series," in Proc. IEEE Int. Conf. Data Sci. Adv. Analytics, 2016, pp. 282–291..

[91] H. Cao, V. Y. F. Tan and J. Z. F. Pang, "A Parsimonious Mixture of Gaussian Trees Model for Oversampling in Imbalanced and Multimodal Time series Classification," IEEE Trans. Neural Networks Learn. Syst., vol. 25, no. 12, pp. 2226-2239, 2014.

Chapter 3

Random rotation-based data augmentation

As explained in the previous chapters, cow behavior monitoring is critical for understanding the current state of cow welfare and developing an effective planning strategy for pasture management, such as early detection of disease and estrus. It is expected that one of the most powerful and cost-effective methods that can attain high accuracy and robustness would be a neural-network-based monitoring system that analyzes time series data from inertial sensors attached to the cows. For this kind of method, however, a significant challenge is improving the quality and quantity of training data to be used in the development of neural network models. This requires one to collect data that can cover various realistic conditions and assign labels to them. As a result, the cost of data collection is significantly high. This Chapter proposes a first version of a data augmentation method aiming to solve two major quality problems encountered in the process of data collection. One is the difficulty and random aspect of data acquisition itself, which only samples a small amount of time compared to a cow's life span. The other is the sensor position changes during actual operation. The method proposed in this chapter can simulate different rotation states of the collar-type sensor device starting from one set of measured acceleration data. At the same time, it generates additional data for the behaviors that occur less frequently.

The verification results show significantly higher estimation performance reaching an average accuracy of over 98% for five main behaviors (feeding, walking, drinking, rumination, and resting) based on learning using long short-term memory (LSTM) network. Compared with the estimation performance without data augmentation, which was insufficient with a minimum of 60.48%, the F_1 score was improved by 2.5%-37.1% for various behaviors. In addition, comparison of different rotation intervals was investigated, and a 30-degree increment was selected based on the accuracy performance analysis. The proposed data augmentation method can improve the accuracy of cow behavior estimation by a neural network model. Moreover, it contributes to a significant reduction of the data collection cost for machine learning and opens many opportunities for new research.

3.1 Considerations on sensor position and collar rotation

In our application, there are two major bottlenecks related to the data, which map closely onto the overall state of the field as introduced in Chapter 1. One is the imbalanced learning problem, i.e., an imbalanced data sample size, which is mainly due to the uncontrollable randomicity of animal behavior during data acquisition. The main problem of learning from imbalanced data is that in the case of underrepresented data and severe category distribution skews, data imbalance can significantly compromise the performance of learning algorithms [1]. Hence, it may seem necessary to measure an enormous amount of time when collecting data on rare behaviors. Another bottleneck lies in practical, real-life application. The position of the sensor attached to a cow's neck may change due to its sliding, caused by body movements, posture changes and interactions with other cows or objects. When the collar rotates, the position of the recording device with respect to the body and head changes; as a consequence, the accelerometer raw data readings change and the accuracy of the algorithm may decreases considerably. This may be considered mostly as a

measurement issue: even if the underlying activity, i.e., cow behavior, remains the same, there is a rotation on a plane which is determined by being orthogonal to the neck axis. As a consequence, two components of the signal may change arbitrarily, even in their sign [2]. Such aspects of what is called invariance are a common issue in measurement situations, and arise quite frequently when considering devices that are applied to living beings, such as wearable devices [3-5]. Although, it may seem possible to fix the position of the device with a weight, relying on the collar to rotate itself so the weight points downwards, this is a very imperfect method. It does not entirely prevent rotations, it may not even be able to apply sufficient torque to rotate the collar back into alignment, and at the same time runs an actual risk of negatively affecting animal welfare because of the additional weight load.

Attempting to address the difficulties of large-scale data acquisition under various realistic conditions and the extensive man-hour cost of label assignment phases, in this chapter I propose a random rotation-based data augmentation method to overcome the two aforementioned problems, and I evaluate it using a NN. Before going through the proposed method, some other data augmentation techniques in the literature are summarized again, as follows. There are several approaches for generating additional samples. These include, for example, data warping, which implies distorting the data in time or amplitude [6], and various sampling methods for imbalanced learning, such as random oversampling and under sampling [7,8]. One concrete example is provided by Le Guennec et al. [9], who combined use window slicing with a warping technique, which extracts multiple slices and accelerates or decelerates in time some of them. Nevertheless, the data generated by scaling may not maintain the correct label in some domains, e.g., the acceleration data for cow monitoring, because some labels are differentiated by the motion intensity. As introduced in the previous chapters, this exemplifies a situation where data augmentation may have a negative effect, if one is not sure of the assumptions underlying the acceptability of a given transformation operation. On the other hand, while it is possible to expand the data size by

applying sampling techniques, little attention has been paid to real-life cases. Unlike data augmentation for images [10] and speech recognition [11], data augmentation for numerical data from inertial sensors has not been systematically investigated yet to the best of my knowledge.

This chapter aims to provide a data augmentation method for cow behavior estimation systems which accurately monitors and assesses multiple cow behaviors while retaining a low cost data collection process. The specific objectives are two-fold: 1) to extend the data of minority class samples to mitigate the difference in the amount of data for different behaviors, and 2) to solve the problem of the decline of recognition performance due to sensor position shifting caused by cow movements, which is an important factor in cow monitoring systems that previous studies have not examined. That is to say, the work reported in this chapter when it was published represented the first attempt to explicitly address the issue that the sensor axes may rotate with respect to the cow's head and body. The main novelty of this study compared to related studies is the proposed method of data augmentation, i.e., data rotation. While this operation by itself is well-known, its application to data augmentation based on a physical reason is new, and the discovery that it can be used for attenuating the class imbalance has a concrete impact. In fact, with the proposed method, imbalanced data distribution between classes can also be avoided, the cost of data acquisition and labeling phase is decreased, and the potential impact of sensor position changes on classification accuracy can be minimized, thereby greatly improving the recognition rate.

3.2 Data acquisition and processing methods

3.2.1 Data acquisition setup

The experiments for sensor data acquisition reported in this chapter were conducted over multiple sessions at the livestock farm in Shinshu University, Nagano, Japan from the end of October to the end of November 2018. For the work reported in this Chapter, the data were collected from two Japanese black cows aged 2 and 10 years. During the experimental work, both cows were healthy as confirmed by veterinary monitoring.

As introduced in Chapter 1, one of the challenges for creating a successful Edge AI device is the need for a compact, reliable device that is capable of storing a large enough amount of raw data for later processing. This is unavoidable because, prior to development of the final application code, it is not possible to compress the data by classification operations or similar, as the development of the classifier in itself requires the raw data. Here, Sony's Spresense was applied as an edge device for the purpose of data collection. Spresense development board includes the multi-core microcontroller CXD5602 (ARM® Cortex®-M4F × 6 cores), which is developed by Sony with a high power efficiency. The clock speed is 156 MHz. As a compact development board, it is also supported by the Arduino Integrated Development Environment (IDE) and the more advanced NuttX-based Software Development Kit (SDK) [12]. Developers can create their applications with these two development environments in a short time. In addition, since the Global Navigation Satellite System (GNSS) is integrated in Spresense development board, it supports Quasi-Zenith Satellite System (QZSS), Global Navigation Satellite System (GLONASS), as well as the Global Positioning System (GPS). This means Spresense also has the ability to support applications in which tracking and precise time-stamping are essential.

Each experimental cow wore a collar fitted with a sensor device that was powered by a mobile battery (5V, 3200 mAh). As illustrated in **Fig. 3.1**, the sensor device consisted of Sony Spresense main and extension boards, and a 3-axis acceleration sensor add-on board (SPRESENSE-SENSOR-EVK-701) and Bluetooth add-on board (SPRESENSE-BLE-EVK-701). Using the Bluetooth module, the sensor device was able to send the raw accelerometer data to an Android terminal or a computer. To prevent behavioral data loss when the signal could not be effectively transmitted (e.g., when the cow was not in the network coverage area), the acceleration data was stored locally in a microSD card so that the behavioral data of the experimental cows could be accessed offline. For this purpose, a specific firmware running on the Spresense board was developed as a part of the project that our lab belonged to.

As previously shown in **Fig. 1.3b**, the position and orientation of the sensor as well as the coordinate frame of the 3-axis acceleration sensor with the X-axis (longitudinal), Y-axis (vertical), and Z-axis (horizontal) were determined by the design of the board and its position in the plastic case attached to a collar. The device was mounted on the cow's neck, and could not be easily interfered with or damaged by the cow. When a cow wears a collar attached to an acceleration sensor, the movements of the cow during the monitoring process cause orientation changes in the acceleration sensor, thus leading to changes in the acceleration values. That is to say, that the accelerometer will record changes in both the static and the dynamic acceleration. The first are due to changes in the body posture, basically, the orientation of the neck with respect to the Earth's gravitational field: when there is no movement, these are visible as non-zero base line (average) in the time series. The second are directly related to the body movement, and represent the second derivative of the trajectory due to the performance of actions such as ruminating and so forth. It is important to consider that, in the sensor time series, these are intermixed and inseparable.

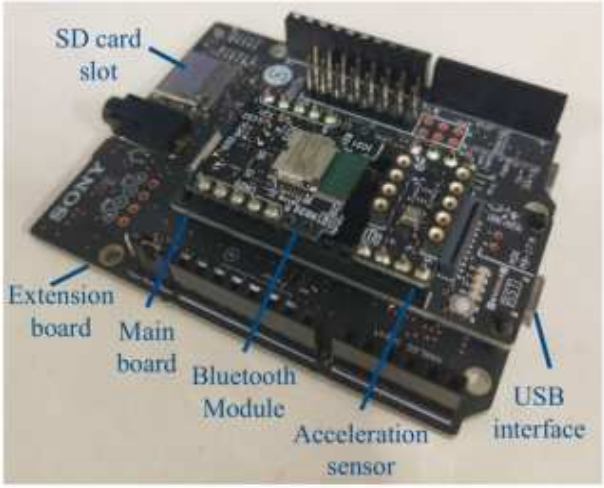


Figure 3.1: Sensor device prototype.

3.2.2 Acceleration data collection

The sensor is KX122-1037 accelerometer by ROHM Co., ltd located in Kyoto, Japan [13]. This accelerometer is built using micro-electro-mechanical (MEMS) technology and has 3-axis, with integrated amplification and analog-to-digital conversion. The used acceleration range was ±4 g. Sensors such as this one support a multitude of ranges, and this setting represented an optimal compromise. Smaller settings could have resulted in saturation, considering that the gravitational acceleration is by definition about 1 g and that dynamical activity overlaps onto it. Larger settings would have reduced the signal-to-noise ratio and have been unnecessary, also considering that cows, because of their large mass, generally do not perform quick movements.

As illustrated in **Fig. 1.3**, the position of the sensor device was initially set to the right side of the cow's neck during the monitoring process. In addition, to minimize the effects of wearing a collar-shaped device on the natural movement pattern of the cows, there was

a gap between the collar-shaped device and the skin of the monitored cow. Again, this represents an engineering compromise. A tight fit would be optimal because it enhances friction and therefore maximizes the coupling between the cow's body and the accelerometer, approximating one rigid body. On the other hand, such a situation would be unacceptable in terms of the discomfort caused to the animal and, besides the welfare implications, it would alter the behavior considerably. Therefore, the issue of collar rotation arises because of this unavoidable physical compromise. While it is outside the scope of this work and therefore will not be discussed, future research may consider the best way to improve the mechanical attachment between cow and accelerometer.

In this study, the sampling frequency of the acceleration sensor data was set to 25 Hz (25 samples per second). As for the other data acquisition settings, this does not represent an exact calculation but a reasonable compromise. In fact, the sensor would support much higher sampling rates. However, these lead to considerably high power consumption and, not less important, generate huge amounts of data that need to be processed. A key issue is that the sampling rate should be at least twice the highest frequency contained in the signal: this is well known as at Nyquist criterion [14]. With this in mind, one may consider that rhythmic movements such as ruminating and drinking are unlikely to contain signals at frequencies beyond a few Hz. At the same time, the Nyquist theorem does not consider the fact that experimental signals also contain noise, therefore, some amount of oversampling may be beneficial. I therefore considered the setting of 25 Hz as a reasonable compromise between a sufficiently fast sampling and not wasting power and data. Other frequency like 20 Hz may also work fine, therefore, this setting is not critical. However, too low frequency like 5 Hz may miss important information, and too high like 50 Hz probably adds no benefit for classification. The sampling frequency was selected during the design of the data acquisition hardware and firmware, to find a compromise between recognition performance and battery life. In this thesis, the effect of this setting is, therefore, not considered in detail.

3.2.3 Video Analysis

This study developed a long short-term memory (LSTM) deep NN model, which is a supervised machine learning process, to accurately recognize various states of cows. Video analysis was performed to generate labeled training and test datasets so that the model could gradually establish links between the motion data and motion labels by learning the labeled acceleration data. When new data arrived, the model produced the behavior pattern directly as trained.

The cow behavior was video-recorded in an undisturbed manner using the hand cameras (Sony HDR-AS300) over the 5-week trial period and the active video were around 32h long. The timestamp of the data from the accelerometer attached to the cow's neck and the timestamp of the camera were unified using a procedure that was partly automatic, and essentially relied on GPS time-stamping. A similar monitoring process has been reported in other studies [15–17]. The time of day for the video recording varied from morning (11:00) to the evening (17:00) in an attempt to cover several activities of the cows (feeding, walking, drinking, ruminating, and resting), which were determined and recorded at each time step for every cow. This is also important as differences due to the time of day may also be present, although small, in the individual behaviors. The distinction between the behaviors in the video analysis was based on the following criteria:

- Feeding is the act or process of eating or being fed.
- Walking signifies that the cow moves on grazing land.
- Drinking signifies that the cow drinks water from a drinking bowl.
- Rumination or cud-chewing is the process by which the cow regurgitates previously consumed food and chews it further.
- Resting represents static states, including standing and lying still.
- Others: Behaviors other than the five main behaviors mentioned above.

The final result was a mapping between the time series of the accelerometer readings and manually labeled action tags on a frame-by-frame basis.

3.2.4 Long Short-term Memory Networks (LSTMs)

As discussed in the previous chapters, the focus of this thesis is on data augmentation rather than neural networks in themselves, however, it would be impossible to introduce the augmentation methodology without considering at least in some detail the networks it is applied to. In this chapter, LSTM networks are considered, and the next chapters focus on CNN. Concretely speaking, LSTM were initially used because of their widespread use, however, later they were replaced by CNN to follow the general trend in this direction and also contribute to reducing model size. Here, a brief introduction of LSTM is given.

The activity measured by accelerometers attached to cows, or indeed any other type of sensor, is in the form of sufficiently long time series. With the movement of cows over time, the continuously measured data can characterize different behavior patterns. There are several types of NNs; however, owing to the characteristics of the data being long time series, a compatible NN with the ability to learn a time state during processing is generally required. A recurrent NN (RNN) satisfies this condition; that is, it is theoretically feasible. However, in practice, an RNN exhibits the vanishing gradient problem through layers when handling long sequences, making the network eventually untrainable [18]. Compared with a typical RNN, an LSTM network is characterized by improved memory modules in the network to counter the vanishing gradient problem for a long sequence. Thus far, a considerable amount of research has been conducted on the application of LSTM networks in time series data analysis [19–21].

As said, RNNs work well under appropriate circumstances, such as short sequence analysis, but there are short-term memory problems [22]. If a sequence is long, it is difficult

to transfer information from an earlier time step to a later time step. In the process of learning, RNN has the problem of gradient disappearance. Gradient is the value used to update the weight of neural network. The problem of gradient vanishing is that the gradient will decline as time goes on. If the gradient becomes very small, the learning process will not continue. That is, simple RNN is limited by short-term memory. In order to overcome this problem, in the early 1990s, a Long Short-Term Memory (LSTM) algorithm with "door" mechanism was designed by Sepp Hochreiter and Jürgen Schmidhuber [23,24].

The control flow of LSTM is similar to that of RNN, which processes the data of transmitting information in the process of forward propagation. The difference is that the structure and operation of LSTM unit have changed, where the core concept is the unit state and various door structures. The unit state is equivalent to a path that can transmit relevant information and let the information pass down in the sequence chain, which is shown as carry track in **Fig. 3.2**.

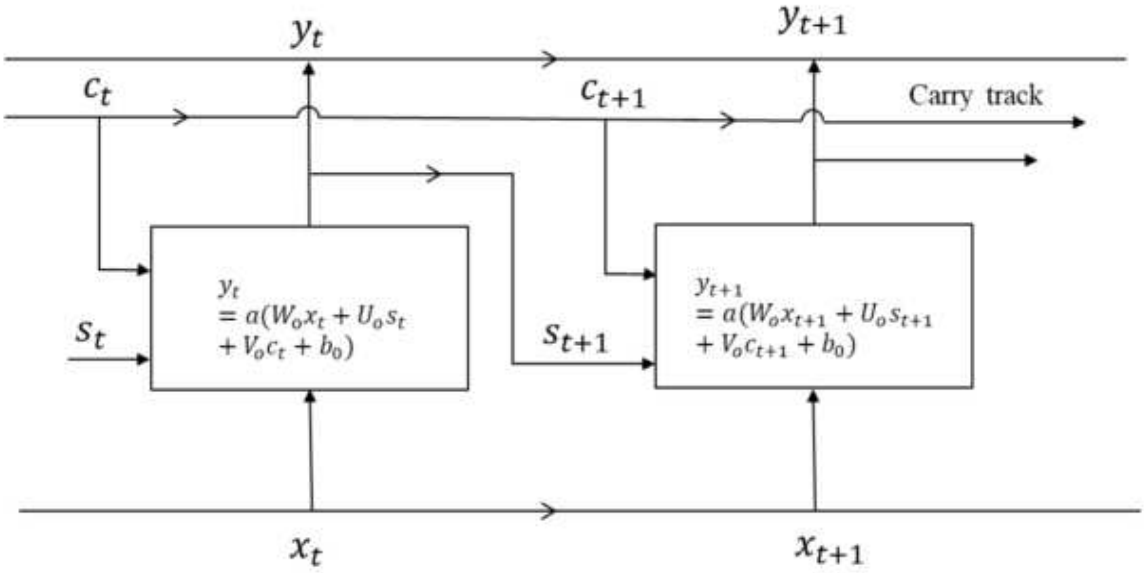


Figure 3.2: Adding a carry track to improve simple recurrent neural network

This can be seen as providing the "memory" of the network. In theory, in the process of sequence processing, the unit state can always carry relevant information. Therefore, the information obtained in the earlier time step can also be transmitted to the unit in the later time step, so as to reduce the impact of short-term memory. There are three kinds of gate structures in LSTM: forgetting gate, input gate and output gate. Forgetting gate can decide which information needs to be retained in the previous step, input gate decides which important information needs to be added in the current input, and output gate decides the next hidden state. In short, the control flow of LSTM network is actually just a few tensor operations and a for loop. Combined with these mechanisms, LSTM networks can selectively retain or forget some information during sequence processing. In short, the role of LSTM cell is to allow the past information to be injected again a while later, thereby fighting the problem of gradient vanishing. The LSTM neural network model layers in use are shown in **Fig. 3.3**, and described in further detail in the next section.

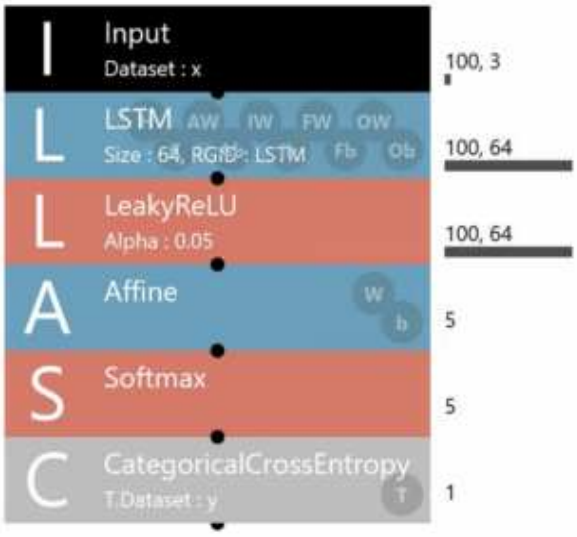


Figure 3.3: Layers of the neural network model. Generated using Neural Network Console by Sony Network Communications Inc [25].

In the system under consideration, the time series data from the accelerometer attached to the cow's neck are analyzed by the LSTM in order to estimate five classes of behavior. The LSTM model has one LSTM layer with 64 units. The model was trained using Adaptive Moment Estimation (Adam) [26] optimizer having an initial learning rate of 0.001 and batch size of 64.

3.2.5 Data processing

As previously explained, the time series data collected from the accelerometer sensor were labeled by video analysis, performed by expert operators using the SyncPlay software from ATR-Promotions Inc., part of Advanced Telecommunications Research Institute International (ATR), Kyoto. This program is especially designed to show together data from video and wearable sensors and enable their annotation. However, it should be noted that after this step the data were still in an irregular data format, because the transitions between behaviors occur at quite random, unpredictable times. This results in snippets of time series having very different lengths. To match the input format of the LSTM deep NN model, the data were cleaned and segmented to separate different behaviors. This was followed by data augmentation. Then, the acceleration data of each behavior pattern were extracted and segmented according to a certain length of time. For an LSTM deep NN model based on sensor data, a consistent time interval is indispensable for unified calculation. In this work, behavioral data from every 4 seconds with 25 Hz sampling frequency were used as an input sample; that is, 100 data rows were set as the length of one segment. These steps were executed in Python 3.6, which also realized the data augmentation processing using code that is provided in Appendix B.1. The structure of the data processing from label assignment to model training and evaluation is illustrated in **Fig. 3.4**. After obtaining the target sequence index file, 70% of the data were randomly selected to train the model, while the remaining 30%

were used as the test dataset. Here I first generated rotated data and then randomly separated the training and test datasets. Then, I could evaluate whether the trained model performs well in rotation angles other than those in the training set, thus reducing the impact of sensor position changes on accuracy in real-world applications. Notably, unlike rotation in image recognition, the acceleration data generated because of the change in the sensor placement may have a completely different relationship. This aspect is clarified in further detail below.

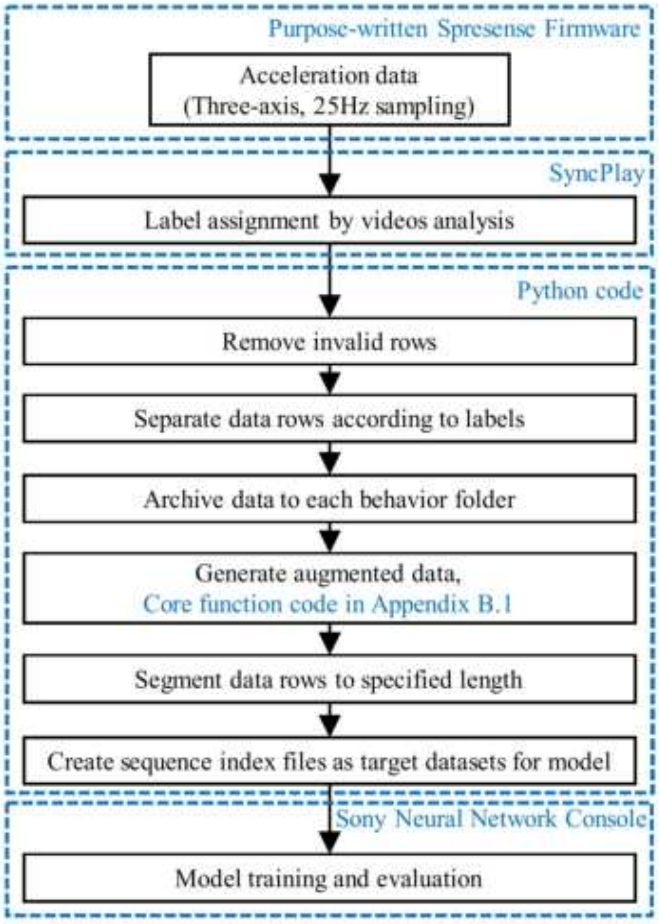


Figure 3.4: Flow of the proposed data processing used in this study. Blue overlays show the software platforms used for implementation.

For this chapter only, and for reasons having to do with experiments deploying the model to the edge device, which are not part of this thesis, the Neural Network Console was used to train the model. This is an effective tool developed by Sony to help design neural networks visually. Through Neural Network Console, the construction, training and evaluation of designed neural networks can be performed with an elegant user interface. In addition, Sony developed a series of built-in optional open-source neural network units based on the platform, which can support various types of network structures. This modular neural network unit style in the Neural Network Libraries and user-defined parameters make deep learning model building simple and efficient, and corresponding hardware requirement can also be effectively supported. Importantly, however, there are no conceptual or substantial algorithmic differences with respect to Python, indicating that the results can be considered interchangeably.

3.3 Classification performance measures

Although there are many indicators to evaluate the classification model for various considerations, generally speaking, the widely accepted evaluation metrics include accuracy, precision and recall [27]. As standard statistical process defined, accuracy, precision and recall can be obtained as:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{3-1}$$

$$\text{Precision} = \frac{TP}{TP+FP} \tag{3-2}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{3-3}$$

Here, "T" refers to True and "F" refers to False, corresponding to real situation. "P" refers to positive and "N" refers to negative, corresponding to our model results. "TP" (True Positives) is the number of instances in which the algorithm correctly classified the target behavior patterns after verification by a visual observer through video analysis. "FP" (False Positives) is the number of instances where the algorithm incorrectly classified other behavior patterns as the target pattern. "FN" (False Negatives) is the number of instances in which the algorithm incorrectly classifies the target behavior pattern as other patterns. "TN" (True Negatives) is the number of instances that other behavior patterns are correctly classified.

Regarding to the definition of precision and recall, I could gather the ideas expressed by these two indicators. That is, precision indicates how precise that our designed model achieves when real situation is true, since some false may be judged as true by the algorithm, and recall indicates how less misclassification for the true one the model made as the real situation is true because some true may be judged as negative by the algorithm.

However, the trend of precision and recall are often in tension. This means that if the precision is improved, recall is probably to decrease. Conversely, if the recall is improved, precision is typically reduced. In order to have a comprehensive understanding of the model and fully evaluate the effectiveness, F_β score is introduced as a balanced parameter that relies on both precision and recall.

$$F_\beta = (1 + \beta^2) \cdot \frac{Precision \cdot Recall}{\beta^2 \cdot Precision + Recall} \tag{3-4}$$

where β represents the relative importance of precision and recall. Respectively, in the case of β>1, false negatives are more emphasized so that recall has a higher weight than precision. In the case of β=1, the weight of recall is the same as that of precision. In the case of

β<1, the effect of false negatives has been attenuated so that the weight of recall is lower that the weight of precision.

In this study, I give equal importance to precision and recall. In this case, F_1 score can be calculated as follows:

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{3-5}$$

3.4 Proposed data augmentation method

Most of the standard algorithms in data science aim at the hypothetical ideal condition, that is, balanced data. However, unbalanced learning problems are more common in real life. If one directly uses a standard algorithm, it will cause a series of problems, which are eventually one cause of insufficient accuracy. This also represents a recurring problem of high importance with wide influence, which is worth more exploration because, as argued in Chapter 1, it affects edge AI across a wide span of applications. Here, both the problems of unbalanced data and insufficient number of data for some behavior classes are present. Hence, at this point the first data augmentation methodology is introduced to solve these problems.

3.4.1 Rationale

The problem caused by insufficient data and a severely skewed category distribution is called the imbalanced learning problem [28]. This problem has high complexity, and instead of a simple application of standard learning algorithms, it requires a deeper understanding of the data and algorithm. Therefore, transformation and augmentation of the original data are required to match the data distribution of standard categories. In this study, the

data size of different behavior patterns of cows was imbalanced. Specifically, the size of the data for the drinking label was much smaller than the size of data for other behaviors, e.g., the size of drinking data was only 1/165 the size of feeding data, as illustrated in **Fig. 3.5**. Directly applying an NN classification model to such imbalanced data produces insufficient accuracy, as illustrated in **Table 3.1**. This table reveals another problem, that is, possible overfitting. The evaluation indices for drinking are all high; in particular, recall indicates that all positive judgments are correct, which suggests that overfitting may have occurred due to the limited drinking data. That is, instead of learning the overall distribution of data, it is possible that the model may have learned the expected output of each input.

In this study, sensor position changes sometimes occurred due to misalignment of the device during the monitoring process. It is best to attach the sensor device at exactly the same position on the cow's neck in each measurement. However, the sensor displacement is inevitable in real-world testing. Not only there is a risk of slipping of the sensor device along the direction of the collar, but after the farm staff detaches the sensor device and attaches it again, even though we try to attach it at exactly the same position, there may be some variation. As illustrated in **Fig. 3.6**, when the position of sensor changes, the displacement in the front-back direction on a cow's neck does not have a significant impact on the accelerometer readings, while the rotation of the sensor does have a significant impact. Therefore, it is desirable that the training data includes the data collected at various rotation amounts. **Fig. 3.7**a illustrates three cases for simplicity; however, the rotation could be random at any angle from 0 to 360 degrees.

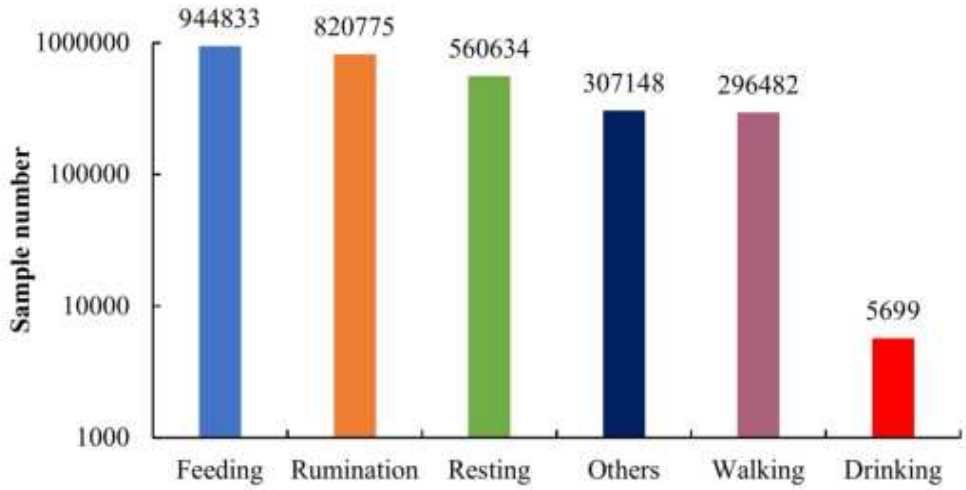


Figure 3.5: Number of row data points for five different behaviors of cows.

Table 3.1: Performance of classification without data augmentation.

	Precision	Recall	F_1-score
Feeding	78.25%	79.14%	78.69%
Walking	75.91%	70.75%	73.24%
Drinking	95.04%	100.00%	97.46%
Rumination	73.62%	78.46%	75.96%
Resting	76.10%	78.68%	77.37%
Others	63.66%	57.60%	60.48%

Therefore, different rotation states of the sensor device were simulated to cover additional data possibilities. Through sufficiently fine rotations at random angles of minority class samples, a balanced class distribution could be achieved. Instead of simply duplicating samples of the minority class, this method considers the actual rotation of the device and contains significantly more data possibilities. It is important to note that this is an augmentation of data considering real-life conditions, it is not the generation of the same data rows within the dataset itself. It is equally important to keep in mind that, from an information

theory viewpoint, no new information is generated, because the original time series could be easily recovered from the rotated ones. However, in practice, because of the rotation, the new time series are entirely different in terms of their value distribution from the initial ones, and this is enough for augmentation to be helpful [29]. The situation is very similar in other works on data augmentation, particularly for image processing, in which no new information is generated but the way the image is presented to the network is altered [30-32].

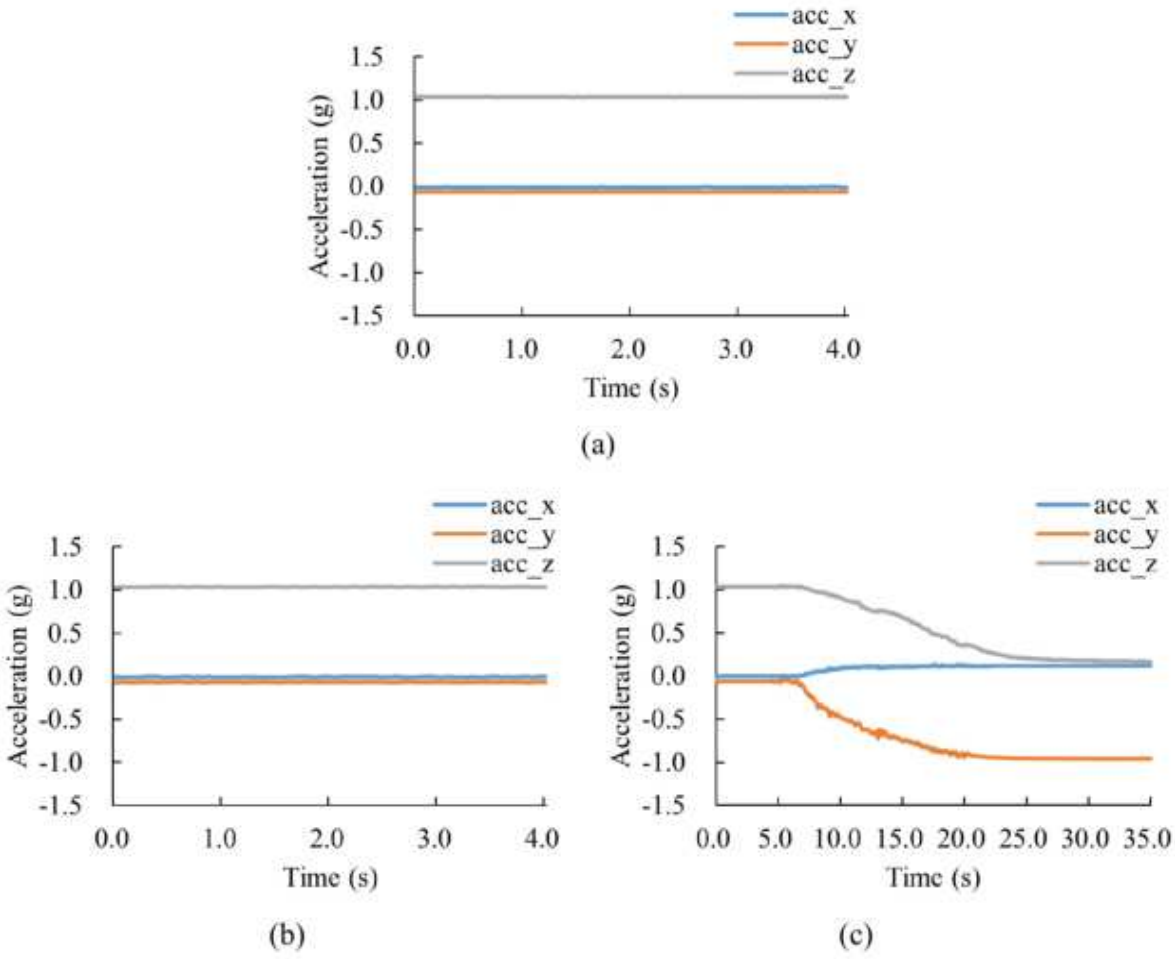


Figure 3.6: Impact of sensor position displacement. (a) Original data. (b) The sensor is moved to the front. (c) Rotating the sensor 90 degrees.

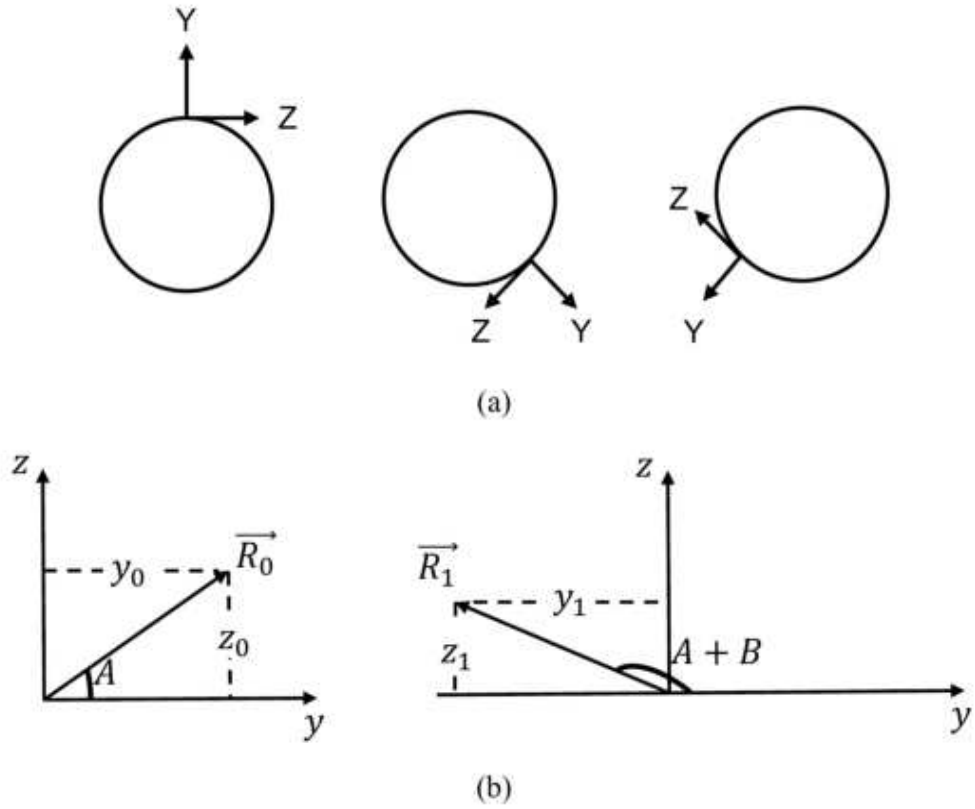


Figure 3.7: Schematic diagram of rotations of sensor device. (a) Coordinate diagram direction during the monitoring process. (b) Diagram for rotation of two-dimensional vectors.

3.4.2 Theoretical Basis and Practical Consideration

Having asserted that data augmentation by simulated rotation is helpful, it is now necessary to clarify the underlying mathematics. Because the measuring device was embedded in a collar worn on the cow's neck, the unknown rotation of the sensor is on the plane perpendicular to the neck, that is, the plane over which the collar lies. Therefore, the vector in the vertical ring direction can be ignored, and the two-dimensional vector of the cross-

section of the collar is used for the theoretical deduction. This helps by reducing the dimensions of the problem [33,34]. In other words, as shown in **Fig. 3.7a**, the problem is a two-dimensional rotation, with only one free variable.

For the purpose of explaining, let us assume that a two-dimensional vector rotates counterclockwise at angle B, as illustrated in **Fig. 3.7b**. The trigonometric relationships between the situation before (left-hand side illustration with the angle A) and after the rotation (right-hand side illustration with the angle $A+B$) are as follows:

$$\begin{cases} y_0 = |\overrightarrow{R_0}|\cos A \\ z_0 = |\overrightarrow{R_0}|\sin A \end{cases} \Rightarrow \begin{cases} \cos A = \frac{y_0}{|\overrightarrow{R_0}|} = \frac{y_0}{|\vec{R}|} \\ \sin A = \frac{z_0}{|\overrightarrow{R_0}|} = \frac{z_0}{|\vec{R}|} \end{cases} \tag{3-6}$$

and

$$\begin{cases} y_1 = |\overrightarrow{R_1}|\cos(A+B) = |\vec{R}|(\cos A \cdot \cos B - \sin A \cdot \sin B) \\ z_1 = |\overrightarrow{R_1}|\sin(A+B) = |\vec{R}|(\sin A \cdot \cos B + \cos A \cdot \sin B) \end{cases} \tag{3-7}$$

Substituting (3-6) into (3-7) yields:

$$\begin{cases} y_1 = |\vec{R}|\left(\frac{y_0}{|\vec{R}|}\cos B - \frac{z_0}{|\vec{R}|}\sin B\right) = y_0\cos B - z_0\sin B \\ z_1 = |\vec{R}|\left(\frac{z_0}{|\vec{R}|}\cos B + \frac{y_0}{|\vec{R}|}\sin B\right) = z_0\cos B + y_0\sin B \end{cases} \tag{3-8}$$

That is,

$$\begin{bmatrix} y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} \cos B & -\sin B \\ \sin B & \cos B \end{bmatrix} \begin{bmatrix} y_0 \\ z_0 \end{bmatrix} \tag{3-9}$$

The two-dimensional rotation matrix is therefore

$$\begin{bmatrix} \cos B & -\sin B \\ \sin B & \cos B \end{bmatrix} \tag{3-10}$$

As expected, a rotation matrix on a two-dimensional plane is obtained. Then, assuming that the acceleration value of the x-axis remains unchanged during the rotation process because it is along the cow's neck, a three-dimensional rotation matrix combined with the x direction can be obtained.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos B & -\sin B \\ 0 & \sin B & \cos B \end{bmatrix} \tag{3-11}$$

Finally,

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos B & -\sin B \\ 0 & \sin B & \cos B \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \tag{3-12}$$

The above formula can extend the data of the three-axis sensor at an arbitrary angle at a specific time step or over a period of time. An example of 45-degree rotation of the accelerations is shown in **Fig. 3.8** to provide an easy understanding of the original data and the data rotated using the proposed augmentation method. The point that should be appreciated here is that the individual time series are in a general sense uncorrelated, that is, their temporal features are not trivially similar. That's because, for example, the large transient around 2.5 s initially appears only on the y-axis (left-side panel), but after the rotation is also clearly visible, with reverse sign, on the z-axis plot (right-side panel).

Finally, there is one more practical consideration that is important for why only rotation around the x-axis is done and we preferred to keep the x-axis accelerations unchanged. It is not only because of collar rotation around the neck, but also because of the cow behavior in itself. In general, like other animals, cows usually keep their head more or less straight, which means, they do not roll it sideways very much during normal life. The situation for the x-axis is different because, in their behaviors, cows more often change the pitch, meaning the vertical orientation pointing their nose downwards or upwards, of their head. Their

head is almost facing the ground when feeding or grazing, however, it is less so when walking, for example. To confirm this, a brief additional analysis was performed, calculating the mean and standard deviation for the x, y and z axes separately across the behaviors. The results are shown in **Fig. 3.9a**. It can be seen that, considering the mean values for the behaviors, there is more variability in the values of the x-axis: the standard deviation of the averages is 0.14 g for the x-axis, and 0.08 g and 0.05 g for the y-axis and z-axis. For the x-axis, the difference between the minimum and maximum mean values due to the behaviors is 0.39 g, but for the y-axis and z-axis, it is just 0.20 g and 0.14 g. More specifically, the head was pointing downwards the most during feeding, with x-axis 0.48±0.14 g, followed by rumination, with x-axis 0.22±0.18 g, then similar values for resting and others, with x-axis 0.15±0.22 g and 0.14±0.31 g, slightly lower for walking, with x-axis 0.12±0.22 g, and smallest for drinking, with x-axis 0.10±0.25 g. A diagram of the cow head pitch is made in **Fig. 3.9b**, and it can be seen that these differences make sense. Because x-axis rotation would destroy them, it was preferred to focus on collar rotation instead.

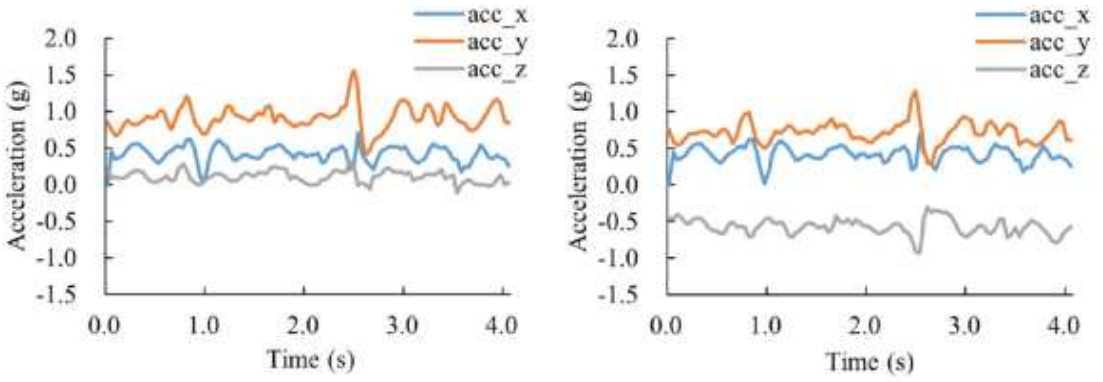


Figure 3.8: Accelerations augmentation example. (a) Original data. (b) Rotated (augmented) data using the proposed rotation method. The rotation amount is 45 degrees.

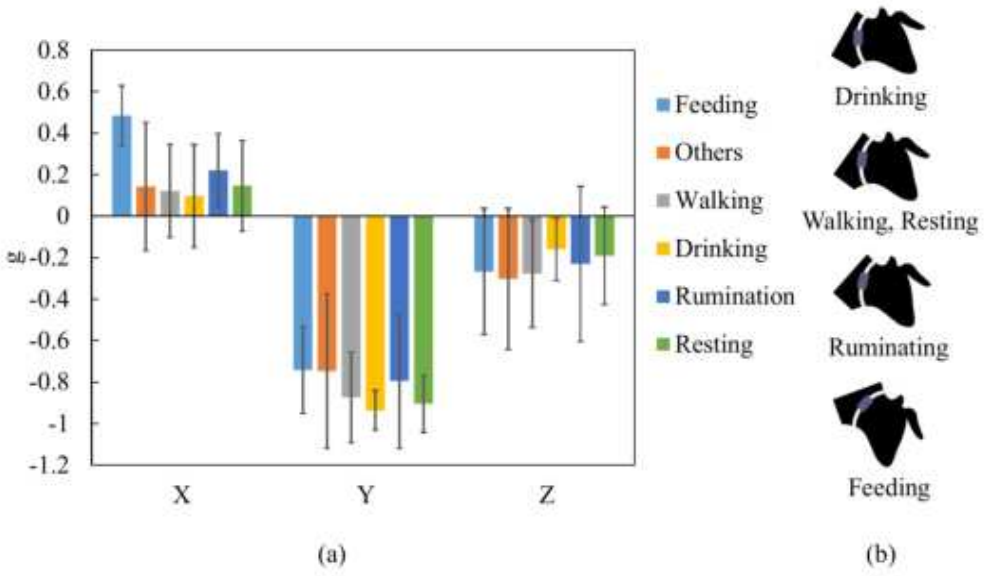


Figure 3.9: Average acceleration values. (a) Comparison of the x, y and z axes (mean±standard deviation), (b) Qualitative illustration of the differences in head pitch postures across the behaviors, based on the order of average x-axis values.

3.5 Experimental results

This section presents the results of estimating cow behavior patterns. A total of 70% of the prepared dataset was randomly selected as the training dataset, while the remaining 30% was used as the test dataset. The LSTM deep NN model was trained with the training dataset, and the effectiveness of the trained model was then examined with the test dataset. This examination is done by comparing the percent agreement between the activity value predicted by the LSTM model in the test dataset and the manually observed activity from the video.

The prepared expanded data was generated by multiplying original accelerations by a rotation matrix. A practical difficulty involves selecting the interval degree of rotation with respect to the original accelerometer data (i.e., extent of expansion). The magnitude difference of the original data is fixed; however, the interval degree of rotation is a variable that must be considered. In this study, the actual variable in data processing was the expansion multiple (i.e., number of parts 360 degrees is divided into); thus, the rotation interval should be a divisor of 360 degrees (e.g., 30 degrees, 36 degrees, 40 degrees, 45 degrees). With a finer rotation interval of less than 30 degrees, the drinking data would have to be further expanded and repeated to match the level of the sample size with the risk of other learning problems, such as overfitting. Conversely, an expansion that was too rough, for instance, 60 degrees or 90 degrees, may not have resulted in high recognition performance for non-rotational angles. Verifying all possible rotation angles requires a long time, which is not feasible.

Fig. 3.10 shows the comparison of precision for five different behaviors in total with several angle intervals of rotation. The precision test is done for every interval with a 5-degree dataset (72-fold expanded version of the original dataset). Finally, 30- degree rotation interval (i.e., 12-fold expansion) was selected as the final interval under the constraint of our predetermined accuracy goal (over 90% for each behavior). Specifically, the rotated data of labels other than drinking were generated in 30-degree increments with respect to the original motion data; that is, 12 sets of rotated data were generated from one set. On the other hand, the data of drinking behavior was expanded 360 times through finer rotation at random angles to achieve a more similar order of magnitude in terms of number of data samples. Thus, a more balanced class distribution was achieved for the model. **Table 3.2** lists the processed results.

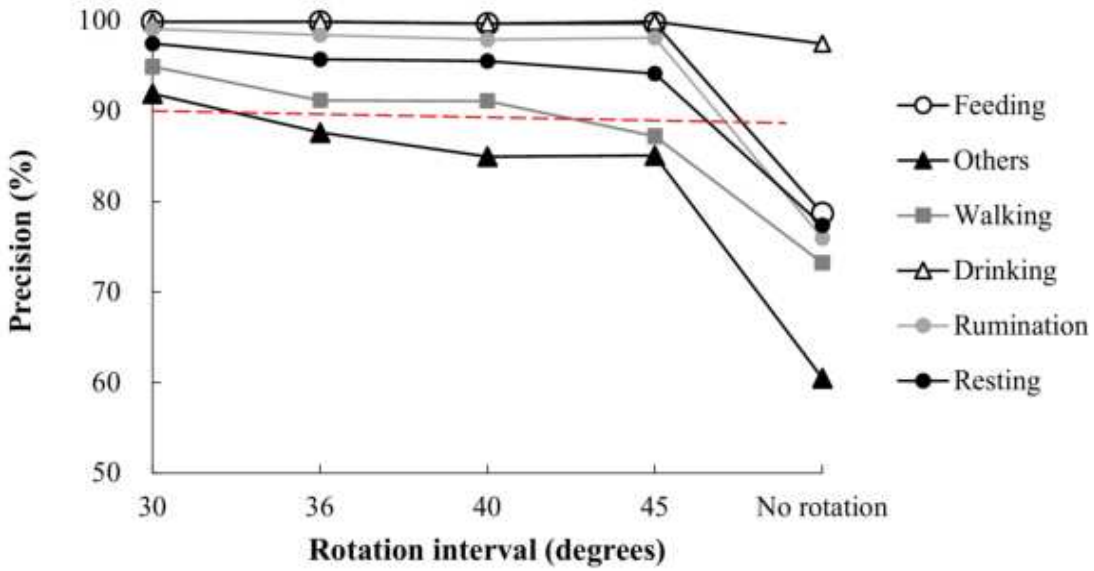


Figure 3.10: Comparison of precision values for each behavior pattern at different rotation intervals. The model was trained with 30-, 36-, 40-, and 45-degree datasets and tested with a 5-degree dataset. The precision result without rotation is also provided.

Table 3.2: Data rows and size without rotation and after rotation.

Label	Number of original data rows	Number of data segments	
		Without rotation	After rotation
Feeding	944833	8635	103608
Walking	296482	1612	19344
Drinking	5699	49	17640
Rumination	820775	6702	80424
Resting	560634	5129	61548
Others	307148	2332	27984

The test results and performance statistics with the updated 30-degree rotation interval expanded dataset are provided in **Table 3.3**. The specific result of cow behavior estimation is generated as a confusion matrix, and is presented in **Table 3.3a**. In a confusion matrix, rows represent actual behaviors while columns represent predicted behaviors. The classification performance is presented in **Table 3.3b**. It can be seen that the recognition rates are all over 97% in both precision and recall. The balanced indicator F_1 score is also satisfactory, which signifies that the deep NN model can be effectively used. Compared with the estimation performance without data rotation, which was insufficient with a minimum of 60.48% (see **Table 3.1**), the recognition rate was improved by 2.5%–37.1%. In Chapter 5, it is explained that the behaviors are different in the time series properties, such as autocorrelation. This variance is probably consequence of that, because how difficult it is to improve is not always the same. The methods explained in the next chapters have lower variance in improvement.

Tables 3.4 and **3.5** are presented to further evaluate the reliability of the selected angle of 30-degrees. **Table 3.4** presents the performance statistics of the estimation results with a 45-degree dataset and the evaluation indices for six behavior patterns were all over 90%, while **Table 3.5** displays an accuracy comparison of 30-degree and 45-degree rotation intervals. From **Table 3.5b**, it can be seen that when training model with a 45-degree dataset and testing with a 30-degree dataset, the precision was 87.41% in walking and 85.44% in other behaviors. In contrast, the model trained with 30-degrees maintained high estimation performance with the evaluation indices of all over 90% for different behaviors (see **Table 3.5a**), which also proves 30-degrees rotation interval is a satisfactory rotation parameter.

Table 3.3: Classification results for a test dataset using long short-term memory model with a 30-degree rotation interval expanded dataset. (a) Confusion matrix. (b) Classification performance.

(a)

	Feeding	Walking	Drinking	Rumination	Resting	Others
Feeding	5170	50	2	12	13	45
Walking	2	5286	0	0	4	0
Drinking	0	0	5292	0	0	0
Rumination	5	10	0	5152	79	46
Resting	7	5	0	54	5139	94
Others	0	1	0	4	67	5213

(b)

	Precision	Recall	F_1 score
Feeding	99.73%	97.69%	98.70%
Walking	98.77%	99.89%	99.32%
Drinking	99.96%	100.00%	99.98%
Rumination	98.66%	97.35%	98.00%
Resting	96.93%	97.11%	97.02%
Others	96.57%	98.51%	97.53%

Table 3.4: Performance of classification on test dataset using long short-term memory model with a 45-degree rotation interval expanded dataset.

	Precision	Recall	F_1 score
Feeding	99.23%	95.01%	97.09%
Walking	96.47%	99.86%	98.13%
Drinking	100.00%	100.00%	100.00%
Rumination	96.37%	94.90%	95.63%
Resting	93.58%	92.91%	93.24%
Others	92.93%	95.69%	94.29%

Table 3.5: Performance of angle verification on a test dataset. (a) Training model with a 30-degree dataset and testing with a 45-degree dataset. (b) Training model with a 45-degree dataset and testing with a 30-degree dataset.

(a)

	Precision	Recall	F_1 score
Feeding	99.93%	97.89%	98.90%
Walking	94.76%	99.94%	97.28%
Drinking	96.08%	100.00%	98.00%
Rumination	98.99%	97.87%	98.42%
Resting	97.61%	97.33%	97.47%
Others	91.87%	98.84%	95.23%

(b)

	Precision	Recall	F_1 score
Feeding	99.72%	95.69%	97.67%
Walking	**87.41%**	99.94%	93.26%
Drinking	94.23%	100.00%	97.03%
Rumination	98.09%	95.02%	96.53%
Resting	93.84%	94.46%	94.15%
Others	**85.44%**	96.10%	90.45%

The results obtained in this study can be used as reference for future applications in which feedback must be collected and potential problems during the field tests must be solved. It should be noted that in this experiment, the training data originated from two sample cows, which may be insufficient for the trained model to ignore individual differences. Hence, with the LSTM deep NN model can then be improved to obtain better results with more data obtained from more cows in future work.

Table 3.6: Comparison of average classification accuracy obtained by machine learning presented in literature.

	Recognition method	Classified classes	Data augmentation technique	Average accuracy (%)
Ref. [35]	Decision tree	3	No	88.33
Ref. [36]	Decision tree	3	No	82.24
Ref. [37]	LSTM network	8	No	88.65
This work	LSTM network	6	Yes	98.43

Although problems may remain in real-life application, the proposed data augmentation method opens up the way for improvements on the results obtained in other studies mentioned in Section 1, as demonstrated in **Table 3.6**. It should be noted that in [37], eight main behaviors (feeding, lying, lying rumination, standing rumination, licking salt, moving, social licking and head butt) of cattle were successfully classified using an LSTM NN model, with an average accuracy of over 88%. This is mainly because, although it did not perform data augmentation, a large amount of teaching data was collected. For each eight behavioral pattern, 45568 data rows were used for the model, developing a balanced data set. However, in this study, there are only 5699 data rows for "drinking" behavior without data augmentation, which is far less than the case for [37], and the data set presented in here is unbalanced. As such, the proposed data augmentation method can generate data for actions that occur less frequently and improve the accuracy of multi-states estimation. Based on the successful recognition of fundamental behaviors in this study, larger number of behavior pattern analysis can become possible. This can help analyze complex situations. For example, the social behavior of cows can also be considered in behavior pattern recognition, such as head-butting, chasing-up, and social licking, which can lead to a more comprehensive understanding. In addition, in [37], the sensor data used were from a 3-axis accelerometer, 3-axis gyroscope, and 3-axis magnetometer. However, this study only uses a 3-axis

accelerometer to reduce the hardware cost and operation energy of the sensor device. Thus, applying the data augmentation method presented here enables us to implement more effective animal behavior monitoring systems at lower cost, both in terms of device and manufacturing costs, and power consumption and battery costs.

3.6 Conclusion

Focusing on changes in activity levels, previous studies used data from an acceleration sensor. Nevertheless, none implemented the method of data rotation as a means of reducing the behavior distribution imbalance. This study proposed a rotation-based data augmentation method for cow behavior estimation systems based on 3-axis acceleration data and evaluated it using NN. The LSTM deep NN model achieved high estimation performance with an average estimation accuracy of over 98% for five main behaviors (feeding, walking, drinking, rumination, and resting). The proposed data augmentation method improved the recognition rates by 2.5%–37.1% compared to the situation without data augmentation. The feasibility and efficiency of the proposed solution to existing problems in data collection were verified in this study. It is considered that these results can reduce the cost of data collection and contribute to the promotion of the use of NN technology in livestock farming.

In summary, this chapter presents a first attempt at realizing the promise of data augmentation put forward in Chapters 1 and 2. The results provided in here demonstrate that random data rotation is sufficient to lead to significantly better classification results compared to using the experimental recordings on their own. At the same time, it should be considered that the variability contained in the dataset was small at this stage, because the research project had just begun. In the next chapter, the results obtained considering a more realistic situation, with more cows but less data per cow, shall be considered. The approach presented in this chapter, while system-specific, provides an initial reference. When there

are other similar situations where a sensor may be misaligned, rotated or otherwise decoupled from the underlying process, it is meaningful to resort to data augmentation by means of a transformation that considers the possible free parameters. In this work, the free parameter was just one, meaning, rotation angle around the neck, but in other situations a spherical rotation may need to be considered, for example.

3.7 Bibliography

[1] H. He and E. A. Garcia, "Learning from imbalanced data," IEEE Trans. Knowl. Data Eng., vol.21, no. 9, pp.1263-1284, 2009.

[2] K. Chen, D. Zhang, L. Yao, B. Guo, Z. Yu, and Y. Liu, "Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities," ACM Computing Surveys (CSUR), vol. 54, no. 4, pp.1–40, 2021.

[3] A. Yurtman and B. Barshan, "Activity recognition invariant to sensor orientation with wearable motion sensors," Sensors, vol. 17, no. 8, p. 1838, 2017..

[4] A. Yurtman, B. Barshan, and B. Fidan, "Activity recognition invariant to wearable sensor unit orientation using differential rotational transformations represented by quaternions," Sensors, vol. 18, no. 8, p. 2725, 2018.

[5] S. Sivaramakrishnan, C. Lee, B. Johnson and A. Molnar, "A Polar Symmetric CMOS Image Sensor for Rotation Invariant Measurement," in IEEE Sensors Journal, vol. 16, no. 5, pp. 1190-1199, March1, 2016.

[6] S. C. Wong, A. Gatt, V. Stamatescu, and M. D. McDonnell, "Understanding data augmentation for classification: when to warp?" 2016 international conference on digital image computing: techniques and applications (DICTA), pp.1-6, 2016.

[7] J. Tanha, Y. Abdi, N. Samadi, N. Razzaghi, and M. Asadpour, "Boosting methods for multi-class imbalanced data classification: an experimental review," J. Big Data, vol. 7, no. 1, pp. 1–47, 2020.

[8] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," ACM SIGKDD Explor. Newsl., vol.6, no. 1, pp.20-29, 2004.

[9] A. Le Guennec, S. Malinowski, and R. Tavenard, "Data augmentation for time series classification using convolutional neural networks," 2016.

[10] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," arXiv Prepr. arXiv1405.3531, 2014.

[11] X. Cui, V. Goel, and B. Kingsbury, "Data augmentation for deep neural network acoustic modeling," IEEE/ACM Trans. Audio, Speech, Lang. Process., vol.23, no. 9, pp.1469-1477, 2015.

[12] "Get started with Spresense development." [Online]. Available at: https://developer.sony.com/develop/spresense/developer-tools. Accessed on Dec 20, 2022.

[13] Kionix, Inc., "± 2g / 4g / 8g Tri - axis Digital Accelerometer," KX122-1037 datasheet, Jan. 2018.

[14] W. Kester, "What the nyquist criterion means to your sampled data system design," Analog Devices, pp. 1-12, 2000.

[15] P. Ahrendt, T. Gregersen, and H. Karstoft, "Development of a real-time computer vision system for tracking loose-housed pigs," Comput. Electron. Agric., vol. 76, no. 2, pp. 169–174, 2011.

[16] H. H. Kristensen and C. Cornou, "Automatic detection of deviations in activity levels in groups of broiler chickens–a pilot study," Biosyst. Eng., vol. 109, no. 4, pp. 369–376, 2011.

[17] X. P. Burgos-Artizzu, P. Dollár, D. Lin, D. J. Anderson, and P. Perona, "Social behavior recognition in continuous video," in 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 1322–1329.

[18] C. Francois, "Deep learning with Python," Manning Publications Company, 2017.

[19] S. Ebrahimi Kahou, V. Michalski, K. Konda, R. Memisevic, and C. Pal, "Recurrent neural networks for emotion recognition in video," Proceedings of the 2015 ACM on International Conference on Multimodal Interaction, pp.467-474, 2015.

[20] H. Ito, N. Saito, C.Y. Huang, S. Hata, A. Okumura, K. Toda, et al., "Development scheme for cattle behavior estimation by deep learning in an edge device," Proceedings of the 2nd international precision dairy farming conference, pp.87-88, 2019.

[21] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," Proceedings of the IEEE conference on computer vision and pattern recognition, pp.1110-1118, 2015.

[22] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies." A field guide to dynamical recurrent neural networks. IEEE Press, 2001.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

[24] S. Hochreiter and J. Schmidhuber, "LSTM can solve hard long time lag problems," in Advances in neural information processing systems, 1997, pp. 473–479.

[25] "Neural Network Console by Sony Network Communications Inc." [Online]. Available at: https://dl.sony.com/. Accessed on Dec 20, 2022.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv: 1412.6980, 2014.

[27] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," Information processing & management, vol. 45, no. 4, pp. 427-437, 2009.

[28] H. Kaur, H. S. Pannu, and A. K. Malhi, "A systematic review on imbalanced data challenges in machine learning: Applications and solutions," ACM Computing Surveys (CSUR), vol. 52, no. 4, pp. 1–36, 2019.

[29] J. F. Wellmann, "Information theory for correlation analysis and estimation of uncertainty reduction in maps and models," Entropy, vol. 15, no. 4, pp. 1464–1485, 2013.

[30] A. Krizhevsky, "Learning multiple layers of features from tiny images,"2009.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of IEEE conference on computer vision and pattern recognition, pp.770–778, 2016.

[32] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[33] H. J. Williams, M. D. Holton, E. L. Shepard, N. Largey, B. Norman, P. G. Ryan, O. Duriez, M. Scantlebury, F. Quintana, E. A. Magowan et al., "Identification of animal movement patterns using tri-axial magnetometry," Movement ecology, vol. 5, no. 1, pp. 1–14, 2017.

[34] M. Gaitan and J. Geist, "Calibration of triaxial accelerometers by constant rotation rate in the gravitational field," Measurement, vol. 189, p. 110528, 2022.

[35] B. Robert, B. J. White, D. G. Renter, and R. L. Larson, "Evaluation of three-dimensional accelerometers to monitor and classify behavior patterns in cattle," Comput. Electron. Agric., vol.67, no. 1-2, pp.80-84, Jun. 2009.

[36] J. A. Vázquez Diosdado, Z. E. Barker, H. R. Hodges, J. R. Amory, D. P. Croft, N. J. Bell, et al., "Classification of behaviour in housed dairy cows using an accelerometer-based activity monitoring system," Anim. Biotelemetry, vol.3, no. 1, p.15, 2015.

[37] Y. Peng, N. Kondo, T. Fujiura, T. Suzuki, Wulandari, H. Yoshioka, et al., "Classification of multiple cattle behavior patterns using a recurrent neural network with long short-term memory and inertial measurement units," Comput. Electron. Agric., vol.157, pp.247-253, Feb. 2019.

Chapter 4

Data Augmentation based on combining multiple empirical methods

After having introduced random rotation as a means of data augmentation in the previous chapter, in this chapter the concept will be extended to other manipulations. Moreover, in terms of the properties of the dataset of experimental recordings that is taken as a starting point, this chapter considers a more realistic scenario. Also, as discussed below, convolutional neural networks (CNN) are becoming more and more popular, because of several factors including the simplicity of their implementation. For this reason, from this chapter onwards, the LSTM network is replaced with a CNN network aiming. The aim, again, is to obtain results that, in the future, may be applicable across different types of Edge AI devices and application scenarios.

Briefly saying, the additional sensor data augmentation methods introduced here fit the characteristics of cattle behavioral data, however, they are less limited to it compared with the assumption of planar rotation around the neck's axis. Using the methods proposed in

this chapter, the classification performance increases from 83.07 to 94.43% with appropriate augmentation steps. In conclusion, the data augmentation approaches presented here can help deep learning performance regarding cattle behavior classification and decrease the overall system cost stemming from data acquisition and labeling. For this reason, these results expand and augment those of the previous chapter.

4.1 Data acquisition and preparation

A dataset containing 6 different cows under normal living conditions was collected using an acceleration sensor without human disturbance, in the same settings as the preceding chapter. As detailed below, there are some important differences, however, that should be considered. Rather than a limitation or an inconsistency, these should be considered carefully in light of the fact that they are rather representative of the development process for an Edge AI device [1,2] . As mentioned in Chapter 1, it is almost an unavoidable fact that concept refinement, data acquisition, hardware and software development are proceded in parallel: if one assumes differently, then the situation is a hypothetically ideal one but not representative of the real world situation. Therefore, the development and improvement of classifiers almost always proceeds at the same time as the acquisition of data, under settings that may change due to refinements and improved experience, especially towards the beginning of a project. In this case, the key difference is that it was realized that recording for a long time a small number of cows is poorly effective. That is because it is the inter-individual variability between cows that fuels the complexity of a project such as this one. Realizing these factors, it was decided to increase the number of cows by a factor of three; however, because the overall resources for data acquisition could not be unlimited, the recording duration for each cow was reduced. Basically, in Chapter 3, 2 cows were recorded

for approximately 979 min each, whereas here, 6 cows were recorded for approximately 59 min each.

Unavoidably, the above results in a situation which is much more challenging for network training, because the variability in the dataset is larger, and the amount of data is smaller. As previously explained, in addition to the difficulties of obtaining a vast amount and many varieties of behavioral data, creating a high-quality dataset necessitates the manual classification and annotation of the collected data, which is time-consuming and laborious [3,4]. Thus, to fully exploit the potential of deep learning and enable it to work effectively on small-scale datasets, accurate and effective augmentation approaches for behavioral data are urgently needed [5]. On the other hand, the behavioral complexity of animals introduces challenges in real-world applications because different activities may generate similar sensor data, e.g., acceleration data. The reason for this phenomenon is that various activities involve similar gestures, for example, standing rumination and standing resting. Additionally, individual differences increase interclass variability. As a result, cattle activity classification is highly challenging, especially when the amount of input data is relatively small [6-8].

The data collection settings, in terms of hardware setup, were kept the same as the previous chapter. Notably, the dataset used in this chapter has been made publicly available at https://doi.org/10.5281/zenodo.5399259 (accessed on 24 September 2021). Another aspect of note is that, in this chapter, the dataset is randomly split into three parts. First, the dataset is divided into learning and independent test sets at an 80/20 guidance ratio, and then 10% of the learning set (8% of the whole dataset) is used as a validation dataset, as shown in **Fig. 4.1**. The training dataset is used to establish an initial behavioral model, the validation dataset is used to tune the model parameters during the training process, and the independent test dataset is used to evaluate the performance of the trained model.

As shown in **Fig. 4.1**, the same specific firmware running on the Spresense board was used to collect the acceleration data, as described in Chapter 3.2.1. Video labeling was also performed in the same way using the SyncPlay software from ATR-Promotions Inc., as described in Chapter 3.2.5. Data separation and splitting were performed using Python, followed by data augmentation and model training. Core function code for augmentation was given in Appendix B.2.

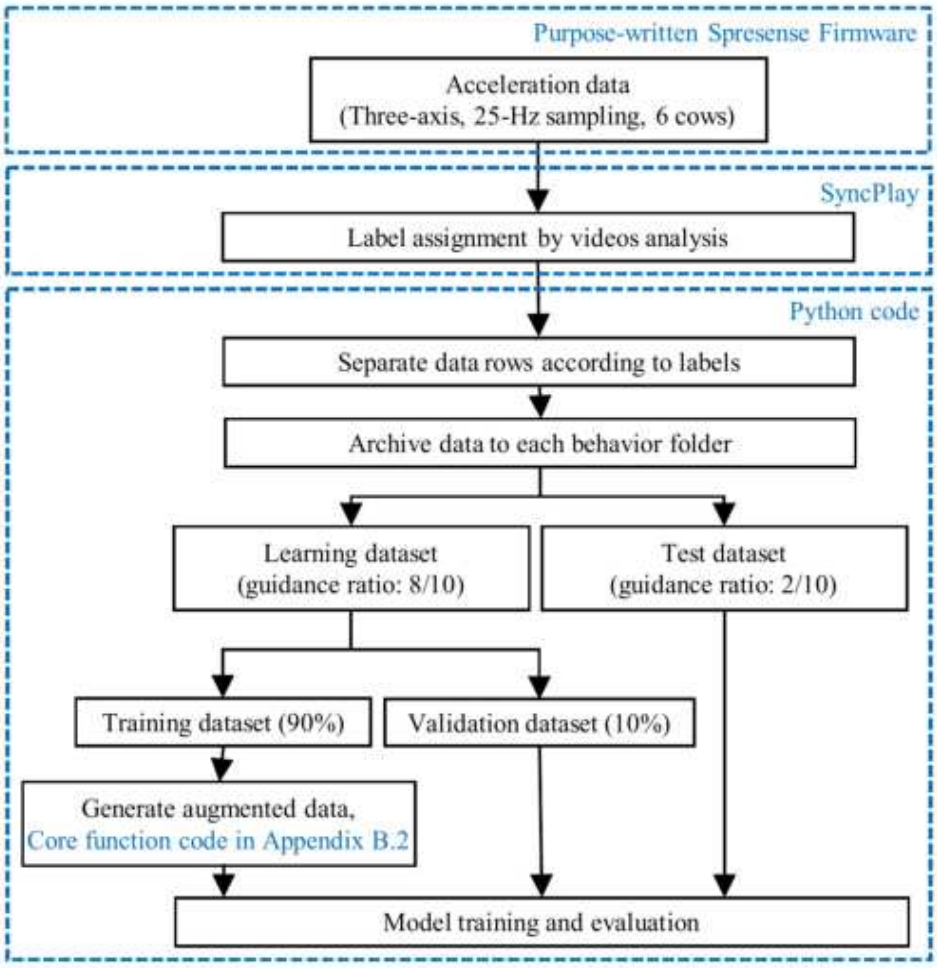


Figure 4.1: Flow of data preparation process in this study. Blue overlays show the software platforms used for implementation.

4.2 The CNN architecture

At this point throughout the thesis, it becomes relevant to consider in some additional detail the type of neural network used for time series classification. The issue of time series classification and anomaly detection is an old and well-known one in AI, having been studied since the early days of the field, several decades ago [9,10]. Historically, the first approach to be considered was the calculation of a feature vector. This means that the time series would be preprocessed using a set of feature-extraction algorithms, each one designed to pick up one or more aspects of the signal content deemed to be relevant. One key aspect of these features is that they should be invariant to irrelevant transformations: for example, translating over time, phase shifting and in some cases even amplitude changes should not affect the features. In the early days of AI, feature selection and extraction was extremely important, with many experiments and reviews dedicated to the topic. I would like to point out that they drew from many areas of signal processing and information theory. For example, the Fourier amplitudes of a signal are a common basis on which to calculate features for classification, since they are time- and phase-insensitive. Otherwise measures such as the entropy are also relevant [11,12].

The approach based on feature extraction has the advantage of keeping the classifier as simple as possible, because the size of the input vector is considerably reduced compared to the raw time series, and because, if proper features are selected, the separability is also much better. For this reason, multi-layer perceptron (MLP) networks with even a small number of layers usually perform well in this kind of context. However, there are two serious drawbacks. The first is that there are no rigorous guidelines for feature selection: as a consequence, human skill and experience play a key role in determining the final accuracy. For example, if a certain problem has one or more features that would be very important in classifying the behaviors, but the human operator does not know about them, they will not

be included in the feature vector and the performance will be poor. The second is that, often, the process of calculating the feature values and assembling them into the feature vector may be computationally demanding, even more computationally demanding than executing the classifier itself [12-14].

Considering the evolution of the field, the next aspect to be considered throughout the field's development was the fact that time series, by definition, unfold over time. That is to say, the vector entries are not in an arbitrary order, but there is a clear inter-dependence from one to the next. In a neural network such as a CNN or an MLP, there is no time-memory or sensitivity to the flow of time, because there are no re-entrant connections. For this reason, RNN networks were introduced and, as detailed in the previous chapter, they lend themselves well to time series classification. In the previous chapter, an LSTM network was found to give good performance, and it was found that data augmentation improves this performance. However, the network was relatively large and, in general, the field of Edge AI seems to be moving towards CNN rather than RNN, because RNN can be more difficult to train and may even suffer stability problems. The rest of this thesis therefore focuses on CNN, leaving the exploration of data augmentation in the context of RNN for future work.

Continuing with explaining the historical trajectory of this field, CNNs are among the most recent type of network introduced. In summary, they effectively also include an MLP, but the feature extraction process, instead of being left outside the training and dependent on the operator, is included in the training itself. That is to say, the network is fed the vectors corresponding to the entire time series, and it is left to the network itself to find the best features to extract and submit to the MLP. The main advantage of this approach is that it is fully automatic, and as such usually finds high-quality solutions. The drawback, however, is that it makes the process sensitive to many sources of variability that an external feature

extractor would deal with, and representative examples include, as said above, time-translation and phase shifting [15,16]. This aspect makes data augmentation very important. One can say, at this point, that the process of data augmentation is meant to aid with dataset size and imbalance. However, it also helps to deal with the sensitivity to irrelevant aspects that originates from the fact that the actual time-vectors enter the training process. With this, I mean that data augmentation can make a CNN less sensitive to the specific point in time that its input data are taken from, because it expands the classifier boundaries. As previously mentioned, in general, the aspects that data augmentation helps to be insensitive to include time translation, phase shifts, changes in sensor axis rotation and so on. This means that the manipulations used for data augmentation should also include the transformations, such as translation, that one wants the classifier to be insensitive to [5]. This aspect will be further expanded below.

Throughout this chapter and the next one, CNNs are utilized for the task of cattle behavior classification starting from limited available labeled data. CNNs are widely used to detect features with filters sliding over time series data and have the advantage of automatic feature extraction with only limited computational complexity [16]. A depth of 8 layers is adopted for the CNN in this chapter to grasp the high variability of the small-scale behavioral data.

To be precise, as detailed for example in Refs. [15] and [16] the architecture of a CNN differs from an MLP is that it contains a deep stack of layers that are used for feature extraction. These contain convolutional, pooling and feature layers. Convolutional layers apply filters that aim to emphasize particular features of the data, such as fluctuations on a particular time-scale, or similar. Further, they also apply a non-linear transformation, which leads to the enhancement of variance of interest and suppression of irrelevant features. Pooling layers, effectively, are necessary to avoid an explosion in the amount of information throughout the network layers. These layers compress the feature maps by operations such

as maximum or averaging. After a sufficient alternation of these two layer types, a feature layer encodes the final set of values that are fed to the classifier: effectively this is an MLP network, which is usually referred to as the "head" of the network.

The design and sizing of the network layers is a task that requires assumptions and compromises between computational load, generalization ability and granularity of the extracted features. Usually, once reasonable values are found for the number of layers and their size, then the network is not significantly sensitive to these settings; this is one of the advantages of CNN networks. For this reason, to avoid an excessive number of tests, throughout this thesis I keep the network size fixed.

The following network configuration was chosen. Firstly, the input of the network is a time series of 125 points each of which actually corresponds to a [x,y,z] triplet. Then, a 1D convolution layer increases the dimensionality to 125×6. This is followed by a rectified linear unit (ReLU) layer, representing the only type of non-linearity used in this network. Then, a further 1D convolution layer brings the dimensionality to 125×12, after which a 1D MaxPooling layer reduces it to 62×12. Then, another 1D convolution layer increases it to 62×18, after which a further 1D MaxPooling reduces it to 31×18. This alternation is continued, leading to 31×24, then 15×24, 15×30 and finally 1×30 via average pooling to create the Feature layer. The aspect that is worth pointing out here is that there is a gradual shift from a representation which is quite elongated over time (125) but shallow in terms of number of features per time point (initially, 3), towards one that eventually has even more features than time points. This shift corresponds to digging deeply into the signal features, that are represented in a higher-dimensional space compared to the initial time series. It is precisely this operation which confers CNN networks their power, but at the same time, requires large amounts of data to determine all parameters. Finally, a dense layer having size 1×30 is fully connected to a 1×5 output layer, which encodes the output classes in terms of

probabilities, with a winner-take-all approach. The detailed layers structure is visible in **Fig. 4.2**.

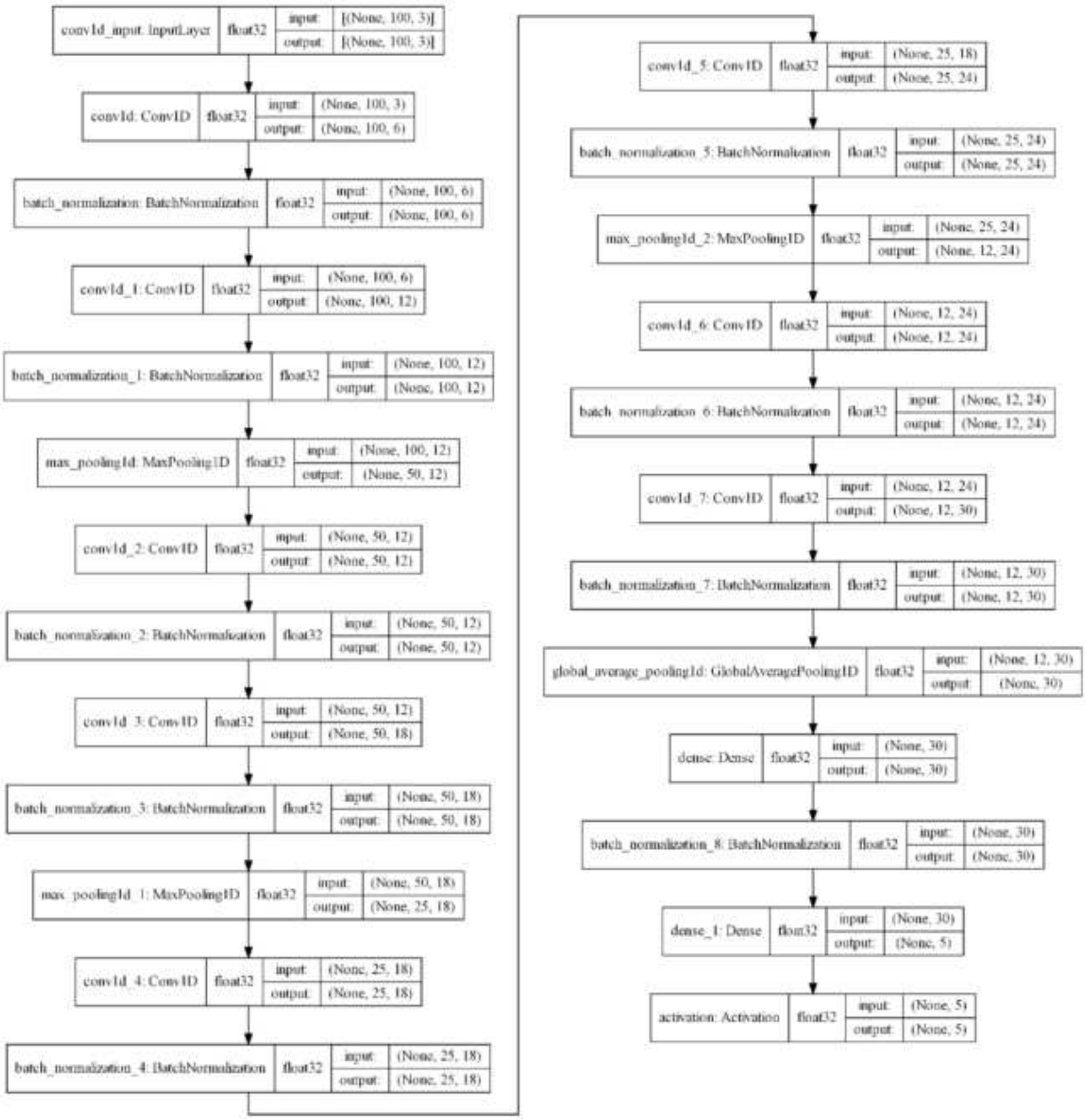


Figure 4.2: Architecture of the convolutional neural network (CNN)

4.3 Implementation aspects of the CNN network

While a detailed analysis of the aspects of implementing the network on an edge device was not conducted in this thesis, in this section, some basic information is provided to show that it can be possible to implement the network on a suitable microcontroller, so that analysis can be performed at the edge in real time.

First of all, the same CNN topology shown in **Fig. 4.2** was implemented using the SONY Neural Network console program, because it provides a reliable means of estimating the occupancy and computation load of the network, which are independent of the implementation platform [17]. The resulting calculations, shown in **Table 4.1**, were checked and confirmed independently with Python scripts developed by others [18,19]. As further discussed below, I am assuming that the network is implemented using single-precision floating point numbers (each occupying 4 bytes). It can be seen that, to store the outputs of all layers, a total of 21,203 values are needed, giving 21,203 × 4 bytes = 82.8 Kbytes. Considering the network parameters (it means, the outputs of training that are fixed during edge device operation), a total of 9,431 values are needed, giving 9,431 × 4 bytes = 36.8 Kbytes. Therefore, it can be said that the memory occupation of this model is 82.8 Kbytes + 36.8 Kbytes = 119.6 Kbytes. Then, considering that the network outputs are evaluated once each time a new time window is acquired, it is possible to calculate how many operations per second are needed on average, with (249,629 operations + 9,576 operations) × 25 Hz / 125 points = 51,841 operations / second. In most cases, one operation such as multiply or accumulate takes one clock cycle, therefore, on the average and rounding for excess, the clock frequency needed is about 60 kHz, or 0.06 MHz (this is a theoretical minimum).

For showing that the CNN model could be implemented on an edge device and does not need a full desktop or server computer or even a single board system like Raspberry PI, I selected three representative target microcontroller platforms. One is Spresense from Sony

Group Corporation, another is STM32F469 from STMicroelectronics NV, and another is ESP32-PICO-MINI-02U from Espressif Inc. The corresponding data about the clock frequency, available on-chip memory, size of an evaluation module board and power consumption were taken from the corresponding specifications; in the case of Spresense, power consumption was taken from additional measurements published by users [20-23]. These values are shown as a summary in **Table 4.2**.

Table 4.1: Memory occupation and calculation load for the CNN network

Layer	Outputs	Parameters	MAC Ops.	IF Ops.
Input	125 × 3 = 375	0	0	0
Transpose	3 × 125 = 375	0	0	0
Convolution	6 × 125 = 750	60	7500	0
ReLU	6 × 125 = 750	0	0	750
BatchNormalization	6 × 125 = 750	24	1500	0
Convolution_2	12 × 125 = 1500	228	28500	0
ReLU_2	12 × 125 = 1500	0	0	1500
BatchNormalization_2	12 × 125 = 1500	48	3000	0
MaxPooling	12 × 62 = 744	0	0	1488
Convolution_3	12 × 62 = 744	444	27528	0
ReLU_3	12 × 62 = 744	0	0	744
BatchNormalization_3	12 × 62 = 744	48	1488	0
Convolution_4	18 × 62 = 1116	666	41292	0
ReLU_4	18 × 62 = 1116	0	0	1,116
BatchNormalization_4	18 × 62 = 1116	72	2232	0
MaxPooling_2	18 × 31 = 558	0	0	1,116
Convolution_5	18 × 31 = 558	990	30690	0
ReLU_5	18 × 31 = 558	0	0	558
BatchNormalization_5	18 × 31 = 558	72	1116	0
Convolution_6	24 × 31 = 744	1320	40920	0
ReLU_6	24 × 31 = 744	0	0	744

BatchNormalization_6	24 × 31 = 744	96	1488	0
MaxPooling_3	24 × 15 = 360	0	0	720
Convolution_7	24 × 15 = 360	1752	26280	0
ReLU_7	24 × 15 = 360	0	0	360
BatchNormalization_7	24 × 15 = 360	96	720	0
Convolution_8	30 × 15 = 450	2190	32850	0
ReLU_8	30 × 15 = 450	0	0	450
BatchNormalization_8	30 × 15 = 450	120	900	0
GlobalAveragePooling	30 × 1 × 1 = 30	0	480	0
Affine	30	930	930	0
ReLU_9	30	0	0	30
BatchNormalization_9	30	120	60	0
Affine_2	5	155	155	0
Total	21,203	9,431	249,629	9,576

Table 4.2: Main characteristics of three possible implementation targets

Target platform	1) SONY Spresense	2) ST Nucleo STM32F469	3) Espressif ESP32-PICO-MINI-02U
Processor type	Arm® Cortex®-M4	Arm® Cortex®-M4	Xtensa dual-core 32-bit LX6
Clock rate	156 or 33 MHz	Up to 180 MHz	Up to 240 MHz
SRAM	1536 Kbytes	384 Kbytes	520 Kbytes
Module size	50 × 20 mm	50 × 20 mm	13 × 11 mm
Power	≈1480 μW / MHz	≈920 μW / MHz	≈900 μW / MHz

It can be seen that target 1) has 1536 Kbytes / 119.6 Kbytes = 12.8 times more memory than necessary, target 2) has 384 Kbytes / 119.6 Kbytes = 3.3 times more memory than necessary, and target 3) has 520 Kbytes / 119.6 Kbytes = 4.3 times more memory than necessary. Then, the model size is not a problem for fitting inside these devices. For all three of them, the physical module size, 50 x 20 mm or 13 x 11 mm, fits easily inside a

collar box. For all three of them, the maximum clock frequency is a lot higher than needed for real-time operating, therefore, this is not a problem. On average, it can be said that the power consumption is (1480 µW / MHz + 920 µW / MHz + 900 µW / MHz) / 3 = 1100 µW / MHz. Then, assuming as a simplification that power scales linearly with clock frequency, 0.06 MHz × 1100 µW / MHz = 66 µW. Actually, this is a very rough approximation as it may be difficult that such low value can be really achieved with these processors, as they are designed to run at higher frequency and this simple calculation ignores many factors. For example, previous works in our lab show the issue of static current consumption and which specified integrated circuits are needed for truly achieving low power in real world [24]. However, this result shows that implementation of the CNN model on a low-power microcontroller is clearly possible.

As discussed in Chapter 6, future work is needed to actually test the model implementing it on microcontrollers like these ones, and measuring the power experimentally. It is important to point out that, in any case, data augmentation influences the training process and therefore changes the numerical values of the trained parameters, however, it has no direct consequence on the number of parameters or number of calculations necessary. In the calculations reported above, it is assumed that single-precision floating point numbers are used (4 bytes) because these are usually precise sufficiently, and half the size of double-precision (8 bytes); that is also because hardware floating point unit present in the target platforms and many edge devices only works for single-precision, not double [25,26]. Quantization into integer values is also a possible way to further reduce the model size and, as a consequence, also reach lower memory requirements and power consumptions, as shown by other works in our lab [24]. In the future, its application to the CNN network should be checked, however, these results already indicate that edge application is possible.

Because this thesis focuses on data augmentation, as explained in Chapter 3.2.2, the data processing for the time being was performed off-line. However, some information is

provided here about the entire system developed by the project that our lab was part of, also referred as "Listening to Silent Voices of Cows with Edge AI Technology". As explained in Ref. [27] and shown in **Fig. 4.3**, according to this project, improved animal welfare management is realized using a solution that uses both edge devices and software running on the cloud. On the edge, each cow to be monitored is fitted with a collar including an accelerometer sensor, GPS receiver, microcontroller and low-power wide area (LPWA) wireless radio. Information about the current behavior, the prevalence of the behaviors, as well as the GPS coordinates of each freely grazing cow can be sent to the cloud by the edge devices. On the cloud, named PETER, the data are stored and processed at higher level, and the available functions can include real-time data, history data, animal welfare index calculations, release of open data, tools to calculate the economic value of each cow, as well as company asset data. These data are made available to the farmer or other user with need to know about animal welfare, and shared with other service providers. Because the development of the PETER cloud is not part of this thesis, more information can be found in Ref. [27]. According to the project, the PETER Edge (Edge AI) could be connected to the cloud using two technologies, that is, SigFox and ELTRES. SigFox is developed by SigFox SA in France, and uses the Industrial, Scientific and Medical radio band to provide long-range communication based on a service provider [28]. ELTRES is developed by SONY Semiconductor Solutions Corporation in Japan, and uses a similar radio band, it is upload-only and can provide extremely long distances of up to 100 km [29].

In the case of SigFox, the packet size is maximum 12 bytes. In the case of ELTRES, it is 128 bits. To support low power operation and meet other requirements of band use, a packet can normally be sent once every 10 to 30 minutes; during such period, 25 Hz × 2 bytes × 3 channels × 600 or 1800 s = 88 to 264 Kbytes of time series data are produced by the accelerometer, so, sending the raw time series is totally impossible. What is sent, instead,

is the information about the prevalence of each behavior. This means that a counter is provided for each classified behavior type, and, every time a time window is analyzed, it is incremented according to the recognized behavior. For example, considering 30 minutes, there are 1800 s × 25 Hz / 125 samples/window = 360 classifications. Because it must be possible to count always the same behavior, this means that $\log_2(360) = 8.5$ bits, that is 9 bits, are needed for the counter of each behavior. Then, for 5 behaviors, this corresponds to a pay load of 9 bits × 5 behaviors = 45 bits. This shows an example of how the classifier can be used to reduce a lot the amount of data to be sent to the cloud, realizing the data reduction necessary for many IoT devices and introduced and discussed in Chapter 1 [30-32]. In general, for PLF, it is not needed to send each window's classification output separately, because the information useful to animal welfare management is contained in the frequencies of the behaviors [33,34]. However, it must be considered that with LPWA, packet delivery is not guaranteed, and data loss happens frequently. Therefore, to reduce the risk of big gaps in the recordings, each time not only the data from the corresponding time period should be sent, but also a moving average of the previous time windows can be sent. Alternatively, the data from the previous time window can be also sent again. In both cases, the amount of data actually transmitted would be twice, so, 45 bits × 2 = 90 bits, which is still below the limit of 128 bits for ELTRES or 12 bytes, that is, 96 bits for SigFox.

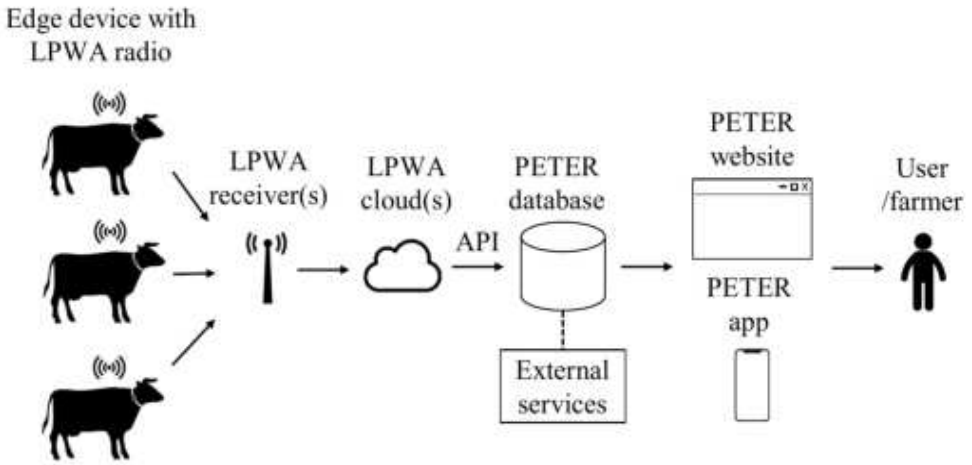


Figure 4.3: Overview of the cow monitoring system. Depending on the realization, LPWA in practice means ELTRES, SigFox or can even be a combination of both.

4.4 Challenges in the behavioral data

As in the previous chapter, I consider five frequent cattle behaviors. These are the same as the previous chapter, however, with one difference. In the remainder of this thesis, the behaviors under consideration are feeding, walking, salting, ruminating, and resting. That is, drinking was dropped, while salting was added. The principal motivation for this change is related to pragmatic considerations: in the experimental settings that were reasonably accessible, drinking was too rare to be meaningfully represented. On the other hand, salting, which was previously considered less relevant, was more easily observable. In the dataset considered for this and the next chapter, the number of data rows for drinking was approximately 75% smaller compared to salting. This difference was due to several factors together, mostly related to the recording arrangements and situations, which kept evolving as the overall research project progressed. Therefore, the total number of behaviors was kept the same, while replacing one for the other. There should be no misunderstanding that both

behaviors are important and have physiological implications, including drinking. However, it can be said that out of all behaviors probably feeding is the most important one, because when cows develop disease, the eating habits change easily [6,7,8]. The situation represents a compromise under the circumstances, which is quite typical for the development process of systems such as the present one. The results provided in this and the in next chapter would, probably, equally well apply to drinking.

Altogether, the datasets collected contain 530,485 data rows sampled at a rate of 25 Hz, i.e., the data are approximately 5.89 h long: this is, actually, relatively tiny compared to other situation of the same kind. By comparison, in Ref. [35], approximately 63 h in total were captured, and a CNN model with long short-term memory (CNN-LSTM) was used to detect five basic behaviors (drinking, ruminating, walking, standing, and lying). In Ref. [36], the active video was approximately 68 h long and applied to an LSTM model. In Ref. [37], the video was approximately 32 h long, and random rotation (ROT)-based augmentation was proposed to augment the data and address the imbalanced learning problem. By comparison, the dataset considered in the previous chapter [37] contained 2,935,571 data rows; however, these originated from two cows only.

It is important to underline that, while less than optimal, the situation considered in this chapter is representative of the developments in this field.

4.5 Proposed data augmentation methods

As discussed in Chapters 1 to 3, data augmentation can be conceived as an injection of prior information about data attributes that are invariant against certain transformations. Augmented data attempt to cover the unknown and unfamiliar spaces of input patterns, thereby alleviating the effect of the overfitting problem and improving the generalization ability of trained NN models. Minor changes in image data, such as scaling and rotating,

are known to have negligible impacts on data labels because such changes are likely to occur in real-world observations [38,39]. However, for inertial sensor data, label-preserving transformations are not immediately apparent and intuitively recognizable. This situation, however, must be considered together with the fact that the time-domain signals are input to the CNN. The implication is that some aspects such as shifting and various basic time-domain alterations should have no impact on the classification. This is the basis on which the developments put forward in this chapter are based.

As was already explained in Chapter 3, one aspect that may introduce label-invariant variability into inertial sensor data involves sensor position differences during monitoring. Therefore, different sensor device rotation states are simulated, and this type of rotation is considered an augmentation approach to cover additional data possibilities while maintaining the correct labels [40]. After the theoretical description provided in Chapter 3, given an original time series x, the transformation formula for rotation (ROT) is shown in the following equations

$$x = x_1, x_2, \dots, x_t, \dots, x_T, \tag{4-1}$$

$$x' = Rx_1, Rx_2, \dots, Rx_t, \dots, Rx_T, \tag{4-2}$$

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \tag{4-3}$$

where $x_t = \left[a_{x_t}, a_{y_t}, a_{z_t}\right]^T$, indicating a 3-axis data point at each time step t, T is the total number of time steps, and R is a 3-dimensional ROT matrix [41]. While the mathematics is the same as the previous chapter, here, unlike the 30-degree ROT interval, a random angle θ distributed uniformly in the range of 0 to 360 degrees is applied. It was found that for the dataset considered in this chapter, the discretization limited the effectiveness of this augmentation step. Probably, this difference is down to the fact that the amount of data per cow

was smaller, and therefore the variance already contained in the initial data was more limited by comparison with Chapter 3.

For the reasons introduced above, perturbing the time position of the activity in the input window is also considered another approach that can introduce label-invariant variability. Considering that cow movements, e.g., rumination, resting, and lying, are mostly repetitive, as well as the fact that window segmentation is arbitrary, the time positions of the activities in the input window may not convey useful information. Therefore, we may augment data by perturbing the time positions in the window. At this point, two remarks appear useful. One is that this situation arises because the time-domain data are entered in the CNN directly. Considering the more traditional MLP-based approaches, features would be extracted using parameters that are based on the Fourier amplitudes, for example, which are by themselves invariant to time and phase shifts. With this remark, one should feel that, as always, there is a compromise in choosing to give up a feature extraction based on known principles and letting the CNN do it. Fewer assumptions are needed, the result may be better, but also more fragile because of possibly limited generalization [15]. That is one of the reasons why data augmentation is so important. Another remark is that, as will become clearer in the next chapter, actually this operation has a deeper meaning from a theoretical standpoint, because it effectively means that the classifier should learn primarily features that are related to the autocorrelation and cross-correlation and variance of the signals, as observed in other applications [42]. The temporal content in itself becomes irrelevant. While for situations such as cow behavior monitoring this is obvious, since the timings do not have any form of synchronization to the behavior, it is important to recognize that it is not always so. One may think of time-locked averaging, such as event-related potential in brain activity, wherein the actual time of occurrence of signal change actually has an important meaning [43].

On another level, recombining two sequences at a random position (REC) is a simple method to randomly perturb the time positions of two or more sequences. To perturb the time positions of the data between windows, we first randomly select two samples, slice the data with a ratio of N: (1-N), with N ranging from 0 to 1, and recombine (that is, concatenate) the segments to create a new sample. This operation is rather simple and crude, and is inspired by the situation using genetics data for example, wherein two sequences may be cut and recombined in a random way [44]. Regarding time series, this operation actually is unphysical, because it introduces a sharp discontinuity, therefore, in principle, it may seem undesirable. That's because in a real-world movement, a point where the first derivative becomes infinitely large cannot exist [45]. However, on a more practical level, this method actually ends up to be useful. The reason is that, because of the filtering action of the convolution layers in the CNN, the effect of the discontinuities is actually negligible [15]. This operation brings into the data a situation of non-stationarity, which means, a situation wherein the properties of a time series change over time. However, that is the norm with animal behavior, particularly because animals often alternate two or more behaviors, even with relatively high frequency; one may consider, for example, feeding and ruminating. It is necessary to make sure that, when a given time window (125 points, or 5 seconds in this study) covers the transition between two behaviors, the classifier will output, ideally, their relative prevalences. If not, it should select the most prevalent behavior, or at least one of those two behaviors that got mixed. But in the absence of a sufficiently representative set of these transitions, it is possible that the classifier may completely misinterpret the data: examples of this situation are well known in the area of computer vision [38,39]. Therefore, creating these artificial recombinations, even though not natural, can be very helpful to ensure that the classifier at least behaves in a sane manner, that is, avoids misclassification into a class that is unrelated to the behaviors that got mixed.

Time reversal (REV) is another method used to perturb the time positions. Assuming that the network learns data features that are time-invariant, a potential strategy is to reverse the samples along the time axes and generate new time series. REV can be defined as follows:

$$x' = x_T, \ldots, x_t, \ldots, x_2, x_1 \tag{4-4}$$

Time does not flow backwards, and so cows do not walk or feed backwards. Nevertheless, the important aspect is that practically all of the behaviors of interest show important periodicity. That is, the actions that cause accelerations are repeated several times, at a characteristic and more or less regular pace. As will be made clearer in the next chapter, however, there is an important aspect to consider. If the element of the movement carrying information is just the period, or the pace, then reversing time will have no effect, by definition. On the other hand, if there are more complex relationships, which are not linear and are corrupted by the reversal of time, then the operation may corrupt the results [46]. One way of thinking about this is playing backwards a short movie of a person walking backwards: will it look the same as a person walking forward? Probably not exactly the same, however, enough similar for the action to be recognizable. In other words, insofar as what matters are the period and pace as represented by the auto- and the cross-correlation, then time-reversal can produce time series that are close enough to realistic ones to be useful, while being different in their numerical content.

Finally, there is a practical aspect which leads to another useful manipulation. Because the transitions between behaviors occur at unpredictable times, the lengths of the time series snippets are irregular. Here, snippet means a segment of time series that is comprised between two behavioral transitions. This implies that many of them may not meet the imposed window size requirement, leading to the wastage of a considerable amount of data. This may occur either because a snippet length is shorter than the window length, so the entire snippet is rejected, or because a snippet length is not an integer multiple of the window

length, leading to rejection of some data towards the end. Even if one sample is missing, all data is lost. In a situation where the dataset size is rather small to start with, it is easy to see that such an approach is not optimal [47]. Under the assumption of repetitive cattle movements, a method to compensate for information loss (CIL) is proposed to fill the insufficient part and augment data by looping the existing period until the length requirement is reached. When looped twice, the augmented time series x' becomes:

$$x' = x_1, \dots, x_t, \dots, x_E, x_1, \dots, x_t, \dots, x_E \tag{4-5}$$

where E is the number of data points in an insufficient sequence; this can be easily repeated for more loops. It can be said that, out of the proposed methods, this is the one for which the theoretical rationale is less strong, because the location at which the repetition takes place is arbitrary. Therefore, it has no relationship to the period of the cattle behavior. Nevertheless, as the results reported in the next subsection show, also this method could help considerably to improve the performance.

In summary, rotating, reversal, recombining two sequences at a random ratio, and the compensatory approach (**Fig. 4.4**) are applied to augment inertial sensor data. The performance of cattle behavior classification achieved when using CNNs in conjunction with the proposed data augmentation approaches is evaluated in the next section.

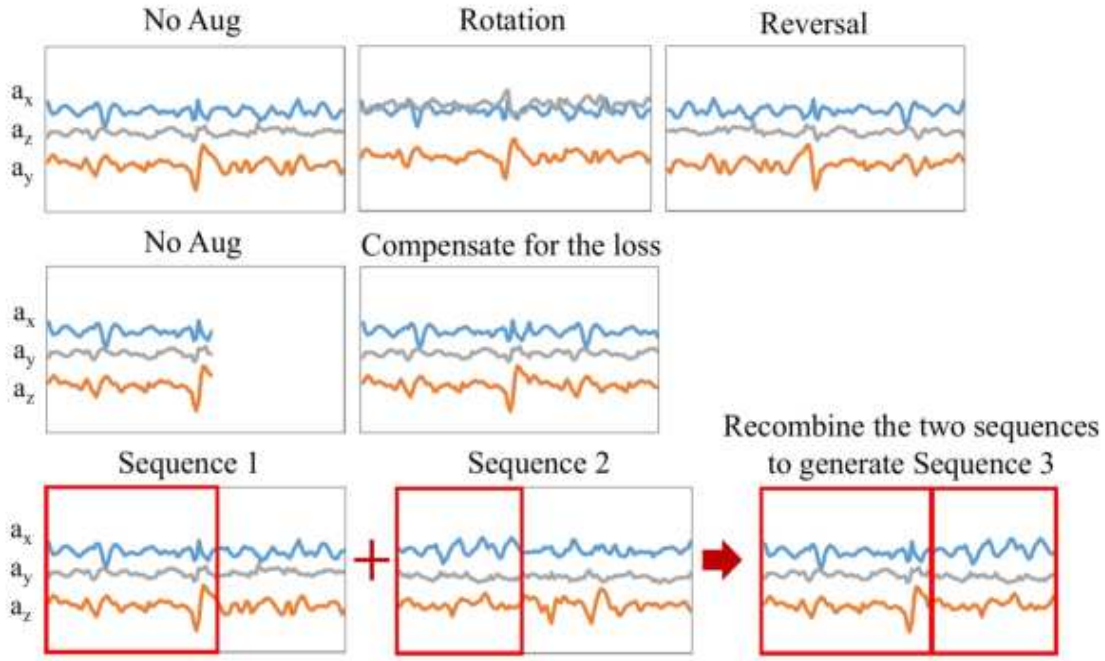


Figure 4.4: Examples of time series generated in a 5 s window with the proposed data augmentation methods: rotating, reversal, compensation for loss, and the recombination of two sequences. Note that the proposed compensatory approach compensates for data loss by looping the existing period. A combination of multiple augmentation methods can also be applied.

4.6 Experimental results

4.6.1 Workflow

As detailed above, cattle behavior classification was performed using a CNN-based behavioral model combined with various data augmentation approaches. All numerical experiments were conducted using 10-second random sliding windows for 1500 epochs. The number of training instances in every scenario shown in **Fig. 4.5** is the same to avoid any

potential source of bias. For the baseline result, the CNN is applied directly to the raw acceleration data without augmentation.

Different random parameter values were applied in the experiment. For ROT, a random ROT matrix was created for every input sequence. For REV, whether to apply or not the method to a given instance was decided randomly with a 50% probability. For the recombination of mixed patterns (REC, **Fig. 4.6**), the recombination ratio was determined as a random number uniformly distributed within the range from 0 to 1. For the compensatory method, only the input sequences with a window possessing a size less than 10 seconds were treated. Notably, the compensatory method augments data by complementing the original data that would otherwise be discarded, while ROT, REV, and REC generate new data by transforming the original training data.

Because the augmentation operations are based on different assumptions and ideas, there is no reason for which they could not be combined. Therefore, in this work, I not only review the accuracy changes induced by every data augmentation approach but I also evaluate the combined augmentation results of the other three data augmentation methods after compensating for the loss. The workflow is shown in **Fig. 4.5**. In such a situation, the danger is to deviate excessively from the real-world data and end up lowering the final performance rather than enhancing it. As the results shown below indicate, this was surely not the case.

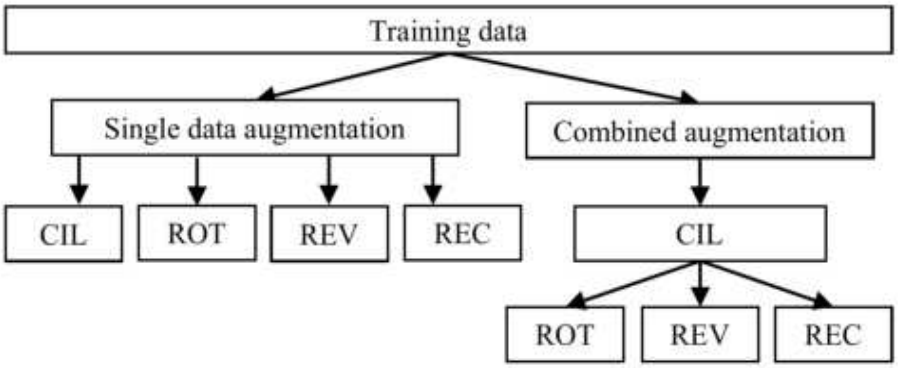


Figure 4.5: Various augmentation scenarios used in this study.

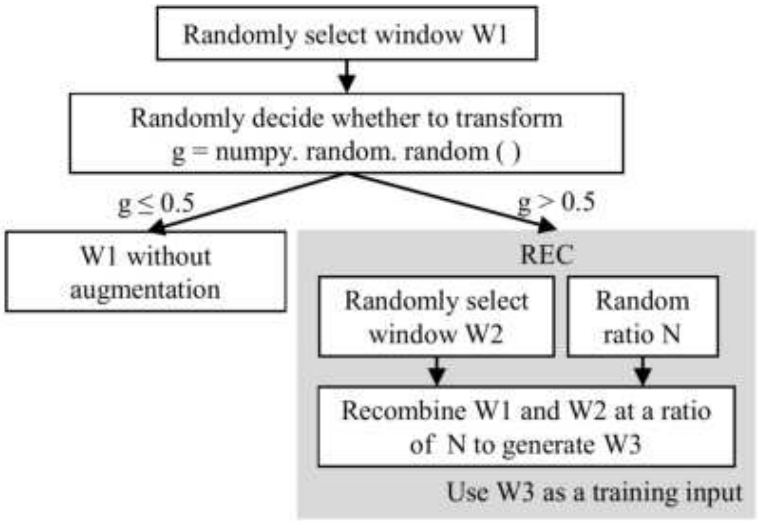


Figure 4.6: Flow of proposed REC augmentation used in this study.

4.6.2 Results

The main results are presented in **Fig. 4.7**, where the numbers in the figure show the average F_1 score value for each scenario. Because of the presence of multiple forms of data manipulations and their combinations, to avoid an excessive number of results, making them difficult to interpret, here I focus only on the F_1 score. Furthermore, the average across

classes is considered: the rationale for this choice is that there is no trivial correspondence between how frequently a behavior is present, and its importance. Weighting more heavily the most frequent behaviors corresponds to assuming that their importance is greater than the least frequent ones, but it is not necessarily so. Therefore, in the absence of other criteria, I argue that the preferred approach is simply to take the average.

Considering the results in **Fig. 4.7**, it is immediately evident that even a single manipulation can substantially increase the average F_1 score. All of the observed single increases across REC, ROT, REV and CIL were significant in terms of application impact, ranging from 4.5% to >6%. It is noteworthy that the relative differences between the behaviors were only partly preserved, and, for example, CIL led to a significant drop in the recognition of feeding behavior.

The results were even more promising when combining two approaches, ranging from 89% for CIL with REC, to 94% for CIL with REV. The corresponding performance increase, more than 10%, is by any standard markable when considered with respect to previous studies in this area. It needs to be pointed out that, for this combination, the performance was evidently higher than the baseline for all behaviors. In other words, the detrimental effect of applying CIL alone for the recognition of feeding was more than compensated by combining it with REV. The numerical values of the performance are also given in **Table 4.3**. The slightly lower F_1 score for Ruminating may be related to the fact that it can occur in different postures, such as standing and laying down. So, it can be said that it in part overlaps Walking and Resting.

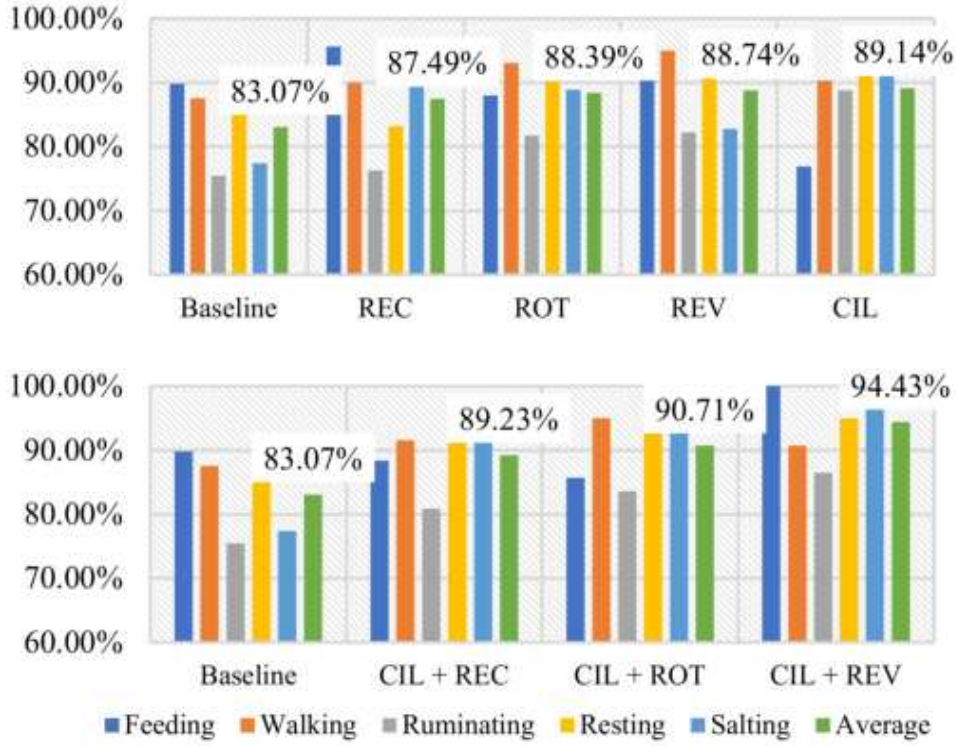


Figure 4.7: The results of cattle activity classification obtained with various data augmentation methods; the top and bottom panels correspond, respectively, to single- and double-data augmentation. The numbers in the figure show the average F_1 score value for each scenario.

Table 4.3: Classification performance on the test dataset with CIL + REV.

	Precision	Recall	F_1 score
Feeding	100.00%	100.00%	100.00%
Walking	87.50%	94.23%	90.74%
Ruminating	98.77%	76.92%	86.49%
Resting	90.86%	99.38%	94.93%
Salting	100.00%	100.00%	100.00%

Table 4.4: Comparison of study parameters and accuracy. Superscript * next to the year indicated an as-yet unpublished study.

Ref.	Year	Behaviors	Data size	Sampling rate (Hz)	Window size	Model type	CNN layers	Augm.	Accuracy
[48]	2016	2	862,500	10	60	CNN	1	No	84.0%
[49]	2018	3	530,000	Unknown	Unknown	CNN	Unknown	No	91.0%
[36]	2019	8	364,544	20	64	CNN	2	No	84.6%
[50]	2021*	3 or 4	187,937	50	160	CNN	3 9	No	83.3% 84.5%
[51]	2021	3	124,560,000	10	900	CNN	4	No	83.0%
[52]	2022*	5	211,720	5	50	CNN	3	No	94.0%
[53]	2022*	3	72,810,000	25	2,250	CNN	2	Yes	93.0%
[54]	2022*	5	124,560,000	10	900	CNN	4	Yes	73.0%
[55]	2022*	5	15,300,000	50	150	CNN	3	No	92.8%
[56]	2022	5	80,319	4	20	CNN	5	No	90.0%
[57]	2023	7	140,000	10	128	CNN	3	No	83.8%
This work	2022	5	530,485	25	125	CNN	8	Yes	94.4%

Prior to ending this chapter, it is useful to present a systematic overview of the present literature and attempt some comparison with it. **Table 4.4** shows the relevant works that were identified during a literature search performed in January 2023. Only studies based on CNN networks are included. Results obtained using simpler classifiers such as multi-layer perceptrons and decision trees, or other deep networks like LSTM, are not reported: those classifiers are so different that a fair comparison would be difficult. To be specific, all works on CNN that I could find were included, regardless of whether they were already published as journal papers, or still preprints or theses. First of all, it can be seen that most CNN works are very recent: 1 is from 2023, 5 are from 2022, 2 are from 2021 and 3 before that year. There is a great variability in all aspects. The number of behaviors to be classified ranges from 2 to 8; the median is 5, which is the same as in this chapter. The data size, intended as the number of data rows, ranges from about 80,000 to more than 100 million; the median is

about 530,000, which is almost exactly the same as in this chapter. The sampling rate goes from 4 Hz to 50 Hz, median 10 Hz, which is 2.5 times lower than in my research. The window size is minimum 20 and maximum 2,250 points, median 139, which is similar to here. The number of convolutional layers spans from 1 to 9, median 3, which is less than half of the network proposed. It should be stated that the parameters in this table were derived as accurately as possible from the references, however, not all data are always given clearly. It can be said that the wide variability in all aspects, together with the fact that most papers were published last year, indicates that this is a field that is still emerging.

Because of all these differences, making a comparison of accuracy is difficult to be done fairly. Attempting to always calculate the overall accuracy from the papers, some approximate comparisons could anyway be attempted. It can be seen that the accuracy reported goes from 73% to 94%, and the median is 84.5%, which is considerably lower than the best case of 94.4% obtained in this Chapter. In fact, the present results are better than all other 11 references in the table. Even considering the possible risk of unfair comparison, for example due to different dataset quality, it can be said that, overall, the present methods seem to allow a good position with respect to the other existing works, including the very recent ones. In addition, the situation is even more positive for the results given in the next chapter, reaching 96%. To be precise, with respect to the 11 references cited, one difference is also the network depth, meaning, the number of convolutional layers: here, it was 8, which is higher than the median 3 in the existing papers. Therefore, the better performance may partially be due to this. However, it is important to clarify that there is a sure effect of data augmentation. The deeper CNN used in this and the next chapter, when trained without augmentation, reached a best accuracy of 83% and 90%, respectively, depending on the labelling. Applying data augmentation to the same network enabled increasing 83% to 94%

in this chapter, and 90% to 96% in the next chapter. Therefore, it is clear that data augmentation in this and the next chapter was essential to exceed the performances of the 11 cited references.

Nevertheless, there remains the general question of why such a large variability in the achieved accuracy is observed between the studies in **Table 4.4**. One possibility to try and understand the variability in terms of the other study parameters is using a correlation approach, that is, search for correlation between each parameter and the achieved accuracy [58]. To avoid an unfair situation, the two studies with data augmentation, Refs. [53] and [54], need to be excluded from this analysis. Because the parameter values are quite scattered, and some of them are integer numbers such as the number of behaviors, a rank-order calculation known as Spearman correlation should be applied [58]. Then, first of all considering the number of behaviors, the correlation coefficient obtained is $r = 0.30$. The p-value is $p = 0.4$, therefore, the correlation is not statistically significant because $p > 0.05$, so no final conclusions can be drawn. However, because the correlation coefficient is positive anyway, it seems that a large number of behaviors to classify does not decrease the accuracy. On the contrary, a positive correlation may be present. This is surprising, but one hypothesis is that it could be because the studies with more behaviors generally tend to be of higher quality. Next, the data size can be considered, which gives $r = 0$ and $p = 1$. In this case, there is totally absent correlation, and therefore, data size also cannot explain the variability, which is also quite surprising considering the importance of data for neural network training, as explained in Chapters 1 and 2. However, also in this case, the determining factors are likely to be hidden behind the numbers, meaning, data quality rather than quantity may be the key. For example, the huge dataset used in Refs. [51] and [54] was not obtained with careful labelling by human experts. Because of the importance of data quality, in the course of the present research, between Chapter 3 and Chapters 4 and 5 it was decided to switch from a larger data to a smaller but higher-quality one. As it is impossible to judge

the labelling quality without inspecting the raw data, this hypothesis can only remain a speculation. Then, considering the sampling rate next, $r = -0.21$ and $p = 0.6$ are obtained. The effect is not significant, but there seems to be a weak trend that higher sampling rate leads to worse accuracy. One possibility is that this is because above the median value of 10 Hz, since most power is found < 1 Hz, for faster sampling the data do not contain meaningful additional information but actually complicate the training process because the input sequence to the CNN network is bigger. A surprisingly strong effect is observed for window size, because $r = -0.66$ and $p = 0.06$, therefore the effect is almost statistically significant. The negative correlation seems to again imply that more data points in each window reduce the accuracy. To attempt understanding the reason, it is useful to consider the number of points divided by sampling rate, giving the window length in seconds, which is median 5 seconds. In that case, still $r = -0.39$, which is a moderately large value. One explanation could also be because longer windows have a higher probability of including a mixture of different behaviors. While this should not be relevant if labelling is done properly, one wonders about the possible influence in situations of lower quality labelling. Last, the effect of the number of layers should be considered, giving $r = -0.03$, $p = 0.9$, which means there is not any relationship. Therefore, in the end, unfortunately these analyses do not provide a clear explanation for the variability. The most likely answer is that there are other factors, hidden behind these parameters, which have an impact. These factors may be related to the quality of the data acquisition and the quality of the labelling, which are difficult to quantify. There is also the possibility that different settings in the network training parameters which are not consistently reported in these studies, such as batch size, learning rate and so on, have an influence. For example, Ref. [54] and Ref. [56] both aim to classify 5 behaviors, the data size is about ~1500 times larger for Ref. [54], but the accuracy, 73%, is much lower than 90% in Ref. [56]. It should be considered that Ref. [54] is a master's thesis, so, there is a possibility that the methods are not applied with the same skill and care as a published

paper. In the same line of thinking, for example, Ref. [52] achieves 94%, Ref. [53] achieves 93% and Ref. [55] achieves 92.8% accuracy. While these values are very similar, the number of behaviors, data size, sampling rate and window size are quite different between these studies. It is clear that unaccounted factors are happening.

Besides the above, it is meaningful to discuss in further detail the only two other studies that have used data augmentation. One of them, Ref. [54], uses an extremely large public dataset of more than 100 million data rows, but reaches a very modest test set accuracy of 73%, the lowest of all. A paper published from the same dataset but with several differences such as the number of behaviors, Ref. [51], attained 83% without data augmentation. The data augmentation techniques used were based on a generic library (Tsaug library:https://tsaug.readthedocs.io/en/stable/references.html) including randomly permuting the axes, adding noise, quantizing, scaling, time wrapping and other similar operations. In Chapter 2, it was indicated that operations such as these ones, especially scaling, may corrupt the labels. Therefore, it cannot be excluded that data augmentation not only did not improve but may even have lowered the accuracy in this study. However, this is impossible to establish since the accuracy without data augmentation is not reported. Furthermore, because it is a master's thesis, it was not peer reviewed and the possibility of mistakes cannot be excluded. The other paper using data augmentation, Ref. [53], attained a much higher accuracy, close to the present report, that is, 93%. In fact, the data augmentation technique used in that work is exactly the random rotation introduced in my paper associated with this chapter, which the authors clearly cite and indicate. Importantly, the original dataset size, about 73 million points, is very much higher than the present dataset, which indirectly suggests that the method is probably also helpful on different and bigger datasets. In addition to the above, it should be noted that some studies have attained high accuracy without data augmentation despite a small data size, for example, Ref. [56]. As discussed, it seems not possible at this stage to find a conclusive answer to individual differences such as this one,

but factors related in data acquisition and labeling and not captured by the numerical parameters are likely to be the key. From this view point, having an internal comparison between with and without data augmentation on the same datasets and classifiers as done in all chapters of the present thesis, appears very important to allow properly demonstrating the actual influence of data augmentation. As said above, the most relevant conclusion that can be drawn from the table is at the overall level, based on the median: the parameters of the present study are close to the median, so it is quite "typical". Without data augmentation, the accuracy attained in this chapter is 83%, which is very close to the median across the studies 84.5%, however, introducing data augmentation, it is considerably higher, 94%.

It should be underlined again that one problem with the field is that most papers provide very limited details about the classifier and the dataset is generally not freely available. Therefore, accurate and reliable comparisons are still difficult to do as of today. In the future, as the field become mature, some standard datasets and classifiers are likely to appear as was the case for some other challenging areas such as computer vision. For the time being, given these results, it seems possible to state that, at a general level, the methods introduced in this chapter compare positively with the literature. That is also suggested by the fact that the present work has already been cited and used by another research group independently.

4.7 Conclusion

This chapter introduced a number of advancements with respect to Chapter 3. First, the application scenario was made to be a more representative one of developments in this area, with a large number of cows and fewer data per cow, leading to a more variable and therefore challenging dataset. Second, the LSTM network was replaced with a more compact CNN network, more in line with current trends towards this type of classifier, owing to its compactness and ease of training. Third, the rotation operation was improved by removing

the discretization in large steps and allowing a continuous random angle. Fourth, and most importantly, the rotation operation was complemented by three others, namely recombination, reversal and compensation for information loss, substantially boosting the advantage of data augmentation.

Two additional considerations are useful at this point. One is that rotation was actually not the best performing manipulation for augmentation, since reversal and compensation for information loss performed better. The other is that compensation for information loss combined with reversal performed best. These findings remark the empirical nature of data augmentation. The operation that was physically best motivated, meaning rotation, could eventually be disregarded, while the one that had the weakest theoretical underpinning, meaning compensation for information loss, performed the best. One aspect of interest, in this sense, is that recombination, reversal and compensation for information loss are not in any way system-specific. This last statement leads towards the next chapter, where it will be shown that the best performance in data augmentation can actually be obtained via the methods that are the most abstract and free from any system-specific considerations, such as rotation around the cow's neck.

4.8 Bibliography

[1] J. Chin, V. Callaghan, and S. B. Allouch, "The internet-of-things: Reflections on the past, present and future from a user-centered and smart environment perspective," Journal of Ambient Intelligence and Smart Environments, vol. 11, no. 1, pp. 45–69, 2019.

[2] M. Merenda, C. Porcaro, and D. Iero, "Edge machine learning for AI-enabled IoT devices: A review," Sensors, vol. 20, no. 9, p. 2533, 2020.

[3] Y. Roh, G. Heo, and S. E. Whang, "A survey on data collection for machine learning: a big data-ai integration perspective," IEEE Transactions on Knowledge and Data Engineering, vol. 33, no. 4, pp. 1328–1347, 2019.

[4] R. Cowie, C. Cox, J.-C. Martin, A. Batliner, D. Heylen, and K. Karpouzis, "Issues in data labelling," in Emotion-oriented systems. Springer, 2011, pp. 213–241.

[5] H. He and E. A. Garcia, "Learning from imbalanced data," IEEE Transactions on knowledge and data engineering, vol. 21, no. 9, pp. 1263–1284, 2009.

[6] D. N. Ledgerwood, C. Winckler, and C. B. Tucker, "Evaluation of data loggers, sampling intervals, and editing techniques for measuring the lying behavior of dairy cattle," J. Dairy Sci., vol. 93, no. 11, pp. 5129–5139, Nov. 2010.

[7] S. Kuankid, T. Rattanawong, and A. Aurasopon, "Classification of the cattle's behaviors by using accelerometer data with simple behavioral technique," in Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific, 2014, pp. 1–4.

[8] T. M. Hill et al., "Evaluation of an ear-attached movement sensor to record rumination, eating, and activity behaviors in 1-month-old calves," Prof. Anim. Sci., vol. 33, no. 6, pp. 743–747, 2017.

[9] G. A. Susto, A. Cenedese, and M. Terzi, "Time series classification methods: Review and applications to power systems data," Big data application in power systems, pp. 179–220, 2018.

[10] A. Abanda, U. Mori, and J. A. Lozano, "A review on distance based time series classification," Data Mining and Knowledge Discovery, vol. 33, no. 2, pp. 378–412, 2019.

[11] B. D. Fulcher, "Feature-based time series analysis," arXiv preprint arXiv:1709.08055, 2017..

[12] M. Barandas, D. Folgado, L. Fernandes, S. Santos, M. Abreu, P. Bota, H. Liu, T. Schultz, and H. Gamboa, "Tsfel: Time series feature extraction library," SoftwareX, vol. 11, p. 100456, 2020.

[13] G. Ciaburro and G. Iannace, "Machine learning-based algorithms to knowledge extraction from time series data: A review," Data, vol. 6, no. 6, p. 55, 2021.

[14] T. Henderson and B. D. Fulcher, "An empirical evaluation of time series feature sets," in 2021 International Conference on Data Mining Workshops (ICDMW). IEEE, 2021, pp. 1032–1038.

[15] B. Zhao, H. Lu, S. Chen, J. Liu, and D. Wu, "Convolutional neural networks for time series classification," Journal of Systems Engineering and Electronics, vol. 28, no. 1, pp. 162–169, 2017.

[16] C.-L. Liu, W.-H. Hsaio and Y.-C. Tu, "Time series classification with multivariate convolutional neural network", IEEE Transactions on Industrial Electronics, vol. 66, no. 6, pp. 4788-4797, 2019.

[17] "Neural Network Console by Sony Network Communications Inc." [Online]. Available at: https://dl.sony.com/. Accessed on Dec 20, 2022.

[18] "TF 2.0 Feature: Flops calculation · Issue #32809." [Online]. Available at: https://github.com/tensorflow/tensorflow/issues/32809. Accessed on Jan 5, 2023.

[19] "How to determine needed memory of Keras model?" [Online]. Available at: https://stackoverflow.com/questions/43137288/how-to-determine-needed-memory-of-keras-model. Accessed on Jan 4, 2023.

[20] "Spresense 6-core microcontroller board with ultra-low power consumption by Sony Group Corporation." [Online]. Available at: https://developer.sony.com/develop/spresense/specifications. Accessed on Jan 4, 2022.

[21] “STM32F469/479 product specifications by STMicroelectronics NV.” [Online]. Available at: https://www.st.com/en/microcontrollers-microprocessors/stm32f469-479.html. Accessed on Jan 4, 2022.

[22] “ESP32-PICO-MINI-02U Datasheet.” [Online]. Available at: https://www.espressif.com/sites/default/files/documentation/esp32-pico-mini-02_datasheet_en.pdf Accessed on Jan 4, 2022.

[23] “SPRESENSE メインボードとカメラの消費電力を測定してみた” [Online]. Available at: https://makers-with-myson.blog.ss-blog.jp/2020-02-21 Accessed on Jan 4, 2022.

[24] J. Bartels et al., “TinyCowNet: Memory- and Power-Minimized RNNs Implementable on Tiny Edge Devices for Lifelong Cow Behavior Distribution Estimation," in IEEE Access, vol. 10, pp. 32706-32727, 2022.

[25] T. Hrycej, B. Bermeitinger, and S. Handschuh, “Training neural networks in single vs double precision,” arXiv preprint arXiv:2209.07219, 2022.

[26] “Cortex-M4 Technical Reference Manual FPU instruction set” [Online]. Available at: https://developer.arm.com/documentation/ddi0439/b/BEHJADED. Accessed on Jan 4, 2022.

[27] H.Ito, K. K. Tokgoz, L. Minati, C. Li, T. Ohashi, and K.-i. Takeda, “Listening to Silent Voices of Cows with Edge AI Technology,” in Annual Meeting Record, I.E.E. Japan, 2022.

[28] “What is Sigfox 0G technology?” [Online]. Available at: https://build.sigfox.com/sigfox. Accessed on Jan 4, 2022.

[29] “Core Technologies.” [Online]. Available at: https://www.sony-semicon.com/en/eltres/technology.html. Accessed on Jan 4, 2022.

[30] L. Pioli, C. F. Dorneles, D. D. de Macedo, and M. A. Dantas, "An overview of data reduction solutions at the edge of iot systems: a systematic mapping of the literature," Computing, pp. 1–23, 2022.

[31] J. D. A. Correa, A. S. R. Pinto, and C. Montez, "Lossy data compression for IoT sensors: A review," Internet of Things, p. 100516, 2022.

[32] M. Merenda, C. Porcaro, and D. Iero, "Edge machine learning for AI-enabled IoT devices: A review," Sensors, vol. 20, no. 9, p. 2533, 2020.

[33] I. Dittrich, M. Gertz, and J. Krieter, "Alterations in sick dairy cows' daily behavioural patterns," Heliyon, vol. 5, no. 11, p. e02902, 2019.

[34] R. Lardy, M. M. Mialon, N. Wagner, Y. Gaudron, B. Meunier, K. H. Sloth, D. Ledoux, M. Silberberg, A. d. B. des Roches, Q. Ruin et al., "Understanding anomalies in animal behaviour: data on cow activity in relation to health and welfare," AnimalOpen Space, vol. 1, no. 1, p.100004, 2022.

[35] D. Wu, Y. Wang, M. Han, L. Song, Y. Shang, X. Zhang, and H. Song, "Using a cnn-lstm for basic behaviors detection of a single dairy cow in a complex environment," Computers and Electronics in Agriculture, vol. 182, p. 106016, 2021.

[36] Y. Peng, N. Kondo, T. Fujiura, T. Suzuki, H. Yoshioka, E. Itoyama et al., "Classification of multiple cattle behavior patterns using a recurrent neural network with long short-term memory and inertial measurement units," Computers and Electronics in Agriculture, vol. 157, pp. 247–253, 2019.

[37] C. Li, K. K. Tokgoz, A. Okumura, J. Bartels, K. Toda, H. Matsushima, T. Ohashi, K.-i. Takeda, and H. Ito, "Data augmentation for cow behavior estimation systems based on neural network technology," in International Workshop on Smart Info-Media Systems in Asia, 2020.

[38] J. Schmidhuber, "Deep learning in neural networks: An overview," Neural networks, vol. 61, pp. 85–117, 2015.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of IEEE conference on computer vision and pattern recognition, pp.770–778, 2016.

[40] H. J. Williams, M. D. Holton, E. L. Shepard, N. Largey, B. Norman, P. G. Ryan, O. Duriez, M. Scantlebury, F. Quintana, E. A. Magowan et al., "Identification of animal movement patterns using tri-axial magnetometry," Movement ecology, vol. 5, no. 1, pp. 1–14, 2017.

[41] C. A. León, J. C. Massé, and L. P. Rivest, "A statistical model for random rotations," Journal of Multivariate Analysis, vol. 97, no. 2, pp. 412–430, 2006.

[42] S. Dey, S. S. Roy, K. Samanta, S. Modak and S. Chatterjee, "Autocorrelation Based Feature Extraction for Bearing Fault Detection in Induction Motors," 2019 International Conference on Electrical, Electronics and Computer Engineering (UPCON), 2019, pp. 1-5.

[43] A. Craik, Y. He, and J. L. Contreras-Vidal, "Deep learning for electroencephalogram (eeg) classification tasks: a review," Journal of neural engineering, vol. 16, no. 3, p. 031001, 2019.

[44] L. Yang, Y. Gao, M. Li, K.-E. Park, S. Liu, X. Kang, M. Liu, A. Oswalt, L. Fang, B. P. Telugu et al., "Genome-wide recombination map construction from single sperm sequencing in cattle," BMC genomics, vol. 23, no. 1, pp. 1–9, 2022.

[45] M. Roggero, "Discontinuity detection and removal from data time series," in VII Hotine-Marussi Symposium on Mathematical Geodesy. Springer, 2012, pp. 135–140.

[46] R. Hegger, H. Kantz, "Practical implementation of nonlinear time series methods: The TISEAN package," Chaos, vol. 9, pp. 413, 1999.

[47] N. Gürsakal, F. M. Yilmaz, and E. U ̆gurlu, "Finding opportunity windows in time series data using the sliding window technique: The case of stock exchanges," Econometrics, vol. 24, no. 3, pp. 1–19, 2020.

[48] K. T. Kasfi, A. Hellicar, and A. Rahman, "Convolutional neural network for time series cattle behaviour classification," in Proceedings of the Workshop on Time Series Analytics and Applications, 2016, pp. 8–12.

[49] M. Lee, "Iot livestock estrus monitoring system based on machine learning," Asia–Pac. J. Convergent Res. Interchange, vol. 4, no. 3, pp. 119–128, 2018.

[50] L. Wang, R. Arablouei, F. A. Alvarenga, and G. J. Bishop-Hurley, "Animal behavior classification via accelerometry data and recurrent neural networks," arXiv preprint arXiv:2111.12843, 2021.

[51] D. Pavlovic, C. Davison, A. Hamilton, O. Marko, R. Atkinson, C. Michie, V. Crnojevi´c, I. Andonovic, X. Bellekens, and C. Tachtatzis, "Classification of cattle behaviours using neck-mounted accelerometer-equipped collars and convolutional neural networks," Sensors, vol. 21, no. 12, p. 4050, 2021.

[52] P. Balasso, C. Taccioli, L. Serva, L. Magrin, I. Andrighetto, and G. Marchesini, "Deep learning performance in predicting dairy cows' behaviour from a tri-axial accelerometer data," Research square preprint, 2022.

[53] V. Bloch, L. Frondelius, C. Arcidiacono, M. Mancino, and M. Pastell, "Cnn and transfer learning-based classification model for automated cow's feeding behaviour recognition from accelerometer data," bioRxiv preprint, 2022.

[54] O. Marko, "Classification of different cattle behaviors using advanced machine learning algorithms." Master thesis, University of Novi Sad, 2022.

[55] Z. Zhao, "Apply machine learning on cattle behavior classification using accelerometer data," Ph.D. dissertation, Virginia Tech, 2022.

[56] G. Castagnolo, D. Mancuso, S. Palazzo, C. Spampinato, and S. Porto, "Cow behavioural activities classification by convolutional neural networks," in Proceeding of the European Conference on Precision Livestock Farming (ECPLF), p. 753-760, Aug 2022.

[57] M. Liu, Y. Wu, G. Li, M. Liu, R. Hu, H. Zou, Z. Wang, and Y. Peng, "Classification of cow behavior patterns using inertial measurement units and a fully convolutional network model," Journal of Dairy Science, vol. 106, no.2, p. 1351-1359, 2022.

[58] R. Peck and J. L. Devore, Statistics: The exploration & analysis of data. Cengage Learning, 2011.

Chapter 5

Improving abstraction by combining Fourier surrogates and sampling schemes

In the previous chapters, it was shown that considerable accuracy improvement can be obtained by augmenting the dataset via considering a system-specific feature such as collar rotation state, or through combining empirical manipulations like recombination. In this chapter, a final step is taken towards a method that aims to more abstract and supported by theory. As I show below, the idea is to combine three aspects. One is surrogate data generation. Another is a sampling scheme that allows adjusting the distribution. And the final aspect is performing all operations in an integrated way. This ensures that new data are available for each step of the training process. The improvement is maximized when the dataset is balanced via the application of a suitable sampling scheme and the negative influence of data duplication is reduced via using Fourier surrogates. With the proposed approach, the overall accuracy is improved from 90% to 96%, while the classification accuracy of an under-represented behavior, namely grazing, is elevated from 45% to 91%.

In Chapter 4, the idea of time reversal was introduced. Basically, it was shown that considering a time series backwards can help the classifier training. By definition, this operation keeps perfectly the distribution of acceleration values, and the amplitudes of the Fourier spectra. However, the phases are changed, because the time goes backwards. It is known that time reversal can be used as a very basic way to check if a time series contains non-linear properties: scientists studying non-linear behaviors have for long time used this simple manipulation. From the literature on non-linear systems, it can be seen that this approach was made more general by finding other ways to change the phases while keeping the value distribution and amplitude spectrum [1]. Those are usually known as methods to generate surrogate data. A well-known software package in which time reversal was used, and then replaced with surrogates methods is TISEAN (TIme SEries ANalysis), that was introduced in the year 2000 [2]. Then, from reading such literature, the idea comes naturally that those methods could also be applied in this application, to replace and extend time reversal that was introduced in Chapter 4. As explained and justified below, the key difference and advantage is that, because the phases using such methods are random, a potentially unlimited number of time series can be generated, in contrast with time reversal, which can only be applied once.

5.1 Concept and generation of surrogate time series

In the same way that rotation and warping were initially meant for use in the domain of image processing, also the notion of surrogate data comes from another area. The underlying idea is to take a time series and generate a new one that preserves some of its features, but not all of them. The concept originates in the field of non-linear science, where the purpose is deciding if a time series is from a non-linear system, or not [1]. Surrogate data usually aim to keep all linear aspects of a signal while destroying the non-linear ones. In a

practical sense this means keeping the same frequency spectrum and distribution of values, but taken from an underlying random system. Then, there can be no fixed causal relationships across time [3]. In the case of time series classification, the neural network may learn several aspects of the signal, but it is impossible to tell which ones from the start, if feature extraction is done by the CNN itself. In general way, the frequencies of the spectrum, the mean and the variance are all important aspects to keep; as I will show below, this is also the case for this study. Then, the idea of surrogate data becomes relevant to generate a possibly unlimited number of new time series, all uncorrelated to each other in time, but keeping some of the underlying features. It should be seen that, if one chooses the approach of surrogate data, then the need for system-specific assumptions such as rotation in Chapter 3, or empirical considerations as in Chapter 4, may be reduced. These assumptions may be replaced by the surrogates generation procedure.

From the review by Schreiber, it can be seen that all surrogate data are random, but not all random data are acceptable as surrogates [4]. When taking just random values for each time point, one has white noise. This is not good, because the autocorrelations in the original signal normally carry important information. Then, one can filter the random signal so that it has the same frequency content as the start. Another way is to keep the initial signal's Fourier amplitudes, but replace the phases with random numbers. It can be seen that by using the inverse Fourier transform the result is a signal uncorrelated in the time domain to the initial one, but having properties similar to it. However, its mean, variance and so on can be different. Another possibility is to randomly shuttle the initial signal: this keeps the distribution of values, but it effectively generates white noise. The purpose of surrogate generation algorithms is to find a balance between the two situations, keeping as carefully as possible the frequency content as well as the distribution of values. The procedure is explained below. It is important to point out that, even in the presence of many signals that are considered together, multivariate generalizations of the technique can be used.

In this section, I will give the theoretical derivation of the multivariate integrated Amplitude Adjusted Fourier Transform (AAFT) [4,5] and propose the use of it as a time series augmentation method for wearable sensor data and how it can be used for data augmentation in the task of cattle behavior classification. The motivation for surrogate data technique-based augmentation initially stems from the observation that time reversal performs well in Chapter 4. Because this operation destroys the original non-linear content of a signal but keeps its linear features, a natural extension of the idea is to use surrogate data. The important benefit is that, while time reversal can only be performed once (otherwise it gives back the initial input), surrogate data provide an unlimited number of time series that could be used for network training.

So, following the steps by Schreiber and Schmitz [4,5], assume that the acceleration time series x can be expressed as a linear autoregressive time series. For x_t, i.e., the data at sample t, it can be described as follows:

$$x_t = \phi_0 + \sum_{i=1}^{p} \phi_i x_{t-i} + \varepsilon_t \quad (1)$$

where $\phi_0, \ldots, \phi_p$ are coefficients associated with prior values of x_t; ε_t is an independent and identically distributed noise sequence with zero mean, i.e., $\varepsilon_t \sim N(0, \sigma^2)$ where $N(0, \sigma^2)$ denotes Gaussian normal distribution with the mean 0 and variance σ^2; p indicates the order of the regression.

In the Fourier domain, the amplitude of the Fourier transform of x_t and from there, the power spectrum, could be obtained by the equation (1) according to the Wiener-Khinchin theorem. Since the phase is not specified in the above equation, there are many possible realizations that preserve the power spectrum. This is the reason why surrogate data are a source of endless time series. Because of the key role of the Fourier transform, surrogates generated using this approach are also informally called "Fourier surrogates".

As mentioned above, the techniques of surrogate data generation were originally designed for a different analytical purpose, that is, testing whether there is nonlinearity in irregular fluctuations of a time series. The basic idea of the surrogate techniques is that the time series we want to examine shows irregular fluctuations. One possible reason for this is the presence of nonlinear dynamics. Surrogate data can eliminate or destroy the nonlinearities in data while preserving various linear statistical properties. Detecting nonlinearity using surrogate techniques have its statistical basis where the null hypothesis we set is that the time series data observed is resulted from a stationary stochastic linear process with random inputs [1,3]. If there is no nonlinearity in the original data, there should not be a large (significant) difference between the surrogate data and the original one in any possible signal statistics, because the surrogate data do not have nonlinearity but have the same linear characteristic as the original one. Conversely, if there is a large difference in some statistics, the null hypothesis will be rejected, meaning there is probably nonlinearity in the original data. One example is a time series showing a period-3 behavior, that is, for example, the repetition of a large-amplitude cycle, an intermediate one, and a small one as 123123123… and so on. This kind of property is destroyed by surrogate generation, but we can guess it is not very relevant to animal behavior. Then, if linear features are most important, then surrogate time series can be a good source of training data.

As introduced by Schreiber and Schmitz [5] and summarized by Lee et al. [6], give a time series $\boldsymbol{x} = \{x_1, x_2, \dots, x_N\}$. The specific mathematical processing of surrogate data generation can be outlined as follows:

1. Sort $\boldsymbol{x}$ in ascending order, with $\boldsymbol{s}$ the sorted data and i_l the index in $\boldsymbol{x}$. Here the l^{th} element of $\boldsymbol{s}$ is the ${i_l}^{th}$ element of $\boldsymbol{x}$, i.e., $s_l = x_{i_l}$.

2. Generate a random array $\boldsymbol{r_k} = \{r_k\} = \{r_1, r_2, \dots, r_N\}$, $r_k \sim U(0,1)$, where $U(0,1)$ is the uniform distribution on $[0,1]$.

3. Sort $\boldsymbol{r_k}$ in ascending order, with $\boldsymbol{r} = \{r_{k_l}\}$ the sorted data.

4. Order $\boldsymbol{r}$ by i_l to get a reordered time series $\boldsymbol{y} = \{y_i\}$.

5. Apply Discrete Fourier Transform to $\boldsymbol{y}$, i.e., $\{\mathrm{Y(w)}\} = \mathcal{F}(\boldsymbol{y})$

6. Use a random angle $\varphi \in U[0,\ 2\pi)$ and obtain $Y'(w) = Y(w)e^{j\varphi}$.

7. Apply Inverse Discrete Fourier Transform to $Y'(w)$, obtain $\boldsymbol{y}' = \{y_i'\} = \mathcal{F}^{-1}(\{Y'(w)\})$.

8. Sort $\boldsymbol{y}'$ in ascending order, with $\boldsymbol{y}^*$ the sorted data and i_i' the index in $\boldsymbol{y}'$. Reorder time series $\boldsymbol{s}$ by i_i' and obtain a surrogate time series data $\boldsymbol{x}' = \{x_i'\}$

The above steps describe the Amplitude Adjusted Fourier Transform (AAFT). It can be seen that both the distribution of values and the Fourier amplitudes are considered, however, at different times, meaning, steps 1-4 and 5-7. This means that only an approximate similarity is kept to the original time series. The Iterated AAFT (IAAFT) indicated in **Fig. 5.1a** will iterate the surrogate process until there is a close match in both the autocorrelations and distribution with respect to the original signal [1,2-5]. Specifically, the amplitude distribution and spectrum of surrogate time series are iteratively corrected in an alternative manner, targeting a close overlap to the original time series as the goal. The distribution adjustment procedure is done by rank ordering, and the Fourier power spectrum adjustment is done by taking the Fourier transform, replacing the squared amplitudes $\left|A_k{}^{(i)}\right|$ where i indicates i th iteration, by the desired one $|A_k|$ while the phase $\psi_{k,m}$ is kept the same, and then transforming back. Because the process may need to iterate several times, the computational load is higher compared to rotation or the empirical manipulations considered before.

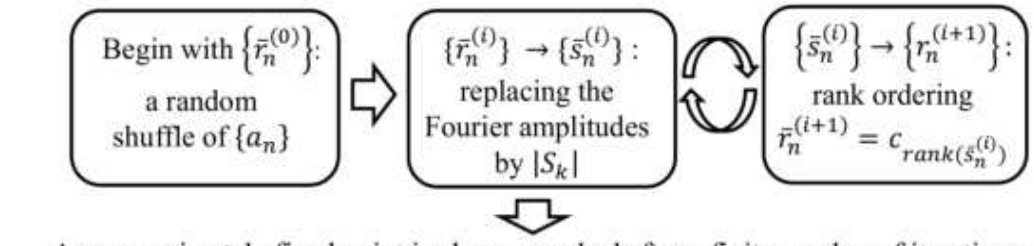
Further, as again summarized by Schreiber [5], considering the sensor data we collected from cows have multi-variables, in addition to matching their individual spectra and distributions exactly, we preserve their cross-correlation function as good as possible. In

the cases of multivariate signals, as shown in **Fig. 5.1b**, the amplitude distribution could be easily applied by rank re-order for each channel individually. For the spectrum, some changes of the processing procedure in Fourier domain are needed. The phases $\psi_{k,m}$ have to be replaced with new ones suitable for all the three axes together, that means, trying to keep their cross-correlations. Specifically, the replacement should be minimal in the least-squares sense, i.e., it should minimize $h_k = \sum_{m=1}^{3} \left|\exp\left[i\phi_{k,m}\right] - \exp\left[i\psi_{k,m}\right]\right|^2$, where $\phi_{k,m}$ donates the phase applied to x-, y-, and z-axis in the spectrum adjustment procedure, $\psi_{k,m}$ donate the phase used in univariate cases. In addition, to preserve the cross-spectrum, the phase differences in x-y, y-z, and x-z should be kept the same, i.e., $\exp\left[i(\phi_{k,m2} - \phi_{k,m1})\right] = \exp\left[i(\rho_{k,m2} - \rho_{k,m1}\right)$. With this, the amplitude distribution and spectrum are kept and there is also close match in the cross-spectrum of x-, y- and z-axis.

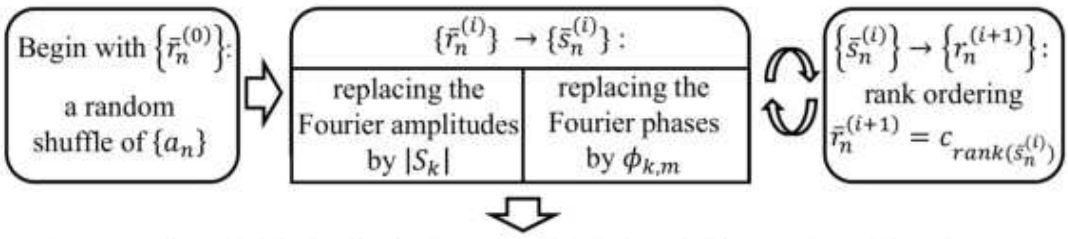
Because the process of surrogate time series generation involves a random aspect, it can be repeated as many times as the user wants. Basically, it is like endless source of data with certain features. Therefore, one may choose how much surrogate data is necessary in each case, depending on initial size, computation time and so on.

For completeness, it should be added that other methods to realize surrogate time series also exist. For example, a well-known alternative to the Fourier transform is the Wavelet transform: instead of representing the data as amplitudes and phases of sine waves, it decomposes them according to kernels having different width, such as the so-called "Mexican hat". The Wavelet transform is powerful because it can show in a clear way different levels of detail, and because it can represent how certain features of the frequency content change over time. For this reason, it is often used to analyze non-stationary signals like speech [7]. Recently, it has been proposed that the Wavelet transform could replace the Fourier transform in surrogate generation, and that this could allow an even better keeping of the original signal features. For example, it seems that methods based on the Wavelet transform are

used in earth science, when the purpose is checking the presence of non-linear behaviors in very complicated time series [8,9]. Such methods keep more features than the value distribution and autocorrelation. Interestingly, it has even been proposed to mix Fourier and Wavelet transforms for surrogate generation [10]. Because in this research the purpose is simply generating more data that are useful, the Fourier transform is preferred, also because it is easier to understand, and its calculation time is much lower. However, future work on data augmentation could consider Wavelet-based surrogates.



(a)



(b)

Figure 5.1: The principle of iteratively refined surrogates . (a) Univariate IAAFT (b) Multivariate IAAFT. Provided the original data $\{a_n\}$, $\{|S_k|\}$ denotes the Fourier amplitudes in the initial data and $\{c_k\}$ sorting the same according to ascending order. At the i th iteration stage, sequence $\{\bar{r}_n^{(i)}\}$ has the correct value distribution, while $\{\bar{s}_n^{(i)}\}$ has the correct Fourier amplitudes.

5.2 Considerations on integrated data augmentation

Besides the replacement of empirical operations with surrogates, and the sampling scheme described below, one new aspect of this chapter is the integrated augmentation. In recent years, data augmentation in the image domain has become a common practice and has been well explored. For example, many well-established deep learning architectures for image classification, such as AlexNet [11] and residual networks (ResNets) [12], and very deep convolutional networks (VGGs) [13], use data augmentation approaches as a standard practice in the process of model training. By contrast, even recent reviews of time series data augmentation did not report any study using integrated augmentation [14-16]. Therefore, in this chapter I will present the first results.

In particular, this chapter introduces new ideas and their application as integrated augmentation methods. Namely, to tackle the task of accurately classifying cattle behavior while faced with limited data availability, this chapter suggests the integrated and synergistic use of multiple approaches, which deliver a substantial increase in classification performance. It combines three key ideas having a possibly broader usefulness:

- First, I propose to use random selection of time series snippets during each training epoch to provide a built-in source of variability that aids the determination of classifier boundaries yielding a high generalization ability.
- Second, I propose to apply a form of biased sampling which aims to offset the dataset imbalance problem, further aiding the training process in determining the boundaries between the most and least represented behaviors. Also in this case, the sampling is performed for each training epoch, making sure that different data are always selected insofar as possible.

- Third, I propose to combine the above with the generation of Fourier transform-based surrogates to alleviate the issue of data duplication encountered when repeatedly sampling over the least represented behaviors. This is especially important for the behaviors with limited recorded data available.

5.3 Data and proposed processing methods

5.3.1 Data Acquisition

The dataset considered in this chapter is the same as Chapter 4. However, as a part of the ongoing development of the system and improvement of researcher skills, the labeling operations were repeated. Therefore, the distribution of behaviors was not identical, and is shown in **Fig. 5.2**, which depicts the relative prevalence of data samples across the activities.

Some examples of the time series are shown in **Fig. 5.3**. In can be seen that in the case of Resting, the activity level is low. In all other cases it's higher, with the biggest accelerations observed for Grazing. All signals are clearly irregular, which is an indication for using surrogate data that attempt to preserve all the linear signal features as closely as possible. It can also be easily seen that the x, y, z components contain frequencies which are very different depending on the activity. Then, the idea of keeping the frequency content while generating surrogate data appears immediately relevant.

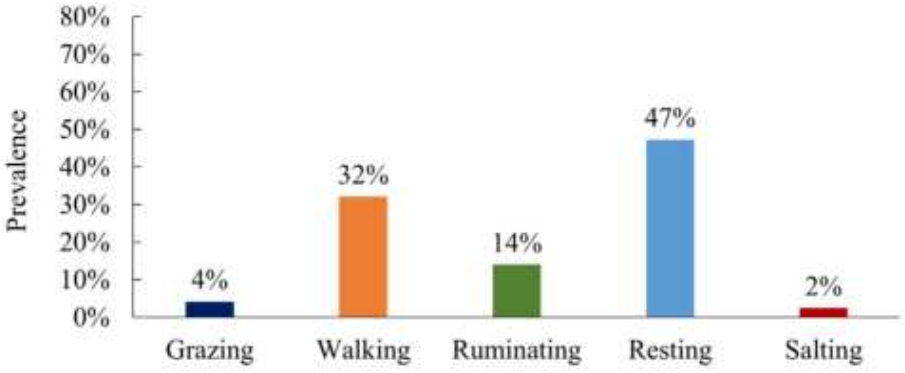


Figure 5.2: Relative behavior prevalence (normalized)

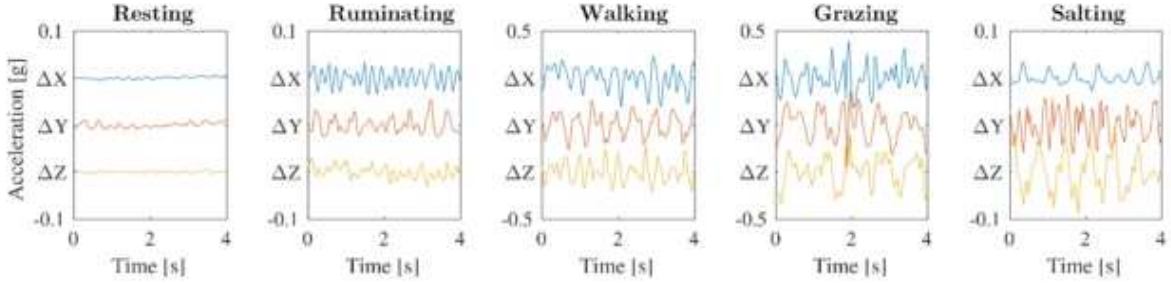


Figure 5.3: Representative time series excerpts for the behavior classes. Mean subtracted for visualization purposes. Units of g.

5.3.2 Machine Learning Model

The network type and configuration was also kept identical as in Chapter 4. However, because of the integrated biased sampling scheme in model training and evaluation, there were some changes in implementation. All these steps were performed using custom-developed source code in Python language using Keras with the backend of TensorFlow (version 2.4.0) [17]. An initial behavior classifier model was established using the training set, then the network parameters were heuristically tuned with the validation set in the process of model training. After learning was completed, the effectiveness of the classifier model was

examined based on the remaining data, which was viewed as an independent test set. The performance measure in term of overall accuracy was the agreement of the behaviors predicted by the classifier using a winner-take-all approach with the manually labeled tags from video analysis. As regards the accuracy for the individual behavioral classes, it was assessed by means of the F_1 score.

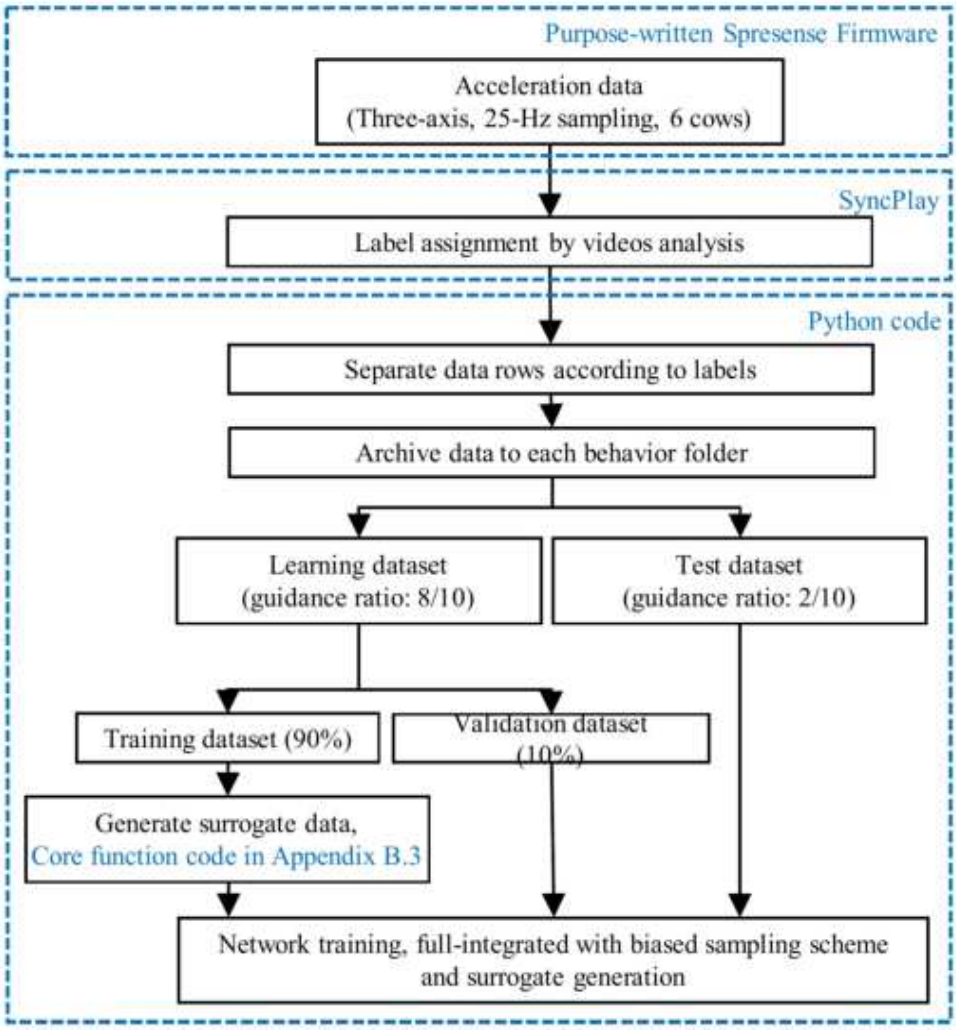


Figure 5.4: Data processing flow. Blue overlays show the software platforms used for implementation.

The detailed data processing flow is shown in **Fig. 5.4**. As in Chapters 3 and 4, a specific firmware running on the Spresense board described in Chapter 3.2.1 was used to collect the acceleration data, and video labelling was done using the SyncPlay software from ATR-Promotions Inc., as described in Chapter 3.2.5. Data separation and splitting for this Chapter were performed using Python code, followed by the model training with integrated augmentation. The core function code for surrogate data generation was provided in Appending B.3.

5.3.3 Data Augmentation procedure

Fig. 5.5 shows the data flow supporting the experimental design. Following data segmentation according to contiguous behavior labels, stratified splitting was performed, retaining 72% of the overall data for training and the remaining 20% and 8% for test and validation, respectively. Thereafter, the analysis split into three analogous branches: a first one involving only original data, a second one involving only surrogates, and a third one involving an even mixture of the two (identified, respectively, with "O", "S" and "M"). This branching pertained to the training data only, and only original data were used for validation and testing. Within each branch and as shown in **Fig. 5.6**, a further split into three sampling approaches was present, namely, considering adjacent windows (all data entered during each training epoch, retaining the original class distribution), considering one window per data segment (extracting one snippet per segment starting from a different random location for each epoch, retaining the original class distribution), and considering n windows per data segment so as to approximately balance the distribution (identified, respectively, with "A", "1" and "n"). In other words, the study was a 3-by-3 design according to surrogate usage and windowing/sampling approach.

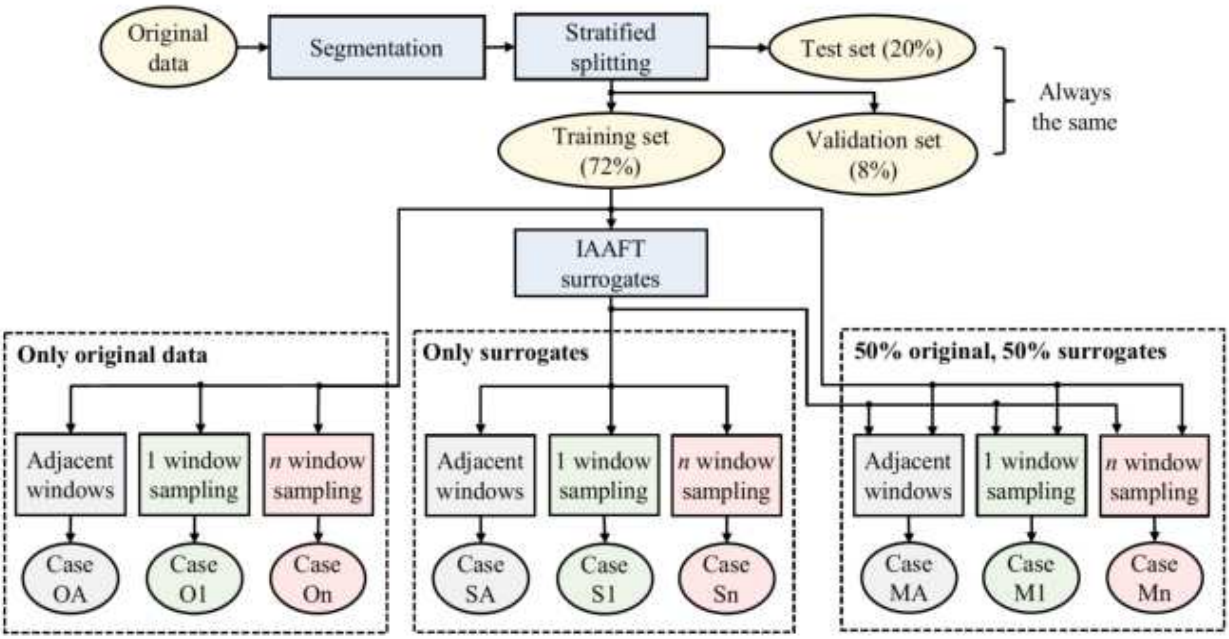


Figure 5.5: Study design for the comparisons, showing the 3-by-3 split according to surrogate usage and sampling scheme.

All aspects of the data augmentation were seamlessly integrated with the training process; in other words, no precalculation was performed, with the surrogate and sampling performed independently for each training epoch. Surrogate time series were generated through the IAAFT method, separately for each segment of original data, to ensure no labels were mixed. The method is described in detail Chapter 5.1, and in brief, consists of the iterative adjustment of amplitude distributions and Fourier spectra, starting from a shuffle of the initial data points. The Fourier transform is calculated at each iteration, retaining the phases but replacing the amplitudes with those of the original time series. Then, the values of the iterated time series are replaced with those from the original time series, according to their ranks. The process is iterated until both the initial signal's value distribution and autocorrelation are sufficiently preserved. The obtained time series is entirely uncorrelated in the time domain, which prevents data duplication. Because, for tri-axial data, significant

information may be contained in the crosscorrelations, I applied a multivariate extension of this method which also preserves crosscorrelations.

The exact modality of using surrogates for data augmentation varies across studies, and the literature remains scant. For example, Lee et al. [18] have used surrogate data intermixed with original accelerometric and neurophysiological recordings, drawing the training and test data from the resulting pool. Schwabedal et al. [19] adopted a similar approach, however applying surrogates only to training data alongside crossvalidation. On the other hand, in a later study, Lee et al. [6] proposed using surrogates as the exclusive basis for training and validation, reserving the entirety of experimental recordings for testing. In this chapter, I more systematically consider three possibilities: training only on original data, training only on surrogates, and training on an evenly mixed pool. In all cases, validation and testing are performed exclusively on original data, which is motivated by the fact that these vectors need to remain unchanged throughout the training process. As a consequence, it is possible to address explicitly the effect of surrogate data inclusion.

Finally, while previous works have used surrogate time series purely based on empirical evidence that they can aid the training process, as shown in **Fig. 5.7**, here I adopted a deductive approach to understanding precisely which retained features render the surrogate

time series usable for training. Starting from the original data, the non-linear structure is firstly destroyed by the IAAFT method itself. Thereafter, I switched from a multivariate approach to a univariate one, thereby ceasing to retain the crosscorrelations. Next, I relinquished the IAAFT iterative approach and simply randomized the Fourier phases, leading to time series that retains the autocorrelation but not the value distribution. Finally, I also subtracted the average and normalized the variance to unity: this aspect is important since significant information about the behavior can be conveyed by the average and variance, which are in part represented within the Fourier amplitudes, even when the value distribution is not adjusted.

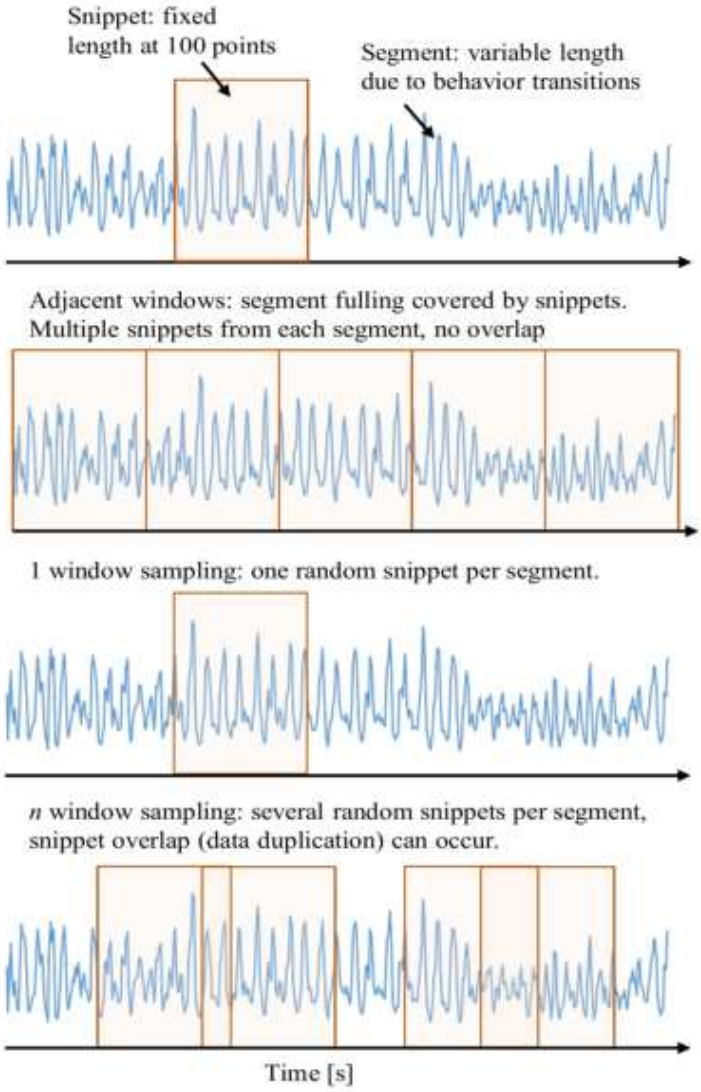


Figure 5.6: Sampling schemes used in deriving snippets (fixed length 4 s time-intervals submitted to the CNN) from segments (variable length 8-48 s time-intervals of homogeneous behavior).

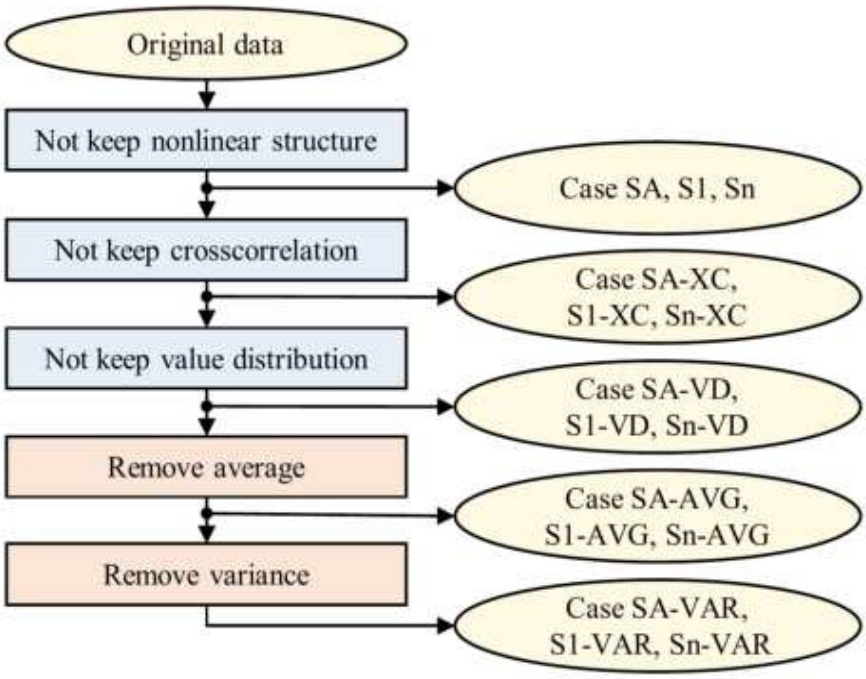


Figure 5.7: Deductive steps used to determine the elements of surrogate data supporting high training performance.

5.4 Experimental results

5.4.1 Classification performance across the sampling and surrogate schemes

The classification accuracy values, aggregated and separate for each behavior class, are given in **Table 5.1**, and the corresponding confusion matrices are shown in **Fig. 5.8**. Considering the original data only and dividing up all segments into adjacent windows (case OA), an overall accuracy of 90% was obtained. With respect to this, the most significant improvement, to 95%, was obtained by introducing random sampling (case O1), so that one window is extracted starting from a different random time-point at each training epoch; this

improvement was particularly notable considering the least-represented behaviors, namely Grazing and Salting, which improved from 45% to 83% and from 83% to 98% respectively.

Introducing a biased sampling so that the number of windows entered in the training process is rendered approximately even across behaviors (case On), a further improvement to 96% was recorded; even though it was quantitatively smaller, it was consistent across all classes.

Considering next the training performed exclusively on the surrogates, the principal finding was that the accuracy levels across the three sampling schemes were closely comparable to those obtained when using the original data, namely 90% for both cases OA and SA, 95% and 94% for cases O1 and S1 respectively, and 96% for both cases On and Sn. When considering the accuracy for the Grazing class, the performance was actually better when training using the surrogates rather than the original data, namely, with 45% vs. 52% for cases OA and SA, 83% vs. 85% for cases O1 and S1, and 86% vs 89% for cases On and Sn. This could plausibly be ascribed to statistical aspects such as improved stationarity.

The most favorable performance was obtained via mixing the original data and surrogates, randomly chosen during each epoch. At the level of overall classification accuracy, the scores of case Mn were comparable to those obtained using only original data or only surrogates, namely, with 96% (case On) and 96% (case Sn). However, considering the individual behaviors, the score for Grazing was maximized, reaching 91% for case Mn as opposed to 86% for case On. Due to a saturation effect, this was not evident for Salting. The advantage of mixing surrogates and original data plausibly stems from their different statistical properties in terms of which features are retained and data variability.

Table 5.1: Performance of the classification results on a test dataset

Case	OA	O1	On	SA	S1	Sn	MA	M1	Mn
Grazing	45%	83%	86%	52%	85%	89%	68%	87%	91%
Walking	89%	95%	96%	88%	95%	98%	85%	96%	97%
Ruminating	88%	92%	95%	89%	91%	93%	83%	91%	92%
Resting	95%	96%	97%	94%	95%	97%	90%	96%	97%
Salting	83%	98%	100%	79%	93%	100%	78%	93%	100%
Overall accuracy	90%	95%	96%	90%	94%	96%	86%	95%	96%

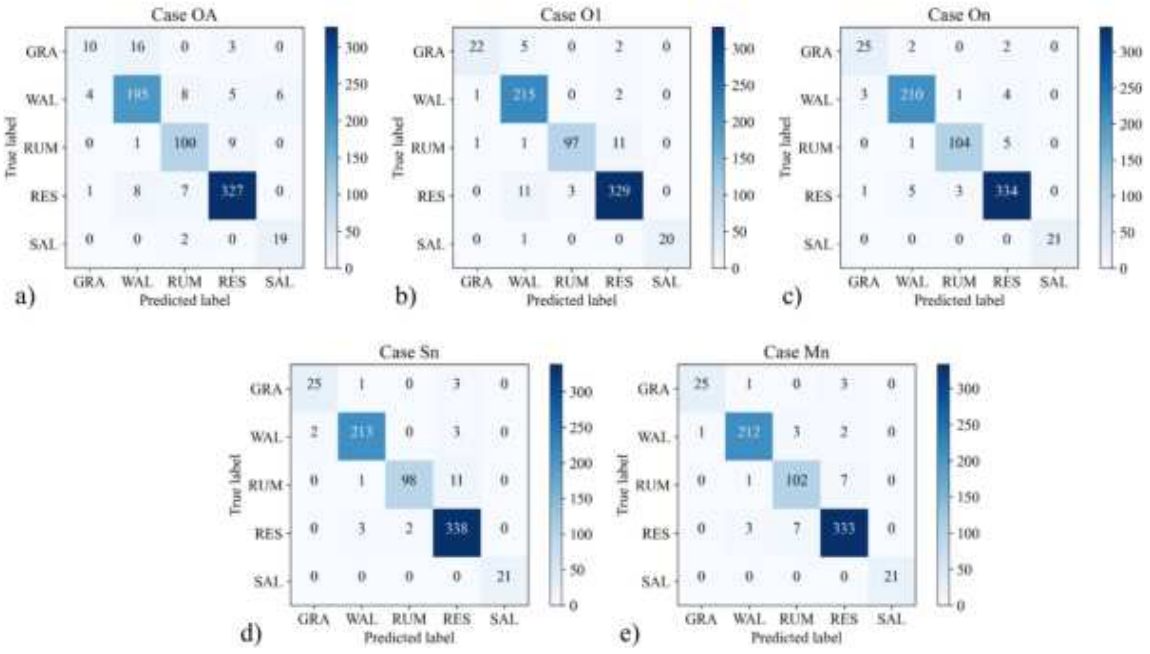


Figure 5.8: Confusion matrices for a selection of sampling and surrogate schemes (test data).

5.4.2 Analysis of the relevance of surrogate time series

Considering the ability of surrogates to sustain high training performance, the following results were noted (see **Table 5.2**); for brevity, they are presented concerning case Sn, but similar considerations apply to cases SA and S1. Starting from the attained score of 96%, switching to a univariate approach not retaining the crosscorrelation had a small effect on performance, which remained high at 95%. Removing the iterative process in the IAAFT algorithms and retaining the Fourier amplitudes without the value distribution did not further reduce the performance.

On the other hand, removing outright the average by means of subtracting it from the data had a more complex effect. The overall accuracy for case Sn remained similar, down to 94%, however, for case SA, the accuracy for the grazing class collapsed to 4%. Overall, removing the variance information by normalization had a stronger and more generalized detrimental effect, reducing the overall accuracy down to 80%.

Table 5.2: Performance of the classification results on a test dataset

Case	SA-XC	S1-XC	Sn-XC	SA-VD	S1-VD	Sn-VD	SA-AVG	S1-AVG	Sn-AVG	SA-VAR	S1-VAR	Sn-VAR
Grazing	57%	80%	84%	50%	87%	84%	4%	78%	85%	5%	49%	64%
Walking	87%	95%	96%	88%	96%	96%	84%	94%	94%	63%	78%	70%
Ruminating	85%	89%	91%	81%	89%	91%	78%	94%	93%	60%	82%	82%
Resting	93%	96%	96%	90%	95%	97%	92%	96%	96%	84%	86%	85%
Salting	79%	93%	100%	95%	95%	100%	67%	95%	98%	83%	92%	98%
Overall accuracy	89%	94%	95%	87%	94%	95%	84%	95%	94%	73%	82%	80%

Altogether, these results suggest that the bulk of relevant information was contained in the autocorrelation, which was the only feature retained after this cascade, and in the variance. Therefore, to further characterize the dynamical features retained by the IAAFT surrogate generation process which support the use of the surrogates, the value distribution, autocorrelation and crosscorrelation were considered in detail across the behavior classes.

Firstly, I note that, as shown in **Fig. 5.9**, the five behaviors were characterized by markedly different distributions of acceleration values; for convenience, they are illustrated here after detrending. For Resting, a narrow Lorentz-like distribution was observed, with near-complete overlap between the three axes (standard deviations 0.03 g, 0.03 g, and 0.02 g for X, Y, and Z, respectively). For Ruminating, the distributions were marginally broader, and less peaked around zero, albeit retaining a comparable standard deviation (0.02 for all three axes). By contrast, for Walking, the acceleration distributions for all the three axes were markedly broader and more Gaussian-like (standard deviations 0.10 g, 0.13 g, and 0.11 g). For Grazing, the situation was comparable, albeit with greater noise due to the smaller amount of data (standard deviations 0.11 g, 0.10 g, and 0.08 g). For Salting, the variability was intermediate, and there was an evident difference between a Lorentz-like distribution for the X axis and Gaussian-like distributions for the Y and Z axes (standard deviations 0.03 g, 0.04 g, and 0.05 g). In summary, the Resting and Ruminating behaviors appeared closely comparable and well-separated from Walking and Grazing, which were similar to each other, whereas Salting represented an intermediate condition. Therefore, the relevant information contained in the value distribution consisted mainly of different variances, supporting the distinction between these classes.

Secondly, I note that, as shown in **Fig. 5.10**, across the three axes, the five behaviors were characterized by visibly different autocorrelation and crosscorrelation profiles, wherein the latter tended to be smaller. On the whole, Resting was associated with a relatively slow and monotonic autocorrelation decay, which was comparable for the three axes;

its crosscorrelation dwelled around zero, except for the YZ combination, which was strongly anti-correlated. These features plausibly stem from the absence of regular movements alongside occasional rotations of the head during Resting. By comparison, the autocorrelation envelope for Ruminating showed a faster decay, which was additionally associated with a prominent periodic oscillation peaking at a lag around 10 samples, particularly for the Y axis; this periodicity could also be appreciated for the crosscorrelation. Ruminating knowingly involves prolonged chewing and associated rhythmic neck movements, which may explain the observed pattern. Conversely, Walking was associated with the fastest autocorrelation decay, alongside minimal periodicity and weakest crosscorrelation. These features plausibly reflect the fact that, regardless of leg movements, the head movements are minimized in this condition due to staring forward. Grazing exhibited properties intermediate between Ruminating and Walking, as regards to both the autocorrelation decay and the strength of crosscorrelation; compared to Ruminating, this behavior was associated with a somewhat slower periodicity, peaking at a lag around 15 samples, again in line with behavioral expectations of slower chewing in this condition. Finally, Salting was markedly different from all other behaviors, in that it was hallmarked by a very strong periodicity at an intermediate frequency, again peaking at a lag around 15 samples, which was visible on all three axes for autocorrelation and all three axis combinations for crosscorrelation. Since Salting involves large repetitive "sliding" movements associated with licking, this pattern was expected. Therefore, the most important distinguishing feature appeared to be autocorrelation, followed by the value distribution. Notably, the separability of the behaviors was different and complementary between them, since Resting and Ruminating, Walking and Grazing had similar value distributions but markedly different autocorrelation profiles.

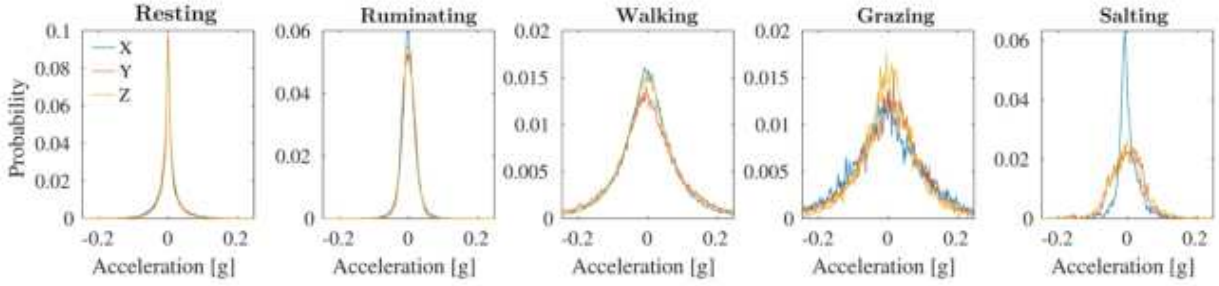


Figure 5.9: Value distributions across the behavior classes.

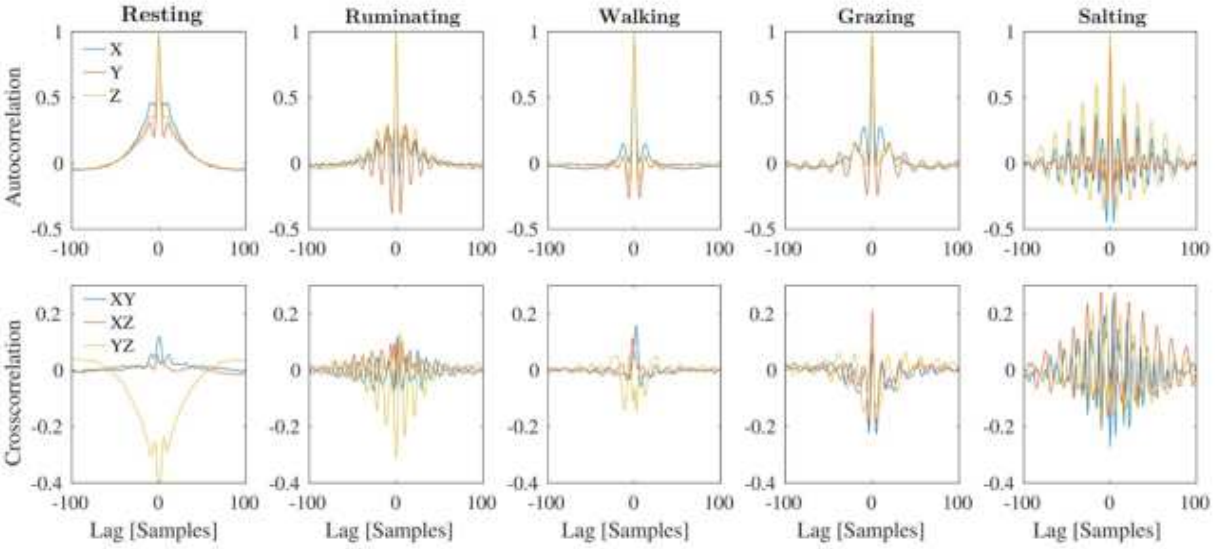


Figure 5.10: Autocorrelation and crosscorrelation across the behavior classes.

5.5 Additional datasets for confirmation

5.5.1 Purpose and data sources

Throughout this thesis, I have stated several times that the augmentation method becomes more abstract from Chapter 3 to Chapters 4 and 5. In this section, I provide some initial indication that the final method obtained, the one which I state is most abstract, seems

works effectively also on other datasets and therefore could, potentially, in the future be used widely in IoT and Edge AI. The main motivation for undertaking these analyses was to confirm that the efficacy of sampling and surrogates is not because of some specific feature of the cow data or other unexpected issue with the data. However, it is necessary to say that the analyses in this section are not enough to be considered an actual proof of generality. To claim general usefulness on any type of sensor time series, much additional work will have to be done in the future considering sensors other than accelerometers, and the influence of many application aspects. More humbly, these results intend to give some initial confirmation that such general use might actually be possible and, therefore, should be investigated in the future.

Three datasets are considered. The first one is a dataset of human behaviors recorded using a tri-axial accelerometer located in a mobile phone [20,21]. The behaviors consist of walking, jogging, going upstairs, going downstairs, sitting and standing. This dataset is related to the cow data, because the data are also accelerations, however, the behaviors are human, so, entirely different. The second dataset, on the contrary, is entirely different and consists of 3 electroencephalography signals, recorded from the back of the brain while the eyes are open or closed. These conditions are linked to changes in the frequency content [22,23]. The third dataset is also completely different and consists of recordings from an electrical motor and mechanical gears, obtained using accelerometers, a tachometer and a microphone. In this case, the conditions correspond to possible faults, and the accelerometers record vibrations rather than behaviors [24].

As presented in Chapter 1, the field of the IoT and Edge AI is so broad, because it includes applications in agriculture, farming, industrial manufacturing, transport, medicine, environment monitoring and so on. All these applications use many types of different sensor, then, it would be difficult to cover them all without testing a large number of datasets [25-27]. Therefore, the three datasets considered in this section should be considered only as an

initial step towards demonstrating general usefulness. The first one shows another type of behavioral classification, using similar type of data as the cow recordings, that is, human accelerometer recordings. The second one uses a totally different type of signal, that is, electrical recordings of brain activity. The third one is based on multiple signals together, mainly acceleration, however, used to monitor vibration instead of behavior. While this selection is not complete, it can be said that these three datasets cover several typical applications related to consumer, biomedical and industrial IoT. In the future, more work is needed to prove if the methods proposed can be used generally or not.

For these three datasets, the model settings are kept the same as for the cow data, since the purpose is to evaluate the improvement due to sampling and surrogates, not to optimize a model itself. This seems acceptable because, as indicated below, good performance is eventually achieved in all cases.

5.5.2 First additional dataset

The first dataset, as I wrote, is about human behaviors recorded using a tri-axial accelerometer located in a mobile phone. It is known as the WISDM: WIreless Sensor Data Mining dataset and considered the most popular and represented in its kind [20,21]. This application is common for fitness tracking and other wearable devices [28,29]. In total, it contains 1095199 data rows, acquired at 20 Hz over a period of ≈15.2 h for 6 classes. It can be seen in **Fig. 5.11** that the classes are heavily imbalanced, which is similar to the situation for the cow data. Walking and jogging are most common, with 39% and 31%, followed by going upstairs and downstairs at 11% and 9%, while sitting and standing are rarer, with 5% and 4%. For this dataset, the window size is set to 100 points and 80% of data are used for training and validation, 20% for testing.

After applying the same procedures indicated in **Fig. 5.5**, the F_1 scores and overall accuracy reported in **Table 5.3** are obtained. For this, only four cases are considered, to reduce the amount of data and focus on evaluations to the key questions.

Initially, the comparison between the original data (case OA) and 1-window sampling (case O1) can be considered, and it can be seen that the overall accuracy increases from 48.6% to 81.7%. Also, the F_1 scores of all behaviors increase markedly, especially for sitting, standing and going up and down the stairs. Then, the comparison between 1 (case O1) and n windows (case On) can be made. It can be seen that the overall accuracy increases very weakly, from 81.7% to 81.8%. Moreover, the F_1 scores increase only slightly or decrease, and the only notable increase is for going upstairs. Finally, the comparison between n windows while using only the original data (case On) and the same while also using surrogates can be made. It can be seen that the overall accuracy increases again substantially, from 81.7% to 88.7%. Besides weak drops for going downstairs and standing, the F_1 scores generally increase, particularly for walking, jogging and going upstairs, reaching an improvement of about 10% in the case of jogging.

The initial confusion matrix, shown in **Fig. 5.12a**, is clearly unusable, but the final one, shown in **Fig. 5.12b**, indicates a system that performs improved classification. As the best result is obtained combining surrogate and sampling, the conclusions are exactly the same as the cow data.

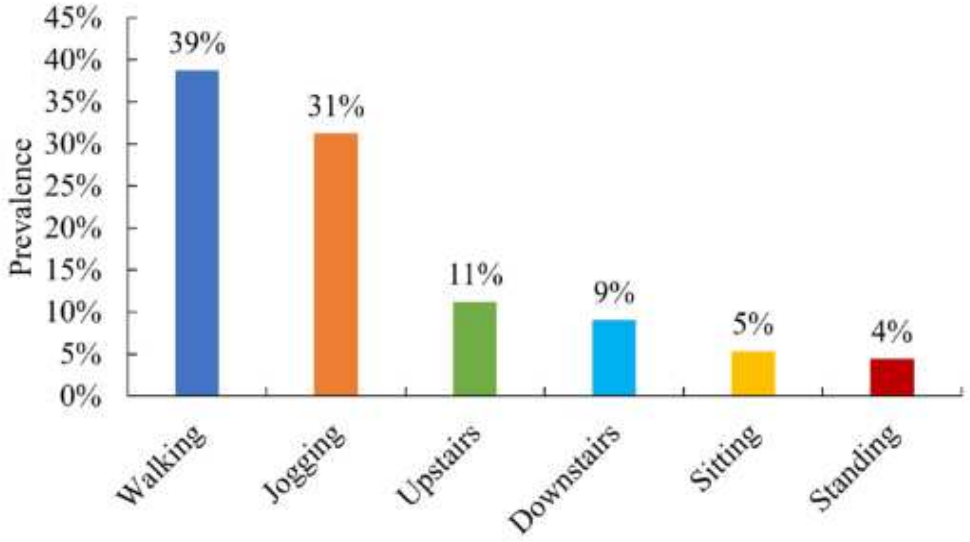


Figure 5.11: Relative prevalence (normalized) in the human behavior dataset.

Table 5.3: F_1 scores and overall accuracy in the human behavior dataset

	OA	**O1**	**On**	**Mn**
Walking	65.0%	84.4%	82.7%	90.7%
Jogging	66.7%	84.8%	84.9%	94.6%
Upstairs	5.6%	62.3%	70.7%	76.4%
Downstairs	29.6%	77.6%	70.8%	69.0%
Sitting	N/A	100.0%	96.3%	100.0%
Standing	37.8%	98.4%	98.0%	95.5%
Overall	48.6%	81.7%	81.8%	88.7%

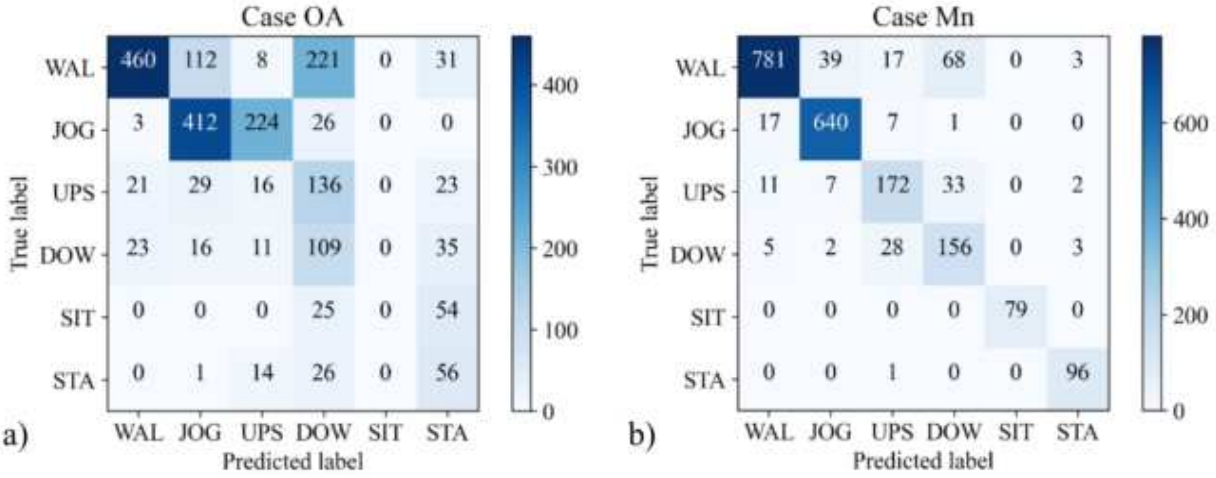


Figure 5.12: Confusion matrices for the original data (a), case OA, and the augmented data (b), case Mn.

5.5.3 Second additional dataset

The second dataset is about brain electrical activity recorded using the electroencephalography (EEG), which is sensitive to voltage changes on the head surface. It is known as the *BCI2000 EEG Motor Movement/Imagery Dataset* and also considered the most popular in its area [22,23]. This dataset contains mainly signals recorded while a person is moving or imagining to move. However, because these are extremely difficult to analyze, I focus instead on the comparison between eyes open and closed. It is known that when a person closes their eyes, activity around 10 Hz, known as alpha band, increases towards the back of the head, known as occipital pole [30]. One minute of eyes open and one minute of eyes closed data were taken for 20 subjects, for three electrodes known as O1, Oz and O2, giving a total of 234,336 data rows acquired at 160 Hz over ≈24.4 min. In this case, the window size was set to 640 points, and 80% of data were used for training and validation, 20% for testing. To test the suggested data augmentation approach, as seen in **Fig. 5.13**, the data

were made imbalanced by keeping only about 20% of the eyes open condition. As suggested for this data, a band-pass filter between 2 Hz and 40 Hz was applied, and the mean signal was subtracted [22,23,30].

Initially, the comparison between the original data (case OA) and 1-window sampling (case O1) was considered, and it can be seen that the overall accuracy increases from 56.3% to 91.7%. This is similar to the first additional dataset, even though in this case there are only two classes. Also, the F_1 scores of both conditions, especially eyes open, increase markedly. Then, the comparison between 1 (case O1) and n windows (case On) can be made. It can be seen that the overall accuracy increases more weakly, from 91.7% to 93.1%. The F_1 scores increase slightly. Also in this aspect the situation is similar to the previous dataset. Finally, the comparison between n windows while using only the original data (case On) and the same while also using surrogates can be made. It can be seen that the overall accuracy increases again, from 93.1% to 95.8%. The F1-scores also increase, especially for eyes open.

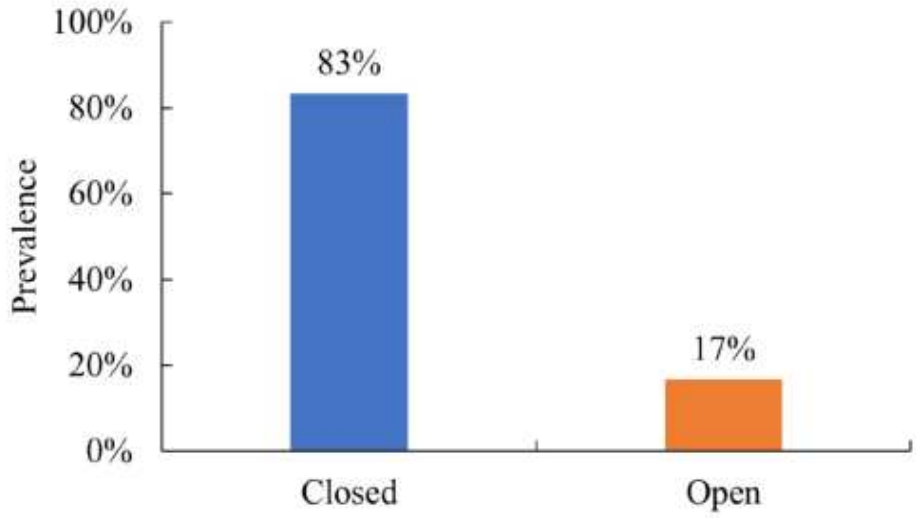


Figure 5.13: Relative prevalence (normalized) in the EEG dataset.

The initial confusion matrix, shown in **Fig. 5.14a**, indicates more misclassifications that the final one, shown in **Fig. 5.14b**. Also in this case, it is concluded that combining surrogate and sampling is best, in agreement with the cow data and the first additional dataset.

In addition, because inter-individual differences are an important topic in human studies, training and evaluation were repeated, completely separately, for each individual participant [22,23]. On average, the overall accuracy across individuals was 53.7±9.4% for case OA, 97.7±4.7% for case O1, 98.2±3.3% for case On, 98.0±3.8% for case Mn. Then, the percentage of cases in which the best performance was attained for a given case was calculated (it should be considered that the same best value can be attained in several cases, so, it is counted for all of them). This was 0% for case OA, 80% for cases O1 and On, and 75% for case Mn. These results show that, when considering each person separately, the biggest difference in accuracy was due to reducing the dataset imbalance using sampling. Because already very high accuracies were obtained using 1- and n-window sampling (cases O1 and On), there was a ceiling effect, so the addition of surrogates did not bring additional advantage. When training and evaluating the model on all participants together, the accuracies were lower, so there was room for surrogates to provide an additional improvement.

Table 5.4: F_1 scores and overall accuracy in the EEG dataset

	OA	**O1**	**On**	**Mn**
Closed	69.6%	95.0%	95.9%	97.6%
Open	22.2%	75.0%	76.2%	85.7%
Overall	56.3%	91.7%	93.1%	95.8%

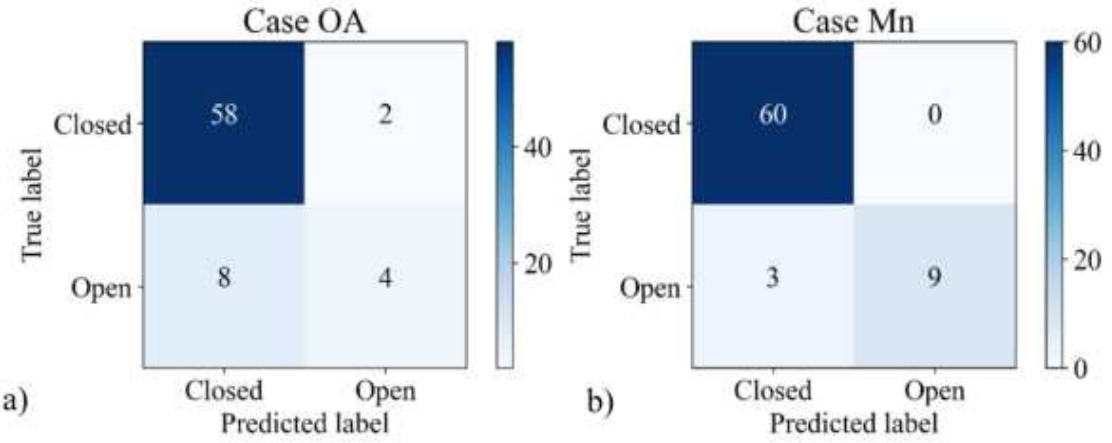


Figure 5.14: Confusion matrices for the original data (a), case OA, and the augmented data (b), case Mn.

5.5.4 Third additional dataset

The third dataset, instead, is related to industrial applications of IoT and was recorded using a machinery that simulates different types of mechanical fault of a large electric motor. It is known as the *MAFAULDA Machinery Fault Database* and considered a key reference in its field [24]. This dataset contains time series recorded from two triaxial accelerometers placed on different points of the mechanical device, a tachometer and a time series from the sound as receive by a microphone. It contains more than 97 million data rows, sampled at 25 kHz, therefore it is much bigger than the cow dataset and the first and second additional datasets. It contains seven operating conditions, one is normal, and the other represent various types of mechanical faults such as horizontal or vertical misalignment obtained by moving the motor, imbalance of the load weight, and so-called overhang and underhang, which indicate failure of the ball bearings. Compared to the cow and human behavior datasets, in this dataset the classes are different because they represent failures that generate strong vibrations. These vibrations do not change over time like natural behavior but remain

present, still, they have to be classified to support maintenance [31]. As can be seen in **Fig. 5.15**, the distribution of classes is also very imbalanced, with 26% and 29% for overhang and underhang, 17% for imbalance, 15% and 10% for horizontal and vertical misalignment, and only 3% for normal operation. This is because the authors of the dataset were interested in capturing each possible failure extensively even though, in real life, normal operation actually takes place most of the time.

After applying the same procedures indicated in **Fig. 5.5**, the F_1 scores and overall accuracy reported in **Table 5.5** are obtained. Also here, only four cases are considered, to reduce the amount of analyses to the key questions.

Initially, the comparison between the original data (case OA) and 1-window sampling (case O1) can be considered, and it can be seen that the overall accuracy increases from 91.9% to 92.6%, which is less than the previous two additional datasets. The F_1 score of the rarest behavior, normal operation, increases markedly, while the other only have small changes. Then, the comparison between 1 (case O1) and n windows (case On) can be made. It can be seen that the overall accuracy increases similarly, from 92.6% to 94.6%. Some F_1 scores decrease only a little, while the scores for normal, horizontal and vertical increase by up to about 10%. Finally, the comparison between n windows while using only the original data (case On) and the same while also using surrogates can be made. It can be seen that the overall accuracy increases again substantially, from 94.6% to 99.7%. No F_1 scores decrease, while particularly large increases are found for normal and horizontal, and all values approach 100%.

The initial confusion matrix, shown in **Fig. 5.16a**, indicates a not perfect classification, but the final one, shown in **Fig. 5.16b**, indicates a system that performs near-perfect classification. Finally, also in this case the same way as the previous ones, the best result is obtained combining surrogate and sampling.

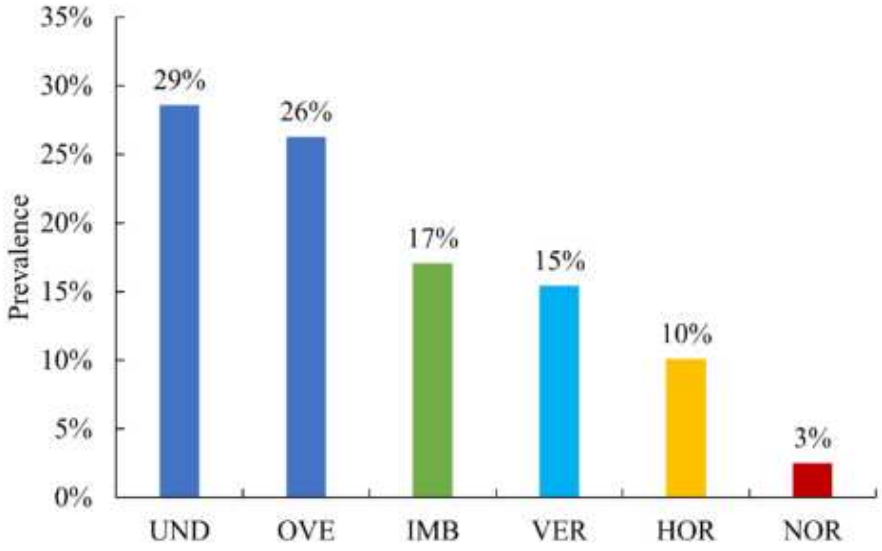


Figure 5.15: Relative prevalence (normalized) in the motor failure dataset. UND: Underhang, OVE: Overhang, IMB: Imbalance, VER: Vertical, HOR: Horizontal, NOR: Normal.

Table 5.5: F_1 scores and overall accuracy in the motor failure dataset

	OA	**O1**	**On**	**Mn**
Normal	11.0%	58.2%	68.7%	97.1%
Horizontal	75.1%	76.0%	87.5%	99.8%
Vertical	89.3%	87.9%	97.1%	99.6%
Imbalance	95.1%	96.8%	92.3%	99.5%
Overhang	99.3%	98.5%	97.9%	100.0%
Underhang	95.6%	95.0%	97.4%	99.8%
Overall	91.9%	92.6%	94.6%	99.7%

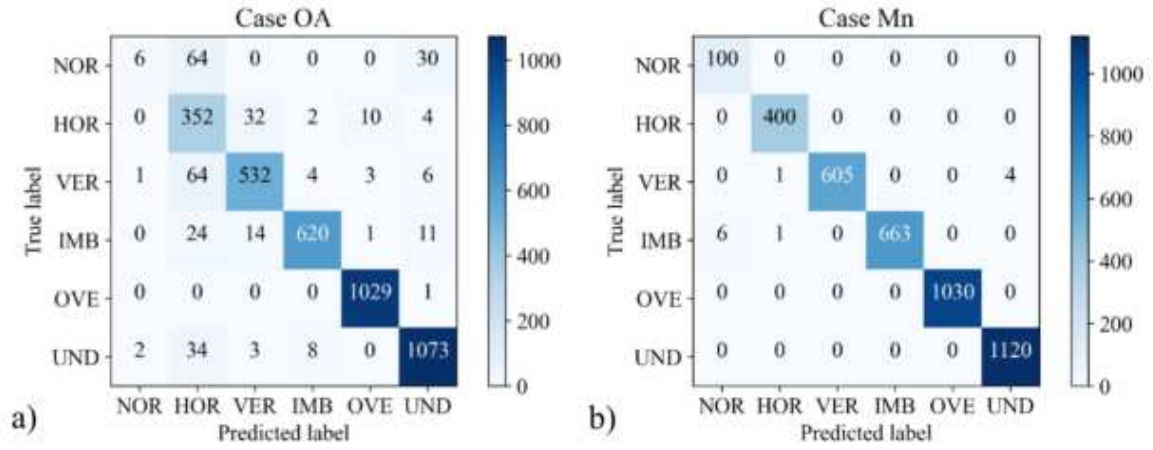


Figure 5.16: Confusion matrices for the original data (a), case OA, and the augmented data (b), case Mn.

5.6 Conclusion

This Chapter proposed using a new combination of time series augmentation methods especially suitable for short, low-dimensional sensor time series, aiming to address the challenges stemming from small dataset size and dataset imbalance. The case of cattle behavior recognition using a CNN-based classifier was considered, however, the results were shown to be relevant beyond the specific example under consideration. The key finding is that performance is maximized when combining a suitable random sampling scheme with surrogate data and integrating it within the training process to realize so-called online augmentation. These results especially demonstrate that dynamically sampling time series snippets during each epoch can facilitate the training process by expanding the classifier boundaries. Furthermore, introducing a biased coverage that compensates for the imbalanced original distribution also enhances the performance, not only for the least represented behavior classes. The issue of limited dataset size is effectively addressed through Fourier surrogates,

which support high training performance while ensuring the absence of any duplication in the time-domain data submitted to the training algorithm. Extending previous works wherein this technique was used empirically, I show how a deductive approach can be used to dissect and identify explicitly which properties retained from the original time series are most relevant. On the whole, using the proposed method improved the average performance from 90% to 96%, and the classification accuracy of grazing from 45% to 91%, without requiring any modification to the classifier architecture. In addition, it was found that the combination of surrogates with sampling gave the best result always also in other datasets of human behavior, EEG and machine failure.

5.7 Bibliography

[1] H. Kantz and T. Schreiber, Nonlinear time series analysis. Cambridge university press, 2004, vol. 7.

[2] R. Hegger, H. Kantz, "Practical implementation of nonlinear time series methods: The TISEAN package," Chaos, vol. 9, pp. 413, 1999.

[3] J. Theiler, S. Eubank, A. Longtin, B. Galdrikian, and J. D. Farmer, "Testing for nonlinearity in time series: the method of surrogate data," Physica D: Nonlinear Phenomena, vol. 58, no. 1-4, pp. 77–94, 1992.

[4] T. Schreiber and A. Schmitz, "Surrogate time series," Physica D: Nonlinear Phenomena, vol. 142, no. 3-4, pp. 346–382, 2000.

[5] T. Schreiber and A. Schmitz, "Improved surrogates data for nonlinearity tests," Phys. Rev. Lett., vol. 77, pp. 635-638, 1996.

[6] T. K.-M. Lee, H. Chan, K. Leo, E. Chew, L. Zhao, and S. Sanei, “Surrogate data for deep learning architectures in rehabilitative edge systems,” in Proc. Signal Process., 2020, pp. 30–35.

[7] A. Graps, "An introduction to wavelets," in IEEE Computational Science and Engineering, vol. 2, no. 2, pp. 50-61, Summer 1995.

[8] C. J. Keylock, “Hypothesis testing for nonlinear phenomena in the geosciences using synthetic, surrogate data,” Earth and Space Science, vol. 6, no. 1, pp. 41–58, 2019.

[9] C. J. Keylock, “A wavelet-based method for surrogate data generation,” Physica D: Nonlinear Phenomena, vol. 225, no. 2, pp. 219–228, 2007.

[10] C. Keylock, “Improved preservation of autocorrelative structure in surrogate data using an initial wavelet step,” Nonlinear Processes in Geophysics, vol. 15, no. 3, pp. 435–444, 2008.[11] A. Krizhevsky, “Learning multiple layers of features from tiny images,” 2009.

[12] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vision Pattern Recognit., pp. 770–778, 2016.

[13] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.

[14] B. K. Iwana and S. Uchida, “An empirical survey of data augmentation for time series classification with neural networks,” PloS One, vol. 16, no. 7, p. e0254841, 2021.

[15] Q. Wen et al., “Time series data augmentation for deep learning: A survey,” in Proc. Int. Jt. Conf. Artif. Intell., pp. 4653–4660, 2021.

[16] Y. Ge, X. Xu, S. Yang, Q. Zhou, and F. Shen, ”Survey on sequence data augmentation”, J. Front. Comput. Sci. Technol.,” vol. 15, no. 7, pp. 1207-1219, 2021.

[17] M. Abadi et a., “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015, software available from tensorflow.org. [Online]. Available at: http://tensorflow.org”. Accessed on Dec 20, 2022.

[18] T. E. K. Lee, Y. Kuah, K.-H. Leo, S. Sanei, E. Chew, and L. Zhao, "Surrogate rehabilitative time series data for image-based deep learning," in Proc. 27th Eur. Signal Process. Conf., 2019, pp. 1–5.

[19] J. T. Schwabedal, J. C. Snyder, A. Cakmak, S. Nemati, and G. D. Clifford, "Addressing class imbalance in classification problems of noisy signals by using fourier transform surrogates," arXiv preprint arXiv:1806.08675, 2018.

[20] "WISDML: WIreless Sensor Data Minning." [Online]. Available at: https://www.cis.fordham.edu/wisdm/dataset.php. Accessed on Dec 20, 2022.

[21] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," ACM SigKDD Explorations Newsletter, vol. 12, no. 2, pp. 74–82, 2011.

[22] "EEG Motor Movement/Imagery Dataset" [Online]. Available at: https://physionet.org/content/eegmmidb/1.0.0/. Accessed on Dec 20, 2022.

[23] G. Schalk, D. J. McFarland, T. Hinterberger, N. Birbaumer, and J. R. Wolpaw, "Bci2000: a general-purpose brain-computer interface (bci) system," IEEE Transactions on biomedical engineering, vol. 51, no. 6, pp. 1034–1043, 2004.

[24] "Machinery Fault Database" [Online]. Available at: http://www02.smt.ufrj.br/~offshore/mfs/page_01.html. Accessed on Dec 20, 2022.

[25] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, "The industrial internet of things (iiot): An analysis framework," Computers in industry, vol. 101, pp. 1-12,2018.

[26] S. Greengard, The internet of things. MIT press, 2021.

[27] J. H. Nord, A. Koohang, and J. Paliszkiewicz, "The internet of things: Review and theoretical framework," Expert Systems with Applications, vol. 133, pp. 97–108, 2019.

[28] W. C.-C. Chu, C. Shih, W.-Y. Chou, S. I. Ahamed, and P.-A. Hsiung, "Artificial intelligence of things in sports science: weight training as an example," Computer, vol. 52, no. 11, pp. 52–61, 2019.

[29] Y. Lin et al., "Artificial Intelligence of Things Wearable System for Cardiac Disease Detection, " 2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS), 2019, pp. 67-70.

[30] W. Hohaia, B. W. Saurels, A. Johnston, K. Yarrow, and D. H. Arnold, "Occipital alpha-band brain waves when the eyes are closed are shaped by ongoing visual processes," Scientific Reports, vol. 12, no. 1, pp. 1–10, 2022.

[31] Y. Lee, C. Kim, and S. J. Hong, "Industrial internet of things for condition monitoring and diagnosis of dry vacuum pumps in atomic layer deposition equipment," Electronics, vol. 11, no. 3, p. 375, 2022.

Chapter 6

Conclusion and Future Work

6.1 Overview of contributions

This thesis aimed to provide a significant and concrete improvement to the accuracy of classifying cattle behaviors based on tri-axial accelerometer time series. The specific objectives were two: (1) improve the overall classification accuracy, and (2) improve the classification accuracy of the least frequent behaviors, providing a system which is not biased towards better recognizing the more frequent behaviors. At a more general level, this thesis represents a comprehensive journey through the development of a data augmentation method tailored for a specific application in farming IoT but potentially, in the future more broadly useful. The developments presented in this work are also representative of the evolution of these techniques during a phase of rapid expansion of data augmentation from the domain of computer vision to other applications.

This thesis considered three related but distinct approaches:

- A basic one, based on system-specific considerations about what physical transformations the classification should be invariant to, i.e., collar rotation.

- An intermediate one, combining this approach with other empirical arguments regarding the recombination of data from different time points and other alterations considered not to alter the recognizability of a behavior.
- A more abstract one, removing any empirical or system-specific considerations and replacing them with methods based on theoretical considerations about signal content.

The novelty is summarized in **Table 6.1**. It can be seen that there are several different aspects of novelty. As regards the sensor rotation, the algorithm is new in its application to data augmentation and class imbalance mitigation. As regards the recombination and reversal, the novelty consists of important improvements, specifically, the usage of multiple windows to exact greater variability from the data, and the usage of reversal in time instead of amplitude. By contrast, the idea of compensating for data loss is entirely new. About the usage of IAAFT surrogates (also referred to as Fourier surrogates), the previous work is extremely limited, and this study is the first using a multivariate approach. Finally, both 1- and n-window sampling and integrated augmentation are entirely new ideas in time series augmentation. It can be seen that these aspects of novelty complement each other at different levels.

They are, however, distinguished by a progression at the levels of both abstraction and complexity. Facing the challenges posed by realizing data augmentation methods that are both powerful in boosting accuracy and practically applicable, this thesis provides a diversified toolkit which can, in the future, be applied to similar problems in animal behavior classification and beyond. At a more general level, this thesis demonstrates that multiple, conceptually different approaches can be combined usefully for the purpose of data augmentation: on the one hand with system-specific operations such as axis rotation, and on the other with generic manipulations that make no particular assumptions about the signal

content. In these senses, it arguably represents the most systematic examination of these approaches to date.

Table 6.1: Summary of the aspects of novelty.

Algorithm or concept	Novelty status
Sensor rotation	**New** when applied towards data augmentation and compensating for imbalanced dataset.
Recombination	**Improvement** over a known method (multiple windows)
Reversal	**Improvement** over a known method (time not magnitude)
Compensate data loss	**Entirely new** idea
IAAFT surrogates	Very limited previous work not applying them coherently, this is **first systematic study**, also substantial improvement because of multivariate approach.
1- and n-window sampling	**Entirely new** idea
Integrated augmentation	**Entirely new** for time series data

Considering the several possible application scenarios, it is possible to ask the question: which data augmentation approach should be chosen? A try to answer this question is provided by the flow chart shown in **Fig. 6.1**. It can be seen that there are two steps. The first step asks if it is necessary, besides addressing class imbalance, to achieve invariance to a physical parameter. In this case, the answer was a yes and the parameter was collar rotation. In such cases, it may be better to simulate variation of that parameter, Approach 1. If this is not necessary, one may decide based on computational load and dataset size, because the

surrogate plus sampling approach can be computationally very loading. Then, in the presence of a large dataset one may prefer the simpler empirical methods, in Approach 2, otherwise Approach 3 should be preferred. This flow chart is only an attempt to organize the results, but it is difficult to give final indications. In the present case, for example, even though collar rotation was present, the best results were obtained using the Fourier surrogates. So, it may always be useful to try more than one approach.

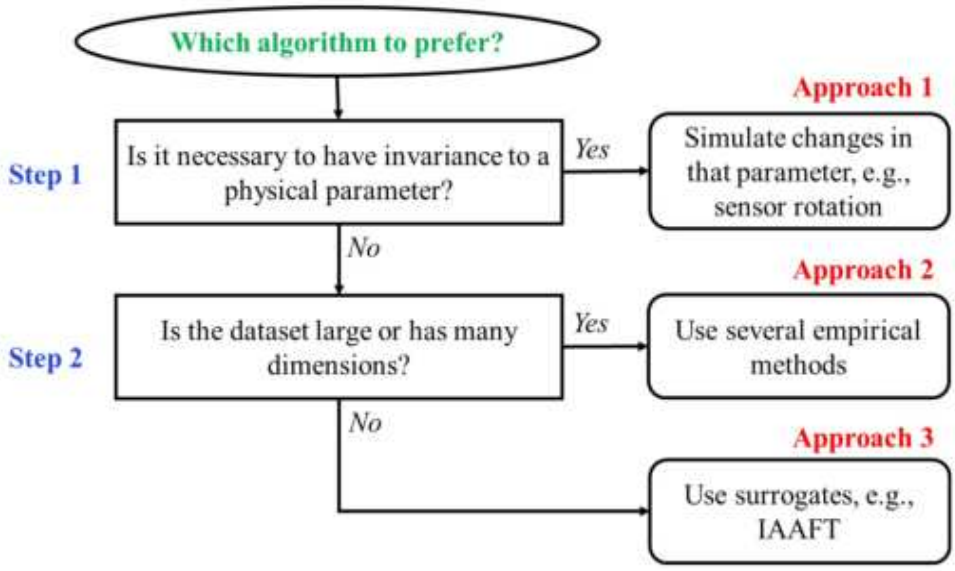


Figure 6.1: Conceptual flow chart for the selection of the most suitable data augmentation approach.

In practice, for the application that I focus on, the first method provided an accuracy improvement of 2.5 - 37.1% for various behaviors, starting from a baseline of the overall accuracy of 77% and reaching 98%; however, it was based on a relatively big LSTM network and trained only on 2 cows. The second method provided an improvement of overall accuracy from 83% to 92%; it was based on a more compact CNN network and trained on 6 cows. The third method provided a further improvement and reached an overall accuracy

of 96%; importantly, for the most problematic behavior, namely grazing, the accuracy increased from 45% to 91% and there was not any change in layer number and size from the second method.

From the perspective of the specific engineering challenge being handled, the present thesis offers solutions reaching a classification accuracy that would otherwise not be accessible. Because precision livestock farming requires observing as early as possible some subtle changes in the frequency of the cattle behaviors, the increased accuracy has a major impact on the viability of the system in concrete applications. As the efforts in this field progress, the data augmentation techniques proposed in this work have an important potential to offer in predicting the high level status of cattle. This study focused on the most prevalent behaviors, which are also those for which deviations from normality is most sensitive. Extending this to a large multitude of classes, also including postures for example, should be a straightforward extension of this work. The eventual goal is attaining the earliest possible disease detection, and the best sensitivity to calving, estrus, or abnormal behavior under stress. Not less importantly, the high quality behavior classification supported by the techniques introduced in this thesis will also allow the generation of high quality sets. Then, this will enable a much better understanding of the complex relationships between behavioral alterations and disease states [1-5].

6.2 Discussion of future work

6.2.1 Testing for other applications

As the additional datasets have initially shown, the methods presented in this thesis seem, on various levels, perhaps applicable also to many other scenarios in edge devices. On the one hand, tri-axial accelerometers have become nearly used everywhere and are

commonly found in wearable devices such as smartwatches and ear buds [6]. Moreover, they are widely used across a range of systems for animal behavior monitoring, extending beyond cattle for example to pigs and horses, and beyond farming itself, encompassing even pet animals such as dogs [7-9]. For such sensors, all the methods developed in this thesis are in principle applicable. The results obtained using the human behavior dataset confirm that the surrogates and sampling methods might also be useful for other datasets, as they consistently give improved accuracy. In fact, the notion of random sensor rotation can in some cases be extended from rotation around an axis to an arbitrary spherical angle. The present methods should be deployed and tested in such applications, where a substantial performance improvement is expected with a minimum or no adaptation required to the technology.

On the other hand, set aside for the axis rotation, all the other methods might be applicable also for other time series that are not related to movement but still record some behavior that should be recognized. This was initially confirmed by the analyses performed using the surrogates and sampling on the electroencephalography and machine failure additional datasets. One additional example may include the identification of electrical loads based on their current absorption, or the classification of vehicles based on their weight or metal mass moving over a sensor [10,11]. For all these applications, methods such as time series reversal and recombination might be usable, and the surrogate-based approach has immediate relevance because it is the most abstract one. Importantly, even though the results in this thesis indicate that autocorrelation may be more important than cross-correlation, the situation could be different for other datasets, and the method fully caters for this possibility.

It should be pointed out that several types of cow breeds exist, and can be quite different in their weight as well as temperament, for example, more or less aggressive and active.

While the basic behaviors overall are the same, of course, it is possible that physical differences between the breeds, for instance due to weight, as well as subtle or big differences in how the behaviors are played out, are present. The detailed investigation of these aspects is a topic for veterinary science, and several analyses have been published [12-14]. It appears likely that the classifier may need to be retrained when moving from one breed to another. On another hand, the proposed techniques could also be useful for monitoring the behaviors of other species, such as sheep or dogs [15-17]. In this case, it needs to be considered that the types of behaviors can be different, for example, rumination and salting may not exist and feeding can look quite different. Clearly, in such cases the systems needs a specific dataset for training. Overall, because the techniques proposed especially in Chapter 5, are quite abstract and work on other datasets (including human behavior), there is no reason to expect they could not be applied beyond cattle monitoring. This, actually, also applies to Chapter 3 and 4 insofar as the sensor is mounted on a collar, which remains the most common approach not only for cattle, because collar rotation is not only a specific problem for cows [18]. One interesting aspect is the possibility of transfer learning, that has been proposed by another paper in this area [19,20]. That means, the models trained for one or more cow types of interest could be used as a starting point to train specifically towards other cow breeds; in this case, the starting point information greatly helps to reach high performance even on a different dataset. To a more limited extent, transfer learning could also be applicable between species, even though the output layer of the classifier would in many cases be different due to the presence/absence of other behaviors.

To completely express the capability of the methods presented in this thesis in impacting the usefulness of Edge AI technologies and therefore society, future works should comprehensively chart the performance of these methods across diverse applications, including monitoring other cattle breeds and/or other animal species. There is the need to analyze in the future in a systematic and extensive way whether the results proposed in this thesis are

also helpful more in general for other applications, where many aspects can be different, including the type of sensor, content of the signal and behaviors, and so on.

6.2.2 Further advancements in surrogate generation

Throughout this thesis, simple trigonometric or algebraic manipulations were applied at first, followed by Fourier surrogates. As discussed in the previous chapter, this transition is due to going from an approach starting from considerations about the system, to an approach based on the abstract aspects of signal content. Of course, there is a very large number of other approaches to the generation of new time series for data augmentation that could be considered. In this sense, the present thesis should be taken as laying a track towards future work, not as an exhaustive examination of all possible time series augmentation techniques.

For example, surrogate generation by means of Fourier transformation, although convenient, was initially developed for a different application. Other approaches may be superior in terms of capturing particular signal features, such as situations in which two behaviors alternate within the same time window [21]. An example would be generative adversarial networks, which, however, are complex in themselves to train [22]. Other possibilities to generate additional time series could be vector autoregressive models, or dynamical systems [23]. There is the possibility of merging together approaches such as Fourier surrogates with time-domain manipulations such as recombination, and so on.

Even through the performance attained in this thesis by means of Fourier surrogates was satisfactory, future work should consider a wide range of generation approaches and systematically compare them. In this sense, the development of a software framework could be very helpful to allow easy integration with the network training process.

6.2.3 Consideration of other classifiers

The numerical experiments presented in this thesis mainly considered a convolutional neural network (CNN), which is increasingly prevalent in time series classification problems owing to the fact that it does not require a separate feature extraction step, simplifying edge implementation in both hardware and software compared to feature based approaches such as multi-layer perceptrons (MLP) [24]. However, the initial analysis was based on a long short-term memory (LSTM) network, another popular classifier for time series problems. These are only two examples among the many possible classifiers usable for this type of applications [25]. Others include multi-layer perceptron (MLP) networks, which are considerably smaller than CNNs but require feature extraction and feature vector construction, recurrent neural networks (RNN), which may offer among the most compact models possible, and spiking neural networks (SNN), which have particular advantages for low-power implementation [26-28].

Of course the benefits of data augmentation are not unavoidably tied to CNN-based classifiers. There is actually no specific connection. Then, there is a need for future work to explore in a more systematic manner the benefits and issues of all the proposed techniques across these network types. One may actually imagine an exploration plane for future work, having on one axis the data augmentation type, and on the other axis the classifier type. This is, in fact, a largely unexplored field. On the one hand, there remains the open question whether data augmentation helps more or less depending on the network size. One could expect that it is more important for large networks with many trainable parameters, because their training is more difficult but they are more able to get high accuracy in principle. However, this has not been proven and was not studied in this thesis. On the other hand, there is the question of the classifier type. That means, different types of neural networks such as MLP, RNN and SNN in addition to CNN, but also other types of classifiers that are

not neural networks, such as support vector machines (SVM) and decision trees (DT). Right now, it is unknown whether data augmentation is more suitable for some of them or not, and this needs to be addressed. This is also important because some classifiers tend to be larger to implement than others, and this has implications for the hardware realization on an edge device

In fact, because CNN, MLP, RNN and SNN classifiers have different strong points, they are likely to all remain in wide use for the foreseeable future, with particular applications leveraging the strong aspect of one or the other depending on the circumstance. Therefore, evaluating and confirming the actual generality or otherwise of the proposed data augmentation algorithms is priority. It should be reminded that a key aspect was found to be the integration of data augmentation with network training, therefore, there is a need to create a software framework to facilitate this. In future work, such a framework should allow combining in a seamless way any combination of network type and data augmentation type.

Another aspect is the topology of the network to consider. In this thesis, the best network type was selected at each stage, and its topology was then kept fixed. That's because otherwise the number of possible combinations of setting would be not manageable. The deep learning network used in Chapter 5 was very robust because it could also give high performance on totally different data sets. However, future work should still examine if the best topology changes after data augmentation. Because the data augmentation gives new data but not change the features of the behaviors, probably a significant change will not be seen.

6.2.4 Hardware and system-level implications

The remarkable results obtained for accuracy throughout this thesis confirm that the technology of data augmentation is a key enabler of high performance levels in time series classification. However, surely there are many other aspects that should be considered at the level of designing an edge AI system. Future work should address the impact of data augmentation on power consumption, in terms of the possibility of attaining similar performance with smaller networks [29]. Further, as mentioned above, future work should check the impact of data augmentation on the performance of different network types. For example, if data augmentation allows replacing a CNN network with a typical RNN or SNN network, the impact on power consumption could be considerable.

On a practical note, in Chapter 4, it was proposed that the CNN network can be implemented on three popular microcontroller platforms. Moving forward, it will be useful to perform those implementations and conduct comparisons in terms of measured power consumption and other aspects.

Not less important, this thesis has demonstrated that data augmentation, when implemented in a rigorous way founded on theoretical principles, is also an effective way of understanding exactly which aspects of a signal contain the information necessary for good classification. It was found that, in the present case, mean, variance and autocorrelation are the key aspects. Therefore, future work should use this approach as a way of driving which features are extracted for hardware-based classification. While this may be unnecessary for CNN, RNN and SNN networks, the extraction of a compact and robust set of features is extremely important for MLP networks and other, small-size classifiers such as decision trees.

6.3 Bibliography

[1] P. Giuseppe and C. Giovanni, "Biological rhythm in livestock," Journal of Veterinary Science, vol. 3, no. 3, pp. 145–158, 2002.

[2] M. A. von Keyserlingk and D. M. Weary, "A 100-Year Review: Animal welfare in the Journal of Dairy Science—The first 100 years," Journal of Dairy Science, vol. 100, no. 12, pp. 10 432–10 444, 2017..

[3] J. Hancock, "Studies of grazing behaviour in relation to grassland management I. Variations in grazing habits of dairy cattle," The Journal of Agricultural Science, vol. 44, no. 4, pp. 420–433, 1954.

[4] S. Ayadi, A. Ben Said, R. Jabbar, C. Aloulou, A. Chabbouh, and A. B. Achballah, Dairy Cow Rumination Detection: A Deep Learning Approach. Springer International Publishing, 2020, vol. 1348.

[5] P. Llonch, E. Mainau, I. R. Ipharraguerre, F. Bargo, G. Tedó, M. Blanch, and X. Manteca, "Chicken or the Egg: The reciprocal association between feeding behavior and animal welfare and their impact on productivity in dairy cows," Frontiers in Veterinary Science, vol. 5, no. DEC, 2018..

[6] E. S. Jeon, A. Som, A. Shukla, K. Hasanaj, M. P. Buman, and P. Turaga, "Role of data augmentation strategies in knowledge distillation for wearable sensor data," IEEE Internet of Things Journal, 2021.

[7] V. M. Suresh, R. Sidhu, P. Karkare, A. Patil, Z. Lei, and A. Basu, ''Powering the IoT through embedded machine learning and Lora,'' in Proceedings of IEEE 4th World Forum on Internet of Things (WF-IoT), Feb. 2018, pp. 349–354.

[8] J. P. Dominguez-Morales, L. Duran-Lopez, D. Gutierrez-Galan, A. Rios-Navarro, A. Linares-Barranco, and A. Jimenez-Fernandez, "Wildlife monitoring on the edge: A

performance evaluation of embedded neural networks on microcontrollers for animal behavior classification," Sensors, vol. 21, no. 9, p. 2975, Apr. 2021.

[9] P. Kumpulainen, A. V. Cardó, S. Somppi, H. Törnqvist, H. Väätäjä, P. Majaranta, Y. Gizatdinova, C. H. Antink, V. Surakka, M. V. Kujala et al., "Dog behaviour classification with movement sensors placed on the harness and the collar," Applied Animal Behaviour Science, vol. 241, p. 105393, 2021.

[10] M. Bourdeau, P. Basset, S. Beauchêne, D. Da Silva, T. Guiot, D. Werner, and E. Nefzaoui, "Classification of daily electric load profiles of non-residential buildings," Energy and Buildings, vol. 233, p. 110670, 2021.

[11] X. Chen, X. Kong, M. Xu, K. Sandrasegaran and J. Zheng, "Road Vehicle Detection and Classification Using Magnetic Field Measurement," IEEE Access, vol. 7, pp. 52622-52633, 2019.

[12] C. M. Pauler, J. Isselstein, J. Berard, T. Braunbeck, and M. K. Schneider, "Grazing allometry: anatomy, movement, and foraging behavior of three cattle breeds of different productivity," Frontiers in Veterinary Science, p. 494, 2020.

[13] R. Prendiville, E. Lewis, K. Pierce, and F. Buckley, "Comparative grazing behavior of lactating Holstein-Friesian, Jersey, and Jersey × Holstein-Friesian dairy cows and its association with intake capacity and production efficiency," Journal of Dairy Science, vol. 93, no. 2, pp. 764–774, 2010.

[14] N. H. Sæther, K. E. B e, and O. Vangen, "Differences in grazing behaviour between a high and a moderate yielding norwegian dairy cattle breed grazing semi-natural mountain grasslands," Acta Agriculturae Scand Section A, vol. 56, no. 2, pp. 91–98, 2006.

[15] P. Chakravarty, G. Cozzi, A. Ozgul, and K. Aminian, "A novel biomechanical approach for animal behaviour recognition using accelerometers," Methods in Ecology and Evolution, vol. 10, no. 6, pp. 802–814, 2019.

[16] P. Chakravarty, M. Maalberg, G. Cozzi, A. Ozgul, and K. Aminian, “Behavioural compass: animal behaviour recognition using magnetometers,” Movement Ecology, vol. 7, no. 1, pp. 1–13, 2019.

[17] D. -N. Tran, T. N. Nguyen, P. C. P. Khanh and D. -T. Tran, "An IoT-Based Design Using Accelerometers in Animal Behavior Recognition Systems," IEEE Sensors Journal, vol. 22, no. 18, pp. 17515-17528, 15 Sept.15, 2022.

[18] A. da Silva Santos, V. W. C. de Medeiros, and G. E. Gonc ̧alves, “Monitoring and classification of cattle behavior: A survey,” Smart Agricultural Technology, p. 100091, 2022.

[19] V. Bloch, L. Frondelius, C. Arcidiacono, M. Mancino, and M. Pastell, “Cnn and transfer learning-based classification model for automated cows feeding behaviour recognition from accelerometer data,” bioRxiv preprint, 2022.

[20] K. Weiss, T. M. Khoshgoftaar, and D. Wang, “A survey of transfer learning,” Journal of Big data, vol. 3, no. 1, pp. 1–40, 2016.

[21] J. Lucio, R. Valdés, and L. Rodríguez, “Improvements to surrogate data methods for nonstationary time series,” Physical Review E, vol. 85, no. 5, p. 056202, 2012.

[22] S. Demir, K. Mincev, K. Kok, and N. G. Paterakis, “Data augmentation for time series regression: Applying transformations, autoencoders and adversarial networks to electricity price forecasting,” Applied Energy, vol. 304, p. 117695, 2021.

[23] Y. S. Perl, C. Pallavicini, I. P. Ipiña, M. Kringelbach, G. Deco, H. Laufs, and E. Tagliazucchi, “Data augmentation based on dynamical systems for the classification of brain states,” Chaos, Solitons & Fractals, vol. 139, p. 110069, 2020.

[24] B. Zhao, H. Lu, S. Chen, J. Liu and D. Wu, "Convolutional neural networks for time series classification," Journal of Systems Engineering and Electronics, vol. 28, no. 1, pp. 162-169, Feb. 2017.

[25] Y. Liu, Z. Su, H. Li and Y. Zhang, "An LSTM based classification method for time series trend forecasting," in Proceedings of 14th IEEE Conference on Industrial Electronics and Applications (ICIEA), 2019, pp. 402-406.

[26] A. Botalb, M. Moinuddin, U. M. Al-Saggaf and S. S. A. Ali, "Contrasting Convolutional Neural Network (CNN) with Multi-Layer Perceptron (MLP) for Big Data Analysis," in Proceedings of International Conference on Intelligent and Advanced System (ICIAS), 2018, pp. 1-5.

[27] M. Hüsken and P. Stagge, "Recurrent neural networks for time series classification," Neurocomputing, vol. 50, pp. 223–235, 2003.

[28] H. Fang, A. Shrestha and Q. Qiu, "Multivariate Time Series Classification Using Spiking Neural Networks," in Proceedings of 2020 International Joint Conference on Neural Networks (IJCNN), 2020, pp. 1-7.

[29] P. Ganesh, Y. Chen, X. Lou, M. A. Khan, Y. Yang, H. Sajjad, P. Nakov, D. Chen, and M. Winslett, "Compressing large-scale transformer-based models: A case study on bert," Transactions of the Association for Computational Linguistics, vol. 9, pp. 1061-1080, 2021.

Appendix A Publication List

A.1 Journal paper

[1] **Chao Li**, Korkut Kaan Tokgoz, Masamoto Fukawa, Jim Bartels, Takumi Ohashi, Kenichi Takeda, Hiroyuki Ito, "Data Augmentation for Inertial Sensor Data in CNNs for Cattle Behavior Classification," IEEE Sensors Letters, vol. 5, no. 11, pp. 1-4, 2021.

[2] **Chao Li,** Korkut Kaan Tokgoz, Naohiko Saito, Ayuka Okumura, Kazuhiro Toda, Hiroaki Matsushima, Takumi Ohashi, Kenichi Takeda, Hiroyuki Ito, "A Data Augmentation Method for Cow Behavior Estimation Systems Using 3-Axis Acceleration Data and Neural Network Technology," IEICE Trans. Fundamentals, vol. E105-A, no. 4, 2022.

[3] **Chao Li**, Ludovico Minati, Korkut Kaan Tokgoz, Masamoto Fukawa, Jim Bartels, Sihan A, Ken-ichi Tekeda, Hiroyuki Ito, "Integrated Data Augmentation for Sensor Time series in Cattle Behavior Recognition: Roles of Sampling, Balancing and Fourier Surrogates," IEEE Sensors Journal, vol. 22, no. 24, pp. 24230-24241, 2022.

A.2 International conference

[4] **Chao Li**, Korkut Kaan Tokgoz, Naohiko Saito, Ayuka Okumura, Kazuhiro Toda, Hiroaki Matsushima, Takumi Ohashi, Kenichi Takeda, Hiroyuki Ito, "Data Augmentation for Cow Behavior Estimation Systems Based on Neural Network Technology," Proceedings of the International Workshop on Smart Info-Media Systems in Asia, pp.95-100, Dec. 2020.

A.3 Domestic conference

[5] **Chao Li**, Hiroyuki Ito," Animal Behavior Estimation by Inertial Sensor and Machine Learning Technologies", IEICE-ICD Summer Academic Workshop, 2019.

[6] **Chao Li**, "Data Extension for a Real-time Cattle Behavior Estimation System Using Edge AI Technology", Outstanding Presentation Award in EISESiV-COI (Tokyo Tech's Center of Innovation) annual conference, 2020.

[7] **Chao Li**, Korkut Kaan Tokgoz, Ayuka Okumura, Jim Bartels, Masamoto Fukawa, Kazuhiro Toda, Hiroaki Matsushima, Takumi Ohashi, Kenichi Takeda, Hiroyuki Ito, "Data Augmentation of Inertial Sensor Data for Cattle Activity Classification using a Long-Short Term Memory Network," IEICE-ICD Young Researchers Forum, Mar. 2021.

[8] **Chao Li**, Korkut Kaan Tokgoz, Ayuka Okumura, Jim Bartels, Masamoto Fukawa, Kazuhiro Toda, Hiroaki Matsushima, Takumi Ohashi, Kenichi Takeda, Hiroyuki Ito, "Data Augmentation for Improving Deep Learning in Animal Behavior Classification," IEICE LSI and System Workshop, May. 2021.

[9] **Chao Li**, Hiroyuki Ito., "Cattle Behavior Recognition Using Deep Learning Methods on Limited Sensory Data", ICD-CAS Young Researchers Forum, December 2021.

[10] **Chao Li**, "Data Augmentation for Inertial Sensor Data in Deep Learning for Animal Behavior Estimation", Outstanding Presentation Award in EISESiV-COI (Tokyo Tech's Center of Innovation) annual conference, 2022.

A.4 Co-author

A.4.1 Journal paper

[11] Jim Bartels, Korkut Kaan Tokgoz, Asihan Tesrendashi, Masamoto Fukuwa, Shohei Otsubo, **Chao Li**, Ikumi Rachi, Kenichi Takeda, Ludovico Minati, Hiroyuki Ito, "TinyCowNet: Searching for Memory and Power Minimized RNNs, Implementable on Tiny Edge Devices for Lifelong Cow Behavior Distribution Estimation," IEEE Access, vol. 10, pp. 32706-32727, 2022.

[12] Ludovico Minati, Jim Bartels, **Chao Li**, Mattia Frasca, Hiroyuki Ito, "Synchronization phenomena in dual-transistor spiking oscillators realized experimentally towards physical reservoirs," Chaos, Solitons and Fractals, vol. 162, art. 112415, 2022.

[13] Ludovico Minati, **Chao Li**, Jim Bartels, Parthojit Chakraborty, Zixuan Li, Natsue Yoshimura, Mattia Frasca, Hiroyuki Ito, "Accelerometer time series augmentation through externally driving a non-linear dynamical system," Chaos, Solitons and Fractals, vol. 168, art. 113100, 2023.

A.4.2 International conference

[14] Hiroyuki Ito, Naohiko Saito, Chenyo Huang, Shogo Hata, Ayuka Okumura, Kazuhiro Toda, Hiroaki Matsushima, Junko Asakawa, **Chao Li**, Takumi Ohashi, and Kenichi Takeda, "Development Scheme for Cattle Behavior Estimation by Deep Learning in

an Edge Device," The 2nd International Conference on Precision Dairy Farming, June 18-19, 2019.

[15] Jim Bartels, Korkut Kaan Tokgoz, Masamoto Fukuwa, Shohei Otsubo, **Chao Li**, Ikumi Rachi, Kenichi Takeda, Hiroyuki Ito, "A 216 μW, 87% Accurate Cow Behavior Classifying Decision Tree on FPGA with Interpolated Arctan2," IEEE International Symposium on Circuits and Systems (ISCAS), Daegu, Korea, May 2021, pp. 1-5.

A.4.3 Domestic conference

[16] Hiroyuki Ito, **Chao Li**, Korkut Kaan Tokgoz, Kenichi Takeda, "Cattle Behavior Estimation by Edge AI Technology", IEICE General Conference 2020, AI-1-5, March 19, 2020.

[17] Hiroyuki Ito, **Chao Li**, Korkut Kaan Tokgoz, Takumi Ohashi, Kenichi Takeda, "A Study of Cow Monitoring System for Automatic Evaluation of Animal Welfare", IEICE Society Conference 2020, AI-1-3, Sept. 16, 2020.

[18] Jim Bartels, Korkut Kaan Tokgoz, Masamoto Fukawa, Shohei Otsubo, **Chao Li**, Ikumi Rachi, Kenichi Takeda, Hiroyuki Ito, "LBCE: A Low Power Behavior Classification Engine using Time-Multiplexed GRU", IEICE-ICD Young Researchers Forum 2020, Dec. 17-19, 2020.

[19] Ikumi Rachi, Korkut Kaan Tokgoz, Masamoto Fukawa, **Chao Li**, Jim Bartels, Asihan Tesrendashi, Sangyeop Lee, Kenichi Takeda, Hiroyuki Ito, "加速度センサを用いたウシ行動推定におけるサンプリングレートの検討," IEICE LSI and System Workshop, May. 2021.

[20] Hiroyuki Ito, Korkut Kaan Tokgoz, Ludovico Minati, **Chao. Li**, Takumi Ohashi, and Kenichi Takeda, “Listening to Silent Voices of Cows with Edge AI Technology,” in Annual Meeting Record, I.E.E. Japan, 2022.

Appendix B Data Augmentation Processing Core Function Code Flow

B.1 Random rotation-based data augmentation

```
def get_rotation_matrix(rad):
    rot = np.array([
            [1, 0, 0],
            [0, np.cos(rad), -np.sin(rad)],
            [0, np.sin(rad), np.cos(rad)],
        ])
    return rot

def rotation_interval(division_num):
    for state in each_status_row_count.keys():
        current_dir = data_path + "/" + state
        files = os.listdir(current_dir)
        files_file = [f for f in files if os.path.isfile(os.path.join(current_dir, f))]
        output_states_dir_path = base_path + "/../data/dst/rotated/" + state
```

```
    os.makedirs(output_states_dir_path, exist_ok=True)
    print("")
    current_file_num = 1
    for file_name in files_file:
      print("\r\033[K----- {2} {0}/{1} files -----".format(str(current_file_num),
     len(files_file), state), end="")
      current_file_num = current_file_num + 1
      file_path = current_dir + "/" + file_name
      target_data = pandas.read_csv(file_path, encoding='shift_jis')
      accs = target_data[['5', '6', '7']].values
      for i in range(division_num):
        degree = i*360/division_num
        rotation_matrix = get_rotation_matrix(np.pi*degree/180)
        rotated_accs = []
        for acc in accs:
          rotated_accs.append(list(np.dot(rotation_matrix, acc)))
        pd.DataFrame(rotated_accs).to_csv(output_states_dir_path + "/" + file_name +
    '_' + str(degree) + ".csv")

def rotation_random(times):
  rand_list = np.random.choice(list(range(360)), size= times, replace=True)
  print(rand_list)
  state = “Drinking”
  data_path = base_path + "/../data/dst/states"
  current_dir = data_path + "/" + state
  files = os.listdir(current_dir)
```

```
files_file = [f for f in files if os.path.isfile(os.path.join(current_dir, f))]
output_states_dir_path = base_path + "/../data/dst/rotated/" + state
os.makedirs(output_states_dir_path, exist_ok=True)
print(" success finding drinking for random rotation")
current_file_num = 1
for file_name in files_file:
    print("\r\033[K-----  {2}  {0}/{1}  files  -----".format(str(current_file_num),
      len(files_file), state), end="")
    current_file_num = current_file_num + 1
    file_path = current_dir + "/" + file_name
    target_data = pandas.read_csv(file_path, encoding='shift_jis')
    accs = target_data[['5', '6', '7']].values
    for i in range(len(rand_list)):
        degree = rand_list[i]
        rotation_matrix = get_rotation_matrix(np.pi*degree/180)
        rotated_accs = []
        for acc in accs:
            rotated_accs.append(list(np.dot(rotation_matrix, acc)))
        pd.DataFrame(rotated_accs).to_csv(output_states_dir_path + "/" + file_name + '_'
      + 'R' + str(degree) + '_' + str(i) + ".csv")
print("drinking \n\nComplete!")
```

B.2 Data processing with multiple empirical methods

```
def compensate_data_loss(specified_length):
    df_pairs = [extract_and_split(df, label)
                for label in range(5)]
    df_valid_concat = pd.concat([df_valid
                                 for df_valid, df_train in df_pairs],
                                 ignore_index=True)
    df_train_concat = pd.concat([df_train
                                  for df_valid, df_train in df_pairs],
                                  ignore_index=True)
    df_train_concat_s = df_train_concat.sample(frac=1)
    df_valid_concat_s = df_valid_concat.sample(frac=1)
    train_x_all = []
    train_y_all = []
    for i in range(df_train_concat_s.shape[0]):
        train_x_all.append(np.load(df_train_concat_s.iloc[i,0]))
        train_y_all.append(df_train_concat_s.iloc[i,1])
    valid_x_all = []
    valid_y_all = []
    for i in range(df_valid_concat_s.shape[0]):
        valid_x_all.append(np.load(df_valid_concat_s.iloc[i,0]))
        valid_y_all.append(df_valid_concat_s.iloc[i,1])
    filtered_x_y_pairs = [(x, y)
                          for x,y in zip(train_x_all, train_y_all)
                         ]
```

```
    train_x_filtered_temple = [x
                        for x,y in filtered_x_y_pairs]
    train_y_filtered = [y
                    for x,y in filtered_x_y_pairs]
    train_x_filtered = []
    for x,y in zip(train_x_filtered_temple,train_y_filtered):
        if len(x) < specified_length:
            train_x_filtered.append(np.tile(x,(int(specified_length/len(x))+1, 1)))
        else:
            train_x_filtered.append(x)
    return train_x_filtered, train_y_filtered

def pickup_data_recombination(arr):
    r_count_fir = random.randint(0, (len(arr)-length))
    arr_split_fir = arr[r_count_fir:r_count_fir+length:1]
    r_count_sec = random.randint(0, (len(arr)-length))
    arr_split_sec = arr[r_count_sec:r_count_sec+length:1]
    if random.random()> 0.5:
        r = random.random()
        len1, len2 = int(length * r), length - int(length * r)
        ran1,    ran2    =    random.randint(0,    len(arr_split_fir)-len1),random.randint(0,
          len(arr_split_sec)-len2)
        arr_split = np.vstack((arr_split_fir[ran1:ran1+len1],arr_split_sec[ran2:ran2+len2]))
    else:
        arr_split = arr_split_fir
    return arr_split
```

```
def pickup_data_rotation(arr):
    r_count = random.randint(0, (len(arr)-length))
    rotation_interval = 1
    degree = random.randint(0, (360/rotation_interval)-1)*rotation_interval
    def get_rotation_matrix(rad):
        rot = np.array([[1,0,0],
                        [0,np.cos(rad), -np.sin(rad)],
                        [0,np.sin(rad), np.cos(rad)]])
        return rot
    rotation_matrix = get_rotation_matrix(degree)
    arr_split = arr[r_count:r_count+length:1]
    return [np.dot(rotation_matrix, acc) for acc in arr_split]

def pickup_data_reversal(arr):
    r_count = random.randint(0, (len(arr)-length))
    arr_split = arr[r_count:r_count+length:1]
    arr_split_flip = np.flip(arr_split, axis = 0)
    if random.random()> 0.5:
        return arr_split
    else:
        return arr_split_flip
```

B.3 Fourier surrogates-based data generation

```
def iaaft_multi_augm(train_x):
    train_x_augm = []
    def iaaft_multi(X):
        max_it = 500
        pp = X.shape[0]
        dim = X.shape[1]
        if dim >1:
            Y = np.fft.fft(X,axis=0)
            Yamp = np.abs(Y)
            Porig = np.angle(Y)
            rn = np.zeros((pp,dim))
            E = np.array([])
            for k in range(0,dim):
                index = [i for i in range(pp)]
                np.random.shuffle(index)
                rn[:,k]= X[index,k]
            Xsorted = np.sort(X, axis =0)
            prev_err = 1000000
            err = prev_err -1
            c =1
            Pcurr = Porig
            while (prev_err >err) & (c<max_it):
                Prn =np.angle(np.fft.fft(rn, axis =0))
                goal = Prn-Porig
```

```
            AUX1 = (np.sum(np.cos(goal),axis=1,keepdims=True))
            alpha                                =(np.int64(AUX1!=0))                                *
(np.arctan((np.sum(np.sin(goal),axis=1,keepdims=True)) / (np.sum(AUX1 + (AUX1 ==
0),axis=1,keepdims=True))))
            alpha = np.matlib.repmat(alpha, 1, dim)
            alpha     =     alpha     +     np.matlib.repmat(math.pi*(np.sum(np.cos(alpha-
goal),axis=1,keepdims=True)<0),1, dim)
            Pcurr = Porig + alpha
            l_a = np.array(-1, dtype = complex)
            sn = np.real(np.fft.ifft(Yamp*(np.cos(Pcurr)+np.sqrt(l_a)*np.sin(Pcurr)),axis =0))
            sns = np.sort(sn, axis =0)
            INDs = np.argsort(sn, axis =0)
            for k in range(0,dim):
               rn[INDs[:,k],k] = Xsorted[:,k]
            prev_err = err
            A2 = np.abs(np.fft.fft(rn,axis =0))
            err = np.mean(np.mean(np.abs(A2 - Yamp)))
            E =np.append(E, err)
            c = c+1
         Xs = rn
      return Xs
   for x in train_x:
      train_x_augm.append(iaaft_multi(x))
   return train_x_augm
```

```
def iaaft_uni_augm(train_x):
    train_x_augm = []
    def iaaft_uni(X):
        max_it = 500
        pp = X.shape[0]
        dim = X.shape[1]
        if dim ==1:
            X = X[:]
            pp = len(X)
            index = [i for i in range(pp)]
            np.random.shuffle(index)
            rn = X[index]
            Yamp = np.abs(np.fft.fft(X, axis = 0))
            Xsorted = np.sort(X, axis = 0)
            prev_err = 0
            E = np.zeros ((1, max_it))
            c = 1
            prev_err = 1000000
            err = prev_err -1
            while(prev_err > err) & (c < max_it):
                Yrn = np.fft.fft(rn, axis=0)
                Yang = np.angle(Yrn)
                l_a = np.array(-1, dtype = complex)
                sn = np.real(np.fft.ifft(Yamp*(np.cos(Yang)+np.sqrt(l_a)*np.sin(Yang)),axis=0))
                sns = np.sort(sn, axis = 0)
                INDs = np.argsort(sn, axis = 0)
```

```
            rn[INDs[:,0],0] = Xsorted[:,0]
            prev_err = err
            A2 = np.abs(Yrn)
            err = np.mean(np.abs(A2 - Yamp))
            E[0][c-1]=err
            c = c+1
        E = E[:,0:c-1]
        Xs = rn
    return Xs
  for x in train_x:
    Xs_x =iaaft_uni(x[:,0].reshape(-1,1))
    Xs_y =iaaft_uni(x[:,1].reshape(-1,1))
    Xs_z =iaaft_uni(x[:,2].reshape(-1,1))
    Xs_u = np.hstack((Xs_x,Xs_y,Xs_z))
    train_x_augm.append(Xs_u)
  return train_x_augm

def aaft_uni_augm(train_x):
  train_x_augm = []
  def aaft_uni(X):
    max_it = 500
    pp = X.shape[0]
    dim = X.shape[1]
    if dim ==1:
      X = X[:]
      pp = len(X)
```

```
        index = [i for i in range(pp)]
        np.random.shuffle(index)
        rn = X[index]
        Yamp = np.abs(np.fft.fft(X, axis = 0))
        Xsorted = np.sort(X, axis = 0)
        E = np.zeros ((1, max_it))
        c = 1
        prev_err = 1000000
        err = prev_err -1
        Yrn = np.fft.fft(rn, axis=0)
        Yang = np.angle(Yrn)
        l_a = np.array(-1, dtype = complex)
        sn = np.real(np.fft.ifft(Yamp*(np.cos(Yang)+np.sqrt(l_a)*np.sin(Yang)),axis=0))
        E = E[:,0:c-1]
        Xs = sn
    return Xs
for x in train_x:
    Xs_x =aaft_uni(x[:,0].reshape(-1,1))
    Xs_y =aaft_uni(x[:,1].reshape(-1,1))
    Xs_z =aaft_uni(x[:,2].reshape(-1,1))
    Xs_u = np.hstack((Xs_x,Xs_y,Xs_z))
    train_x_augm.append(Xs_u)
return train_x_augm
```