CITY UNIVERSITY OF HONG KONG

香港城市大學

# Towards Robust Animal Activity Recognition Using Deep Learning and Wearable Sensors

# 基於深度學習和穿戴式傳感器的魯棒動物行為識別

Submitted to
Department of Infectious Diseases and Public Health
傳染病與公共衛生學系
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
哲學博士學位

by

MAO Axiu
毛阿秀

September 2023
二零二三年九月

# Abstract

An automated animal activity recognition (AAR) system allows caretakers to continuously and remotely monitor animal behavioral variations, thereby providing rich insights into animal health and welfare and promoting livestock management efficiency. Over the past decades, advancements in deep learning techniques and wearable sensors have driven the rapid development of automated and precise AAR systems. However, when we develop an AAR system based on deep learning and wearable sensors, some technical challenges must be improved before its practical implementation in commercial animal farming. This thesis mainly focuses on four practical challenges of building an AAR system, including multi-modal fusion, class imbalance, data privacy, and energy efficiency. Concretely, (1) Multi-modal fusion. Typically, multiple sensors of different types are attached to an animal's body, or sensors of the same type are attached to different locations on an animal's body, to record multi-modal data and obtain rich information. However, integrating multi-modal data poses a challenge for multi-modal fusion in the development of deep learning-based recognition models, as a model may struggle to generalize the different modalities of sensor data. A conflicting correlation between multiple modalities can easily interfere with multi-modal fusion, resulting in limited recognition performance. (2) Class imbalance. The frequency and duration of different animal behaviors tend to be inconsistent, owing to animals' specific physiologies, thereby leading to a disproportion in the number of samples among behavioral classes and inducing class imbalance. Deep learning methods trained on imbalanced datasets tend to be biased towards majority classes and away from minority classes, which often causes poor model generalizability and high classification error rates for rare categories. (3) Data privacy. Deep learning has dominated

the tasks in AAR due to the high performance achievable with the help of large-scale training datasets. However, in reality, constructing a large corpus of centralized datasets across different sources (e.g., farms) results in data ownership and privacy problems, and poses a significant risk of commercial information leakage for producers and stockholders. Compared to such a traditional centralized manner, distributed learning paradigm without exchanging private data can provide a promising solution in the future privacy-preserving AAR system. (4) Energy efficiency. Animal activities are generally monitored over a long period (e.g., a few weeks or several months), which requires sensing devices to continuously collect and transmit data. As most embedded sensing devices are battery-powered, factors affecting the energy consumption and battery life of sensing devices must be carefully considered. The literature has proven that higher sampling rates come at a cost in real-world deployments that rely on long-term operations. Considering practical benefits, existing works have often lowered the sampling rate of sensors to reduce energy costs. However, when the sampling rate falls below a threshold, the AAR performance degrades rapidly due to many relevant signals being missed.

In this thesis, I am devoted to investigating corresponding solutions to above-mentioned challenges, aiming to enhance the robustness of an automated AAR system. First, to improve the capability of AAR based on imbalanced multi-modal data, I develop a cross-modality interaction network (CMI-Net) for multi-modal fusion and adopt class-balanced (CB) focal loss for alleviating the class imbalance problem. Specifically, the CMI-Net consists of a dual CNN trunk architecture to extract modality-specific features and a cross-modality interaction module (CMIM) to achieve deep inter-modality interaction. In particular, the CMIM based on an attention mechanism adaptively recalibrates each modality's temporal- and axis-wise features by leveraging multi-modal information. Thus, it enables the CMI-Net to effectively capture complementary information and suppress unrelated information from multiple modalities. In addition, the CB focal loss is employed to supervise the network training, and it can force the network to pay more attention not only to samples of minority classes, diminishing their influence from being overwhelmed during optimization, but also to samples that are hard to distinguish.

Second, I introduce a new distributed learning strategy, i.e., federated learning (FL), to achieve automated AAR based on decentralized data over different farms while protecting data privacy and ownership. I adequately consider two challenges (i.e., client-drift during local training and local gradient conflicts during global aggregation) resulting from data heterogeneity between multiple farms when directly applying FL to AAR tasks. To tackle these two challenges, I propose a novel FL framework called FedAAR that comprises a prototype-guided local update (PLU) module for local optimization and gradient-refinement-based aggregation (GRA) module for global aggregation. Specifically, the PLU module encourages all clients to learn consistent feature knowledge by imposing a global prototype guidance constraint to local optimization, reducing the divergence between client updates. The GRA module eliminates conflicting components between local gradients during global aggregation, effectively guaranteeing that all refined local gradients point in a positive direction to improve the agreement among clients.

Third, I present a novel approach, dubbed teacher-to-student information recovery (T2S-IR), to achieve energy-efficient AAR at low sampling rates while maintaining desirable performance. The T2S-IR effectively leverages the knowledge obtained from high-sampling-rate data, to assist in recovering the missing information in features extracted by the classification network trained on low-sampling-rate data. Specifically, I first utilize high-sampling-rate data for training teacher classification and reconstruction networks sequentially. Then, I train a student classification network using low-sampling-rate data, while promoting its performance by exploiting the knowledge learned by trained teacher networks via two novel modules, namely the reconstruction-based information recovery (RIR) module and the correlation-distillation-based information recovery (CDIR) module. Particularly, the RIR module exploits the pre-trained teacher reconstruction network to compel the student classification network to learn complete and descriptive features. The CDIR module enforces the feature maps of student network to mimic internal correlations within feature maps of pre-trained teacher classification network along temporal and sensor axes directions. The enhanced student network can be directly applied to infer different animal activities in practical scenarios

with low sampling rates.

In conclusion, this thesis outlines four practical challenges associated with the development of AAR systems based on deep learning and wearable sensors, including multi-modal fusion, class imbalance, data privacy, and energy efficiency. Correspondingly, I have presented a series of strategies to address these challenges, including the CMI-Net combined with CB focal loss to achieve multi-model fusion and mitigate the class imbalance problem, the FedAAR to perform automated AAR by uniting decentralized data while preserving data privacy across farms, and the T2S-IR to maintain the favorable performance of AAR at low sampling rates. Extensive experiments conducted on public datasets acquired for horses or/and goats using tri-axial accelerometers and tri-axial gyroscopes have verified the effectiveness of the proposed methods, which exhibit superior performance to the state-of-the-art algorithms on various tasks.

**CITY UNIVERSITY OF HONG KONG**

**Qualifying Panel and Examination Panel**

| | |
|---|---|
| Surname: | MAO |
| First Name: | Axiu |
| Degree: | Ph.D. |
| College/Department: | Department of Infectious Diseases and Public Health |

The Qualifying Panel of the above student is composed of:

*Supervisor(s)*

| | |
|---|---|
| Dr. LIU Kai | Department of Infectious Diseases and Public Health<br>City University of Hong Kong |

*Qualifying Panel Member(s)*

| | |
|---|---|
| Dr. ZHANG Zijun | Department of Data Science<br>City University of Hong Kong |
| Dr. PARKES Rebecca Sarah Victoria | Department of Infectious Diseases and Public Health<br>City University of Hong Kong |

This thesis has been examined and approved by the following examiners:

| | |
|---|---|
| Dr. LIU Kai | Department of Infectious Diseases and Public Health<br>City University of Hong Kong |
| Prof. SPARAGANO Olivier | Department of Infectious Diseases and Public Health<br>City University of Hong Kong |
| Prof. LU Mingzhou | College of Artificial Intelligence<br>Nanjing Agricultural University |
| Prof. LI Jiangong | College of Animal Science and Technology<br>China Agricultural University |

# Acknowledgements

I am grateful for this opportunity to express my heartfelt appreciation to all those who have helped, supported, and inspired me during my Ph.D. study at City University of Hong Kong (CityU). Without their invaluable guidance and encouragement, this thesis would not have been possible. The journey towards earning a doctorate degree is challenging, but the experiences and adventures I have undergone are truly rewarding.

First of all, I would like to express my deepest gratitude and respect to my supervisor, Dr. LIU Kai, for his invaluable guidance, unwavering support, and patience throughout my research project. His exceptional expertise, enthusiasm, and dedication to excellence are the constant source of inspiration to me. His insightful comments, constructive feedback, and valuable suggestions play a crucial role in refining my research questions and methodology. His unwavering encouragement keeps me motivated and focused throughout this research journey. I immensely appreciate his efforts in training me to become a better researcher. The skills and knowledge I have acquired under his guidance will undoubtedly benefit me for a lifetime.

I am also greatly thankful to my qualifying panel members, Dr. ZHANG Zijun and Dr. PARKES Rebecca Sarah Victoria, for their time and effort spent on assessing my research and annual reports. Their expert guidance and mentorship are invaluable, and I am honored to have the opportunity to learn from them. In addition, the deep appreciation then goes to our wonderful collaborator, Dr. McELLIGOTT Alan, whose research topics and style have influenced and benefited me greatly, and whose constructive suggestions and comments are valuable.

I extend my sincerest gratitude and appreciation to my thesis committee members, Prof. SPARAGANO Olivier, Prof. LU Mingzhou, and Prof. LI Jiangong. Their thoughtful and constructive comments and feedback are invaluable in shaping my research and ensuring it meets the highest standards of academic excellence.

In addition, I am incredibly honored and privileged to work with an exceptionally talented and dedicated group of individuals in our SmartAM research group. They are HUANG Endai, HE Zheng, LYU Li, GUO Zhaojin, SZE Cheryl Natalie, and NIDHI Mahejabeen Hossain. I am deeply grateful to each of them for their contributions to my research and for the many unforgettable memories we shared together. They are not only my colleagues but also my friends, and I will always cherish the relationships we built during our time at CityU. I would also like to thank all co-authors for their hard work to dedicate my publications, and all good friends at CityU for sharing valuable time and experience.

Finally, I devote the most special gratitude to my family and boyfriend ZHU Meilu for their unconditional love and support. Their constant encouragement and understanding are the cornerstone of my academic success. Their belief in me motivates me to pursue excellence and strive for greatness. I am truly blessed to have them in my life.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1

# 1 Introduction

## 1.1 Background

The behavior of animals provides rich insights into their mental and physical states and is among the most crucial indicators of animals' health, welfare, and subjective states [1, 2]. However, animal behavior monitoring largely relies on manual observations that are time-consuming, labor-intensive, and involve the subjective judgments of individuals [2]. Therefore, investigating and developing an automated, quantifiable, and precise measurement system for animal behavior, particularly for animal health and welfare monitoring, is vital. Such intelligent animal activity recognition (AAR) systems allow caretakers to continuously and remotely monitor animal behavioral variations, thereby reducing workloads and costs in veterinary clinics and promoting livestock management efficiency [3].

Over the past decades, advancements in digital technologies (e.g., computer vision, wearable sensors, and acoustic analysis systems) have driven the rapid development of automated and precise AAR systems. In particular, wearable sensors, such as accelerometers, gyroscopes, magnetometers, pressure sensors, and global navigation satellite systems, have gained popularity in animal monitoring applications owing to their light weight, compact size, low power consumption, high reliability, exceptional stability, and effortless integration. The kinetic characteristics (e.g., acceleration and angular velocity), pressure, and geo-location

information of animals with different behaviors can be accurately measured at a certain sampling rate (e.g., 10, 25, 50, and 100 Hz) using these sensors, which are generally attached to specific animal body parts (e.g., ears, necks, halters, or legs). Subsequently, advanced intelligent computing techniques are used to process and analyze the recorded data to classify various animal behaviors, such as the walking and rumination of cattle [4], the trotting and cantering of horses [2], and the eating and petting behaviors of dogs [5].

Deep learning, as one of the most promising data processing and analysis techniques, has been successfully adopted in diverse fields, including wearable sensor-aided AAR, owing to its excellent automated feature-extraction ability [6, 7]. Specifically, deep learning involves multiple layers of neural networks and thus enables the learning of features from raw sensor data with less preprocessing than other methods and allows for the hierarchical representation of features from low to high levels [8]. The most common deep learning methods include fully connected feedforward neural networks, convolutional neural networks (CNNs), recurrent neural networks, and their variants [8–10]. These methods can be stacked into different layers to form deep learning models that can enhance system performance, flexibility, and robustness. Deep learning models combined with wearable sensors have exhibited promising performance in distinguishing daily animal activities [2, 11–14].

Despite the increased and successful application of deep learning techniques with wearable sensor data for AAR, some technical challenges associated with deep learning, such as its performance and cost, need to be improved before its practical implementation in commercial animal farming. As our primary concern, the recognition performance of deep learning models is heavily reliant on the availability of high-quality training data [4, 15]. However, external factors in the data collection process may increase the data complexity, potentially interfering with the network's feature learning and thus resulting in limited recognition performance [8]. Meanwhile, some strict data collection conditions or high annotation expenses often lead to data limitations, which prompts us to construct a centralized dataset across diverse sources [16, 17]. This inevitably raises concerns over data privacy and poses a significant risk of commercial

information leakage for producers and stockholders. In addition, long-term animal monitoring is necessary for practical wearable sensor-aided AAR systems, making it crucial to develop energy-efficient behavior detection methods [2, 18]. This directly relates to the economic benefits of the farming industry. In this thesis, I focus on four practical challenges associated with the development of AAR systems based on deep learning and wearable sensors, including multi-modal fusion, class imbalance, data privacy, and energy efficiency, aiming to find feasible solutions to promote the robustness of automated AAR systems.

## 1.2 Motivations

In this section, I delve into the motivations behind the above-mentioned challenges, including multi-modal fusion, class imbalance, privacy, and energy efficiency.

### 1.2.1 Multi-modal Fusion

In general, using a single wearable sensor may not capture all relevant information and result in incomplete or noisy data. This easily leads to difficulty in accurately classifying similar activities and a lack of redundancy, making the AAR system more vulnerable to errors and failures. To obtain richer information about animal activities, multiple sensors are often attached to the animal's body in terms of sensor type or placement location. As a result, multi-type data exhibiting diverse modality characteristics, known as multi-modal data, can be recorded during data collection. However, integrating multi-modal data inevitably poses a challenge of multi-modal fusion when developing deep learning-based recognition models, because the model may struggle to generalize the different modalities of sensor data [11]. The conflicting correlation among multiple modalities can easily interfere with the multi-modal fusion, resulting in limited recognition performance [8]. Thus, there is an urgent need for favourable methods that can handle well on multi-modal fusion.

### 1.2.2 Class Imbalance

The frequency and duration of different animal behaviors tend to be inconsistent, owing to animals' specific physiologies, and annotating rare or infrequent behaviors (e.g., the drinking behavior of grazing cattle) is difficult because they occur occasionally or for short durations [6]. This leads to a disproportion in the number of samples among behavioral classes and induces class imbalance [19–21]. Deep learning methods trained on imbalanced datasets tend to be biased towards majority classes and away from minority classes, which often causes poor model generalizability and high classification error rates for rare categories [22, 23]. Therefore, in the context of class imbalance, developing a fair classification model that does not favor one behavior over another is necessary.

### 1.2.3 Data Privacy

In recent years, deep learning models have dominated AAR tasks due to the high performance achievable with the help of large-scale training datasets [4, 15]. However, in reality, building a big dataset for one farm or institution is difficult, and limited training data easily cause model overfitting, resulting in unsatisfactory classification performance [24, 25]. As shown in Fig. 1.1, training models based on data within a single site always have low accuracies due to data limitations. In contrast, the accuracy significantly increases when using centralized data from both sites. Thus, data collaboration across diverse sources (e.g., farms) is increasingly desired for learning a global model [16, 17]. However, constructing a large corpus of centralized datasets across different farms inevitably brings data ownership and privacy issues. Compared to such a traditional centralized manner, distributed learning paradigm without exchanging private data can provide a promising solution in the future privacy-preserving AAR system.

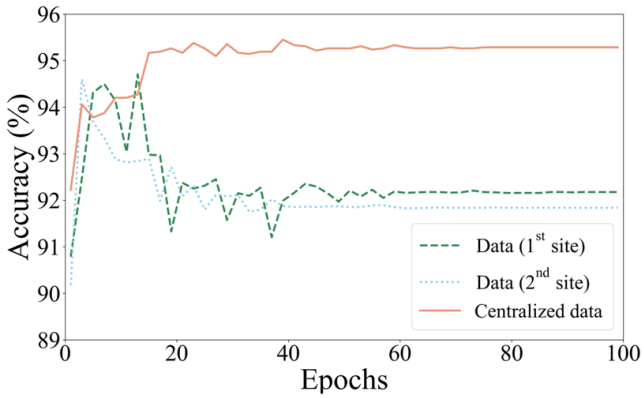



Fig. 1.1 Test accuracies based on data within a single site and centralized data.

### 1.2.4 Energy Efficiency

Energy efficiency is a non-negligible challenge in the practical deployment of automated wearable sensor-aided AAR systems. In general, animal activities are monitored over a long period (e.g., a few weeks or even several months), which requires sensing devices to collect and transmit data continuously [18]. As most embedded sensing devices are battery-powered, factors affecting the energy consumption and battery life of sensing devices must be carefully considered, such as sampling rate, transmit rate, and routing methods [26–29]. Amongst, the literature has proven that higher sampling rates come at a cost in real-world deployments that rely on long-term operations [30]. Considering practical benefits, existing works have often lowered the sampling rate to reduce energy costs and extend the battery life [28, 31, 32]. However, when the sampling rate falls below a threshold, the AAR performance degrades rapidly due to many relevant signals being missed. Hence, finding an effective method for improving the performance of AAR at low sampling rates is necessary and will significantly benefit the development of energy-efficient sensor-aided AAR systems.

## 1.3 List of Contributions

This thesis has presented and validated the solutions to tackle the above-mentioned challenges, i.e., multi-modal fusion, class imbalance, data privacy, and energy efficiency. The main contributions of this thesis are summarized as follows:

- To improve the capability of AAR based on imbalanced multi-modal data, I develop a cross-modality interaction network (CMI-Net) for multi-modal fusion and adopt a class-balanced (CB) focal loss for alleviating the class imbalance problem. In particular, the attention mechanism is introduced into the CMI-Net, enabling the network to effectively capture complementary information and suppress unrelated information from multiple modalities. The CB focal loss is employed to supervise the network training, which forces the model to pay more attention not only to samples of minority classes but also to samples that are hard to distinguish.
- To achieve automated AAR based on decentralized data while protecting data privacy and ownership, I introduce a new distributed learning strategy, i.e., federated learning (FL). I sufficiently take into account two challenges (i.e., client-drift during local training and local gradient conflicts during global aggregation) resulting from data heterogeneity between multiple farms when directly applying FL to AAR tasks. To tackle these two issues, I put forward a novel FL framework called FedAAR that comprises a prototype-guided local update (PLU) module and gradient-refinement-based aggregation (GRA) module. Specifically, the PLU module encourages all clients to learn consistent feature knowledge by imposing a global prototype guidance constraint to local optimization. The GRA module eliminates conflicting components between local gradients during global aggregation.
- To achieve energy-efficient AAR at low sampling rates while maintaining desirable performance, I propose a novel approach, dubbed teacher-to-student information recovery (T2S-IR). The T2S-IR effectively leverages the knowledge obtained from

high-sampling-rate data, to assist in recovering the missing information in features extracted by the classification network trained on low-sampling-rate data. Specifically, I first utilize high-sampling-rate data to train a teacher classification network and a reconstruction network sequentially. Then, I train a student classification network using low-sampling-rate data, while promoting its performance by exploiting the knowledge learned by trained teacher networks via two novel modules, namely the reconstruction-based information recovery module and the correlation-distillation-based information recovery module. In practice, the trained student network can be directly applied to infer different animal activities using low-sampling-rate data.

## 1.4 Thesis Organization

The rest of this thesis is organized as follows. A detailed literature review is given in Chapter 2. Chapter 3 presents the CMI-Net combined with CB focal loss to achieve multi-modal fusion and alleviate the class imbalance problem for AAR tasks based on imbalanced multi-modal data. In Chapter 4, I develop the FedAAR to achieve automated AAR using decentralized data from multiple farms and to address, in particular, two major challenges resulting from data heterogeneity when directly applying FL in AAR tasks. Chapter 5 proposes the T2S-IR method to achieve energy-efficient AAR at low sampling rates while maintaining desirable performance. Finally, I summarize this thesis and draw a conclusion in Chapter 6. In addition, some limitations and potential future works are also presented for advancing robust AAR systems.

# Chapter 2

# 2 Literature Review

In this chapter, I first briefly overview the application of wearable sensors with deep learning techniques for animal activity recognition (AAR). Then, I comprehensively present some potential grand challenges associated with developing AAR systems based on deep learning and wearable sensors. In addition, I review the related techniques or works related to the four challenges concerned in this thesis.

## 2.1 AAR Using Wearable Sensors and Deep Learning

### 2.1.1 Wearable Sensors for AAR

Advancements in wearable sensors and communication technologies have significantly enhanced the effectiveness of remote tracking of individual animal behaviors in various environments, and such behaviors can be tracked on a larger scale than was previously achievable [33]. Herein, I introduce five types of wearable sensors commonly used in AAR-related research: accelerometer, gyroscope, magnetometer, global navigation satellite system (GNSS), and ultra-wideband (UWB). The studies on these wearable sensors are also presented in Table 2.1.

**Accelerometer**

The tri-axial accelerometer is the most commonly used sensor in animal behavior monitoring tasks [31, 33, 34]. It can measure acceleration values along three perpendicular spatial axes, enabling the capture of animal motion dynamics. The measurement unit of acceleration is meters per second squared ($m/s^2$). Attaching multiple accelerometer devices to different parts

of an animal is an effective approach for expanding the spectrum of well-predicted behaviors [31], and it has been demonstrated to enhance recognition performance [13, 35].

### Gyroscope

A tri-axial gyroscope measures orientation and angular velocity along three orthogonal spatial axes. The unit of angular velocity is degrees per second (°/s). Gyroscopes are usually integrated with accelerometers and attached to the same location as an accelerometer on animal bodies, such as the neck [36], halter [37], back [38], tail [39], or leg [40], and operate at the same sampling rate as an accelerometer, typically ranging from a few Hz to several hundred Hz [31, 34]. Gyroscopes can provide information that complements that captured by accelerometers, thereby improving the prediction of some behaviors that are difficult to predict using only accelerometers [28].

### Magnetometer

The tri-axial magnetometer is another commonly used sensor in AAR tasks. It allows for the detection of changes in the magnetic field at a particular location and the measurement of rotation angle values (pitch, roll, and yaw). The measurement unit is Tesla (T). A magnetometer is typically assembled with an accelerometer and a gyroscope into an inertial measurement unit (IMU), which can simultaneously capture linear acceleration, angular velocity, and rotation angle. IMUs have been used to obtain a deep understanding of animal behaviors [12, 41–44]. For example, Hosseininoorbin et al. obtained superior results by using an IMU compared with using an accelerometer, gyroscope, or magnetometer alone or a combination of any two of these devices [41].

### GNSS

The GNSS is a satellite-based navigation system that provides location and time information to users worldwide. It has been combined with motion sensors to track the geo-location and movement patterns of various animals, particularly those that graze on pastures, to obtain a higher behavioral prediction accuracy than is possible with other methods [45–47]. For example, a recent study examined the use of data from both accelerometers and GNSSs for classifying cattle behaviors [48]. Using a combination of data from both sensors resulted in considerably higher prediction accuracy than using data from only one sensor, particularly for infrequent but important behaviors such as walking and drinking. Furthermore, the incorporation of the GNSS helps to distinguish between distinct behaviors with similar movement patterns, as these

behaviors tend to occur in different functional areas, such as feeding in designated feeding areas and drinking at water troughs [45].

**UWB**

The UWB is a radio technology that can use a very low energy level for short-range, high-bandwidth communications over a large portion of the radio spectrum. Real-time location systems based on UWB technology are typically utilized as indoor positioning systems to identify and locate groups of animals in free-stall barns [49–51]. A recent study incorporated the UWB-based indoor location data with accelerometer data to enhance cattle behavior monitoring systems in a free-stall barn [49]. The novelty of their approach lies in restricting the number of behaviors considered by the accelerometer based on the functional area in which the cow is located (feeding, lying, drinking), effectively reducing the confusion between behaviors (e.g., eating concentrate vs. drinking) with close patterns. In addition, combining these two data sources enables the tracking of social interactions among cows, which play a vital role in their health and welfare [49].

Table 2.1 Existing animal activity recognition (AAR)-related studies involving different wearable sensors.

| **Sensor**[#] | **Reference** |
|---|---|
| Accelerometer | [2, 4, 5, 11, 13, 14, 26, 35, 52–58] |
| Accelerometer and gyroscope | [6, 36–40, 59–62] |
| IMU (accelerometer, gyroscope, and magnetometer) | [12, 41–44] |
| Accelerometer and GNSS | [48, 63] |
| Accelerometer and UWB | [49] |

[#] GNSS: Global navigation satellite system; IMU: Inertial measurement unit; UWB: Ultra-wideband.

## 2.1.2 Deep Learning-based Methods for Wearable Sensor-aided AAR

Deep learning, known for its excellent capability in automated feature extraction [7], is becoming the mainstream data analysis technique in the field of AAR in recent years. Existing works have demonstrated that deep learning-based methods own promising performance in distinguishing daily animal activities [2, 11, 12, 14]. This section presents an extensive

overview of deep learning-based methods for wearable sensor-aided AAR (Table 2.2), in terms of the taxonomy of deep learning algorithms, i.e., fully connected feedforward neural network (FFNN), convolutional neural network (CNN), recurrent neural network (RNN), and hybrid model.

Table 2.2 Deep learning techniques for AAR tasks with wearable sensors.

| **Model**[#] | **Description** | **Reference** |
|---|---|---|
| FFNN | Fully connected, feedforward, multi-layer neural network | [14, 41, 42, 48, 52, 53, 58] |
| CNN | Convolutional, hierarchical, shared-weight neural network | [2, 4, 6, 11, 13, 14, 35, 36, 38, 39, 54, 55, 59, 62] |
| RNN | Recurrent, feedback, temporal neural network | [12, 26, 43, 44, 60] |
| Hybrid | Combination of some deep models | [5, 37, 40, 57, 61] |

[#] CNN: Convolutional neural network; FFNN: Fully connected feedforward neural network; RNN: Recurrent neural network.

**FFNN**

An FFNN, also known as multi-layer perceptron (MLP), is a type of artificial neural network. The FFNN typically consists of an input layer, multiple hidden layers, and an output layer, with each layer composed of many interconnected neurons. The output of each neuron in one layer is input to every neuron in the subsequent layer, allowing the network to learn complex non-linear relationships between inputs and outputs. In addition, an FFNN can handle large datasets and high-dimensional input spaces because it can be easily scaled up by adding more hidden layers and neurons.

Table 2.3 presents studies in which FFNNs have been used for wearable sensor-aided AAR. Coelho et al. [53] compared an FFNN with a generalized linear model and a random forest model in terms of cattle grazing behavior detection based on accelerometer data. The random forest model yielded the highest accuracy (76%), followed by the FFNN and the generalized linear model (74% and 59%, respectively). The lower accuracy of the FFNN than the random forest model is attributable to overfitting caused by the large number of parameters used in the FFNN. Hosseininoorbin et al. [41] and Dominguez-Morales et al. [42] have examined the performance of FFNNs based on nine-axis motion data from neck-attached IMU sensors, and obtained a favorable F1-score of 89.3% and a high accuracy of 97.96% in recognizing different activities of horses and cattle, respectively. Arablouei et al. [52] compared an MLP with several

machine-learning algorithms (e.g., linear regression, support-vector machine, and decision tree) for identifying cattle behaviors, based on a cattle dataset acquired using neck-attached tri-axial accelerometers. The results demonstrated that the MLP performed the best, with a 93.4% overall accuracy in classifying behaviors, including grazing, ruminating, and resting. According to these findings, Arablouei et al. selected the MLP as the classification model to distinguish more specific cattle behaviors (e.g., walking and drinking) in their subsequent two papers [48, 58]. Arablouei et al. [58] developed an end-to-end deep learning model, which consists of infinite-impulse-response and finite-impulse-response filters for feature extraction together with an MLP for classification. This model yielded a high accuracy (95.68%) on the same dataset used in [52]. Arablouei et al. [48] devised an MLP-based multi-modal fusion approach that combines both accelerometer and GNSS data to achieve accurate cattle behavior classification. Experiments were conducted on two new real-world datasets collected using smart cattle collar tags and ear tags, and the method obtained a higher accuracy (88.47%) on the collar-based dataset.

**CNN**

CNNs, one of the most researched deep learning algorithms, have been used successfully in a wide range of applications, such as natural language processing, image classification, speech recognition, and time series analysis [8, 64]. A CNN is generally designed to automatically learn and extract hierarchical features from raw input data (e.g., sensor values) using multiple layers of convolutional and pooling operations. The convolutional layers apply a set of filters with different kernel sizes and strides to capture local temporal dependencies between neighboring input values. The pooling layers operate by sliding a window over the input feature map and computing a summary statistic (e.g., maximum and average values) within each window to generate a downsampled output feature map, which makes the network translation invariant to changes and variations. Then, the features learned from sensor data are fed into several fully connected layers for final activity classification. In addition, CNNs use weight sharing to reduce the number of parameters in a network, which helps to prevent overfitting and improve model generalization. Owing to these remarkable benefits, CNNs have been increasingly adopted and are the most-used type of deep learning model in the field of wearable sensor-aided AAR [1,3,38,39,52,53,58,62].

Table 2.4 summarizes studies on the use of CNNs for wearable sensor-aided AAR. CNN-based approaches exhibit excellent performance in the classification of animal behaviors, with accuracies exceeding 90% in most cases. Eerdekens et al. [2, 13, 54, 55] have conducted several

studies on the use of CNNs in combination with accelerometer data to recognize the behaviors of horses and dogs. These studies have consistently demonstrated the excellent performance of CNNs, with prediction accuracies exceeding 97%. In particular, Eerdekens et al. [2, 13] have validated the superior performance of hybrid CNNs with statistical features as input compared with techniques based on raw sensor data, consistent with the findings in [11]. Kleanthous et al. [11] devised three CNN models with distinct architectures to classify sheep activities, including grazing, active behavior, and inactive behavior. Instead of directly inputting raw sensor data into a network, the authors employed a feature engineering approach to extract hand-crafted features from the raw data for network training. This approach effectively improved model generalization, resulting in higher classification accuracy (98.55%) than the approach based on raw sensor data. In Chapter 3, I developed a new CNN architecture to identify various horse activities. I evaluated the resulting CNN model using a publicly available dataset obtained using accelerometers and gyroscopes fixed to the horse's neck and obtained a high classification accuracy (90.68%) [6]. This model was subsequently utilized as the baseline model in my recent studies with different application objectives [59, 65], as presented in Chapter 4 and Chapter 5. Pavlovic et al. [62] designed a CNN architecture to accurately categorize cattle behaviors using data generated from neck-mounted accelerometer collars. Their CNN architecture underwent a rigorous hyperparameter search process, and the optimal model achieved an impressive overall F1 score of 82%. Their CNN architecture was also applied in a recent study [35], in which it was validated for cattle behavior classification using data collected from both neck- and leg-attached accelerometers, and achieved a high F1 score (93.9%). Li et al. [4] investigated the use of CNNs with various data-augmentation techniques to detect cattle behaviors using motion data collected using neck-mounted tri-axial accelerometers. Their CNN model achieved a high classification accuracy (94.43%) for cattle behaviors. Minati et al. utilized the same dataset to develop a CNN with two convolutional layers and obtained a higher classification accuracy (96%) [14]. Kasnesis et al. [38] and Hussain et al. [39] have investigated the use of CNNs for recognizing dog activities based on motion data obtained from accelerometers and gyroscopes attached to various body parts, such as the back, neck, and tail. In [38], a deep late-fusion CNN was designed and yielded a high accuracy (93.68%) in classifying dog activities. In [39], a CNN model with one-dimensional convolution was devised, with features extracted from raw sensor data used as input. This method obtained a high accuracy (96.85%) in recognizing dog behaviors. Recently, CNNs have also been utilized to distinguish behaviors in chickens [66] and lactating sows [36]. In [66], the CNN model achieved an average accuracy of nearly 100% in classifying activities of individual hens using motion data from body-worn IMU sensors. The high classification accuracy in this study

can be attributed to its focus on coarsely categorizing different activity levels in chickens rather than classifying specific behaviors. In [36], an accuracy of 87.33% was obtained in detecting static behaviors (i.e., nursing, lying, and sleeping) and active behaviors (i.e., eating, drinking, and moving) of lactating sows.

**RNN**

An RNN is designed to model sequential data, such as natural language text and time series (e.g., sensor data). Unlike feedforward neural networks, RNNs have a feedback loop in their hidden unit, which allows them to maintain a memory (i.e., hidden state) of previous inputs and use it to inform the processing of current inputs. At each time step, an RNN takes an input vector and combines it with the previous hidden state to produce a new hidden state, which is then used in the next time step. This recurrent connection enables the network to capture the temporal dependencies in the input data. However, RNNs are difficult to train and suffer from the problem of vanishing or exploding gradients, which limits their applicability in modeling long-term activity sequences and temporal dependencies in sensor data. Several variants of RNN have been devised to solve these problems, such as long short-term memorys (LSTMs) [67] and gated recurrent units (GRUs) [68]. An LSTM and a GRU introduce memory cells that can maintain information over long periods and varieties of gates that can control information flow in and out of cells. Compared with an LSTM, a GRU has a simpler structure with fewer parameters, resulting in a simpler training procedure and faster execution speed.

Table 2.5 summarizes recent studies on RNNs for wearable sensor-aided AAR. Peng et al. [12, 43, 44] have utilized LTSMs for three cattle-behavior recognition studies based on distinct datasets and targeting different applications. These datasets included nine-axial-motion data acquired from collar-attached IMU sensors. Peng et al. [44] conducted the pioneering research on utilizing LSTMs in wearable sensor-aided AAR tasks. They developed an LSTM model that attained an accuracy of 88.7% in distinguishing eight cattle behaviors, such as feeding, licking salt, and headbutting; thus, the model can be used to assess the health and welfare of cattle. In [43], the LSTM-based methods achieved an accuracy of 79.7% in identifying seven behaviors prior to calving, such as feeding, lying normally (collected during 72–24 h before calving), and standing normally (collected during 72–24 h before calving). Wu et al. [12] developed a novel LSTM model called deep residual bi-directional LSTM to classify cattle behavioral patterns for the early identification of bovine dermatomycosis. Their deep residual bi-directional LSTM exhibited a significantly higher classification accuracy (94.9%) than a basic LSTM. Among these three studies, Peng et al. [43, 44] compared LSTM models with CNN models and revealed

that the former could achieve superior cattle behavior classification performance, consistent with the findings in [26]. Wang et al. [26] designed multiple LSTM and GRU architectures with varying depths and widths and compared their performances in classifying cattle behaviors with those of two state-of-the-art CNN-based classification models. The experiments were conducted on collar-attached and ear-attached accelerometer data reported in [48]. The considered RNN models achieved comparable or superior accuracy to the CNN models while requiring less computational and memory resources. A two-layer bi-directional GRU model with 128 hidden units exhibited the best performance, with the highest accuracy of 89.5% and 80% on the collar- and ear-based datasets, respectively. The accuracy values were significantly higher than those obtained using MLP models in [48]. Hussain et al. [60] employed an LSTM-based method to detect dog activities based on motion data from accelerometers and gyroscopes, which were mounted on the necks and tails of dogs. Their model achieved a good test accuracy (94.25%) in classifying activities of dogs.

**Hybrid model**

A hybrid model is a combination of several deep learning models, with the models being combined to improve performance or eliminate the shortcomings of individual models.

Table 2.6 presents various hybrid deep learning models for AAR tasks. One emerging hybrid deep-learning model is the combination of CNN and LSTM, i.e., CNN–LSTM model. Liseune et al. [40] provided a good example of combining CNN and LSTM in cattle-behavior classification tasks, based on motion data from accelerometers and gyroscopes. It was revealed that the CNN–LSTM model outperformed pure CNN and LSTM models. Similar results were also reported in a recent study [61], in which the CNN-LSTM model exhibited a 93.4% accuracy in identifying dog behaviors. Chambers et al. [5] applied FilterNet, a CNN–LSTM model devised in [69], to detect daily dog behaviors using neck-mounted accelerometer data. FilterNet displayed high classification accuracies (> 94%) for all considered behaviors, such as drinking (99%), eating (94.8%), and object licking (98%). Arablouei et al. [57] devised a new hybrid deep learning method called GRU-MLP to classify cattle behaviors and validated it using previously adopted collar-attached [26] and ear-attached [48] accelerometer data. Classification accuracies of 88.1% and 80.6% were obtained on collar-based and ear-based datasets, respectively. This GRU–MLP model exhibited performance comparable with that of the pure GRU model devised in [26] while requiring significantly smaller numbers of parameters (0.026 vs. 0.496 M) and fewer computational operations (6.4 vs. 101.2 M). The model exhibited this performance mainly because the authors adopted the knowledge

distillation technique, in which the residual neural network was taken as the teacher model to improve the performance of the student model, that is, the GRU–MLP model.

## 2.2 Public Datasets for Wearable Sensor-aided AAR

Herein, I provide a comprehensive list of publicly available datasets collected via wearable sensor-aided AAR over the past five years (Table 2.7). Table 2.7 presents several attributes of the datasets: the species and number of animals; the types, placements, and sampling rates of sensors; considered activities; and the recording duration or number size of annotated samples. To the best of our knowledge, before 2018, only one dataset had been made public by Kamminga et al. [70]. We can observe from Table 2.7 that the number of publicly released datasets increases over time, and cattle are the most extensively studied species. For all of the datasets, an accelerometer is the adopted wearable sensor and the neck is the sensor location for data collection, which are consistent with the statement that most studies have focused on using collar-borne accelerometers [34]. This list can serve as a valuable resource for readers who wish to further explore the field of AAR.

Table 2.3 Studies on FFNN-based methods for wearable sensor-aided AAR.

| Sensor | Placement | Species | Activities | Accuracy (%) | Reference |
|---|---|---|---|---|---|
| Accelerometer | Halter | Cattle | Grazing, non-grazing | 74 | [53] |
| IMU | Neck | Cattle | Grazing, walking, ruminating (standing), ruminating (lying), standing, lying, drinking, grooming, and others | 89.3 (F1-score) | [41] |
| IMU | Neck | Horse | Motionless, walking, and trotting | 97.96 | [42] |
| Accelerometer | Neck | Cattle | Grazing, ruminating, resting, and others | 93.40 | [52] |
| Accelerometer | Neck | Cattle | Grazing, walking, ruminating/resting, drinking, and others | 95.68 | [58] |
| Accelerometer and GNSS | Neck \| ear | Cattle | | 88.47 (neck) \| 75.32 (ear) | [48] |

Table 2.4 Studies on CNN-based methods for wearable sensor-aided AAR.

| Sensor | Placement | Species | Activities | Accuracy (%) | Reference |
|---|---|---|---|---|---|
| Accelerometer | Leg | Horse | Standing, walking, trotting, cantering, rolling, pawing, and flank watching | 99 | [54] |
| Accelerometer | Leg | Horse | | 99.93 | [55] |
| Accelerometer | Leg | Horse | | 99.59 | [2] |
| Accelerometer | Neck and chest | Dog | Lying, sitting, standing, walking, running, sprinting, eating, and drinking | 97.87 | [13] |
| Accelerometer | Neck | Sheep | Grazing, active, and inactive | 98.55 | [11] |
| Accelerometer and gyroscope | Neck | Horse | Eating, standing, trotting, galloping, walking with a rider, and natural walking | 90.68 | [6] |
| Accelerometer and gyroscope | Neck | Horse | | - | [59] |
| Accelerometer and gyroscope | Neck | Horse | | - | [65] |
| Accelerometer | Neck | Cattle | Eating, ruminating, and others | 82 (F1-score) | [62] |
| Accelerometer | Neck | Cattle | Feeding, ruminating, and others | 93.9 (F1-score) | [35] |
| Accelerometer | Neck | Cattle | Feeding, walking, salting, ruminating, and resting | 94.43 | [4] |
| Accelerometer | Neck | Cattle | Grazing, moving, resting, ruminating, and salting | 96 | [14] |
| Accelerometer and gyroscope | Back | Dog | Standing, walking, trotting, and running | 91.26 | [38] |
| Accelerometer and gyroscope | Neck and tail | Dog | Walking, sitting, down, staying, feeding, sideways, leaping, running, shaking, and nose work | 96.85 | [39] |
| IMU | body | Chicken | Low-, medium-, and high-intensity behaviors | > 99 | [66] |
| Accelerometer and gyroscope | Neck | Pig | Moving, drinking, eating, nursing, sleeping, and lying | 87.33 | [36] |

Table 2.5 Studies on RNN-based methods for wearable sensor-aided AAR.

| Sensor | Placement | Species | Activities | Accuracy (%) | Reference |
|---|---|---|---|---|---|
| IMU | Neck | Cattle | Feeding, lying, ruminating (lying), ruminating (standing), licking salt, moving, social licking, and headbutting | 88.7 | [44] |
| IMU | Neck | Cattle | Feeding, ruminating (lying), ruminating (standing), lying normal, lying final, standing normal, and standing final | 79.7 | [43] |
| IMU | Neck | Cattle | Feeding, lying, ruminating (lying), itch rubbing (leg), social licking, and itch rubbing (neck) | 94.9 | [12] |
| Accelerometer | Neck \| ear | Cattle | Grazing, walking, ruminating/resting, drinking, and others | 89.5 (neck) \| 80 (ear) | [26] |
| Accelerometer and gyroscope | Neck and tail | Dog | Walking, sitting, down, staying, feeding, sideways, leaping, running, shaking, and nose work | 94.25 | [60] |

Table 2.6 Studies on hybrid methods for wearable sensor-aided AAR.

| Hybrid model[#] | Sensor | Placement | Species | Activities | Accuracy(%) | Reference |
|---|---|---|---|---|---|---|
| CNN-LSTM | Accelerometer and gyroscope | Neck and leg | Cattle | Ruminating, eating, lying, standing up, walking, and inactive behavior | - | [40] |
| CNN-LSTM | Accelerometer and gyroscope | Neck | Dog | Standing, sitting, lying with head raised, lying without head raised, walking, sniffing, and running | 93.4 | [61] |
| CNN-LSTM | Accelerometer | Neck | Dog | Drinking, eating, object licking, self-licking, petting, rubbing, scratching, shaking, sniffing, and others | > 90 | [5] |
| GRU-MLP | Accelerometer | Neck \| ear | Cattle | Grazing, resting, and others | 88.1 (neck) \| 80.6 (ear) | [57] |

[#] GRU: Gate recurrent unit; LSTM: Long short-term memory; MLP: Multi-layer perceptron.

Table 2.7 Public datasets[1] on AAR with wearable sensors.

| ID | Species | Sensor type | Placement | Sampling rate (Hz) | Activities | Duration /number | Reference |
|---|---|---|---|---|---|---|---|
| 01 | Goat (5) | Accelerometer and gyroscope | Neck | 100 | Stationary behavior, walking, trotting, running, eating, fighting, shaking, climbing up, climbing down, rubbing, and food fight | 143.7 (h) | [71] |
| 02 | Horse (6) | Accelerometer and gyroscope | Neck | 100 | Walking (naturally or with a rider), trotting (naturally or with a rider), grazing, standing, galloping (naturally or with a rider), head shake, scratch biting, rolling, eating, fighting, shaking, jumping, rubbing, and scaredness | 93,303 (2-s) | [72] |
| 03 | Cattle (26) | Accelerometer (A) and GNSS | Neck | 59.5 (A) and 1 (GNSS) | Grazing, ruminating (while lying or standing), resting (lying or standing), and walking | - | [63] |
| 04 | Cattle (6) | Accelerometer | Neck | 25 | Feeding, drinking, grazing, ruminating (standing), salt licking, licking, resting (lying or standing), moving, urinating, attacking, escaping, and being mounted | 3.28 (h) | [4] |
| 05 | Cattle (18) | Accelerometer | Neck | 10 | Eating, ruminating, and others | 3,460 (h) | [62] |

[1] List of links: https://github.com/Max-1234-hub/List-of-public-datasets

| ID | Species | Sensor type | Placement | Sampling rate (Hz) | Activities | Duration /number | Reference |
|---|---|---|---|---|---|---|---|
| 06 | Dog (45) | Accelerometer and gyroscope | Neck and back | 100 | Galloping, lying on chest, sitting, sniffing, standing, trotting, and walking | 29.48 (h) | [73] |
| 07 | Sheep (9) | Accelerometer | Neck | 12.5 | Grazing, active behavior (walking and scratching), and inactive behavior (standing and resting) | > 65 (h) | [11] |
| 08 | Sheep (18) | Accelerometer | Neck | 0.1 | Standing, eating, moving, running, and others | 1,685,974 (rows) | [74] |
| 09 | Cattle (8) | Accelerometer and GNSS | Neck and ear | 50 (neck) and 62.5 (ear) | Grazing, drinking, resting, walking, and others | 11962 (5.12-s; neck) and 10879 (4.1-s; ear) | [48] |
| 10 | Cattle (21) | Accelerometer | Neck and leg | 25 | Feeding, ruminating, and others | 809 (h) | [35] |

## 2.3 Potential Challenges

In this section, I broadly discuss potential challenges associated with the three main stages (i.e., data acquisition, model development, and activity inference) of AAR tasks; these challenges are related to annotation scarcity, data privacy, energy efficiency, multi-modal fusion, class imbalance, inter-activity similarity, domain generalization, and open-set recognition (Fig. 2.1).

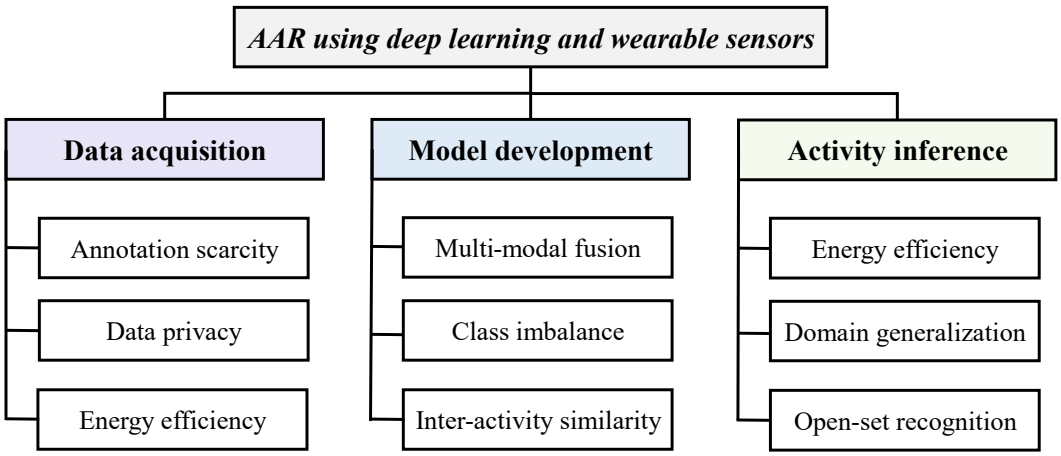



Fig. 2.1 Challenges associated with the data acquisition, model development, and activity inference stages of AAR.

### 2.3.1 Annotation Scarcity

The development of deep learning models is heavily reliant on the availability of large, annotated training datasets. However, in practice, such datasets are often scarce owing to the laborious and time-consuming process of data labeling. Additionally, data labeling requires experienced staff with a certain level of knowledge, making it highly challenging to obtain large, annotated datasets. This annotation scarcity often results in the model's overfitting and poor generalizability problem, limiting the applicability of models in real-world AAR scenarios.

### 2.3.2 Data Privacy

AAR-related studies tend to perform model development on a dataset within a single site, and the high performances of their deep learning models are highly dependent on massive training datasets [14, 35, 36, 38]. However, in reality, building a large dataset for one farm or institution

is difficult, and limited training data can easily cause model overfitting, resulting in unsatisfactory performance [24, 59]. Thus, data collaboration across diverse sources (e.g., farms) is increasingly desired for learning a global model [16, 17]. However, constructing a large corpus of centralized datasets across different farms results in data ownership and privacy problems, and poses a significant risk of commercial information leakage for producers and stockholders. As for this challenge, I will introduce a federated learning technique to solve it, as presented in Chapter 4.

### 2.3.3 Energy Efficiency

Energy efficiency as a non-negligible challenge has been widely investigated in many AAR-related works [2, 57, 62, 65]. This challenge is typically considered from two perspectives, i.e., the power usage of wearable sensors during data acquisition and the memory and computational cost of deep learning models during activity inference. First, animal activities are generally monitored over a long period, which requires sensing devices to continuously collect and transmit data [18]. As most embedded sensing devices are battery-powered, factors affecting the energy consumption and battery life of sensing devices must be carefully considered, such as sampling rate, transmit rate, and routing methods [18, 32, 54]. Second, sensing devices for behavior monitoring typically possess low processing and storage capabilities owing to their limited battery life [2, 18], which limits the on-board implementation of deep learning algorithms on wearable sensors. Thus, the memory and computational cost of deep learning models should be constrained to reduce energy usage in micro-controllers. In this thesis, I plan to decrease the sampling rate of wearable sensors to achieve a reduction in energy costs. The details are given in Chapter 5.

### 2.3.4 Multi-modal Fusion

The capability of deep learning models for AAR is highly related to the sensor modalities involved. Typically, multiple sensors of different types are attached to an animal's body (e.g., an accelerometer and a gyroscope are mounted on the same location), or sensors of the same type are attached to different locations on an animal's body (e.g., two accelerometers are separately placed at the ear and the neck), to collect multi-modal data and obtain richer information. Combining various sensing modalities tends to result in superior performance in animal behavior classification tasks compared with using only one modality [8, 9, 13, 41, 48]. Integrating multi-modal data aims to learn the intra-modality information and inter-modality

correlations, and then integrate the complementary and sometimes redundant information from each modality. However, a conflicting correlation among between multiple modalities can easily interfere with multi-modal fusion, resulting in limited recognition performance [8]. To this end, I propose a promising method in Chapter 3 to handle the multi-modal fusion.

### 2.3.5 Class Imbalance

Class imbalance refers to the situation where one or more classes in a dataset are represented by significantly fewer examples than other classes [75]. In the context of animal behavior recognition, this class imbalance problem inevitably presents due to the inconsistent frequency and duration of different activities resulting from animals' specific physiologies. In particular, some infrequent behaviors, such as the shaking behavior of horses, occur sporadically or for short durations, making them difficult to observe and record and further intensifying the imbalance degree between different categories. When training deep learning models on class-imbalanced datasets, bias towards majority classes can occur, leading to high error rates for rare categories [22, 23]. Therefore, I also investigate related solutions to deal with the class imbalance problem, as described in Chapter 3.

### 2.3.6 Inter-activity Similarity

One of the challenges in AAR tasks is inter-activity similarity; that is, a case in which different animal activities have similar characteristics or movement patterns [6, 37, 43]. This affects the ability of deep learning models to extract distinguishable features that can uniquely represent activities, leading to high confusion in classifying different activities [9]. For example, the ruminating behavior for both lying and standing behaviors of cattle can be easily misclassified as feeding, as both behaviors entail a similar chewing movement [43]. Similarly, a horse's activities of walking with and without a rider are often misclassified as each other, owing to their extreme resemblances in physical movement [6, 76].

### 2.3.7 Domain Generalization

Domain generalization is a challenge in the AAR field. It is the ability of models to generalize well to new and unseen domains, such as different animal species, sensor types, or environmental conditions [11, 58]. In reality, classification models trained on specific domains may exhibit significantly reduced performance when tested on data from a new domain (e.g.,

different animal species) during the inference phase. This phenomenon, known as domain shift, arises because of the discrepancy between training and test distributions. That is, a model may learn domain-specific features that are not transferable across domains or may overfit the training data, leading to poor performance on new and unseen data and limited applicability in real-world scenarios.

### 2.3.8 Open-set Recognition

AAR-related studies have generally developed deep learning-based classification models on training datasets that only cover a limited set of specific animal activities. However, some rare or infrequent activities, such as head shaking, scratch biting, and rolling, which are important indicators of animal health and welfare, may occur in real-world monitoring scenarios but be absent from training datasets. Thus, during the inference phase, these unseen activities are often misclassified into known activity categories in a training dataset, leading to poor performance and potentially missed opportunities for the early detection of health problems. Consequently, AAR is also characterized by the open-set recognition problem [77], which requires models to not only accurately classify known categories but also effectively deal with unknown categories.

## 2.4 Techniques Related to Focused Challenges

This thesis mainly focuses on four practical challenges, including multi-modal fusion, class imbalance, data privacy, and energy efficiency. Some related works or potential techniques for solving these challenges are presented in the following.

### 2.4.1 Solutions for Multi-modal Fusion

Common strategies for multi-modal fusion include early fusion, decision fusion, and feature fusion [78]. The early fusion strategy involves concatenating the input data from multiple modalities into a single vector or tensor and processing them using a unified model. This is the most commonly used method in existing research [13, 41]. However, owing to distribution gaps, early fusion is susceptible to interference between multi-modal information [79]. The decision fusion strategy combines the outputs or decisions from multiple classifiers at the decision-making stage, using techniques such as majority voting, weighted voting, or stacking, to obtain a final decision that is accurate and robust. However, this scheme is often sub-optimal because

rich modality information is gradually compressed and lost in separate processes, and the inter-modality correlations are ignored. A better choice is the feature fusion strategy, which involves the fusion of intermediate information from multiple modalities, followed by the generation of a final prediction using classifiers. This strategy avoids the distribution gap problem and achieves inter-modality interaction [80]. The feature fusion method has been used to process the multi-modal data collected using different sensor types and has confirmed the benefits of multi-modality combination for promoting model performance [38, 48, 61].

### 2.4.2 Solutions for Class Imbalance

Common techniques for addressing the class imbalance problem include resampling and cost-sensitive learning. Resampling involves either oversampling the minority class or undersampling the majority class to balance the class distribution [81]. Cost-sensitive learning is a technique that assigns different misclassification costs to different categories during model training and places a higher penalty on misclassifications in the minority class than on those in the majority class. Cost-sensitive learning can be achieved by adjusting the decision threshold or using class-weighted techniques [82]. In particular, the basic class-weighted method adds a term for each class that is inversely proportional to the frequency of that class and it has been adopted in several studies [6, 39, 60]. However, these approaches often lack good generalizability to real-world datasets because their training dataset may not accurately represent a true data distribution. Therefore, new techniques that do not rely on class distribution should be further explored, such as focal loss [83] and adaptive class suppression loss [84]. In addition, generative adversarial networks can also be utilized to address the class imbalance by generating synthetic data for the minority class and combining them with the original data to create a balanced dataset [85]. In a scenario with extremely imbalanced datasets, the classification tasks can be reformulated as an anomaly detection problem, in which minority-class instances that are dissimilar to the majority class are treated as outliers or anomalies [86].

### 2.4.3 Federated Learning for Data Privacy

Federated learning (FL), which is a new distributed learning paradigm, has emerged as an attractive approach to mitigate the problem of data privacy when constructing a large corpus of centralized datasets from different sources [87–89]. A standard FL system iterates two steps periodically, i.e., local training in each data source (client) and global aggregation in a

trustworthy centre (server), to train a global model. Specifically, during local training, each client downloads the parameters of the global model from the server-side to initialize its local model and then exploits local data to calculate client gradients, which are sent to the server in turn. The server collects and aggregates these local gradients to update the global model. Such a mechanism promotes privacy preservation between independent and decentralized data stores while producing trained models that leverage datasets of all participating clients [16, 90, 91]. Meanwhile, it enables us to train models with less storage and computational capabilities under individual clients. The most popular and easiest FL strategy is FedAvg [87], which aggregates the parameters of all local models by weighted averaging. However, FedAvg often suffered from slow convergence and performance degradation in most non-iid contents [90]. Thus, some state-of-the-art FL strategies such as FedProx [92], FedBN [93], SCAFFOLD [94], and FedLSD [95] are further proposed to tackle the non-iid issue. In recent years, FL has been increasingly designed for various applications due to its privacy-preserving nature, including mobile edge devices, industrial engineering, and health care [96–98].

### 2.4.4 Knowledge Distillation for Energy Efficiency

Knowledge distillation (KD), pioneered by Bucilă et al. [99]and further popularized by Hinton et al. [29], has made it possible to improve the performance of any machine learning algorithms while minimizing deployment and computational resources [27]. The core principle of KD is to transfer the knowledge from a cumbersome model to a small one that is more suitable for deployment, such that the small model's performance can be significantly boosted compared to training the small model alone [29]. Concretely, the large model, referred to as a teacher, is highly regularized via pre-training; and the small model, referred to as a student, is trained and tailored for a specific task. During the training, the student would mimic different kinds of knowledge involved in the pre-trained teacher, mainly including softened outputs [29], probability distribution [100], intermediate feature representations [101], and their variants [27, 102–106]. Specifically, the basic KD proposed by Hinton et al. distilled the logits (the inputs to the final softmax layer) from teacher to student by minimizing the Kullback-Leibler divergence [29]. Considering features contains richer representations, FitNet utilized intermediate features as knowledge transferred between teacher and student [101]. AT further introduced the attention map generated from intermediate features to deliver knowledge [106]. In addition, structural information within the feature space has also been taken as transferred knowledge. RKD measured the distance- and angle-wise relations among the given instances in the teacher's feature space and forced the student to mimic the same relations [105]. ICKD

encouraged the student to match the inter-channel correlation within the teacher's feature maps [103]. In recent years, KD has been applied to various applications such as model compression [27, 29, 105], continual learning [107, 108], and cross-domain transfer learning [109]. Due to the remarkable benefits KD provides, I attempt to explore the possible application of KD in the AAR task based on data having low sampling rates.

## 2.5 Public Datasets Used in This Thesis

In this thesis, I select two open-source datasets (D01 and D02 in Table 2.7), which are collected from horses [72] and goats [71], respectively. The details of each selected datasets are described as follows.

### 2.5.1 Horse Dataset

In this dataset, more than 1.2 million 2-s data samples are collected from 18 individual horses using neck-attached IMUs. The sampling rate is set to 100 Hz for both the tri-axial accelerometer and tri-axial gyroscope and 12 Hz for the tri-axial magnetometer. The majority of the samples are unlabeled, but data from six horses and six activities (i.e., eating, galloping, standing, trotting, walking-natural, and walking-rider are labeled extensively (87,621 2-s samples in total) and have been used to classify horse activities in previous studies [56, 76, 110]. In this thesis, data from the tri-axial accelerometer and tri-axial gyroscope among the 87,621 samples are exploited separately, forming up to two tensors with a size of $1 \times 3 \times 200$ for each sample. Table 2.8 illustrates the number of samples for each horse and activity. The sample number of the six horse individuals (i.e., Happy, Zafir, Driekus, Galoway, Patron, and Bacardi) is 23,625, 11,071, 10,127, 24,602, 12,849, and 5347, respectively. The sample number of the six activities (i.e., eating, standing, trotting, galloping, walking-rider, and walking-natural) is 16,048, 3,939, 5,113, 25,076, 3,327, and 34,118, respectively.

### 2.5.2 Goat Dataset

The goat dataset is a real-world dataset consisting of 42,943 2-s data samples collected from five goats on two farms. Each goat's collar is equipped with six tri-axial accelerometers and tri-axial gyroscopes fixed in different orientations. The sampling rate is set to 100 Hz for both the tri-axial accelerometer and tri-axial gyroscope. A total of five activities are included,

i.e., standing, running, eating, trotting, and walking. In this thesis, the tri-axial accelerometer and gyroscope data from all six sensors of the collar are utilized, forming up to two tensors with a size of 1 × 18 × 200 for each sample. Table 2.9 gives the distribution of samples per goat and activity. The sample number of the five goats (i.e., three domestic pygmy goats from one farm and two larger and more wild goats from another farm) is 13,902, 5,321, 11,954, 7,523, and 4,243, respectively. The sample number of the five activities (i.e., standing, running, eating, trotting, and walking) is 18,531, 373, 15,179, 189, and 8,671, respectively.

Table 2.8 Number of data samples per horse and activity in the horse dataset.

| **Activity** / **Horse** | Eating | Galloping | Standing | Trotting | Walking-natural | Walking-rider | Total |
|---|---|---|---|---|---|---|---|
| Happy | 5,063 | 696 | 1,186 | 7,038 | 746 | 8,896 | 23,625 |
| Zafir | 1,091 | 835 | 347 | 3,559 | 161 | 5,078 | 11,071 |
| Driekus | 2,496 | 323 | 341 | 2,673 | 270 | 4,024 | 10,127 |
| Galoway | 4,331 | 1,043 | 1,750 | 6,423 | 1,402 | 9,653 | 24,602 |
| Patron | 1,951 | 714 | 1,244 | 3,402 | 388 | 5,150 | 12,849 |
| Bacardi | 1,116 | 328 | 245 | 1,981 | 360 | 1,317 | 5,347 |
| Total | 16,048 | 3,939 | 5,113 | 25,076 | 3,327 | 34,118 | 87,621 |

Table 2.9 Number of data samples per goat and activity in the goat dataset.

| **Activity** / **Goat** | Standing/ Lying | Running | Grazing/ Eating | Trotting | Walking | Total |
|---|---|---|---|---|---|---|
| Goat_1 | 4,930 | 41 | 6,651 | 3 | 2,277 | 13,902 |
| Goat_2 | 1,175 | 49 | 2,507 | 13 | 1,577 | 5,321 |
| Goat_3 | 5,031 | 40 | 5,110 | 65 | 1,708 | 11,954 |
| Goat_4 | 4,196 | 128 | 759 | 43 | 2,397 | 7,523 |
| Goat_5 | 3,199 | 115 | 152 | 65 | 712 | 4,243 |
| Total | 18,531 | 373 | 15,179 | 189 | 8,671 | 42,943 |

### 2.5.3 Dataset Usage Distribution in Different Chapters

Table 2.10 gives the usage distribution of all three selected public datasets in different chapters, including Chapter 3, Chapter 4, and Chapter 5. Specifically, the horse dataset is used in all three chapters, and Chapter 5 additionally utilizes the goat datasets.

Table 2.10 Dataset usage distribution in different chapters.

| Chapter | Dataset |
|---|---|
| Chapter 3 | Horse dataset |
| Chapter 4 | Horse dataset |
| Chapter 5 | Horse dataset, goat dataset |

## 2.6 Publication Related to This Chapter

**Mao, A.**, Huang, E., Wang, X., & Liu, K.* (2023). Deep learning-based animal activity recognition with wearable sensors: Overview, challenges, and future directions. Computers and Electronics in Agriculture, 211, 108043.

# Chapter 3

# 3 Precise AAR with Imbalanced Multi-modal Data

In this chapter, I focus on tackling two challenges of multi-modal feature fusion and imbalanced data modeling, aiming to improve the performance of animal activity recognition (AAR). The horse is selected as the target studied subject in this chapter. Specifically, I develop a cross-modality interaction network (CMI-Net) involving a dual convolutional neural network (CNN) architecture and a cross-modality interaction module (CMIM) to improve the recognition performance for horse activities. The CMIM adaptively recalibrates the temporal- and axis-wise features in each modality by leveraging multi-modal information, consequently achieving deep inter-modality interaction. Moreover, to alleviate the class imbalance problem, I adopt a class-balanced focal loss to supervise the training of CMI-Net.

## 3.1 Introduction

The behavior of horses provides rich insight into their mental and physical status and is one of the most important indicators of their health, welfare, and subjective states [111]. However, behavioral monitoring for animals, to date, largely relies on manual observations, which are time-consuming, labor-intensive, and prone to subjective judgments of individuals [112]. The use of sensors and machine learning is well-established in monitoring gait change [113], and for lameness detection as part of the horse veterinary examination, increasing the accuracy of identifying subtle lameness. Lameness has been stated as one of the most expensive health issues in the horse industry [114, 115]. Therefore it is of significant importance to investigate

and develop an automatic, objective, accurate, and quantifiable measurement system for horse behaviors. Such a system will allow caretakers to identify variations in the animal behavioral repertoire in real-time, decreasing the workloads in veterinary clinics and improving the husbandry and management of animals [3, 116].

Over recent decades, automated AAR has been studied widely with the aid of various wearable sensors (e.g., accelerometers, gyroscopes, and magnetometers) and the use of machine learning techniques. For instance, a naïve Bayes (NB) classifier was applied to recognize horse activities (e.g., eating, standing, and trotting) using tri-axial acceleration and obtained 90% classification accuracy [110]. Four classifiers including a linear discriminant analysis, a quadratic discriminant analysis, a support vector machine (SVM), and a decision tree (DT) were utilized to detect dog behaviors (e.g., galloping, lying on chest, and sniffing) based on accelerometer and gyroscope data, and the results revealed that the sensor placed on the back and collar yielded 91% and 75% accuracy at best, respectively [117]. A random forest algorithm was applied to categorize cow activities using tri-axial acceleration and gained high classification accuracy with 91.4%, 99.8%, 88%, and 99.8% for feeding, lying, standing, and walking events, respectively [118]. In horses, the use of receiver operating characteristic curve analysis classified standing, grazing, and ambulatory activities with a sensitivity of 94.7–97.7% and a specificity of 94.7–96.8% [119]. However, to classify animal behaviors accurately using these machine learning methods, feature extraction and method selection are often conducted separately, which requires expert domain knowledge and easily induces feature engineering issues [8]. Moreover, hand-crafted features often fail to capture general and complex features, resulting in low generalizability, i.e., these extracted features perform well in recognizing the activities of some subjects but badly for others.

Along with the recent advances in internet technology and fast graphics processing units, various deep learning approaches have been increasingly and successfully adopted in wearable sensor-aided AAR tasks. Classification models based on deep learning achieve automatic feature learning through data driving and subsequent AAR. For example, feedforward neural networks and long short-term memory (LSTM) models were applied to automatically recognize cattle behaviors (e.g., feeding, lying, and ruminating) using data collected from inertial measurement units [41, 43]. CNNs, which accurately capture local temporal dependency and scale invariance in signals, were developed in automated horse activity classification based on tri-axial accelerometer and gyroscope data [2, 55]. FilterNet, presented based on CNN and LSTM architectures, was adopted to classify important health-related canine behaviors (e.g., drinking, eating, and scratching) using a collar-mounted accelerometer [5].

However, multi-modal data fusion has not been well handled when different sensors are used simultaneously in existing studies. Multi-modal data with different characteristics are often simply processed using common fusion strategies such as early fusion, feature fusion, and result fusion [78]. The early fusion strategy used in previous studies [41, 43], i.e., extracting the same features without distinction of modalities, often causes interference between multi-modal information due to their distribution gap [79]. The result fusion scheme is sub-optimal, because rich modality information is gradually compressed and lost in separate processes, thereby ignoring the inter-modality correlations. As a better choice, the feature fusion strategy fuses the intermediate information of multiple modalities, which avoids the distribution gap problem and achieves inter-modality interaction simultaneously [80, 120]. However, feature fusion is often limited to linear fusion (e.g., simple concatenation and addition) and fails to explore deep multi-modality interactions and achieve complementary-redundant information combinations between multiple modalities [78].

In addition, the collected sensor datasets often present class imbalance problems due to the inconsistent frequency and duration of each activity resulting from specific animal physiology. Deep learning methods trained on imbalanced datasets tend to be biased towards majority classes and away from minority classes, which easily causes poor modal generalizability and high classification error rates for rare categories [22]. Commonly used solutions are mainly divided into two categories, including resampling and reweighting. Resampling attempts to sample the data to obtain an evenly distributed dataset, e.g., oversampling and undersampling [121]. However, oversampling and undersampling come with high potential risks of overfitting and information loss, respectively [22]. Reweighting is more flexible and convenient by directly assigning a weight for the loss function per training sample to alleviate the sen-sitivity of the model to data distribution [75]. This method is further divided into class-level and sample-level reweighting. The former, such as cost-sensitive (CS) loss [122] and class-balanced (CB) loss [123], depends on the prior category frequency, while the latter, such as focal loss [83] and adaptive class suppression (ACS) loss [84], relies on the network output confidences of each instance. In addition, CB focal loss, combining a CB term with a modulating factor, effectively focuses on difficult samples and considers the proportional impact of effective numbers per class simultaneously [123].

To improve the recognition performance for horse activities while tackling the above-mentioned challenges, I develop a CMI-Net to achieve deep inter-modality interaction and adopt a CB focal loss [123] to supervise the training of CMI-Net. The CMI-Net consists of a dual CNN trunk architecture and a joint CMIM. Specifically, the dual CNN trunk architecture

extracts modality-specific features for accelerometer and gyroscope data, respectively, and the CMIM based on attention mechanism adaptively recalibrates the importance of the elements in the two modality-specific feature maps by leveraging multi-modal knowledge. The attention mechanism has been widely utilized in different tasks using multi-modal datasets such as RGB-D images [78, 124]. It has also been adopted to focus on important elements along with channels and spatial dimensions of the same input feature [125, 126]. The favorable performance presented in these studies with the attention mechanism indicated the rationality of the proposed CMIM. In this study, softmax cross-entropy (CE) loss is initially used to supervise the training of CMI-Net. However, softmax CE loss suffers from inferior classification performance, especially for monitory classes [75]. In contrast, CB focal loss, by adding a CB term to focal loss, focuses more on minor-class samples and hard-classified samples and can alleviate the class imbalance problem. Therefore, a CB focal loss [123] is adopted in this chapter.

# 3.2 Materials and Methods

## 3.2.1 Cross-modality Interaction Network

The proposed CMI-Net, where two-modality data (i.e., accelerometer and gyroscope data) are fed into two CNN branches (represented by $CNN_{acc}$ and $CNN_{gyr}$) separately, is shown in Fig. 3.1 (a). The dual CNN is constructed to extract modality-specific features and concatenate these features before the final dense layer. To achieve deep interaction between the two-modality data, a joint CMIM is designed and inserted in the upper layer. The CMIM can capture complementary information and suppress unrelated information from different modalities. The details are described below.

**Dual CNN Trunk Architecture**

The $CNN_{acc}$ and $CNN_{gyr}$ contain four convolutional blocks, three max-pooling layers, one global average-pooling layer, and one fully connected layer, followed by concatenation and one joint fully connected layer. Inspired by the residual unit in the deep residual network that behaves like ensembles and has smaller magnitudes of responses [127], I design a Res-LCB to promote the representation ability and robustness of the model, as demonstrated in Fig. 3.1 (b). The definition is given below.

$$X_{l+1} = RELU(Conv^{1\times1}(X_l) \oplus Conv^{1\times3}(X_l)), \tag{3.1}$$

where $X_l$ and $X_{l+1}$ denote feature maps in the $l$ and $l+1$ layers, respectively, $Conv^{1\times1}(\bullet)$ and $Conv^{1\times3}(\bullet)$ represent 1×1 and 1×3 convolution operations, respectively, $\oplus$ denotes the elementwise addition, and $RELU(\bullet)$ denotes the rectified linear unit activation function [128].

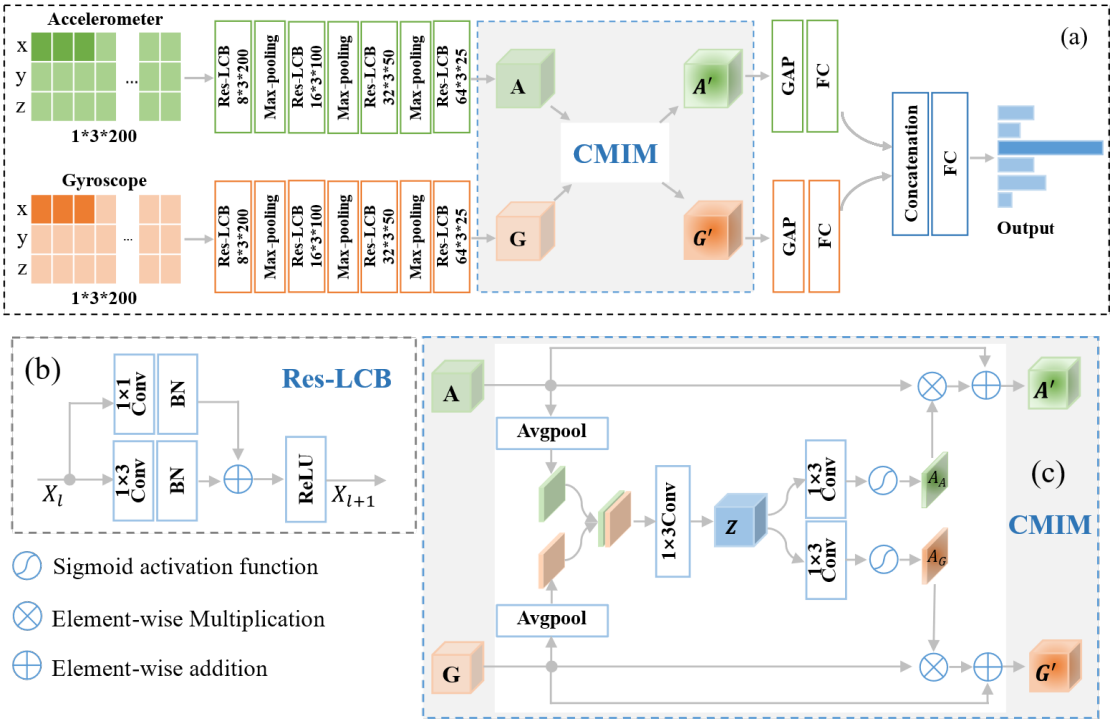



Fig. 3.1 The architecture of the proposed cross-modality interaction network (CMI-Net).

### Cross-modality Interaction Module

Inspired by the multi-modal transfer module that recalibrates channel-wise features of each modality based on multi-modal information [129] and the convolutional block attention module that focuses on the spatial information of the feature maps [125], I devise a CMIM based on an attention mechanism to adaptively recalibrate temporal- and axis-wise features in each modality by utilizing multi-modal information. The detailed CMIM is illustrated in Fig. 3.1 (c).

Let $A \in R^{C\times H\times W}$ and $G \in R^{C\times H\times W}$ represent the features at a given layer of $CNN_{acc}$ and $CNN_{gyr}$, respectively. Here, $C$, $H$, and $W$ denote the channel number and spatial dimensions of features. Specifically, $H$ and $W$ correspond to the axial and temporal signals, respectively. The CMIM receives $A$ and $G$ as input features. I first apply average-pooling operations along channels of the input features, generating two spatial maps. These two maps are then concatenated and mapped into a joint representation $Z \in R^{C'\times H\times W}$. The operation is

shown as follows:

$$Z = RELU(Conv^{1\times3}([Avgpool(A), Avgpool(G)])), \tag{3.2}$$

where $C'$ denotes the channel number of feature $Z$, $Avgpool(\bullet)$ denotes the average-pooling operation, and $[\bullet]$ denotes the concatenation operation. Furthermore, two spatial attention maps $A_A \in R^{1\times H\times W}$ and $A_G \in R^{1\times H\times W}$ are generated through two independent convolutional layers with a sigmoid function $\sigma(\bullet)$ based on the joint representation $Z$:

$$A_A = \sigma(Conv^{1\times3}(Z)),\ A_G = \sigma(Conv^{1\times3}(Z)), \tag{3.3}$$

$A_A$ and $A_G$ are then used to recalibrate the input features, generating two final refined features, i.e., $A' \in R^{C\times H\times W}$and $G' \in R^{C\times H\times W}$:

$$A' = A \otimes A_A \oplus A,\ G' = G \otimes A_G \oplus G, \tag{3.4}$$

where $\otimes$ denotes the elementwise multiplication. Specifically, each convolution operation under this chapter is followed by a batch normalization operation. The increases in channel numbers and decreases in spatial dimensions are implemented through Res-LCB and max-pooling operations, respectively.

### 3.2.2 Optimization

As the most widely utilized loss in the multi-class classification task, softmax CE loss is applied to optimize the parameters of CMI-Net. The formulation of softmax CE loss is defined as

$$L_{CE}(z) = -\sum_{i=1}^{C} y_i log\ (p_i) \tag{3.5}$$

$$\text{with } p_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}}, \tag{3.6}$$

where $C$ and $z = [z_1, \dots, z_C]$ are the total number of classes and the predicted logits of the network, respectively. In addition, $y_i \in \{0,1\}$, $1 \le i \le C$ is the one-hot ground-truth label. However, the models based on softmax CE loss often suffer from inferior classification performance, especially for monitory classes, due to the imbalanced data distribution [75]. Therefore, I further introduce an effective loss function to supervise the training of CMI-Net and alleviate the class imbalance problem, namely, CB focal loss.

CB focal loss, which adds the CB term to the focal loss function, focuses more on not only samples of minority classes, diminishing their influence from being overwhelmed during optimization, but also samples that are hard to distinguish. The CB term is related to the inverse

effective number of samples per class, and focal loss adds a modulating factor to the sigmoid CE loss to reduce the relative loss for well-classified samples and focuses more on difficult samples. The CB focal loss is presented as

$$L_{CB_{FL}}(z) = \frac{1}{E_{n_y}} L_{FL}(z) = -\frac{1-\beta}{1-\beta^{n_y}} \sum_{i=1}^{C} \left(1 - p_i^t\right)^{\gamma} \log\left(p_i^t\right) \tag{3.7}$$

$$\text{with } p_i^t = \frac{1}{1+e^{-z_i^t}}, \tag{3.8}$$

$$z_i^t = \begin{cases} z_i & if\ i = y. \\ -z_i, & \text{otherwise.'} \end{cases} \tag{3.9}$$

where $n_y$ and $E_{n_y}$ represent the actual number and the effective number of the ground-truth label $y$, respectively. The hyperparameter $\beta \in [0,1)$ controls how fast $E_{n_y}$ grows as $n_y$ increases, and $\gamma \geq 0$ smoothly adjusts the rate at which easy samples are down-weighted [75]. The value of $\beta$ is set to 0.9999, and the search space of the hyperparameter $\gamma$ is set to $\{0.5, 1.0, 2.0\}$ [123] in this chapter. In particular, CB loss and focal loss rebalances the loss function based on class-level and sample-level reweighting, respectively. Thus, I also utilize class-level reweighted losses, including cost-sensitive cross-entropy (CS_CE) loss [82], class-balanced cross-entropy (CB_CE) loss [123], and sample-level reweighted losses, including focal loss [83] and ACS loss [84], to validate the effectiveness of the CB focal loss.

### 3.2.3 Datasets and Data Preprocessing

The dataset used in this chapter is the horse dataset [72], as described in Section 2.5.1. A total of 87,621 2-s labeled samples with two modalities (i.e., tri-axial magnetometer and tri-axial gyroscope) are exploited, where each sample consists of two tensors with a size of 1 × 3 × 200. Fig. 3.2 visualizes the distribution of different activity categories. The activities of eating, standing, trotting, galloping, walking-rider, and walking-natural occupy 18.32%, 5.84%, 28.62%, 4.50%, 38.94%, and 3.80% of the total sample number, respectively, producing a maximum imbalance ratio of 10.25. In addition, the input sample of each axis per sensor modality is normalized by removing the mean and scaling to unit variance, which can be formulated as follows:

$$\tilde{S}_i = \frac{S_i - \mu_i}{\sigma_i}, \tag{3.10}$$

where $S_i$ denotes all samples of a particular axis per sensor modality (i.e., X-, Y-, and Z-axis of the accelerometer, and X-, Y-, and Z-axis of the gyroscope), $\tilde{S}_i$ denotes all normalized

samples, and $\mu_i$ and $\sigma_i$ denote mean and standard deviation values in each axis per sensor modality, respectively.

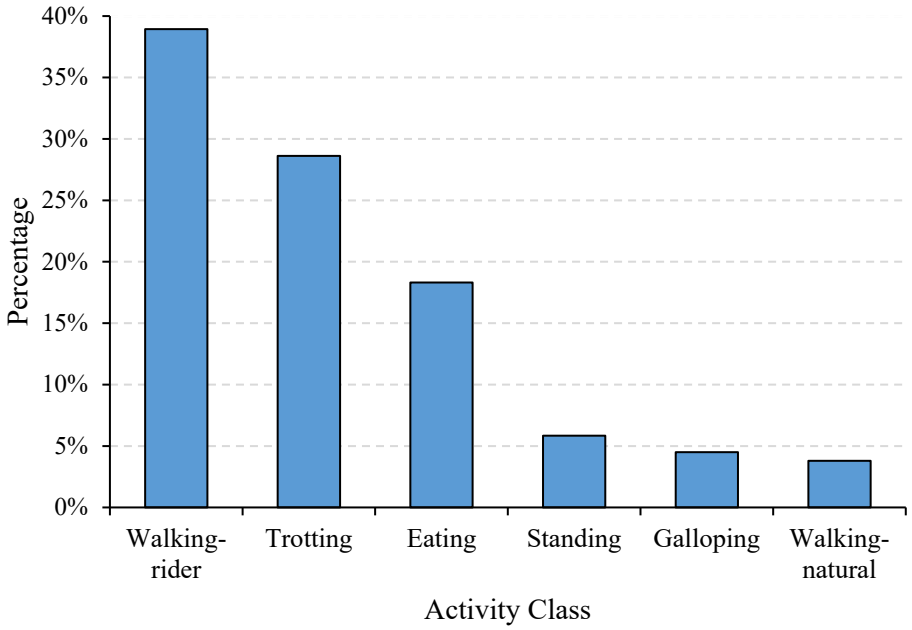



Fig. 3.2 Histogram of activity category distribution.

## 3.2.4 Design of Experiments

**Evaluation Metrics**

The comprehensive performance of the activity classification model is indicated by the following four evaluation metrics, which are defined in Eq. (3.11) to Eq. (3.14). Each indicator value is multiplied by 100 as the result to reflect the difference in indicator values more clearly.

$$Precision = \frac{TP}{TP+FP}, \tag{3.11}$$

$$Recall = \frac{TP}{TP+FN}, \tag{3.12}$$

$$F1-Score = \frac{2TP}{2TP+FP+FN}, \tag{3.13}$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}, \tag{3.14}$$

where $TP$, $FP$, $TN$, and $FN$ are the number of true positives, false positives, true negatives, and false negatives, respectively. In particular, the overall precision, recall, and F1-score are

calculated by using a macro-average [130].

**Implementation Details**

To attain subject-dependent results, the leave-one-out cross-validation method is used, in which four subjects are chosen for training, one for validation, and one for testing each time and rotated in a circular manner. During training, the loss function is added by an L2 regularization term with a weight decay of 0.1 to avoid overfitting. An Adam optimizer with an initial learning rate of $1 \times 10^{-4}$ is employed, and the learning rate decreases by 0.1 times every 20 epochs. The number of epochs and batch size are set to 100 and 256, respectively. The best model with the highest validation accuracy is saved and verified using test data. To evaluate the classification performance of the CMI-Net, I compare it against various existing methods, including three machine learning methods (i.e., NB, DT, and SVM) and two deep learning methods used in horse activity recognition (i.e., CNN and ConvNet7) [55, 76], based on the same public dataset. Specifically, the hand-crafted features used in machine learning are the same as those used by Kamminga et al. [110]. To further explore the performance of the CMIM, I run the network without CMIM and with it inserted after the 1st, 2nd, and 3rd max-pooling layers to obtain four different variants, i.e., Variant0, Variant1, Variant2, and Variant3, respectively. The softmax CE loss is used as the loss function for all variants. All experiments are executed using the PyTorch framework on an NVIDIA Tesla V100 GPU. The developed source code is available at https://github.com/Max-1234-hub/CMI-Net.

## 3.3 Results and Discussion

Overall, experiments conducted on the public dataset demonstrate that the proposed CMI-Net outperforms the existing algorithms. Ablation studies are then carried out to verify the effectiveness of CMIM and that applying the CMIM in the upper layer of CMI-Net could obtain better performance. Different loss functions are adopted to validate that CB focal loss performs better than any class-level or sample-level reweighted loss used alone, and it effectively improves the overall precision, recall, and F1-score, although the overall accuracy decreases due to the imbalanced dataset used. Furthermore, recognition performance analysis is presented to help us probe the predicted performance on each activity using the CMI-Net with CB focal loss. The details are described as follows.

### 3.3.1 Comparisons with Existing Methods

The comparison results of the CMI-Net with three machine learning methods (i.e., NB, DT, and SVM) and two deep learning methods (i.e., CNN and ConvNet7) [55, 76] are illustrated in Table 3.1. The results reveal that the CMI-Net with softmax CE loss outperforms the machine learning algorithms with higher precision, recall, F1-score, and accuracy of 79.74%, 79.57%, 79.02%, and 93.37%, respectively. The reason for this superior performance is the convolution and pooling operations in CNN, which could achieve automated feature learning and aggregate more complex and general patterns without any domain knowledge [131]. The other CNN-based method [55] obtains inferior precision of 72.07% and accuracy of 82.94% compared to DT and SVM. This result is consistent with the "No Free Lunch" theorem [132] because this CNN-based method [55] is developed using leg-mounted sensor data. In addition, the CMI-Net with softmax CE loss performs better than ConvNet7 [76], which obtains lower precision, recall, F1-score, and accuracy of 79.03%, 77.79%, 77.90%, and 91.27%, respectively. This is attributed to the ability of the proposed architecture to effectively capture the complementary information and inhibit unrelated information of multi-modal data through deep multi-modality interaction. In addition, CMI-Net with CB focal loss ($\gamma$ = 0.5) enables the values of precision, recall, and F1-score to increase by 2.76%, 4.16%, and 3.92%, respectively, compared with CMI-Net with softmax CE loss. This reveals that the adoption of CB focal loss effectively improves the overall classification performance.

Table 3.1 Classification performance comparison with existing methods.

| Methods | Precision (%) | Recall (%) | F1-Score (%) | Accuracy (%) |
|---|---|---|---|---|
| Machine learning | | | | |
| Naïve Bayes | 70.90 | 72.41 | 69.42 | 76.60 |
| Decision tree | 75.67 | 73.90 | 74.35 | 88.83 |
| Support vector machine | 73.92 | 71.30 | 72.19 | 89.65 |
| Deep learning | | | | |
| CNN [55] | 72.07 | 76.91 | 73.42 | 82.94 |
| ConvNet7 [76] | 79.03 | 77.79 | 77.90 | **91.27** |
| Our proposed methods[#] | | | | |
| CMI-Net + softmax CE loss | **79.74** | **79.57** | **79.02** | **93.37** |
| CMI-Net + CB focal loss ($\gamma$ = 0.5)[*] | **82.50** | **83.73** | **82.94** | 90.68 |

[#] CB: Class-balanced; CE: Cross-entropy; CMI-Net: Cross-modality interaction network;

[*] The $\gamma$ of value is 0.5, which could refer to Table 3.3.

### 3.3.2 Ablation Studies

**Evaluation of CMIM**

To explore the effectiveness of CMIM and the impact of its position in the network on classification performance, the results corresponding to four different variants are shown in Table 3.2. The proposed CMI-Net with softmax CE loss shows superior performance to Variant0 (i.e., the network without CMIM), indicating the effective performance of the interaction module. Variant1, Variant2, and Variant3 (i.e., networks with CMIM inserted after 1st, 2nd, and 3rd max-pooling layer, respectively) did not perform better in terms of precision and recall compared with Variant0, which obtains precision and recall values of 79.02% and 77.09%, respectively. This might be explained by the fact that modality-specific features learned in the shallow layer are simple and contain noise, which interferes with the process by which CMIM learns complex inter-modality correlations, leading to poor predictions [133]. In addition, my proposed architecture obtains the best performance, which is because applying the CMIM after a deeper layer enables the network to discover more discriminative patterns and suppress irrelevant variations more effectively [7].

Table 3.2 Performance comparison of the proposed CMI-Net with its variants.

| Methods | Precision (%) | Recall (%) | F1-score (%) | Accuracy (%) |
|---|---|---|---|---|
| Variant0# | 79.02 | 77.09 | 76.88 | 91.76 |
| Variant1* | 78.18 | 77.07 | 77.40 | 92.17 |
| Variant2* | 77.50 | 78.44 | 77.91 | 92.92 |
| Variant3* | 78.36 | 76.94 | 77.02 | 92.62 |
| CMI-Net + softmax CE loss | **79.74** | **79.57** | **79.02** | **93.37** |

# Variant0 denotes the network without a cross-modality interaction module (CMIM);

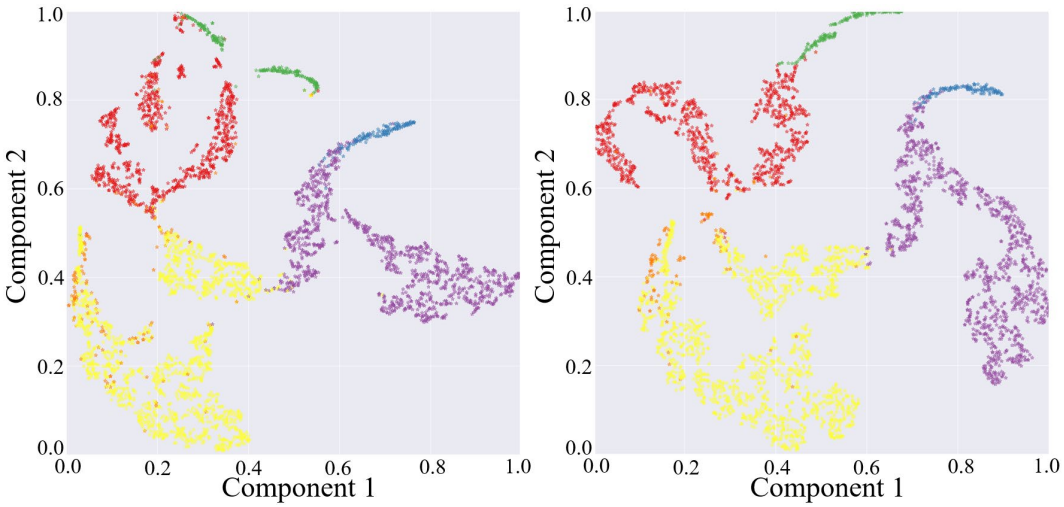
* Variant1, Variant2, and Variant3 denotes the network where the CMIM is inserted after the 1st, 2nd, and 3rd max-pooling layers, respectively.

The results above have proven that the inclusion of the CMIM in the network provides quantifiable improvements in identification performance. This is also reflected in the qualitative visualization of the embeddings and the corresponding clusters (Fig. 3.3), with the help of t-distributed stochastic neighbor-embedding (t-SNE), a technique for visualizing high-dimensional data by giving each data point a location in a two- or three-dimensional map [134]. Figure 3.3 shows the two-dimensional embedded features from the part test dataset after the

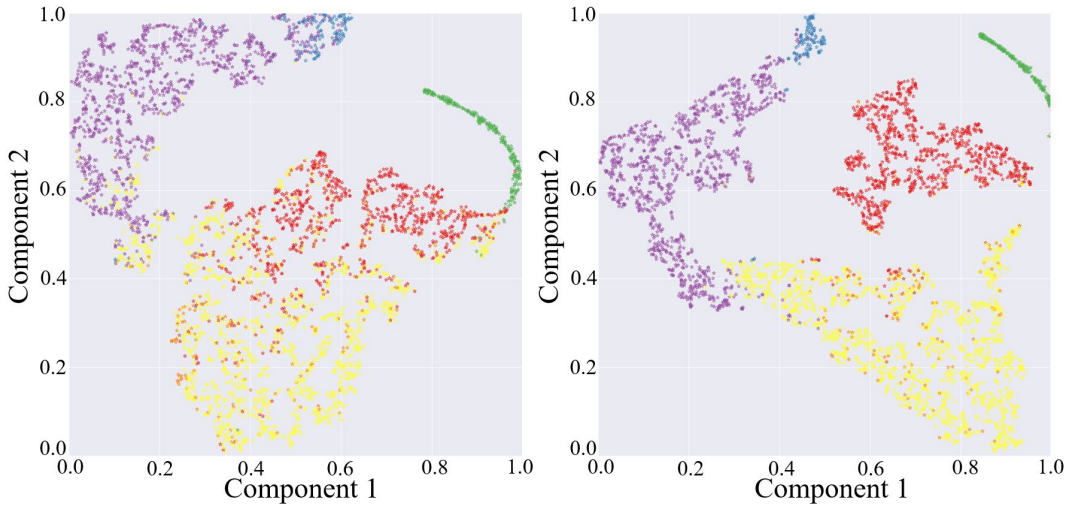
fully connected layers of both CNN branches under the network without and with CMIM by using the t-SNE technique with an init of 'pca' and perplexity of 30. Comparing the left and right columns in Fig. 3.3, it can be observed that more compact clusters are generated under the network with CMIM by reducing the intra-class distance and enlarging the inter-class distance. The core technical point is that the joint interaction module enables adaptive amplification of salient features and suppression of unrelated features based on information from two-modality data. To further provide insights into its contribution, I present two spatial attention maps for features extracted from the tri-axial accelerometer and tri-axial gyroscope data, as illustrated in Fig. 3.4, the value per pixel represents the contribution degree corresponding to each temporal period and each axis, and it is adaptively recalibrated through inter-modality interaction. Therefore, both quantitative and qualitative findings reinforce the suitability of the proposed CMI-Net to tasks using two-modality sensor data.

**Evaluation of CB Focal Loss**

To study the effect of CB focal loss on the optimization of CMI-Net, I show the quantitative performance in Table 3.3 and explore the sensitivity of its hyperparameter $\gamma$. CMI-Net with CB focal loss ($\gamma$ = 0.5) achieves the best precision of 82.50%, recall of 83.73%, and F1-score of 82.94%. This indicates that CB focal loss is beneficial to the improvement of classification performance when the modulation strength is controlled appropriately, whereas negative effects occur if the value of $\gamma$ is too large or too small.




(a) Accelerometer (without CMIM) (b) Accelerometer (with CMIM)



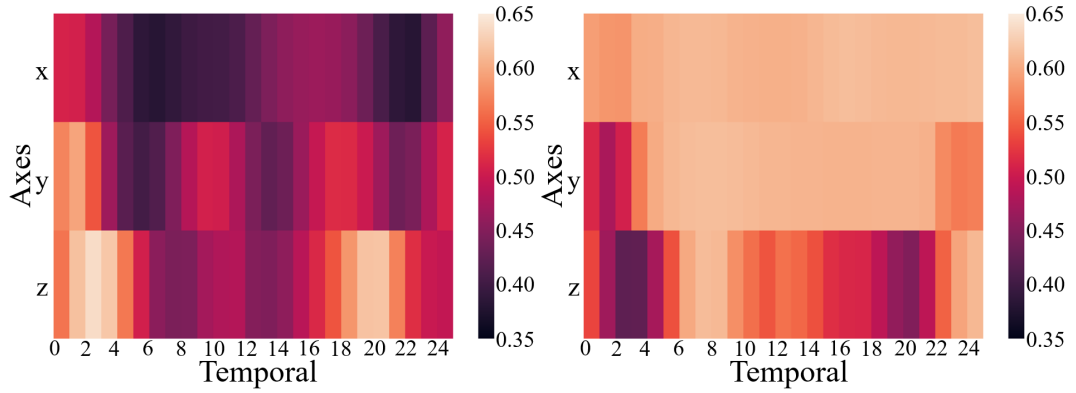

(c) Gyroscope (without CMIM) (d) Gyroscope (with CMIM)

Fig. 3.3 Embedding visualization of the features extracted from tri-axial accelerometer and gyroscope data under network without and with cross-modality interaction module (CMIM), respectively.

To provide further insight into the influence of CB focal loss ($\gamma = 0.5$) on the classification performance, I present the classification results of each activity under CMI-Net with CB focal loss and softmax CE loss, respectively, as shown in Fig. 3.5. It shows that precision, recall, and F1-score of the walking-natural are significantly improved, while other activities varies slightly when using CB focal loss. This explains that the overall classification performance increases mainly due to the increase in walking-natural, since the CB focal loss focuses more on difficult samples and samples of minority classes. However, the overall accuracy of CMI-Net with CB focal loss decreases by 2.69% (Table 3.3), which is related to the different variations of recall values in different activities and the current imbalanced dataset. In particular, the overall accuracy can also be presented as the weighted average of the recall value for each activity according to the sampling frequency of each activity. As shown in Fig. 3.5, the recall increases are 35.92% for walking-natural, 1.17% for standing, and 0.91% for galloping, and the recall decreases are 8.41% for walking-rider, 4.26% for eating, and 0.36% for trotting when using CB focal loss. It can be observed that all activities with increased recall belong to the minority class, while the remaining activities with decreased recall belong to the majority class. Therefore, the overall accuracy decreases, suggesting the necessity to collect a more balanced dataset in the future.

Table 3.3 Performance comparison between the softmax CE loss and CB focal loss with different $\gamma$.

| Loss Functions | Precision (%) | Recall (%) | F1-Score (%) | Accuracy (%) |
|---|---|---|---|---|
| Softmax CE Loss (baseline) | 79.74 | 79.57 | 79.02 | **93.37** |
| CB focal loss ($\gamma$ = 0.1) | 81.31 | 83.60 | 81.97 | 89.57 |
| CB focal loss ($\gamma$ = 0.5) | **82.50** | **83.73** | **82.94** | 90.68 |
| CB focal loss ($\gamma$ = 1) | 80.42 | 82.03 | 81.05 | 89.89 |
| CB focal loss ($\gamma$ = 2) | 78.92 | 78.48 | 77.97 | 91.05 |




(a) (b)

Fig. 3.4 Attention maps for features extracted from the tri-axial accelerometer (a) and gyroscope (b) data.

In addition, experiments under different loss functions are conducted to verify the effectiveness of the CB focal loss, as illustrated in Table 3.4. The contrasting losses mainly include CS_CE loss, CB_CE loss, focal loss, and ACS loss (see Section 3.2.2). I found that CB focal loss combining CB loss and focal loss performs better than any of them used alone, which indicates that adding the CB term to the focal loss function improves the overall classification performance on the imbalanced dataset. In addition, the precision, recall, and F1-score of CS_CE loss and CB focal loss increase by different degrees, while both accuracies decrease compared with softmax CE loss. Specifically, the accuracy is only 83.79%, although the recall reaches the highest value of 85.11%. This is because the recall of walking-rider is only 72.49%, although that of walking-natural is 69.16% (Fig. 3.6). The result further verifies that decreased accuracy occurs when using balancing techniques on the imbalanced dataset. In addition, I found that the recall of majority classes decreases while that of minority classes increases when using CS_CE loss and CB focal loss (Fig. 3.6). This result reveals that both losses effectively

focus on the samples of minority classes during training, but it is inevitable that more samples in majority classes are misclassified as minority classes so that overall accuracy would decrease.

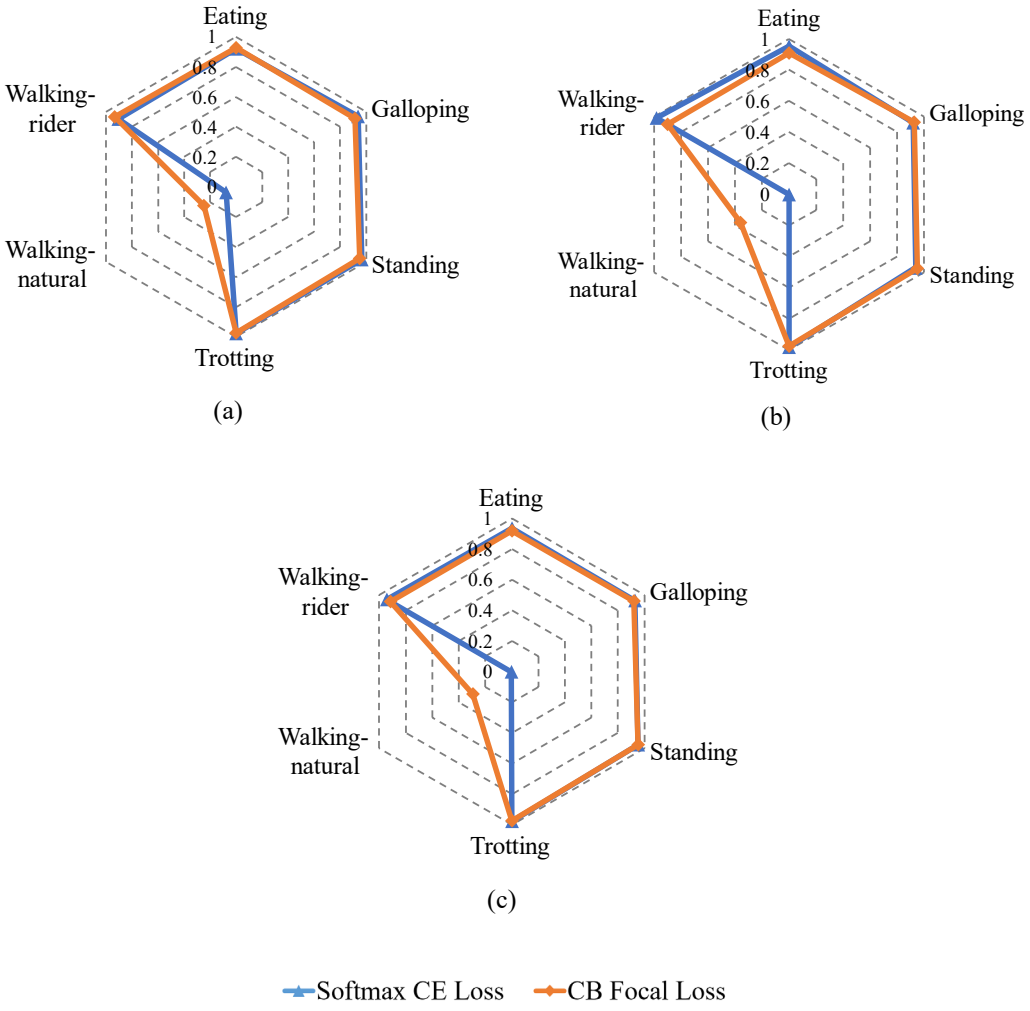



Fig. 3.5 Precision (a), recall (b), and F1-score (c) comparison of each activity under softmax cross-entropy (CE) loss and class-balanced (CB) focal loss.

### 3.3.3 Classification Performance Analysis

In Fig. 3.7, I show the precision and recall confusion matrix aggregating the classification results under 6-fold cross-validation when using CMI-Net with CB focal loss ($\gamma$ = 0.5). The precision and recall values of almost all activities are higher than 90% (i.e., the precision and recall for eating are 92.86% and 90.89%, for galloping are 91.41% and 92.89%, for standing are 95.18% and 95.11%, for trotting are 97.34% and 97.46%, and for walking-rider are 93.49%

and 90.01%, respectively), except for that of the walking-natural activity (Fig. 3.7). The low classification precision and recall of the walking-natural activity occur for two main reasons. The first reason is class imbalance. Walking-natural as the minority class in the dataset only occupies 3.8%, which is much less than 38.94% occupation of the majority class walking-rider, which easily causes the model to be biased towards the majority classes and results in poor minority class recognition performance. The second reason is severe confusion of walking-natural activity with other activities, especially eating and walking-rider activities. As shown in Fig. 3.7, 18.64% and 56.14% of the samples predicted to be class walking-natural have ground truth classes eating and walking-rider, respectively. In addition, 20.38% and 43.13% of the samples with ground truth class walking-natural are misclassified as class eating and walking-rider, respectively. This is because, during eating, the horse is slowly walking so that some samples of eating might contain walking activity [72]. The movement patterns of walking-natural and walking-rider are very similar, which interferes with the learning ability of the network for these two behavioral characteristics (Fig. 3.8). It also reveals that there is no major variability in horse walking patterns in the presence or absence of a rider. This is consistent with a previous study that found no major changes in horse limb kinematics, although the extension of the thoracolumbar region increases during walking with a rider compared with non-ridden walking [135]. In addition, there is confusion between galloping and trotting activities with misclassification of 6.93% of galloping as trotting. This might be related to the misinterpretation by the annotator during labeling, as it is not always clear when the activity transitions occur [72]. Additionally, a sample rate of 100 Hz may limit the distinction in the transition between trotting and cantering or galloping.

Table 3.4 Classification performance comparison with different loss functions.

| **Loss Functions**[#] | **Precision (%)** | **Recall (%)** | **F1-Score (%)** | **Accuracy (%)** |
|---|---|---|---|---|
| Softmax CE loss | 79.74 | 79.57 | 79.02 | **93.37** |
| Class-level | | | | |
| CS_CE loss [82] | **80.47** | **85.11** | **79.91** | 83.79 |
| CB_CE loss [123] | 75.35 | 75.70 | 75.47 | 90.61 |
| Sample-level | | | | |
| Focal loss [83] | 78.84 | 77.99 | 78.25 | **93.30** |
| ACS loss [84] | 77.03 | 76.54 | 76.60 | 92.05 |
| CB focal loss ($\gamma$ = 0.5) | **82.50** | **83.73** | **82.94** | 90.68 |

[#] CS_CE: cost-sensitive cross-entropy; CB_CE: class-balanced cross-entropy; ACS: adaptive class suppression.

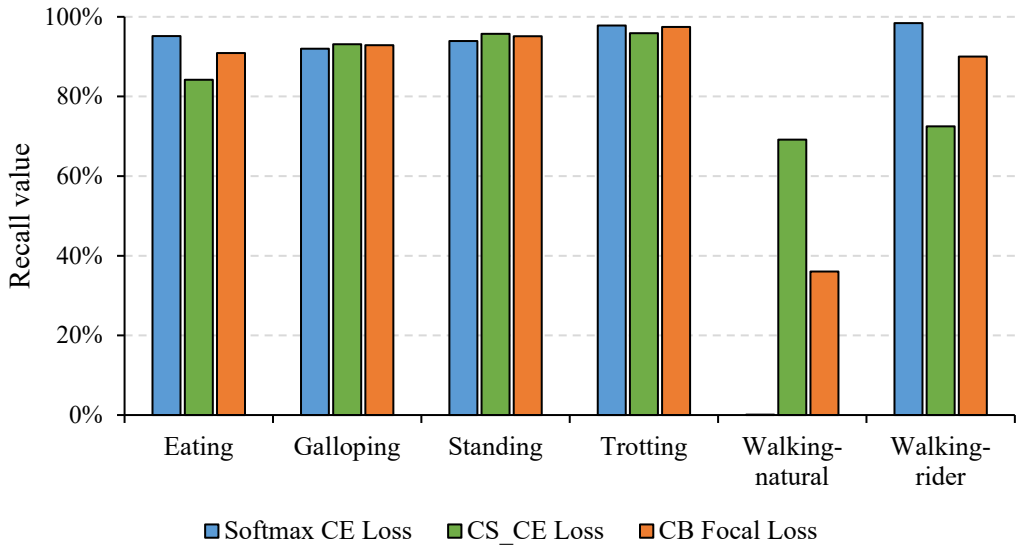



Fig. 3.6 Recall of different activities under different loss functions including softmax CE loss, cost-sensitive cross-entropy (CS_CE) loss, and CB focal loss.

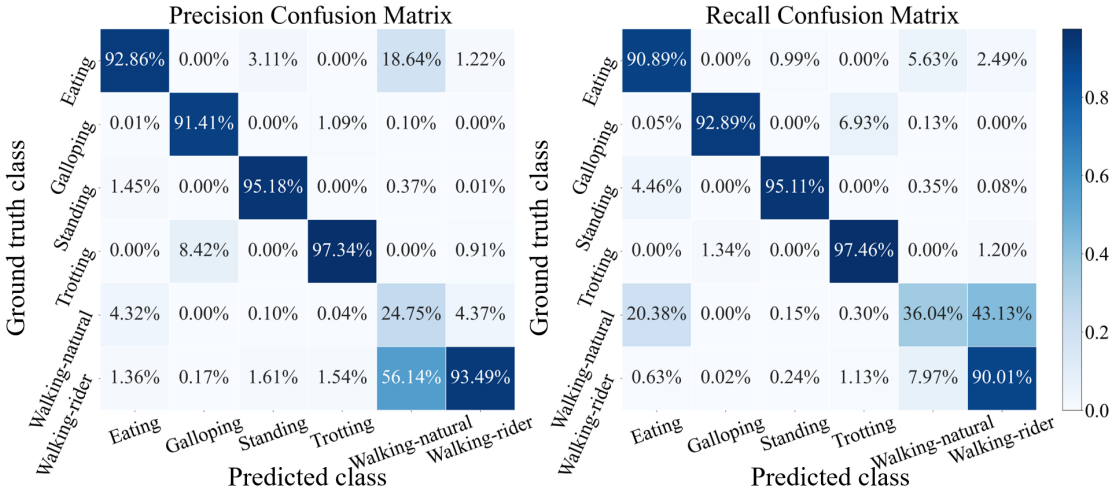



Fig. 3.7 Precision (a) and recall (b) confusion matrix of CMI-Net with CB focal loss ($\gamma$ = 0.5).

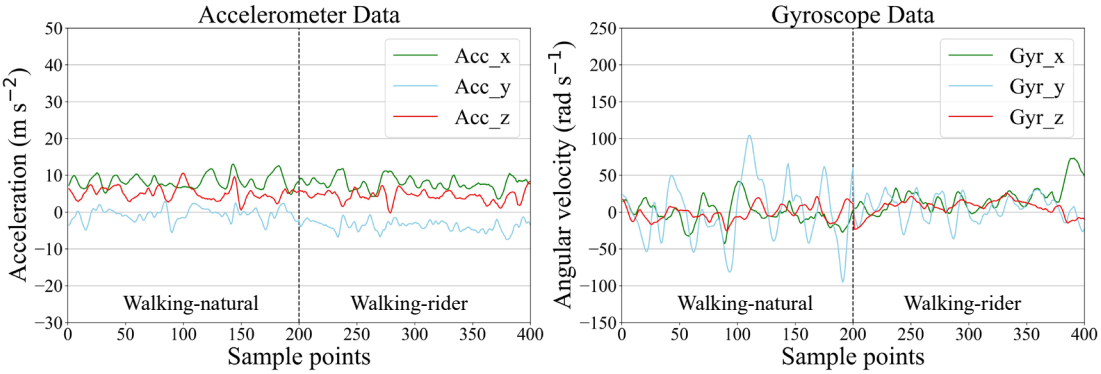



Fig. 3.8 Example of accelerometer and gyroscope data for walking-natural and walking-rider.

## 3.4 Summary

In this chapter, I develop a CMI-Net involving a dual CNN trunk architecture and a joint CMIM to improve horse activity classification performance. The CMI-Net effectively captures complementary information and suppresses unrelated information from different modalities. Specifically, the dual CNN architecture extracts modality-specific features, and the CMIM recalibrates temporal- and axis-wise features in each modality by utilizing multi-modal knowledge and achieves deep inter-modality interaction. To alleviate the class imbalance problem, a CB focal loss is leveraged for the first time to supervise the training of CMI-Net, which focuses more on the difficult samples and samples of minority classes during optimization. The results reveals that the CMI-Net with softmax CE loss outperforms the existing methods, and the adoption of CB focal loss effectively improves the precision, recall, and F1-score while slightly decreasing the accuracy. In addition, ablation studies demonstrates that applying the CMIM in the upper layer of CMI-Net could obtain better performance since high-level features contained more general patterns. CB focal loss also performs better than any class-level or sample-level reweighted losses used alone. In short, the favorable classification performance indicates the effectiveness of the proposed method.

## 3.5 Publication Related to This Chapter

1. **Mao, A.**, Huang, E., Gan, H., Parkes, R. S., Xu, W., & Liu, K.* (2021). Cross-modality interaction network for equine activity recognition using imbalanced multi-modal data. Sensors, 21(17), 5818.
2. **Mao, A.**, Huang, E., Xu, W., & Liu, K.* (2021, October). Cross-modality Interaction Network for Equine Activity Recognition Using Time-Series Motion Data. In Proceedings of the 2021 International Symposium on Animal Environment and Welfare (*ISAEW 2021*).

# Chapter 4

# 4 Privacy-preserving AAR with Decentralized Data

In the above chapter, I propose a novel deep learning-based model, i.e., cross-modality interaction network (CMI-Net), aiming to improve the performance of animal activity recognition (AAR) based on imbalanced multi-modal data. In general, the high performance of deep learning-based classification algorithms also largely relies on the availability of big data, which inevitably brings data privacy issues when collecting a centralized dataset from different farms. To address the data privacy issue, federated learning (FL) provides a distributed learning solution to train a shared model by coordinating multiple farms (clients) without sharing their private data. In this chapter, I investigate the challenges when directly applying FL to AAR tasks, including client-drift during local training and local gradient conflicts during global aggregation. To deal with these problems, I develop a novel FL framework called FedAAR to achieve AAR based on decentralized data over different farms. Concretely, I devise a prototype-guided local update (PLU) module to alleviate the client-drift issue, which introduces a global prototype as shared knowledge to force clients to learn consistent features. To reduce gradient conflicts between clients, I design a gradient-refinement-based aggregation (GRA) module to eliminate conflicting components between local gradients during global aggregation, thereby improving agreement between clients.

## 4.1 Introduction

Monitoring and assessing animal activities provide rich insights into their physical status and

circumstances, as activity is one of the most critical indicators of animal health and welfare [136]. Traditionally, animal activity monitoring largely relies on direct visual and behavioral observation, which is time-consuming and labor-intensive [2]. Over the past decade, automated AAR with wearable sensors, which allows staff to identify variations in the animal behavioral repertoire in real-time, has attracted increasing attention and achieved great success. In the sensor-based AAR systems, wearable sensors are attached to a certain part of the animal body (e.g., ear, neck, halter, back, and leg) to collect motion data (e.g., acceleration and angular velocity), which are then used for classifying animal activities (e.g., feeding, drinking, and resting) with suitable classification algorithms.

In recent years, deep learning has dominated the tasks in AAR due to the high performance achievable with the help of large-scale training datasets [4, 15]. For instance, convolutional neural networks (CNNs) are widely used to automatically classify various animal activities, such as the walking and ruminating of cattle [4], the trotting and cantering of horses [2], and the eating and petting of canines [5]. However, collecting a large corpus of centralized datasets from different sources (e.g., farms) often raises data privacy issues. FL has recently emerged as a distributed learning paradigm, providing an attractive solution to the data privacy problem [87–89]. FL allows learning from distributed clients (data sources) by aggregating the locally trained models without exchanging the client's data, aiming to build a shared model collaboratively while avoiding privacy leakage [16, 90, 91]. However, directly applying FL to AAR tasks often faces two major challenges, i.e., client-drift during local training and local gradient conflicts during global aggregation, which easily increase the difficulty in model convergence and cause extreme performance degradation [137].

First, the movement patterns of individual animals are often drawn from distinct distributions, which inevitably results in data heterogeneity between clients. Such data heterogeneity enlarges the inconsistency of learned features across clients, easily raising drift concerns between client updates since each client model is optimized towards its local objective instead of global optima during local training [94, 138]. To address this issue, some existing methods [92, 94, 95, 138] impose constraints on the local optimization by exploiting a model-level regularization term, aiming to facilitate all local models to approach consistent views. For instance, FedProx restricted local model parameters to be close to global parameters by adding a proximal term in the local training [92]. SCAFFOLD corrected for client-drift by using control variates to overcome gradient dissimilarity in local training [94]. Both FedLSD and FedCAD regarded the distributed global model as a teacher and distilled its predictions on local data to guide local optimization [95, 139]. However, these methods only emphasize constraints

on local models instead of directly forcing clients to learn consistent features, consequently yielding sub-optimal performance.

Second, gradients among clients often possess inconsistent directions and even have conflicting components due to the inconsistency between local optimization objectives in the context of data heterogeneity. Directly aggregating all local gradients in standard FL methods easily leads to mutual interferences among clients' knowledge, further hampering the process of model convergence and exacerbating the risk of model divergence. To alleviate this problem, most existing works [140–142] attempted to modify the global aggregation mechanism by dynamically adjusting aggregation weights to local gradients under different criteria. For example, IDA assigned each client weight based on the inverse distance of its gradient to the averaged gradient across all clients [142]. Precision-weighted FL aggregated local gradients by averaging gradients by the inverse of their estimated variance [141]. ABAVG combined gradients over clients by the accuracies of local models on the validation set at the server-side [140]. However, aggregating local gradients merely by changing their weights still cannot drastically remove conflicting components in gradients and may even lose important information in gradients.

In this chapter, I propose a novel FL framework, namely FedAAR, to achieve automated AAR by uniting decentralized data while tackling the above-mentioned issues. To alleviate the client-drift issue, I design a PLU module, in which I introduce a globally shared prototype (class-wise feature representation) as shared knowledge to constrain local optimization. To reduce conflicts between local gradients, I devise a new GRA module to constantly recalibrate local gradients throughout the training process. The proposed FedAAR is trained based on a public dataset [72], and its generalization performance is compared with that of the state-of-the-art FL strategies and with the centralized learning algorithm.

## 4.2 Materials and Methods

### 4.2.1 Preliminaries for Federated Learning

A FL system coordinates $K$ clients to collectively train a global model while keeping their data stored locally, effectively reducing the potential for violating data privacy. In each client $k$, the local dataset $\{(x_i^k, y_i^k)\}_{i=1}^{N^k}$ is sampled from a distribution $\mathcal{D}^k$, where $y_i^k \in \{1, \cdots, C\}$ corresponds to the ground-truth label of the data instance $x_i^k$, $C$ is the number of label

categories, and $N^k$ is the data number of the $k$-th client. The training of a standard FL system mainly consists of $T$ communication rounds between a global server and $K$ clients, with the detailed procedure of each communication round divided into the following three steps:

**Step 1.** All clients synchronously download a global model $w^{global}$ from a global server.

**Step 2.** Each client $k$ uses the global model $w^{global}$ to initialize its local model $w^k$ (i.e., $w^k_{initial} = w^{global}$) and then conducts local training for $E$ epochs, i.e., minimizing the local optimization objective by using a gradient descent algorithm:

$$\min \frac{1}{N^k}\sum_{i=1}^{N^k} \mathcal{L}_{CE}^k(w^k, x_i^k, y_i^k), \tag{4.1}$$

where $\mathcal{L}_{CE}^k$ denotes the cross-entropy loss function of the $k$-th client. After local training, I can obtain the updated local model $w^k_{updated}$ and local gradient $g^k$ (i.e., the difference between the updated local model $w^k_{updated}$ and the initial local model $w^k_{initial}$).

**Step 3.** Each client $k$ then uploads its local gradient $g^k$ to the global server. These local gradients $\{g^k\}_{k=1}^K$ from all clients are aggregated by directly averaging to generate global gradients $g^{global}$:

$$g^{global} = \frac{1}{K}\sum_{k=1}^{K} g^k. \tag{4.2}$$

Afterwards, the global gradient $g^{global}$ is used to further update the original global model $w^{global}$:

$$w^{global} = w^{global} + g^{global}. \tag{4.3}$$

The updated global model $w^{global}$ will be sent to all clients again in the next communication round (**Step 1**). The above steps are performed repeatedly until the global model achieves convergence.

Implementing the above-described standard FL (e.g., FedAvg) systems heavily relies on the assumption that no data heterogeneity issues occur across clients (i.e., data between clients follow a uniform distribution). However, this assumption does not hold in AAR tasks because the discrepancy of movement patterns among individual animals often results in data heterogeneity, thus giving rise to detrimental effects on the training of FL. Specifically, data heterogeneity across clients inevitably enlarges the inconsistency of learned features between clients, thereby inducing the drift between client updates as each client model is optimized towards its local objective instead of global optima [95]. In addition, local gradients may conflict with each other due to the inconsistent local objectives, resulting in knowledge

interference among clients when directly aggregating all local gradients. Hence, there is a need to reformulate the above optimization process to adapt to AAR tasks.

### 4.2.2 The Federated Learning Framework for AAR

**Overview**

To achieve automated AAR in the context of data heterogeneity between clients, I propose a novel FL framework called FedAAR, as illustrated in Fig. 4.1. The training of FedAAR consists of $T$ communication rounds between a global server and $K$ clients, where the detailed procedures of each communication round include three steps as follows. (1) Each client $k$ first downloads the same global model $w^{global}$ and global prototype $P^{global}$ from a global server simultaneously. (2) Based on the local dataset $\mathcal{D}^k$ in each client $k$, the local model $w^k$ is initialized as the global model $w^{global}$ and then trained in a PLU module (Fig. 4.1 (a)). After local training, the updated local model $w^k_{updated}$ and local prototype $P^k_{updated}$ can be obtained. Then, the local gradient $g^k$ can be calculated as the difference between the updated local model $w^k_{updated}$ and the initial local model (i.e., the downloaded global model $w^{global}$). (3) All local gradients $\{g^k\}_{k=1}^K$ and local prototypes $\left\{P^k_{updated}\right\}_{k=1}^K$ are then uploaded to the global server simultaneously. Afterwards, these local gradients $\{g^k\}_{k=1}^K$ are aggregated to a global gradient $g^{global}$ using a GRA module (Fig. 4.1 (b)) and then used to update the global model $w^{global}$ (see Eq. (4.3)). In addition, local prototypes $\left\{P^k_{updated}\right\}_{k=1}^K$ are aggregated to update the global prototype $P^{global}$. The updated global model $w^{global}$ and global prototype $P^{global}$ are sent to all clients again in the next communication round. The above three processes are repeated until the global model achieves convergence.

**Prototype-Guided Local Update**

To alleviate client-drift issues in the local training, I devise a PLU module Fig. 4.1 (a), which introduces a global prototype to serve as the shared feature knowledge to guide local optimization.

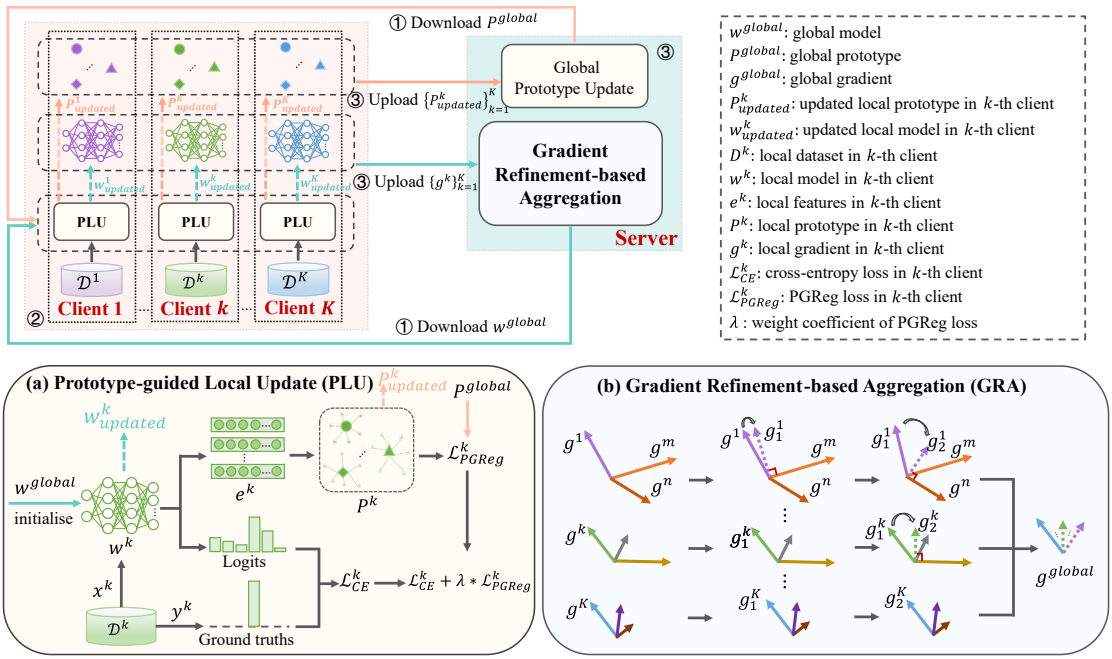



Fig. 4.1 Overall architecture of the proposed FedAAR framework.

First, each client $k$ simultaneously downloads the same global model $w^{global}$ and global prototype $P^{global}$ from the server. Based on the local dataset $\mathcal{D}^k$ in each client $k$, the local model $w^k$ is initialized as the global model $w^{global}$ and then trained in the proposed PLU module. Specifically, given batchwise samples $\{(x_i^k, y_i^k)\}_{i=1}^{\mathcal{B}}$ with $C$ categories in client $k$, I first adopt the feature extractor to extract feature representation $\{e_{c,i}^k\}_{i=1}^{\mathcal{B}_c}$ for samples in each class $c$, where $\mathcal{B}$ and $\mathcal{B}_c$ represent the number of total samples and the $c$-th categorial samples in a batch, respectively. These features are then used to calculate the corresponding class-wise prototype $\{P_c^k\}_{c=1}^C$, where $P_c^k$ is the mean value of feature representations of samples in class $c$, i.e.,

$$P_c^k = \sum_{i=1}^{\mathcal{B}_c} \frac{e_{c,i}^k}{\mathcal{B}_c}. \tag{4.4}$$

Herein, I empirically use the network that removes the last fully connected layer as the feature extractor [143].

Inspired by prototype learning, in which gathering the prototypes across heterogeneous datasets enables the incorporation of feature representations over various data distributions [143, 144], I bring a global prototype $\{P_c^{global}\}_{c=1}^C$ ($P^{global}$) aggregated across clients as consistent feature-level views to guide local training. I propose a new prototype guidance regularization

(PGReg) loss $\mathcal{L}_{PGReg}^{k}$ as follows:

$$\mathcal{L}_{PGReg}^{k} = \sum_{c\in C}\left\|P_c^k - P_c^{global}\right\|_2, \tag{4.5}$$

where $P_c^{global}$ denotes the global prototype of the $c$-th category and $\|\cdot\|_2$ denotes the L2 distance. The PGReg loss sufficiently encourages local prototype $\{P_c^k\}_{c=1}^{C}$ of each client to approach the same global prototype $\left\{P_c^{global}\right\}_{c=1}^{C}$, effectively keeping all clients as having a consistent direction of feature learning. Thus, the total loss function can be reformulated as the linear combination of the original classification loss $\mathcal{L}_{CE}^{k}$ and PGReg loss $\mathcal{L}_{PGReg}^{k}$:

$$\mathcal{L}^{k} = \mathcal{L}_{CE}^{k} + \lambda * \mathcal{L}_{PGReg}^{k}, \tag{4.6}$$

where $\lambda$ is the weight coefficient of PGReg loss and its value equals 0 in the initial state. Then, I conduct the local training (see Eq. (4.1)) to obtain the updated local model $w_{updated}^{k}$. The updated local prototype $\left\{P_{c,updated}^{k}\right\}_{c=1}^{C}$ can be computed based on the feature vectors of correctly classified samples using the updated local model $w_{updated}^{k}$. The local gradient $g^k$ can be calculated as the difference between the updated local model $w_{updated}^{k}$ and the initial local model (i.e., the downloaded global model $w^{global}$). Afterwards, all local gradients $\{g^k\}_{k=1}^{K}$ and local prototypes $\left\{P_{updated}^{k}\right\}_{k=1}^{K}$ are uploaded to the server to update the original global model and global prototype.

At the server-side, to avoid the attack resulting from noise components involved in the updated local prototypes $\left\{P_{updated}^{k}\right\}_{k=1}^{K}$, I devise a novel adaptive global prototype update process. Instead of directly replacing the global prototype with the average values of local prototypes over clients, I define the updated global prototype $\left\{P_c^{global}\right\}_{c=1}^{C}$ ($P^{global}$) as a linear combination of the weighted averaged local prototypes $\{\bar{P}_c\}_{c=1}^{C}$ and the original global prototype $\left\{P_c^{global}\right\}_{c=1}^{C}$:

$$P_c^{global} = \gamma_c * P_c^{global} + (1-\gamma_c) * \bar{P}_c, \tag{4.7}$$

where $\gamma_c$ is an adaptively balanced coefficient controlling the updating degree of the global prototype and $\bar{P}_c = \sum_{k=1}^{K} n_c^k P_{c,updated}^k / \sum_{k=1}^{K} n_c^k$, where $n_c^k$ represents the number of correctly classified samples belonging to the $c$-th category in client $k$. Considering that the updated global prototype may be transferred close to each other due to noise components, inducing similarity increases of inter-class feature vectors [26], I modulate $\gamma_c$ according to the

intra-class and inter-class distance between local prototypes and the original global prototype:

$$\gamma_c = \frac{exp^{d\left(\bar{P}_c, P_c^{global}\right)}}{exp^{d\left(\bar{P}_c, P_c^{global}\right)} + exp^{d\left(\bar{P}_c, P_{c'}^{global}\right)}}, \tag{4.8}$$

where $d(\cdot)$ denotes the Euclidean distance and $P_{c'}^{global}$ is the global prototype of class $c'$, that is the closest class to class $c$, i.e., $c' = \underset{j \in \{1,2,\ldots,C\} \setminus c}{\operatorname{argmin}} d\left(P_c^{global}, P_j^{global}\right)$. Intuitively, when the averaged local prototype $\bar{P}_c$ is farther from the global prototype of the same class $c$ than the global prototype of its closest class (i.e., $d\left(\bar{P}_c, P_c^{global}\right) > d\left(\bar{P}_c, P_{c'}^{global}\right)$), the contribution of $\bar{P}_c$ on the update process (Eq. (4.7)) should be lower than that of $P_c^{global}$. Note that when the value of $P_c^{global}$ is empty at the early training phase, I directly put the updated averaged prototype $\bar{P}_c$ into global prototype $P_c^{global}$.

**Gradient-Refinement-Based Aggregation**

To reduce conflicts among local gradients during global aggregation, I designe a new GRA module Fig. 4.1 (b), which eliminates the conflicting components between local gradients, ensuring all refined local gradients point in a positive direction to improve the agreement across clients.

Given a set of local gradients $\{g^k\}_{k=1}^K$ that are uploaded to the global server, I first characterize any two of these gradients as conflicting when their directions point away from one another (i.e., having a negative cosine similarity). Herein, I aim to reconstruct consensus vectors by refining the conflicting local gradients and keeping the non-conflicting local gradients invariant. Fig. 4.2 visualizes the main step of the local gradient refinement process. Specifically, suppose $g^i$ is the local gradient at the $i$-th client and $g^j$ is selected in a random order from the rest of the local gradients, where $i \in \{1,2,\ldots,K\}$ and $j \in \{1,2,\ldots,i-1,i+1,\ldots,K\}$. The cosine similarity between $g^i$ and $g^j$ can be denoted as $\cos\theta_{i,j}$, where $\theta_{i,j}$ is the angle between $g^i$ and $g^j$. As shown in Fig. 4.2 (a), if $g^i$ conflicts with $g^j$ (i.e., cosine similarity $\cos\theta_{i,j} < 0$), I remove the component of $g^i$ in the direction fully opposite that of $g^j$ and alter $g^i$ by its projection $\tilde{g}^i$ onto the normal plane of $g^j$:

$$\tilde{g}^i = g^i - \frac{g^i \cdot g^j}{\|g^j\|^2} g^j. \tag{4.9}$$

If $g^i$ and $g^j$ are not in conflict (i.e., cosine similarity $\cos\theta_{i,j} > 0$), I retain the original local gradient $g^i$ as unchanged (i.e., $\tilde{g}^i = g^i$), as shown in Fig. 4.2 (b). Afterwards, the updated

$\tilde{g}^i$ is further selectively updated according to the condition of whether there are conflicting components compared to other local gradients. This process is repeated until all of the local gradients are compared.

Supposing $\{\tilde{g}^k\}_{k=1}^K$ is a collection of refined local gradients, I then aggregate them using Eq. (4.2) and Eq. (4.3) to further update the global model $w^{global}$.

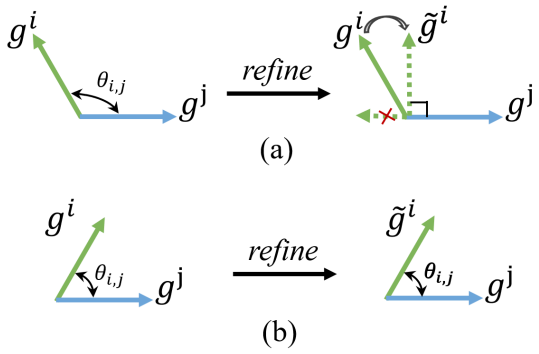



Fig. 4.2 Process of gradient refinement.

## 4.2.3 Datasets and Data Preprocessing

This chapter conducts experiments on the horse dataset [72] described in Section 2.5.1. The dataset is a centralized dataset comprising 87,621 2-s samples that are collected from six horses, where the sample number of each individual (i.e., Happy, Zafir, Driekus, Galoway, Patron, and Bacardi) is 23,625, 11,071, 10,127, 24,602, 12,849, and 5347, respectively. As done in Chapter 3, I also exploit the motion data from the tri-axial accelerometer and tri-axial gyroscope as the input samples, which are normalized before being input into the network (see Eq. (3.10)).

## 4.2.4 Design of Experiments

### Evaluation Metrics

The comprehensive performance of the activity classification model is indicated by four evaluation metrics, including precision, recall, F1-score, and accuracy (see Eq. (3.11) to Eq. (3.14)). Each indicator value is multiplied by 100 as the result to reflect the difference in indicator values more clearly.

**Implementation Details**

In Chapter 3, I establish a cross-modality interaction network (CMI-Net) for horse activity recognition based on accelerometer and gyroscope data. The CMI-Net, consisting of a dual CNN trunk architecture and a joint cross-modality interaction module, has been validated to effectively improve the classification performance for horse activities. Therefore, I select the CMI-Net as the primary model architecture of global and local models to achieve the proposed FedAAR.

During training, I use softmax cross-entropy loss with L2 regularization (a weight decay of 0.15). An Adam optimizer with an initial learning rate of $5 \times 10^{-5}$ is used, and the learning rate decreases by 0.1 times every 20 epochs. The communication rounds $T$ and batch size are set to 100 and 256, respectively. If not specified, the value of local training epochs $E$ is set to 1, and the weighting coefficient $\lambda$ in PLU is set to 0.05 by default. The global model is initialized randomly before being downloaded to clients in the first round. Over the communication rounds, the best global model with the highest test accuracy is saved as the optimal model. To verify the model's generalizability, I perform the leave-one-out-based validation method. Specifically, I separately run three times in each experiment. In each run (time), I randomly select five horses from the original six horses and individually assign these five horses to five farms (clients). Each of these five horses' data serves as each client's data, and all client data are used as training data to train a shared global model collectively. The data from the remaining horse (the sixth horse) are then used as the test data to verify the performance of the trained global model. The final test result of the model performance is presented in the format mean ± std from the three runs. This kind of data allocation can well simulate practical scenarios, i.e., data heterogeneity across farms, since the movement patterns of individual animals are often drawn from distinct distributions. All experiments are executed using the PyTorch framework on an NVIDIA Tesla V100 GPU. The source code is available at https://github.com/Max-1234-hub/FedAAR (accessed on 21 August 2022).

## 4.3 Results and Discussion

Overall, the experimental results demonstrate that the proposed FedAAR outperforms the state-of-the-art FL strategies from both quantitative and qualitative perspectives while exhibiting performance close to that of the centralized learning algorithm. Ablation studies are then carried out to evaluate the effectiveness of the PLU and GRA module on the classification capability.

In addition, a comprehensive investigation of FedAAR's performance is conducted at different levels of three practical conditions (i.e., dataset sizes on local clients, communication frequency between local clients and the global server, and client numbers). The experimental results of FedAAR are compared with those of its corresponding baseline, further validating the performance advantages of my method. In the end, some possible future research directions are proposed. The details are described as follows.

### 4.3.1 Comparisons with State-of-the-art Methods

**Quantitative Comparison**

I compare the performance of the FedAAR with the state-of-the-art FL approaches (i.e., FedAvg, FedPorx, IDA, SiloBN, FedBN, and precision-weighted FL) and with centralized learning. As illustrated in Table 4.1, the proposed framework outperforms all of the selected state-of-the-art FL methods, with the highest average values of 75.23%, 75.17%, 74.70%, and 88.88% in precision, recall, F1-score, and accuracy, respectively. These results demonstrate the promising capabilities of FedAAR for animal behavioral classification. In particular, compared with the precision-weighted FL [141], which obtains relatively good performance among the selected state-of-the-art FL methods, the proposed approach achieves remarkable increments of 3.75%, 9.39%, 8.34%, and 4.22% in the average values of the precision, recall, F1-score, and accuracy, respectively. This can be ascribed to the ability of my architecture to effectively alleviate client-drift concerns in local training and conflicts of local gradients during global aggregation. In addition, centralized learning provides the upper bounds for FL, because the conflicts between clients in the FL framework influence the model aggregation and further induce the performance degradation of the global model. Compared with centralized learning, the performance of my method is close, with 3.64%, 3.26%, and 3.49% lower average values of the recall, F1-score, and accuracy, respectively. This further reveals the favorable performance of my method. It is also worth noting that the proposed approach demonstrates smaller variances than the state-of-the-art works, with 1.01%, 3.92%, 2.49%, and 1.36% variance in the precision, recall, F1-score, and accuracy, respectively, indicating its good stability and robustness.

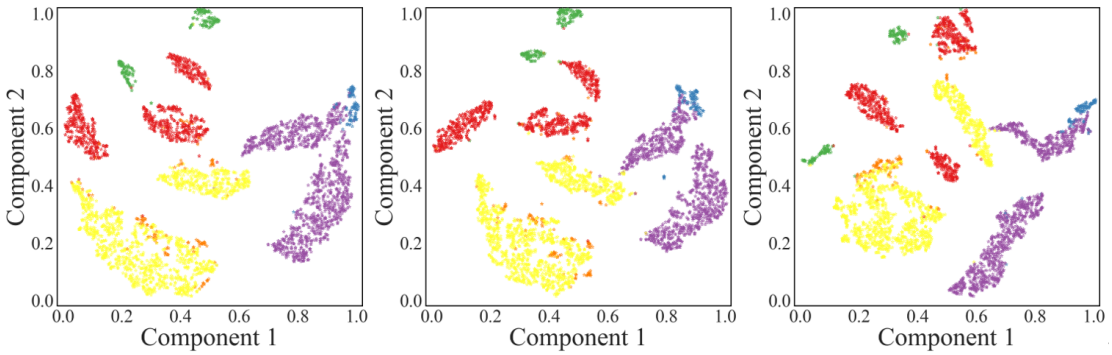
**Qualitative Comparison**

To qualitatively verify the proposed approach, I visualize the feature vectors of the test set before the last fully connected layer within FedAAR and the state-of-the-art FL models, with

the help of t-distributed stochastic neighbor embedding [29]. As illustrated in Fig. 4.3, the two-dimensional embeddings can reflect the distribution of the network features in the feature space and indicate the generalization ability of models, in which each point corresponds to a sample and different colors represent different category labels (ground-truth). Better generalization means that the feature points of samples belonging to the same class cluster closer to each other, whereas the points between different classes are located far from each other. From the embedding visualization, we can observe that the proposed FedAAR displays more compact clusters within the same categories and larger distances between different categories compared to the selected state-of-the-art methods. This reflects the success of my approach in improving the consistency of update directions across clients from both the local optimization and global aggregation perspectives, which is beneficial to the generalization performance promotion of the global model.

Table 4.1 Comparative results (mean ± std) of the proposed FedAAR with state-of-the-art federated learning (FL) methods.

| Method | Precision (%) | Recall (%) | F1-Score (%) | Accuracy (%) |
|---|---|---|---|---|
| Centralized learning | **83.34 ± 10.81** | **78.81 ± 2.40** | **77.96 ± 2.28** | **92.37 ± 3.84** |
| FedAvg [87] | 71.10 ± 4.42 | 64.96 ± 9.81 | 65.40 ± 8.93 | 84.31 ± 3.05 |
| FedProx [92] | 71.10 ± 4.42 | 64.93 ± 9.83 | 65.37 ± 8.96 | 84.30 ± 3.06 |
| IDA [142] | 70.67 ± 5.45 | 64.35 ± 10.68 | 64.27 ± 10.02 | 84.36 ± 3.33 |
| SiloBN [145] | 71.15 ± 2.73 | 64.57 ± 9.41 | 64.96 ± 7.94 | 83.18 ± 2.64 |
| FedBN [93] | 70.90 ± 2.84 | 65.45 ± 8.81 | 65.82 ± 7.14 | 83.72 ± 2.16 |
| Precision-weighted FL [141] | 71.48 ± 3.78 | 65.78 ± 9.15 | 66.36 ± 8.03 | 84.66 ± 2.68 |
| Our FedAAR | **75.23 ± 1.01** | **75.17 ± 3.92** | **74.70 ± 2.49** | **88.88 ± 1.36** |




(a) FedAvg (b) FedProx (c) SiloBN

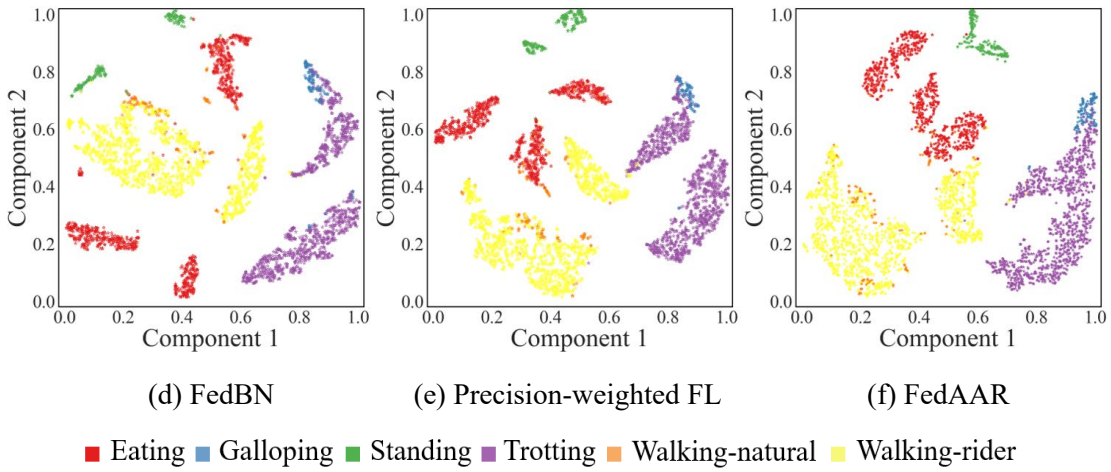



Fig. 4.3 t-distributed stochastic neighbor embedding (t-SNE) visualization of the feature vectors produced by the proposed FedAAR and other federated learning (FL) approaches.

## 4.3.2 Ablation Studies

### Evaluation of PLU and GRA Module

***Quantitative analysis.*** To investigate the effectiveness of PLU and GRA on classification performance, I design four different experimental settings as follows. (1) Baseline: I apply FedAvg as my baseline for the ablative comparison; (2) Baseline + PLU: I use the PLU module instead of the original optimization process in the baseline during local training; (3) Baseline + GRA: I replace the original weighted average mechanism in the baseline with the GRA module during global aggregation; (4) FedAAR: I use the proposed framework involving both PLU and GRA modules simultaneously. The quantitative results are shown in Table 4.2. It is remarkable that the two modules individually yield desirable performance improvements over the baseline, which proves that each of the PLU and GRA modules plays an important role in AAR tasks with data heterogeneity issues. In particular, the GRA module contributes to the tremendous performance improvements, with increases of 3.07%, 9.23%, 8.34%, and 3.47% in the average values of the precision, recall, F1-score, and accuracy, respectively. The variances also significantly decline, by 3.47%, 6.85%, 7.63%, and 2.27% in the precision, recall, F1-score, and accuracy, respectively, when the GRA module is used separately. These experimental results validate the significance of the GRA module in the model's performance and robustness improvements. The inclusion of the PLU module in addition to the GRA module enables all clients to possess consistent guidance directions of feature learning and further obtain relative gains of 1.06%, 0.98%, 0.96%, and 1.10% in the average values of the precision, recall, F1-

score, and accuracy, respectively.

Table 4.2 Evaluation results (mean ± std) of the gradient-refinement-based aggregation (PLU) module and guided local update (GRA) module on classification performance.

| Method | Precision (%) | Recall (%) | F1-Score (%) | Accuracy (%) |
|---|---|---|---|---|
| Baseline | 71.10 ± 4.42 | 64.96 ± 9.81 | 65.40 ± 8.93 | 84.31 ± 3.05 |
| Baseline + PLU | 73.04 ± 2.89 | 65.98 ± 9.34 | 67.10 ± 8.02 | 84.92 ± 3.00 |
| Baseline + GRA | 74.17 ± 0.95 | 74.19 ± 2.96 | 73.74 ± 1.30 | 87.78 ± 0.78 |
| FedAAR | **75.23 ± 1.01** | **75.17 ± 3.92** | **74.70 ± 2.49** | **88.88 ± 1.36** |

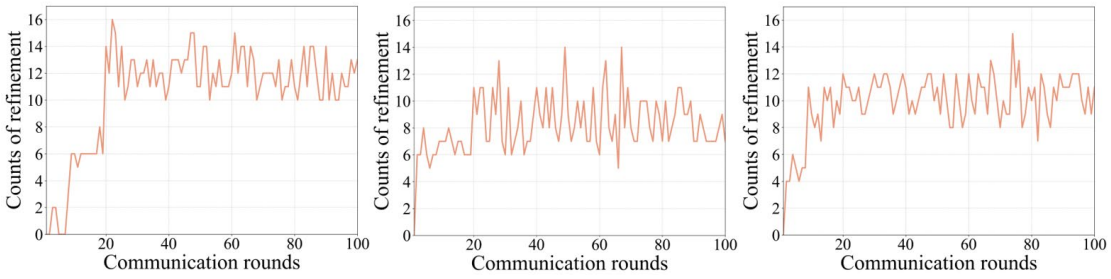
***Analysis of the hyper-parameter $\lambda$ in PLU.*** The hyper-parameter $\lambda$ in Eq. (4.6) represents the weight of newly added PGReg loss, corresponding to the constraint degree of global knowledge on local training. I conduct experiments to evaluate the performance of my approach with different $\lambda$ values (i.e., 0.01, 0.03, 0.05, 0.07, and 0.09), with the results shown in Table 4.3. The FedAAR achieves clearly the best average values in the recall and F1-score when $\lambda$ is set to 0.05, while obtaining a performance in the precision and accuracy comparable to the model with $\lambda$ set to 0.09. Although the precision and accuracy arrive at the highest average values when $\lambda$ is set to 0.09, the model exhibits a poor recall and F1-score. In addition, the average recall and F1-score values first increase and then decrease as $\lambda$ varies from 0.01 to 0.09, which illuminates the likely benefit of properly choosing the value of $\lambda$ for the improvement of overall classification performance.

Table 4.3 Experimental results (mean ± std) of FedAAR with different weighting coefficients $\lambda$ of the prototype guidance regularization loss.

| $\lambda$ | Precision (%) | Recall (%) | F1-Score (%) | Accuracy (%) |
|---|---|---|---|---|
| 0.01 | 74.39 ± 1.07 | 74.56 ± 2.81 | 74.08 ± 1.07 | 88.17 ± 0.61 |
| 0.03 | 75.10 ± 0.82 | **74.87 ± 3.39** | **74.53 ± 1.93** | 88.55 ± 0.80 |
| 0.05 | **75.23 ± 1.01** | **75.17 ± 3.92** | **74.70 ± 2.49** | **88.88 ± 1.36** |
| 0.07 | 74.97 ± 1.08 | 74.66 ± 4.07 | 74.18 ± 2.59 | 88.88 ± 1.65 |
| 0.09 | **75.94 ± 2.14** | 72.50 ± 5.21 | 72.80 ± 3.85 | **88.89 ± 1.67** |

***Visualization of refinements in GRA.*** To provide further insights into the enormous contributions of the GRA module, I visualize the counts of gradient refinement operations during the training process over three runs in Fig. 4.4. It can be observed that various numbers of gradient modulation operations occur in each communication round, which confirms that conflicts among local gradients arise continuously during the training process. In addition, this

observation implies that the framework can constantly and steadily recalibrate the local gradients across clients, effectively enhancing the model's performance. This finding also reinforces the suitability of the proposed GRA module for AAR tasks in the context of data heterogeneity.




(a) (b) (c)

Fig. 4.4 Counts of refinement operations during the training process over three runs from (a) to (c).

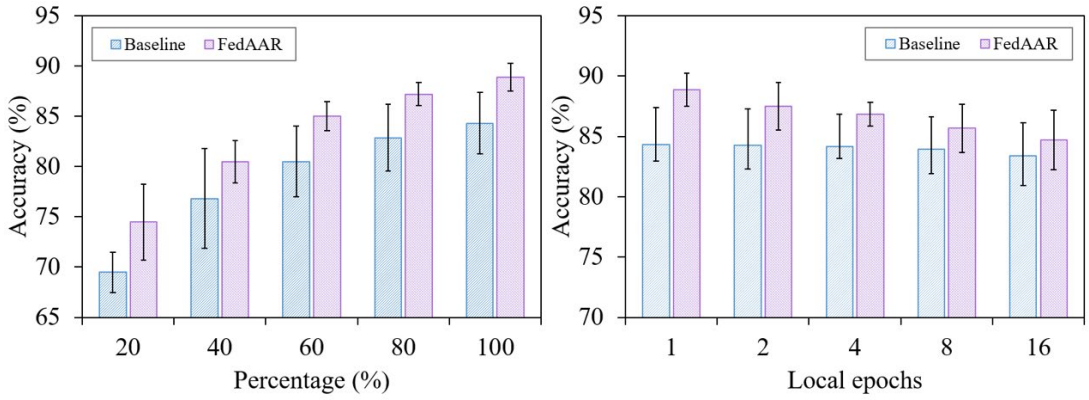
**Analysis of Local Dataset Size**

To observe the behavior of the proposed method over different data capacities, I present in Fig. 4.5 (a) the average test accuracies of FedAAR and the baseline under various local dataset percentages (i.e., 20%, 40%, 60%, 80%, and 100%). The test accuracies of both FedAAR and the baseline decrease gradually as the number of training samples reduces, but the accuracy of FedAAR continues to exceed that of the baseline, validating the performance advantages of my method under scenarios with a small amount of data. In addition, FedAAR conducted on 60% local data still obtains higher accuracy than the baseline performed on full-sized local data, which reveals that my approach can effectively mitigate the performance degradation due to the reduced data amount.

**Analysis of Communication Frequency**

The communication frequency between local clients and the global server may influence learning behavior. I decrease the communication frequency by increasing the local updating epochs and present in Fig. 4.5 (b) the average test accuracies of FedAAR and the baseline on various local updating epochs (i.e., $E$ = 1, 2, 4, 8, 16). As expected [93], both FedAAR and the baseline achieve higher test accuracy under smaller local updating epochs, because aggregating at lower frequencies (i.e., larger local updating epochs) easily results in the models' divergence, especially in the early training stages [87]. Notably, FedAAR with a local epoch of

16 still exhibits higher accuracy than the baseline with one local epoch, demonstrating the superiority and reliability of my approach.




(a) (b)

Fig. 4.5 Test accuracies of FedAAR and its baseline over varying (a) local dataset sizes and (b) local updating epochs.

**Analysis of Client Numbers**

More participated clients may bring more severe drifts among local gradients, which poses a great challenge to the practical application of FL [146]. To further verify the performance advantages of FedAAR compared to the baseline under scenarios with more clients, I simulate the situations with varying client numbers by redistributing the original training data based on the basic setting with five clients (see Section 4.2.4). Specifically, I separately parcel each of five horse datasets into 2, 3, 4, 5, and 6 smaller ones, each serving as the training data of a single client, thus forming five settings with total client numbers of 10, 15, 20, 25, and 30, respectively. A larger number of clients indicates less data at each client, which causes bigger variance in data distributions of clients and more severe drifts between clients. The data from the remaining horse are still used as the test data to validate my method's performance. As shown in Fig. 4.6, the test accuracy consistently decreases as client numbers increase, but FedAAR drops more slowly than the baseline method. This indicates the robustness of my approach under scenarios with more clients and more limited data numbers at each client.

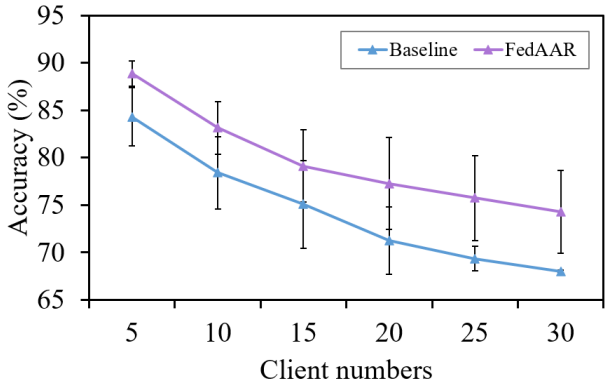



Fig. 4.6 Test accuracies of FedAAR and its baseline over various client numbers.

## 4.4 Summary

In this chapter, I develop a novel FL framework called FedAAR involving a PLU module and a GRA module to achieve automated AAR by uniting decentralized sensor data while avoiding privacy leakage. The PLU module forces all clients to learn consistent class-wise feature representations in local training, effectively reducing drift among client updates. The GRA module eliminates the conflicting components between local gradients during global aggregation, which ensures that all refined local gradients point in a positive direction to improve the agreement among clients. The experimental results reveal that my approach outperforms the state-of-the-art FL methods and achieves performance that is close to that of the centralized learning algorithm. Ablation studies further illuminate the effectiveness of the PLU and GRA modules. In addition, comparative analyses of the performance of FedAAR and the baseline at different levels of three practical conditions (i.e., local data sizes, communication frequency, and client numbers) confirm the performance advantages of my algorithm. These analyses also provide rich insights into the appropriate future applications of my method.

## 4.5 Publication Related to This Chapter

1. **Mao, A.**, Huang, E., Gan, H., & Liu, K.* (2022). FedAAR: A novel federated learning framework for animal activity recognition with wearable sensors. Animals, 12(16), 2142.
2. **Mao, A.**, Huang, E., Gan, H., & Liu, K.* (2022, August). Uniting farms: Federated learning for sensor-based animal activity recognition. In 10th European Conference on Precision Livestock Farming (*ECPLF 2022*).

# Chapter 5

# 5 Energy-efficient AAR with Low-sampling-rate Data

In previous chapters, I investigate and validate attention mechanisms and federated learning techniques in promoting the performance of animal activity recognition (AAR). In fact, there are also other factors impacting the performance of AAR, such as sampling rates of wearable sensors. Normally, achieving good monitoring performance is dependent on the availability of data with high sampling rates, which poses a huge challenge to the energy consumption and memory storage of sensing devices. To this end, existing works have often lowered the sampling rate to reduce energy costs, whereas the recognition performance of animal activity degrades rapidly when the sampling rate falls below a threshold. In this chapter, I propose a novel method, dubbed teacher-to-student information recovery (T2S-IR), aiming to improve the performance of AAR at low sampling rates. This approach effectively leverages the knowledge obtained from high-sampling-rate data, to assist in recovering the missing information in features extracted by the classification network trained on low-sampling-rate data. The workflow of the T2S-IR contains two main steps. (1) I utilize high-sampling-rate data for training teacher classification and reconstruction networks sequentially. (2) Then, I train a student classification network using low-sampling-rate data, while promoting its performance by exploiting the knowledge learned by trained teacher networks via two novel modules, namely the reconstruction-based information recovery (RIR) module and the correlation-distillation-based information recovery (CDIR) module. Specifically, the RIR module employs the pre-trained teacher reconstruction network to enforce the student classification network to learn complete and descriptive features. The CDIR module enforces the feature maps of student network to mimic internal correlations within feature maps of pre-trained teacher classification

network along temporal and sensor axes directions.

## 5.1 Introduction

Advancements in sensing technologies and smart computing techniques are driving the rapid development of automated and precision livestock management [11, 28]. Automated AAR is one sector that benefits considerably from the use of such technologies. In particular, wearable sensors have been widely used as part of an animal activity monitoring system and in conjunction with deep learning to infer individual animals' daily behaviors, such as eating, drinking, and walking [147]. Monitoring these activities in real time allows the early identification of time-sensitive health issues and timely human interventions when necessary, further improving animal health and welfare [2, 25, 59]. For instance, lameness, which is a widespread welfare problem in the field of livestock farming, can be inferred from changes in animal behaviors, such as lying down, standing, and walking [6, 28, 148–150]. Additionally, reductions in feeding and activity levels and increases in resting suggest that animals are in a state of sickness, pain, or heat stress [11, 147, 151, 152].

Despite the remarkable benefits that automated AAR provides, achieving good monitoring performance is highly dependent on the availability of data with high sampling rates. The sampling rate considerably affects the energy consumption and battery life of sensing devices due to the continuous data collection and transmission required [2, 18, 28, 30]. This can be particularly challenging in practical automated AAR systems using wearable sensors, where animals are often constantly monitored over a long period (e.g., several months). Higher sampling rates come at a cost in real-world deployments that rely on long-term operations [30]. For example, Walton et al. stated that increasing the sampling frequency from 16 to 32 Hz reduced the battery life by up to half (1.81 years vs. 3.08 years) [28]. To reduce power usage and extend battery life, a common practice is to operate wearable sensors at lower sampling rates [18, 30]. However, when the sampling rate falls below a threshold, the AAR performance degrades rapidly due to many relevant signals being missed.

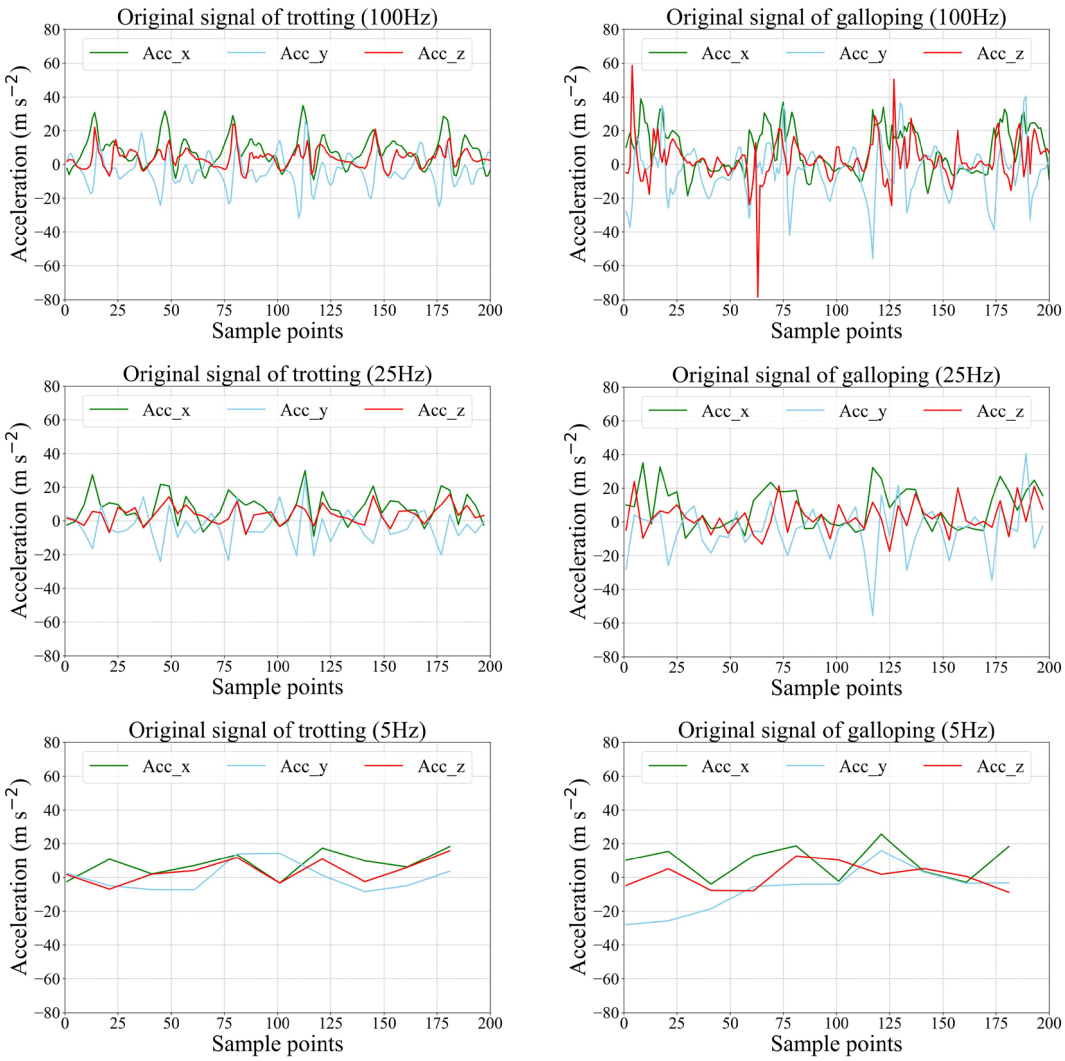
Some approaches have been proposed to address the above problem, which can be divided into two directions. One way is to find an optimal sampling rate that balances the trade-off between the classification performance and energy budget [2, 28, 54, 153]. For instance, Walton et al. evaluated the effects of the sampling rate (8, 16, and 32 Hz) on sheep activity recognition using a tri-axial accelerometer and gyroscope sensor [28]. They found that the highest

classification performance was obtained at a 32 Hz sampling rate, but sampling at 16 Hz produced comparable results, thus suggesting significant benefits of using 16 Hz for real-time monitoring. Eerdekens et al. attempted to reduce the sampling rate from 200 Hz to 25 Hz in their research for horse behavior identification, and the accuracy decreased on average by about 4.75% [54]. The other solution direction is to select an appropriate and optimal classification algorithm to overcome the decrease in accuracy due to a low sampling rate [18, 31]. For example, Benaissa et al. compared the support vector machine (SVM) with naïve Bayesian (NB) and K-nearest-neighbor (KNN) algorithms in identifying dairy cows' behaviors using accelerometer data with varying sampling rates (e.g., 0.25, 0.5, and 1 Hz) [18]. The results reflected that the SVM was less affected by a reduction in the sampling rate than the NB and KNN algorithms and had a higher accuracy at the same sampling rate. However, choosing an optimal sampling rate or algorithm does not fundamentally prevent the degradation of recognition performance when decreasing the sampling rate.

To investigate the underlying reasons why low sample rates lead to degraded performance, I visualize different sensor signals (for a 2-s period) of two horse activities (trotting and galloping) at different sampling rates of 100 Hz, 25 Hz, and 5 Hz, as illustrated in Fig. 5.1. First, we can observe that critical details in sensor signals of different behaviors are missing as sampling rate decreases, such as signal crests and troughs. Generally, they represent significant features to distinguish one behavior from another. For example, the signals of trotting and galloping at 100 Hz are dramatically different, whereas the signals of trotting and galloping at 5 Hz are highly similar. Second, the continuity of a behavior, i.e., the relationships between signals at different time points, is also missing. For instance, in a 100-Hz signal of trotting, the time interval between every two crests is around 350 milliseconds. In comparison, the relationship between every two crests in the 5-Hz signal of trotting is irregular. Therefore, it would be difficult for algorithms to capture the intrinsic movement patterns (e.g., periodicity) of a behavior. Based on the above analysis, there is an urgent need to find an effective method to recover the critical details and behavior coherence lost in low-sampling-rate signals but present in high-sampling-rate signals.

In this chapter, I propose a novel T2S-IR method to improve the performance of AAR at low sampling rates while addressing afore-mentioned issues. The workflow of the proposed T2S-IR contains two main steps. (1) I first use high-sampling-rate data to train the teacher classification and reconstruction networks sequentially. The classification network performs well in recognizing animal activities, and the reconstruction network possesses a good generalizability. (2) Then, I downsample the high-sampling-rate data to obtain low-sampling-

rate data, which are used to train a student classification network. Meanwhile, we exploit the knowledge learned by trained teacher networks to promote the student classification network's performance via two novel modules (i.e., RIR module and CDIR module). In practical scenarios, we should first record data using wearable sensors with a high sampling rate, and then utilize our T2S-IR method to obtain a trained student classification network based on the collected data. Particularly, the obtained student network can be directly applied in wearable sensor-aided AAR tasks with low sampling rates while possessing desirable performance, thereby achieving energy-efficient animal activity monitoring.




(a) Trotting (b) Galloping

Fig. 5.1 Accelerometer signals of two horse activities (trotting and galloping) at different sampling rates of 100 Hz, 25 Hz, and 5 Hz.

## 5.2 Materials and Methods

### 5.2.1 T2S-IR for AAR at Low Sampling Rates

#### Overview

The proposed T2S-IR method aims to obtain a student classification network that has promising performance at low sampling rates with the help of knowledge from high-sampling-rate data. The T2S-IR method contains two main steps: (1) In the first step, I use high-sampling-rate data to train teacher networks that include two subnetworks, i.e., the classification network and reconstruction network. (2) In the second step, I employ the trained teacher networks to guide the training of student classification network with the input of low-sampling-rate data.

#### Training teacher network using high-sampling-rate data

This section introduces the training procedure of the teacher classification network and teacher reconstruction network using high-sampling-rate data. The overall process includes two stages, as illustrated in Fig. 5.2.

***Stage 1.*** I first train a CNN-based teacher classification network, which is composed of convolutional layers and pooling layers for extracting features, and fully connected layers for final classification. Given a high-sampling-rate instance $(x^{high}, y)$ in the training dataset $\chi$ with $C$ classes of animal activities, where $x^{high}$ is an image transformed from the sensor signal (see Section 5.2.2) and $y$ is a one-hot label, I input $x^{high}$ into the classification network $T$ to produce a logit $z^T$. Then, the logit $z^T$ is used to calculate the posterior probability $p_c^T$ of each class activity using softmax function:

$$p_c^T = \frac{e^{z_c^T}}{\sum_{j\in[1,2,\dots,C]} e^{z_j^T}},\ c \in [1,2,\dots,C], \tag{5.1}$$

where $z_c^T$ is the logit value of $c$-th class. To reduce the classification error of all training samples in $\chi$, I minimize the cross-entropy loss $\mathcal{L}_{CE}^T$ between posterior probabilities and the original one-hot labels:

$$\mathcal{L}_{CE}^T = \frac{1}{|\chi|}\sum_{(x^{high},y)\in\chi}\sum_c - y_c \log\,(p_c^T), \tag{5.2}$$

where $y = [y_1, y_2, \dots, y_C]$. Under the supervision of $\mathcal{L}_{CE}^T$, the teacher classification network can capture movement patterns of different activities and perform accurate classification.

***Stage 2.*** After learning the teacher classification network, I employ its feature extractor to train a teacher reconstruction network. The reconstruction network comprises multiple transposed convolutional layers, in which the feature size variation is controlled to be symmetric with that in the teacher's feed-forward feature extractor so that it generates an image with the same dimensions as the original teacher input. Specifically, I can obtain the feature vector $e^T$ (the input to fully connected layers) after inputting $x^{high}$ into the teacher feature extractor, where the parameters of the extractor are frozen in this stage. The feature vector $e^T$ is then fed into the reconstruction network $R$, generating a reconstructed image $R(e^T)$. To attain a reconstruction network $R$ with good regeneration ability, the objective is to minimize the below reconstruction loss $\mathcal{L}_{REC}^T$ between the reconstructed image $R(e^T)$ and the input image $x^{high}$:

$$\mathcal{L}_{REC}^T = \frac{1}{|\chi|}\sum_{(x^{high},y)\epsilon\chi}\left\|x^{high} - R(e^T)\right\|_2^2, \tag{5.3}$$

where $\|\cdot\|_2^2$ denotes the mean squared error loss function.

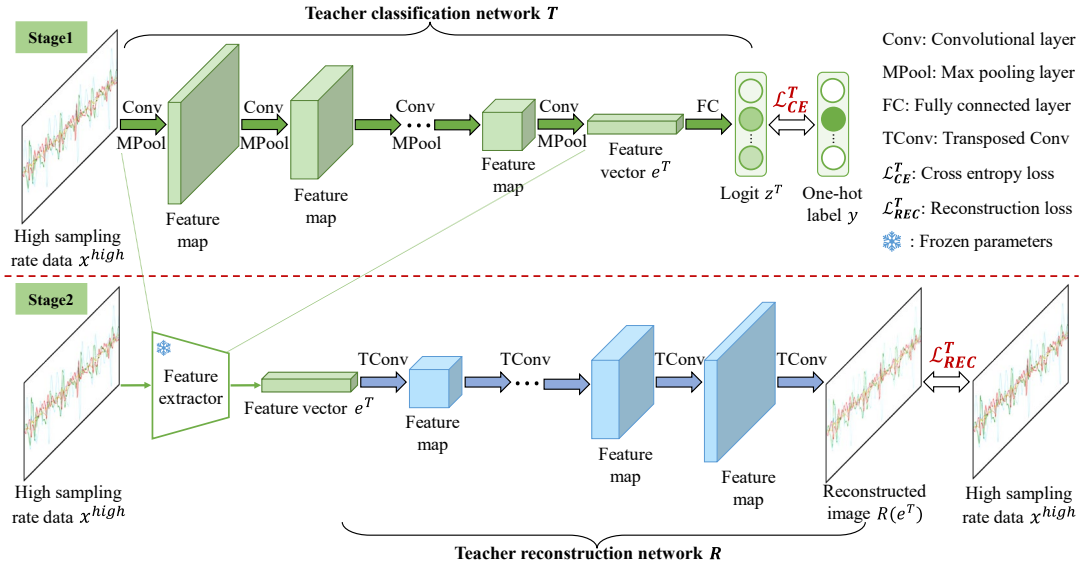



Fig. 5.2 The training workflows of the teacher classification network and teacher reconstruction network.

### Training student network using low-sampling-rate data

With guidance from the above pre-trained teacher networks, this section aims to train a favorable student classification network using low-sampling-rate data. Note that I construct a low-sampling-rate training dataset $\chi'$, where each instance $(x^{low}, y)$ is obtained by

downsampling a corresponding high-sampling-rate instance $(x^{high}, y)$ in dataset $\chi$ (here, $|\chi| = |\chi'|$) at a specific sampling frequency. The high-sampling-rate instance and low-sampling-rate instance have the same class label. In addition, the student classification network has the same architecture as the teacher classification network. Figure 5.3 presents the overall training architecture of the student classification network.

As shown in Fig. 5.3 (a), I feed $x^{low}$ into the student classification network $S$ to yield a logit $z^S$, which then goes through a softmax function to produce the posterior probability $p_c^S$ of each class. The optimization criterion for the classification network is to maximize the prediction probabilities of corresponding class activities, i.e., minimizing the negative log-likelihood loss function $\mathcal{L}_{CE}^S$ between the posterior probability $p^S$ and the one-hot label $y$ of all training samples in $\chi'$:

$$\mathcal{L}_{CE}^S = \frac{1}{|\chi'|}\sum_{((x^{low},y)\in\chi'}\sum_c -y_c \log\left(p_c^S\right), \tag{5.4}$$

$$\text{with } p_c^S = \frac{e^{z_c^S}}{\sum_{j\in[1,2,\dots,C]} e^{z_j^S}},\ c \in [1,2,\dots,C]. \tag{5.5}$$

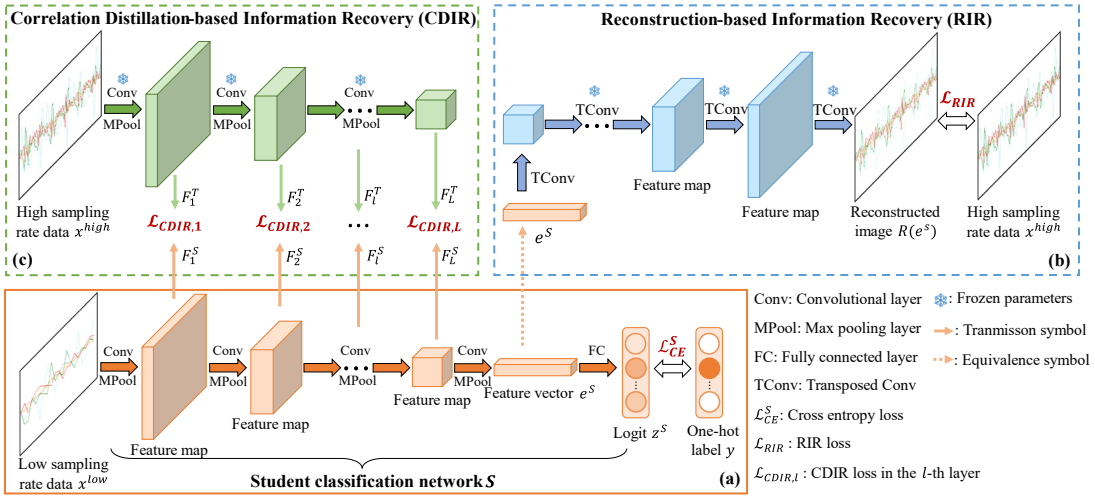



Fig. 5.3 The overall training architecture of the student classification network.

However, solely relying on low-sampling-rate data for training classification networks often leads to poor performance, as the network cannot extract comprehensive and sufficiently discriminative features from the low-sampling-rate data due to a loss of critical information. Inspired by the statement that adding an image reconstruction branch based on the latent representations facilitates the network to generate more descriptive features than a pure

classification network [154], I design an RIR module (Fig. 5.3 (b)) to facilitate the recovery of missing information in features. The RIR module exploits the pre-trained teacher reconstruction network $R$ to encourage the student network $S$ to learn more complete and discriminative features. Concretely, I first get the feature vector $e^S$ after feeding $x^{low}$ into the student classification network $S$ (Fig. 5.3 (a)). Then, I input $e^S$ into the pre-trained teacher reconstruction network $R$ to generate a reconstructed image $R(e^S)$ (Fig. 5.3 (b)), which is impelled to be close to the image $x^{high}$ through an RIR loss $\mathcal{L}_{RIR}$:

$$\mathcal{L}_{RIR} = \frac{1}{|\chi|}\sum_{x^{high}\epsilon\chi}\left\|x^{high} - R(e^S)\right\|_2^2. \tag{5.6}$$

With the constraint of loss $\mathcal{L}_{RIR}$, the student classification network $S$ can exploit existing information within low-sampling-rate data to complete missing details in extracted features since there are inherent correlations within a behavior sample along temporal points and sensor axes [130, 155, 156]. Note that the parameters of the pre-trained reconstruction network $R$ remain constant during the training of student network $S$.

Considering the significance of inherent correlation within a behavior sample for recovering missing information, I further devise a CDIR module (Fig. 5.3 (c)) that can enforce the features of student network to imitate correlations hidden in features of the pre-trained teacher network along time and sensor axes directions. Specifically, I collect feature maps $\{F_l^S\}_{l=1}^L$ of different layers when the image $x^{low}$ is input into student classification network $S$, where $L$ represents the number of layers. Meanwhile, I also feed the corresponding high-sampling-rate image $x^{high}$ into the trained teacher classification network $T$ to yield its feature maps $\{F_l^T\}_{l=1}^L$. Note that the parameters of network $T$ are kept fixed during training. Afterwards, I use pairwise (teacher–student) feature maps of each layer to compute the corresponding layer's CDIR loss $\mathcal{L}_{CDIR,l}$. Figure 5.4 exhibits the calculation procedure of the CDIR loss $\mathcal{L}_{CDIR,l}$ in the $l$-th layer.

In Fig. 5.4, feature maps $F_l^T$ and $F_l^S$ have the same size of $c \times a \times t$, where $c$ is the channel number, $a$ and $t$ refer implicitly to the axis and time dimensions, respectively. Both feature maps $F_l^T$ and $F_l^S$ are input into a temporal correlation distillation (TCD) branch and an inter-axis correlation distillation (ACD) branch to calculate the CDIR loss $\mathcal{L}_{CDIR,l}$. In the TCD branch, I first reshape $F_l^T$ and $F_l^S$ into two two-dimensional feature maps with a size of $t \times ac$, where each vector with length $ac$ is regarded as the feature along the time dimension. I then conduct matrix multiplication on each two-dimensional feature map and its

corresponding transposition, generating temporal correlation matrices $\hat{M}_l^T$ and $\hat{M}_l^S$ of the teacher and student, respectively. Afterwards, I perform the L2 normalization on each horizontal vectors of these two matrices. To enforce the student feature map to mimic temporal correlations within the teacher feature map, I define the following TCD loss $\hat{\mathcal{L}}_l$, which penalizes the negative cosine similarity between the normalized vectors $u$, $v$ of $\hat{M}_l^T$, $\hat{M}_l^S$ through the following formulation:

$$\hat{\mathcal{L}}_l = -\frac{1}{t}\sum_{u\in\hat{M}_l^T, v\in\hat{M}_l^S}\frac{uv}{|u||v|}, \tag{5.7}$$

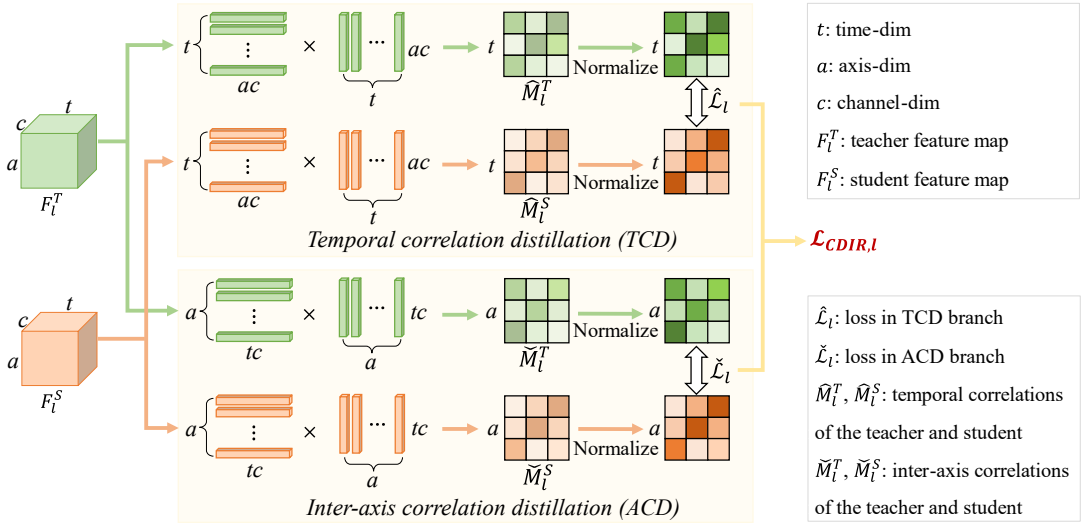



Fig. 5.4 The computation process of correlation-distillation-based information recovery loss in $\boldsymbol{l}$-th layer.

Furthermore, I transfer the inter-axis correlations within feature maps from teacher to student in the ACD branch. Briefly, $F_l^T$ and $F_l^S$ are first reshaped into two two-dimensional feature maps with a size of $a \times tc$, which are used to generate two inter-axis correlation matrices $\check{M}_l^T$ and $\check{M}_l^S$ through matrix multiplication. Then, I use the normalized vectors within $\check{M}_l^T$ and $\check{M}_l^S$ to calculate the ACD loss $\check{\mathcal{L}}_l$ by conducting a similar operation with Eq. ((5.7). Finally, the loss $\mathcal{L}_{CDIR,l}$ in the $l$-th layer is formulated as the loss combination of the TCD branch and ACD branch:

$$\mathcal{L}_{CDIR,l} = \frac{1}{2} * (\hat{\mathcal{L}}_l + \check{\mathcal{L}}_l). \tag{5.8}$$

Since the temporal and inter-axis correlations are included in intermediate features across various network layers, I penalize the total CDIR loss $\mathcal{L}_{CDIR}$ over all $L$ layers to achieve the

correlation distillation of feature maps from teacher to student:

$$\mathcal{L}_{CDIR} = \frac{1}{L}\sum_{l=1}^{L}\mathcal{L}_{CDIR,l}. \tag{5.9}$$

With the constraint of loss $\mathcal{L}_{CDIR}$, the CDIR module compels the student feature maps to imitate the temporal and inter-axis correlations hidden in teacher feature maps across hierarchical layers.

In the end, based on the obtained classification loss $\mathcal{L}_{CE}^{S}$, RIR loss $\mathcal{L}_{RIR}$, and CDIR loss $\mathcal{L}_{CDIR}$, the student classification network $S$ is optimized using the below joint loss function:

$$\mathcal{L}_{total} = \mathcal{L}_{CE}^{S} + \lambda_1 * \mathcal{L}_{RIR} + \lambda_2 * \mathcal{L}_{CDIR}, \tag{5.10}$$

where $\lambda_1$ and $\lambda_2$ are weight factors controlling the effects of the $\mathcal{L}_{RIR}$ and $\mathcal{L}_{CDIR}$. Through the loss in Eq. (5.10), the proposed T2S-IR method adequately leverages the knowledge obtained from high-sampling-rate data to assist in recovering the missing information in low-sampling-rate data, consequently boosting the student classification network's performance. In practical scenarios, the enhanced student network can be directly applied to infer different animal activities using low-sampling-rate data, thus achieving energy-efficient animal activity recognition.

### 5.2.2 Datasets and Data Preprocessing

**Datasets**

The proposed method is tested using two public datasets, i.e., horse dataset [72] and goat dataset [71]. The details are described in Section 2.5. The horse dataset comprises 87,621 2-s labeled samples acquired from six horses using tri-axial accelerometers and tri-axial gyroscopes, in which the sampling rate is set at 100 Hz. The motion data recorded by a tri-axial accelerometer and a tri-axial gyroscope form a tensor of 1 × 6 × 200 for each sample. The goat dataset consists of 42,943 2-s data samples collected from five goats. Each goat's collar is equipped with six tri-axial accelerometers and tri-axial gyroscopes fixed in different orientations, where the sampling rate is set at 100 Hz. The motion data recorded by all six tri-axial accelerometers and tri-axial gyroscopes are used, forming a tensor of 1 × 36 × 200 for each sample.

**Data Preprocessing**

Figure 5.5 displays the preprocessing procedure of sensor signals before they are involved in the training of student classification network. Herein, I take an example of a 2-s signal sample

with six sensor axes. The signal having a high sampling rate is first downsampled to obtain a low-sampling-rate signal sample. Then, the two samples are normalized using the same procedure as in Eq. (3.10). Since the duration of each sample is fixed at a consistent value (i.e., two seconds), the number of recordings along time direction is different between high- and low-sampling-rate samples. In other words, data are missing in time intervals in the low-sampling-rate sample compared to the high-sampling-rate sample. The missing data directly lead to the extracted features losing information critical to the network in making correct predictions. In efficiently recovering the missing information, I upsample the low-sampling-rate sample through nearest-neighbor interpolation [157] to keep its shape the same as high-sampling-rate sample. Afterwards, the high-sampling-rate sample and upsampled low-sampling-rate sample are imported into the teacher network and student network, respectively.

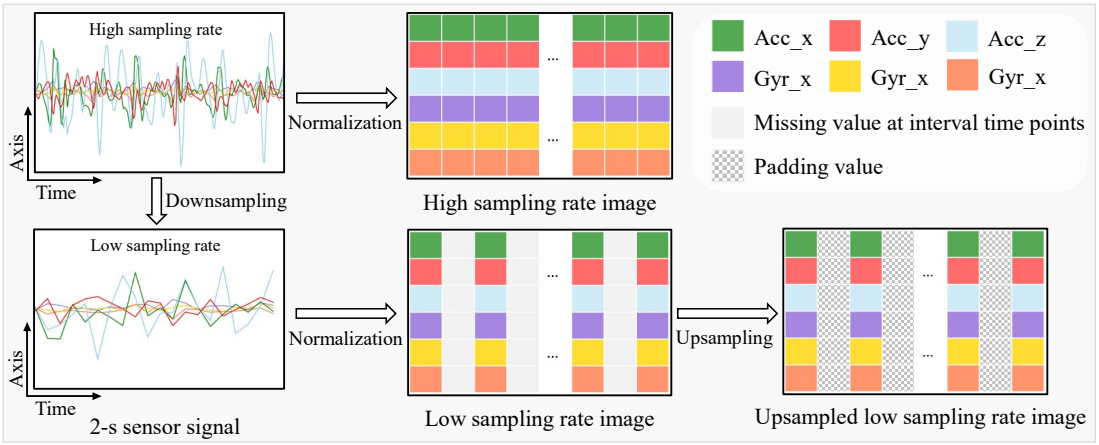



Fig. 5.5 Illustration of data preprocessing.

## 5.2.3 Design of Experiments

### Evaluation Metrics

The comprehensive performance of the activity classification model is indicated by four evaluation metrics, including precision, recall, F1-score, and accuracy (see Eq. (3.11) to Eq. (3.14)). Each indicator value is multiplied by 100 as the result to reflect the difference in indicator values more clearly.

### Implementation Details

This chapter uses a CNN-based classification network called CMI-Net, which has been

validated in Chapter 3 to improve the classification performance of horse activities, as the architecture of teacher and student classification networks. With reference to the studies of Eerdekens et al. [2, 54], I downsample the original data sampled at 100 Hz to sampling frequencies of 50, 25, 12.5, 10, 5, and 2 Hz. The performances of CMI-Net trained at these different sampling rates are used as reference values (i.e., baseline results). The sampling rate at which the CMI-Net performes best is selected as the high sampling rate, and the corresponding data are used to train teacher networks. With the guidance of trained teacher networks, I train a student classification network using data with lower sampling rates. To validate the proposed T2S-IR method in improving the student model's performance, I compare it against the baseline method (i.e., training the student classification network only using low-sampling-rate data) and various existing knowledge distillation (KD) methods, i.e., basic KD [29], AT [106], RKD [105], and ICKD [103]. In addition, the leave-one-out cross-validation method (see Section 3.2.4) is applied to verify the generalizability of the classification model.

During training, I add an L2 regularization (with a weight decay of 0.1 for the horse dataset and 0.005 for the goat dataset) to the loss function to avoid overfitting. An Adam optimizer with an initial learning rate of $1 \times 10^{-4}$ is used, and the learning rate is reduced by a factor of 10 every 20 epochs. The total running epoch number and batch size are set at 100 and 256, respectively. As in previous studies [27, 105], I adopt a grid search of hyperparameters (i.e., $\lambda_1$ and $\lambda_2$ in Eq. (5.10)) and set exact values for the horse dataset and goat dataset. All experiments are executed using the PyTorch framework on a single NVIDIA GeForce RTX 3090 graphics processing unit. The source code is available at https://github.com/Max-1234-hub/T2S-IR.

## 5.3 Results and Discussion

Overall, the experimental results demonstrate that the proposed T2S-IR method remarkably boosts the model trained using data having low sampling rates and outperforms existing KD methods. Ablation studies are carried out to evaluate the effectiveness of the RIR and CDIR modules in terms of classification capability. Additionally, the recognition analysis is analyzed to probe the predictive performance advantages of the T2S-IR method for each activity compared with the baseline method. This section ends with the proposal of future works.

### 5.3.1 Baseline Performance

Figure 5.6 presents the performance of the baseline model (CMI-Net) trained on two public datasets with different sampling rates, i.e., 100, 50, 25, 12.5, 10, 5, and 2 Hz. The results for the horse dataset (Fig. 5.6 (a)) show that as the sampling rate gradually drops from 100 Hz, the performance increases slightly until it peaks at 25 Hz; that is, a precision, recall, and F1-score of 82.92%, 85.12%, and 83.50%, respectively. This result is consistent with the results of several previous studies on horse behavior recognition using wearable sensors [2, 54, 55], which found that reducing the sampling rate within a certain range leads to higher performance. This phenomenon might be explained by the fact that too-high sampling rates raise the level of irrelevant noise in the data, thereby misleading the final behavior classification. Subsequently, the performance, as expected, starts to degrade constantly as the sampling rate continues to drop, because less information is included in the data obtained at a lower sampling rate. Additionally, the results for the goat dataset (Fig. 5.6 (b)) show that the model performance peaks for data obtained at a sampling rate of 100 Hz; that is, a precision, recall, F1-score, and accuracy of 75.50%, 90.48%, 80.69%, and 91.85%, respectively. The performance fluctuates mildly as the sampling rate decreases from 100 to 12.5 Hz and then declines continuously as the sampling rate fell below 12.5 Hz. This reflects that the information contained in the data tends to be saturated once the sampling rate reaches 12.5 Hz, and there would be a trade-off between performance and energy consumption in the practical selection of the sampling rate [28, 31, 153]. According to the obtained performance and the selection criteria described in Section 5.2.3, I choose 25-Hz data and 100-Hz data to train teacher networks on the horse and goat datasets, respectively.

### 5.3.2 Comparisons with Existing Methods

To evaluate the effectiveness of the proposed T2S-IR method, I carry out comparison experiments with the baseline method and existing KD methods (i.e., basic KD, AT, RKD, and ICKD) based on the horse and goat datasets.

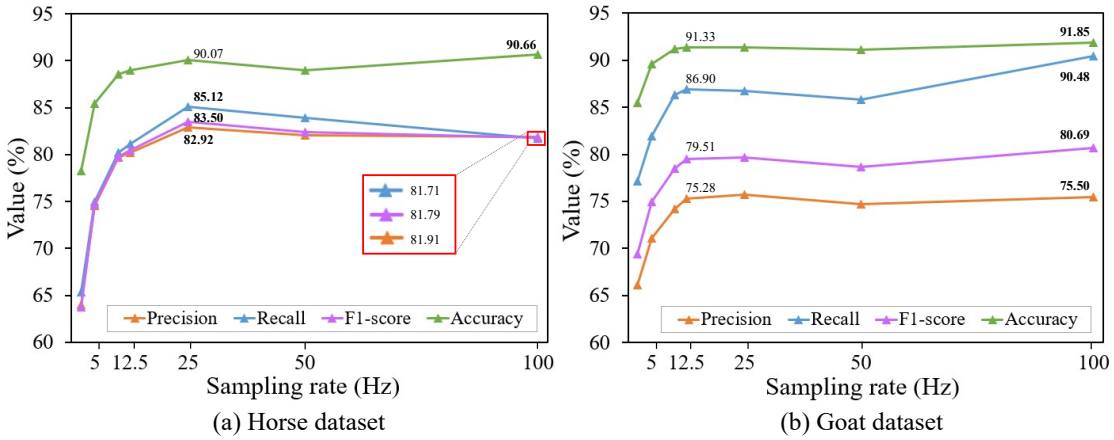



Fig. 5.6 Classification performance of the baseline method for the horse dataset (a) and goat dataset (b) at different sampling rates (i.e., 100, 50, 25, 12.5, 10, 5, and 2 Hz).

**Comparative results obtained on the horse dataset**

Table 5.1 presents the experimental results obtained on the horse dataset when using the data with a sampling rate of 12.5 Hz ($\lambda_1$= 0.1 and $\lambda_2$=1.1) and 5 Hz ($\lambda_1$=0.1 and $\lambda_2$=1.4) as inputs of student classification network. The results reveal that all KD approaches outperform the baseline in terms of all evaluation metrics, with the baseline having a precision, recall, F1-score, and accuracy of 80.18%, 81.12%, 80.48%, and 89.01%, respectively, for the 12.5-Hz data and 74.53%, 74.94%, 74.67%, and 85.44%, respectively, for the 5-Hz data. This demonstrates the promising capabilities of the knowledge transfer mechanism in alleviating the performance degradation resulting from the use of data having low sampling rates. The proposed method outperforms the existing KD methods with a precision, recall, F1-score, and accuracy of 83.51%, 84.70%, 83.93%, and 91.20%, respectively, for the 12.5-Hz data and 76.77%, 76.36%, 76.45%, and 87.05%, respectively, for the 5-Hz data. This can be ascribed to the ability of my architecture to effectively recover the missing information in student features by sufficiently exploiting the knowledge obtained from teacher data. Moreover, my method even outperforms the teacher classification network with relative percentage-point gains of 0.59%, 0.43%, and 1.13% in the precision, F1-score, and accuracy, respectively, for the 12.5-Hz data. This finding is consistent with the results of several previous works [101, 105]. A possible explanation is that I transfer the knowledge from the extracted features using the CDIR module, and the noise in high-sampling-rate data has already been removed during the feature extraction using deep learning. This impressive result is also in line with practical requirements that aim to relieve the burden of energy costs while maintaining desirable performance.

Table 5.1 Comparison of the teacher-to-student information recovery (T2S-IR) method against the baseline and existing knowledge distillation methods on the horse dataset at low sampling rates (i.e., 12.5 and 5 Hz).

| Method | 12.5 Hz | | | | 5 Hz | | | |
|---|---|---|---|---|---|---|---|---|
| | Prec* (%) | Rec (%) | F1 (%) | Acc (%) | Prec (%) | Rec (%) | F1 (%) | Acc (%) |
| Teacher# | 82.92 | 85.12 | 83.50 | 90.07 | 82.92 | 85.12 | 83.50 | 90.07 |
| Baseline | 80.18 | 81.12 | 80.48 | 89.01 | 74.53 | 74.94 | 74.67 | 85.44 |
| KD [29] | 81.54 | 82.12 | 81.68 | 89.99 | 76.01 | 75.05 | 75.25 | **87.20** |
| AT [106] | 82.45 | 82.83 | 82.61 | 91.17 | 75.07 | 75.76 | 75.17 | 86.47 |
| RKD [105] | 82.14 | 82.56 | 82.28 | 90.74 | 75.69 | 76.34 | 75.87 | 86.78 |
| ICKD [103] | 81.36 | 82.43 | 81.77 | 90.15 | 75.15 | 74.59 | 74.39 | 86.02 |
| Our T2S-IR | **83.51** | **84.70** | **83.93** | **91.20** | **76.77** | **76.36** | **76.45** | 87.05 |

# For comparison, I also report the performance of the teacher classification network trained on data having a sampling rate of 25 Hz. The values refer to the results in Fig. 5.6 (a);

* Prec: precision; Rec: recall; F1: F1-score; Acc: accuracy.

**Comparative results obtained on the goat dataset**

Comparison experiments are also conducted on the goat dataset, where the data with a sampling rate of 5 Hz ($\lambda_1$=1 and $\lambda_2$=1.5) and 2 Hz ($\lambda_1$=0.1 and $\lambda_2$=2.5) are used to train the student classifiaction network. Table 5.2 shows that all KD methods again outperform the baseline method, which has a precision, recall, F1-score, and accuracy of 71.06%, 81.90, 74,96%, and 89.64%, respectively, for the 5-Hz data and 66.07%, 77.12%, 69.39%, and 85.51%, respectively, for the 2-Hz data. This further confirms the favorable abilities of KD in improving student models without adding network complexity. The proposed T2S-IR method outperforms the baseline model with relative percentage-point gains in precision, recall, F1-score, and accuracy of 7.6%, 4.44%, 6.9%, and 0.79%, respectively, for the 5-Hz data and 4.29%, 2.35%, 3.99%, and 2.28%, respectively, for the 2-Hz data. Although basic KD has higher precision and accuracy than my method for the 2-Hz data, it came at the cost of extremely low recall values, which reflects that the samples are more likely to be misclassified when using basic KD. Additionally, my approach has desirable performance boosts over the teacher model in the precision and F1-score of 3.16% and 1.17%, respectively, for the 5-Hz data. This result reinforces the finding that the proposed method is suited to practical scenarios with limited energy and memory sources.

Table 5.2 Comparison of the T2S-IR method against the baseline and existing knowledge distillation methods on the goat dataset at low sampling rates (i.e., 5 and 2 Hz).

| Method | 5 Hz | | | | 2 Hz | | | |
|---|---|---|---|---|---|---|---|---|
| | Prec (%) | Rec (%) | F1 (%) | Acc (%) | Prec (%) | Rec (%) | F1 (%) | Acc (%) |
| Teacher[#] | 75.50 | 90.48 | 80.69 | 91.85 | 75.50 | 90.48 | 80.69 | 91.85 |
| Baseline | 71.06 | 81.90 | 74.96 | 89.64 | 66.07 | 77.12 | 69.39 | 85.51 |
| KD [29] | 77.36 | 85.49 | 80.65 | 89.78 | **72.36** | 72.82 | 71.70 | **88.64** |
| AT [106] | 75.64 | 83.04 | 78.48 | 89.20 | 69.12 | 78.91 | 71.44 | 87.46 |
| RKD [105] | 75.81 | 85.87 | 79.45 | 89.90 | 70.56 | 78.49 | 72.54 | 87.76 |
| ICKD [103] | 76.55 | 83.98 | 79.52 | 89.57 | 68.22 | 78.09 | 71.05 | 87.87 |
| Our T2S-IR | **78.66** | **86.34** | **81.86** | **90.43** | 70.36 | **79.47** | **73.38** | 87.79 |

[#] For comparison, I also report the performance of the teacher classification network trained on data having a sampling rate of 100 Hz. The values refer to the results in Fig. 5.6 (b);

### 5.3.3 Ablation Studies

**Evaluation of RIR and CDIR modules**

To deeply probe the effects of the two main modules, i.e., the RIR and CDIR modules, I conduct experiments on the two public datasets for the four following configurations. First, the Baseline configuration is the basic classification model without any knowledge transfer operation (i.e., both $\lambda_1$ and $\lambda_2$ in Eq. (5.10) are set to zero). Second, the Baseline + RIR configuration includes the RIR module in the original classification task to supervise the student training (i.e., $\lambda_1$ is set to zero). Third, the Baseline + CDIR configuration includes the constraints from the CDIR module in the training of the student classification model (i.e., $\lambda_2$ is set to zero). Fourth, the T2S-IR (Baseline + RIR + CDIR) configuration implements the proposed T2S-IR method involving the RIR and CDIR modules simultaneously. The experimental results obtained on the horse and goat datasets are given in Tables 5.3 and 5.4, respectively. It is seen that the two modules individually contribute to the tremendous performance advantage over the baseline, which demonstrates that each of the RIR and CDIR modules plays a critical role in the AAR tasks using low-sampling-rate data. In particular, the performance gains are greater for data having a higher sampling rate than for data having a lower sampling rate; that is, 12.5 Hz vs. 5 Hz on the horse dataset (Table 5.3) and 5 Hz vs. 2 Hz on the goat dataset (Table 5.4). The reason may be that more information is contained in the high-sampling-rate data, which is more beneficial for recovering missing information. In other words, while relying on the knowledge

from teacher data to recover the missing information in student features, it is also necessary to ensure that the student data have enough information of its own.

Table 5.3 Evaluation results of the reconstruction-based information recovery (RIR) module and correlation-distillation-based information recovery (CDIR) module in terms of the classification performance on the horse dataset at a low sampling rate.

| Method | 12.5 Hz | | | | 5 Hz | | | |
|---|---|---|---|---|---|---|---|---|
| | Prec (%) | Rec (%) | F1 (%) | Acc (%) | Prec (%) | Rec (%) | F1 (%) | Acc (%) |
| Baseline | 80.18 | 81.12 | 80.48 | 89.01 | 74.53 | 74.94 | 74.67 | 85.44 |
| Baseline + RIR | 82.51 | 83.30 | 82.73 | 90.56 | 76.02 | 75.08 | 75.22 | **87.12** |
| Baseline + CDIR | 83.16 | 84.34 | 83.53 | 90.90 | 76.45 | 74.99 | 75.57 | 86.45 |
| T2S-IR | **83.51** | **84.70** | **83.93** | **91.20** | **76.77** | **76.36** | **76.45** | 87.05 |

**Visualization of correlation matrices**

To qualitatively verify the effectiveness of the CDIR module, I visualize the temporal and inter-axis correlation matrices obtained from teacher feature maps across different layers, as shown in Fig. 5.7 and Fig. 5.8. Figure 5.7 shows that within a specific duration, the strength of temporal correlations varies periodically along the time dimension; that is, the temporal correlations become more apparent as the layer number increases. This observation further confirms the existence of periodicity within the movement patterns of animals. Similarly, there are correlations between axis-dimension features across different layers in Fig. 5.8, further supporting the introduction of the CDIR module.

Table 5.4 Evaluation results of the RIR module and CDIR module in terms of the classification performance on the goat dataset at a low sampling rate.

| Method | 5 Hz | | | | 2 Hz | | | |
|---|---|---|---|---|---|---|---|---|
| | Prec (% ) | Rec (%) | F1 (%) | Acc (%) | Prec (% ) | Rec (%) | F1 (%) | Acc (%) |
| Baseline | 71.06 | 81.90 | 74.96 | 89.64 | 66.07 | 77.12 | 69.39 | 85.51 |
| Baseline + RIR | 77.95 | 85.43 | 81.04 | 90.06 | 69.07 | 79.00 | 72.10 | **87.80** |
| Baseline + CDIR | 77.19 | 86.04 | 80.67 | 89.78 | 69.17 | 77.23 | 71.77 | 87.62 |
| T2S-IR | **78.66** | **86.34** | **81.86** | **90.43** | **70.36** | **79.47** | **73.38** | 87.79 |

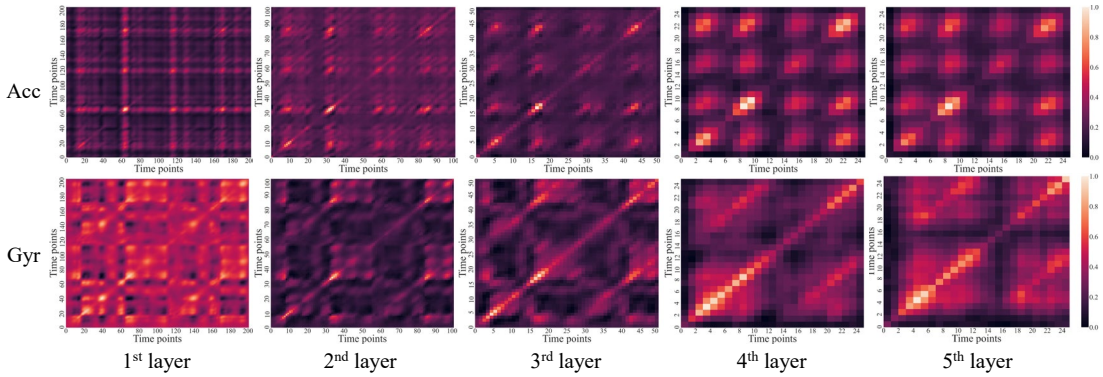



Fig. 5.7 Visualization of the temporal correlation matrices across different layers under the teacher network.

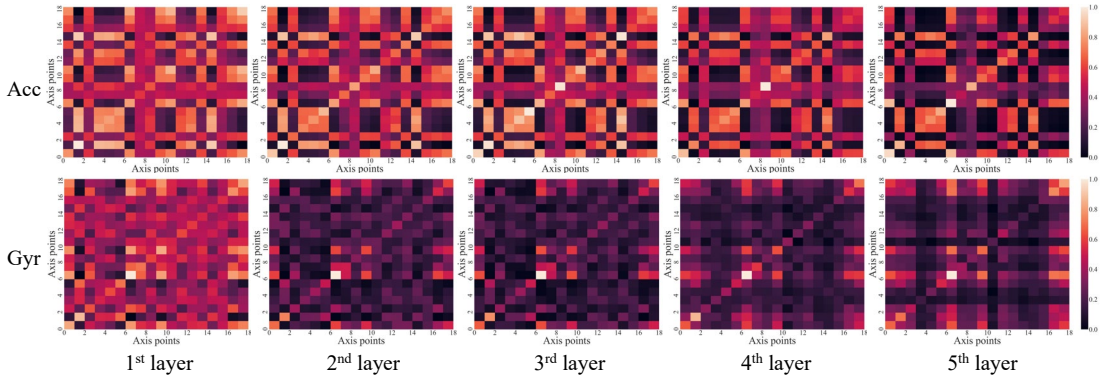



Fig. 5.8 Visualization of the inter-axis correlation matrices across different layers under the teacher network.

### 5.3.4 Classification Performance Analysis

Among the evaluation metrics, recall represents the percentage of correctly classified samples [25]. To further provide insights into the promising capability of the proposed T2S-IR method in improving classification performance, I display the recall confusion matrices of the baseline method and my method, as in Figs. 5.9 and 5.10. Compared with the baseline (Figs. 5.9 (a) and 5.10 (a)), the proposed T2S-IR method (Figs. 5.9 (b) and 5.10 (b)) has improved recall values for almost all activities to varying degrees. Especially on the data with a slightly higher sampling rate, my method has considerably higher increments in recall values for minority classes; that is, 11.93% for naturally walking horses at a sampling frequency of 12.5 Hz (Fig. 5.9) and 14.81% for trotting goats at a sampling frequency of 5 Hz (Fig. 5.10). This demonstrates that my method effectively addresses low-performance recognition due to

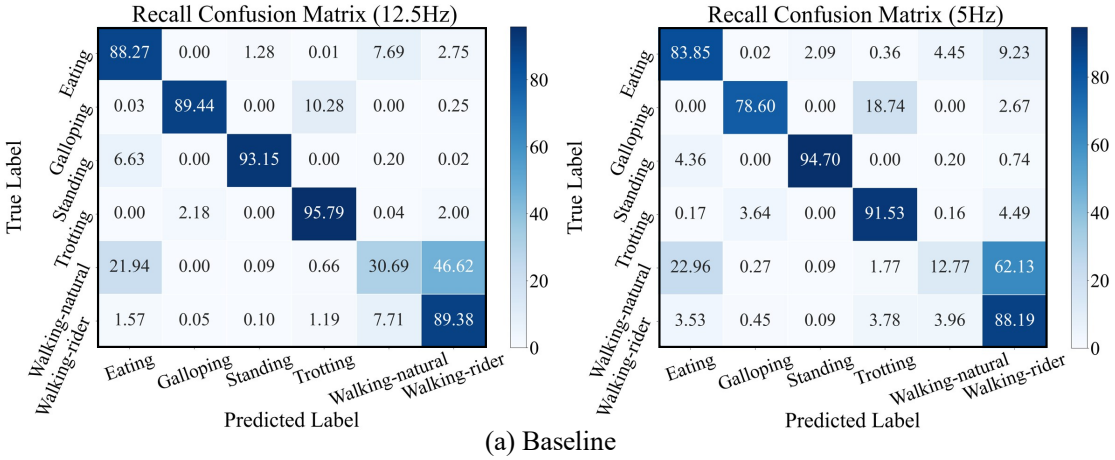
insufficient sample numbers and has promising potential for application in practical scenarios with data imbalance problems. Additionally, the recall confusion matrix of my method on the horse dataset (Fig. 5.9 (b)) shows that except for walking-natural, all activities have recall values of at least ~90% for the 12.5-Hz data and at least ~80% for the 5-Hz data. The recall confusion matrix of my method trained on the goat dataset (Fig. 5.10 (b)) shows that except for trotting, all activities have recall values of at least ~80% for the 5- and 2-Hz data. These results again demonstrate the advantage of applying my method in scenarios with low sampling rates.

In practical scenarios, the trained student classification network can be directly applied to identify animal activities based on low-sampling-rate data. Fig. 5.11 randomly visualizes the test result of the trained student model on continuous sensor data (lasting for around 3.32 hours) collected from a single goat. Notably, the data of the selected goat are unseen during the model training. When the sampling rate was set to 5 Hz, the model achieved favorable classification capability for goat's activities with recall, F1-score, and accuracy of 78.02%, 91.3%, 82.48%, and 95.02%, respectively (Fig. 5.11 (a)). It can be observed that the activities standing, grazing, and walking were sometimes confused with each other, which can be explained by the fact that goat is often standing while chewing, or walking while eating [71]. This phenomenon became more evident in the test result based on 2-Hz data, where the model can still obtain high identification accuracy of 91.29% (Fig. 5.11 (b)). We can also find that walking was more prone to being misclassified as trotting when decreasing the sampling rate from 5 Hz to 2 Hz.

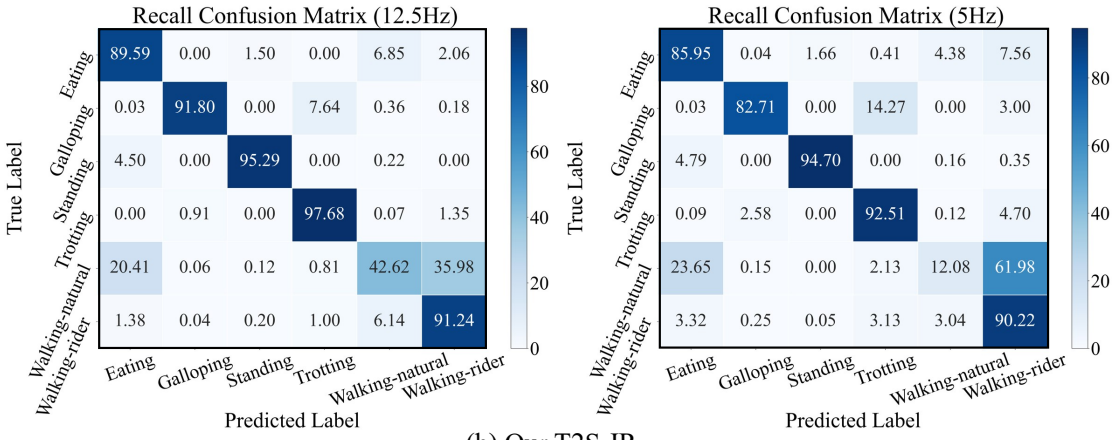
## 5.4 Summary

This chapter develops a novel method named T2S-IR to enhance the performance of AAR using data obtained at low sampling rates. The T2S-IR makes use of the knowledge from high-sampling-rate data, to help recover the missing information in features extracted by the classification network trained on low-sampling-rate data. Specifically, I utilize high-sampling-rate data to train teacher classification and reconstruction networks, which are then employed to guide the training of student classification network based on low-sampling-rate data. During the training of the student network, two newly designed modules, i.e., RIR module and CDIR module, can help recover missing information using pre-trained teacher networks, further extracting more complete and discriminative features. The experimental results reveal that the approach remarkably boosts the performance of a model trained on data having low sampling rates while outperforming existing KD algorithms. Ablation studies are also performed to show the effectiveness of each module in the proposed approach. Additionally, classification

performance is analyzed for each activity to further confirm the advantages of my method's predictive performance when using low-sampling-rate data. My method thus has great application potential for energy-efficient wearable sensor-aided AAR systems, especially in scenarios with limited energy sources.




(a) Baseline




(b) Our T2S-IR

Fig. 5.9 Recall (unit: %) confusion matrix of the baseline method (a) and the proposed teacher-to-student information recovery (T2S-IR) method (b) on the horse dataset with low sampling rates.

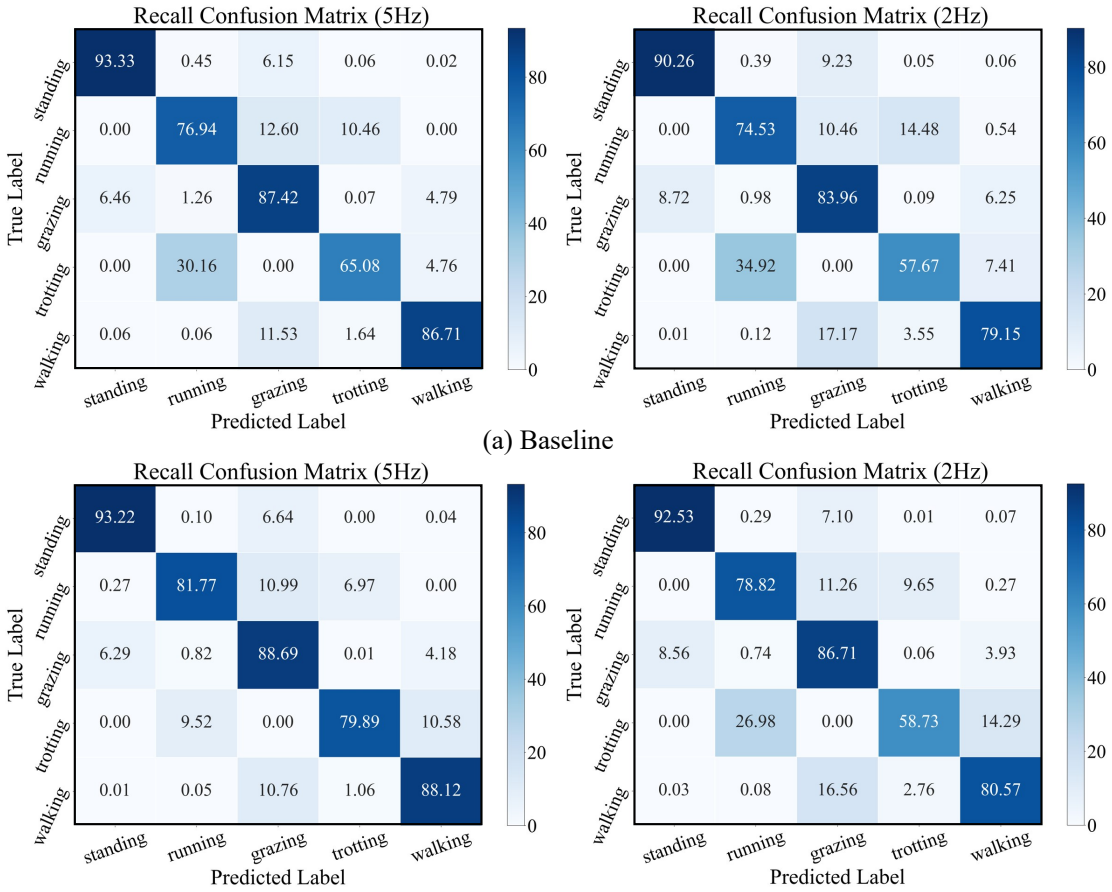
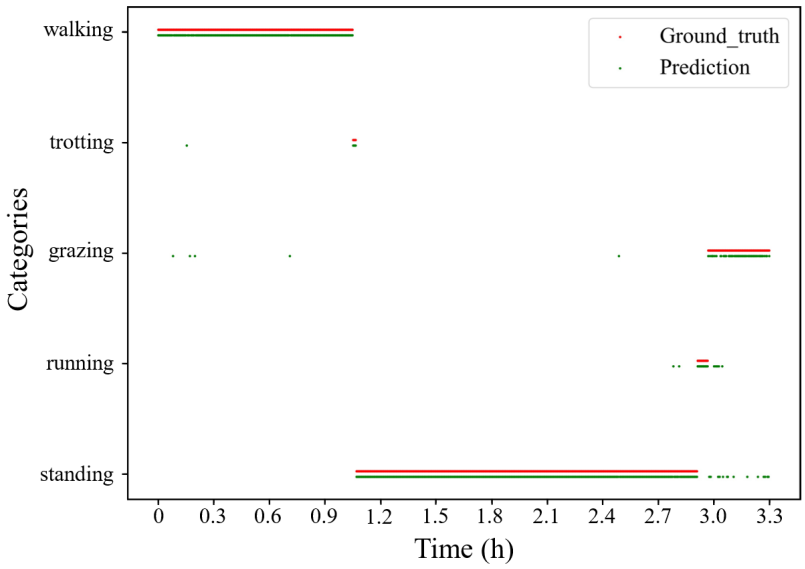
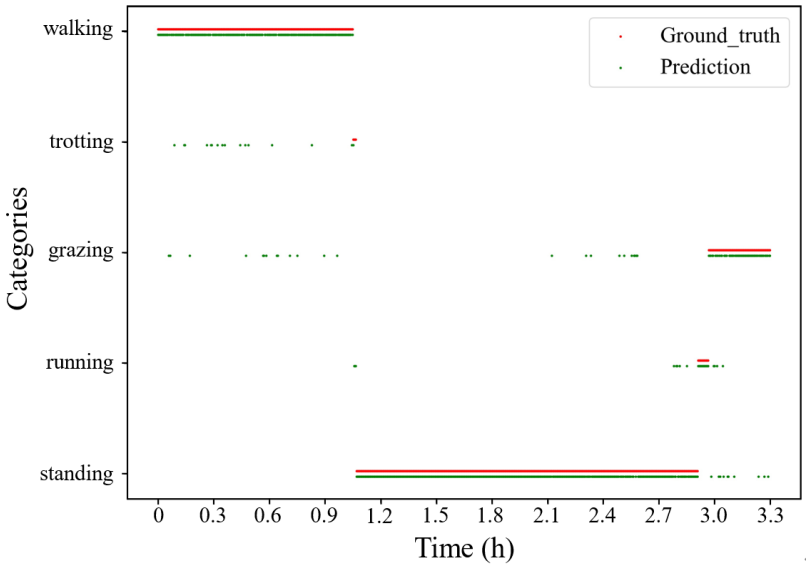



Fig. 5.10 Recall (unit: %) confusion matrix of the baseline method (a) and the proposed T2S-IR method (b) on the goat dataset with low sampling rates.




(a) Test on 5-Hz data




(b) Test on 2-Hz data

Fig. 5.11 Test on continuous sensor data (over a period) collected from a single goat under a sampling rate of 5 Hz (a) and 2 Hz (b), respectively.

## 5.5 Publication Related to This Chapter

1. **Mao, A.**, Zhu, M., Huang, E., Yao, X., & Liu, K.* (2023). A Teacher-to-Student Information Recovery Method Toward Energy-Efficient Animal Activity Recognition at Low Sampling Rates. Computers and Electronics in Agriculture, Accepted.
2. **Mao, A.**, Huang, E., Zhu, M., & Liu, K.* (2023, May). Robust Animal Activity Recognition Using Wearable Sensors: A Correlation Distillation-based Information Recovery Method toward Data Having Low Sampling Rates. In 2nd U.S. Precision Livestock Farming Conference (*USPLF 2023*).

# Chapter 6

# 6 Conclusion

This chapter summarizes my proposed approaches elaborated in the above-mentioned chapters. In addition, I also discuss some potential improvements for future research.

## 6.1 Summary

Over the past decades, advancements in deep learning techniques and wearable sensors have driven the rapid development of automated and precise animal activity recognition (AAR) systems. In this thesis, I present a comprehensive review of recent research on AAR based on wearable sensors and deep learning algorithms. Before implementing AAR systems based on wearable sensors and deep learning in commercial animal farming, some relative technical challenges must be addressed, including multi-modal fusion, class imbalance, data privacy, and energy efficiency. To tackle these challenges, I have presented a series of strategies, including the cross-modality interaction network (CMI-Net) combined with class-balanced (CB) focal loss to achieve multi-model fusion and mitigate the class imbalance problem, the FedAAR to perform automated AAR by uniting decentralized data while preserving data privacy across farms, and the teacher-to-student information recovery (T2S-IR) to maintain favorable performance of AAR at low sampling rates. Extensive experiments conducted on public datasets acquired for horses or/and goats using tri-axial accelerometers and tri-axial gyroscopes have verified the effectiveness of my proposed methods, which exhibit superior performance to the state-of-the-art algorithms on various tasks. Overall, I summarize this thesis as follows.

In Chapter 2, I comprehensively summarize recent research on AAR based on wearable

sensors and deep learning techniques. Five commonly used sensor types for monitoring animal activities are included, i.e., accelerometers, gyroscopes, magnetometers, global navigation satellite systems, and ultra-wideband. The accelerometer is the most frequently used type of sensor in AAR tasks, and most studies have either utilized accelerometers alone or combined them with other sensors to record animal motion data. The majority of studies have chosen the neck as the preferred sensor location, and sensors placed on the neck have been proven to be more accurate in detecting daily animal activities (e.g., feeding, ruminating, and resting) than sensors placed at other locations. Cattle are the most studied animals, mainly because compared with other animals, cattle are larger, so are suitable for wearing sensors, and they have higher economic benefits in agriculture. Only a few studies have focused on activities that are rare but important for animal health, such as licking, itch rubbing, scratching, and shaking. These activities should be further investigated in future work. In addition, the applications of different deep learning approaches in wearable sensor-aided AAR are also reviewed, according to the taxonomy of deep learning algorithms, i.e., Fully connected feedforward neural network, CNN (Convolutional neural network), RNN (Recurrent neural network), and hybrid models. CNNs are currently the most widely used models in AAR-related research. They have exhibited excellent performance in most cases, with accuracies higher than 90%. Considering the advantage of RNNs in modeling time series, a few studies have combined a CNN with an RNN to process sequential data from wearable sensors, and the combined model outperforms the single models. Therefore, future research can focus on combining a CNN and an RNN to further explore the effectiveness of this approach. In addition, we provide a comprehensive list of publicly available datasets collected via wearable sensor-aided AAR over the past five years. This list can serve as a valuable resource for readers who wish to further explore the field of AAR.

In Chapter 3, to promote the ability of AAR based on imbalanced multi-modal data, I develop a CMI-Net for multi-modal fusion and adopt CB focal loss for mitigating the class imbalance problem. Specifically, the CMI-Net consists of a dual CNN trunk architecture to extract modality-specific features and a cross-modality interaction module (CMIM) to achieve deep inter-modality interaction. In particular, the CMIM based on the attention mechanism adaptively recalibrates each modality's temporal- and axis-wise features by leveraging multi-modal information. Thus, it enables the CMI-Net to effectively capture complementary information and suppress unrelated information from multiple modalities. In addition, the CB focal loss is employed to supervise the network training, which forces the network to pay more attention not only to samples of minority classes, diminishing their influence from being

overwhelmed during optimization, but also to samples that are hard to distinguish.

In Chapter 4, I investigate the applications of federated learning (FL) in the field of wearable sensor-aided AAR, aiming to develop a global model based on decentralized data over different farms while protecting data privacy and ownership. I adequately consider two challenges (i.e., client-drift during local training and local gradient conflicts during global aggregation) resulting from data heterogeneity between multiple farms when directly applying FL to AAR tasks. To address these two challenges, I propose a novel FL framework called FedAAR that involves a prototype-guided local update (PLU) module for local optimization and a gradient-refinement-based aggregation (GRA) module for global aggregation. Specifically, the PLU module introduces a global prototype as shared knowledge to force clients to learn consistent features, reducing the divergence between client updates. The GRA module eliminates conflicting components between local gradients during global aggregation, effectively guaranteeing that all refined local gradients point in a positive direction to improve the agreement among clients.

In Chapter 5, I present a novel approach named T2S-IR to achieve energy-efficient AAR at low sampling rates while maintaining desirable performance. The T2S-IR effectively leverages the knowledge obtained from high-sampling-rate data, to assist in recovering the missing information in features extracted by the classification network trained on low-sampling-rate data. Concretely, I first utilize high-sampling-rate data for training teacher classification and reconstruction networks sequentially. Then, I train a student classification network using low-sampling-rate data, while enhancing its capability by exploiting the knowledge learned by trained teacher networks via two novel modules, including the reconstruction-based information recovery (RIR) module and the correlation-distillation-based information recovery (CDIR) module. Particularly, the RIR module exploits the pre-trained teacher reconstruction network to compel the student classification network to learn complete and descriptive features. The CDIR module enforces the feature maps of student network to mimic internal correlations within feature maps of pre-trained teacher classification network along temporal and sensor axes directions. The trained student network can be directly applied to infer different animal activities in practical scenarios with low sampling rates.

## 6.2 Limitations and Future Works

Automated AAR based on wearable sensors and deep learning techniques is critical for

achieving precise animal monitoring and management. In addition to the challenges addressed in this thesis, some limitations and practical challenges still require us to explore and solve, including annotation scarcity, inter-activity similarity, domain generalization, energy efficiency, and open-set recognition (see Section 2.3). Correspondingly, I propose potential future research directions in the following to address these challenges, aiming to promote the robustness of automated AAR.

### 6.2.1 Few-shot Learning for AAR with Scarce Annotated Dataset

As described in Section 2.5.1, only a few samples (less than 15%) are labeled among the original 1.2 million 2-s data samples within the horse dataset, which is common due to the time-consuming and labor-intensive annotation operation. To address the annotation scarcity issue, most AAR-related works employ data augmentation techniques to expand data sizes through the transformation of existing annotated data via rotation, jittering, scaling, noise addition, and other transformation techniques [4, 14, 36, 55]. However, augmentation techniques tend to be tailored to specific datasets, restricting their generalizability to other datasets. Transfer learning is another effective strategy to solve the problem of annotation scarcity, since it allows models pre-trained on large-scale datasets to extract general features that can be fine-tuned on the target dataset with limited labeled data [158]. A recent study on cow's feeding behavior recognition has validated the transfer learning to increase classification accuracy based on a small number of labeled datasets [35]. However, these above-described methods are not applicable to the extreme case in which there are a few labeled samples available. Few-shot learning is a promising option for this case, as it aims to learn a model or an optimizer from a set of base tasks, and the learned model generalizes well to new tasks with few labeled training samples. Thus, I am interested in designing algorithms based on few-shot learning to achieve precise AAR when the annotated dataset is extremely scarce.

### 6.2.2 Multi-type Sensors for Addressing Inter-activity Similarity

As displayed in Chapter 3, the developed classification algorithm easily confuses walking-natural with walking-rider because of the high similarity of movement patterns between these activities (see Fig. 3.8). Currently, there is a lack of studies on how inter-activity similarity affects the development of deep learning in wearable sensor-aided AAR. This raises urgent concerns for finding feasible solutions to increase the capability of distinguishing different animal activities with high similarity. One potential solution is to employ fine-grained activity

recognition, which seeks to recognize subtle differences between similar activities by using more detailed features and modeling techniques than other approaches [161]. However, this method relies on complex algorithms and large amounts of data and is noise-sensitive. A better solution is the simultaneous application of multi-type wearable sensors (e.g., an accelerometer and a geographical positional system (GPS) device), which can help capture richer multi-modal information about animals than a single device and further enhance the fault tolerance of AAR systems. Furthermore, advanced multi-modal fusion techniques can be applied to combine data from multiple sensors, further improving the accuracy and reliability of AAR systems. For instance, the feeding and drinking of cattle have similar movement patterns, and thus a GPS device can be added to record geo-location information, as these two activities are likely to occur in different functional areas, that is, a feeding area and at a water trough [45]. In future work, I will utilize multiple types of sensors to collect data from animals, seeking to improve the accuracy and robustness of deep learning models in distinguishing different animal activities with high similarity.

### 6.2.3 Unseen Domain Generalization with Domain-agnostic Learning

As presented in Chapter 3, the deep learning-based classification model is developed only based on a specific horse dataset, which contains limited horse individuals raised in a single site. The applications of the developed model for different horse individuals, horse species, or raising sites remain to be explored, which inevitably induces a non-negligible challenge called domain generalization. In reality, classification models trained on specific domains often exhibit significantly reduced performance when tested on data from a new domain, owning to the domain shift problem between training and test data. A common approach to address this problem is transfer learning, which allows transmitting the knowledge learned from diverse source datasets to target datasets with different domains and further improves performance. However, transfer learning may not be applicable in situations where the source and target domains are substantially different. That is, a pre-trained model may not capture the relevant features in the target domain, and fine-tuning the model may result in overfitting or underfitting due to the mismatch between the source and target datasets. In such a case, domain-agnostic learning is a suitable option, which can enhance a model's generalizability by enabling the model to learn feature representations that are robust or invariant to domain shifts. That is, it enables a model to effectively capture characteristics shared by the source and target domains by learning domain-invariant features, thus improving its generalizability to new domains. On this point, I will investigate the effectiveness of domain-agnostic learning in AAR tasks, aiming

to learn a model that can generalize well across multiple domains.

### 6.2.4 Energy-efficient AAR with Light-weight Models

In Chapter 5, I attempt to decrease the sampling rate of wearable sensors to achieve energy-efficient AAR; meanwhile, I propose the T2S-IR method to alleviate the performance degradation resulting from the reduction of sampling rates. However, only adjusting the configuration parameters of wearable sensors is not enough. The memory and computational cost of deep learning models also considerably affect energy consumption, and particularly this should be noticeable when embedding the models in sensing devices. Thus, constructing light-weight deep learning models is a necessary and worthy explored direction for achieving energy-efficient AAR. Popular techniques for constructing light-weight deep learning models include knowledge distillation [29], weight quantization [159], and network pruning [160]. In particular, knowledge distillation can enable the transfer of knowledge from a cumbersome model to a small model that is more suitable for deployment while maintaining desirable performance. Weight quantization seeks to reduce memory footprint and computational complexity by representing model weights with a smaller number of bits than the original representation. Network pruning involves removing or pruning the connections, weights, or neurons in a neural network. As implementing the above-described solutions may sacrifice a model's performance, it is crucial to identify a good trade-off between energy efficiency and recognition performance in automated AAR systems. Therefore, I will improve these methods or explore new ones to construct light-weight models while ensuring desirable performance.

### 6.2.5 Open-set Recognition with Generative Adversarial Network

As implemented in Chapter 3, Chapter 4, and Chapter 5, the proposed deep learning-based methods are always validated based on category-limited training datasets, which is also an ordinary operation in existing AAR-related studies. During the inference phase, trained models typically classify samples as activity categories seen in the training phase. However, some rare but important activities, which may occur in real-world monitoring scenarios but be absent from training datasets, are also misclassified into known activity categories in a training dataset. As a result, open-set recognition, which requires models to not only accurately classify known categories but also effectively deal with unknown categories, is an urgently explored research direction. The generative adversarial network is a powerful algorithm that can help solve the open-set recognition problem by modeling the distribution of both known and unknown classes.

This model can generate new samples that resemble the known classes and identify samples dissimilar to the known classes as potential outliers and label them as belonging to unknown classes. Therefore, I plan to design customized algorithms based on the generative adversarial network, which can simultaneously perform classification well on known and unknown activity categories.

# References


1. Mao A, Huang E, Wang X, Liu K (2023) Deep learning-based animal activity recognition with wearable sensors: Overview, challenges, and future directions. Comput Electron Agric 211:108043. https://doi.org/10.1016/j.compag.2023.108043
2. Eerdekens A, Deruyck M, Fontaine J, Martens L, Poorter E De, Plets D, Joseph W (2021) A framework for energy-efficient equine activity recognition with leg accelerometers. Comput Electron Agric 183:106020. https://doi.org/10.1016/j.compag.2021.106020
3. Astill J, Dara RA, Fraser EDG, Roberts B, Sharif S (2020) Smart poultry management: Smart sensors, big data, and the internet of things. Comput Electron Agric 170:105291. https://doi.org/10.1016/j.compag.2020.105291
4. Li C, Tokgoz K, Fukawa M, Bartels J, Ohashi T, Takeda KI, Ito H (2021) Data augmentation for inertial sensor data in CNNs for cattle behavior classification. IEEE Sensors Lett 5:1–4. https://doi.org/10.1109/LSENS.2021.3119056
5. Chambers RD, Yoder NC, Carson AB, Junge C, Allen DE, Prescott LM, Bradley S, Wymore G, Lloyd K, Lyle S (2021) Deep learning classification of canine behavior using a single collar-mounted accelerometer: Real-world validation. Animals 11:1–19. https://doi.org/10.3390/ani11061549
6. Mao A, Huang E, Gan H, Parkes RS V, Xu W (2021) Cross-modality interaction network for equine activity recognition using imbalanced multi-modal data †. Sensors 21:5818
7. Lecun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436–444. https://doi.org/10.1038/nature14539
8. Nweke HF, Teh YW, Al-garadi MA, Alo UR (2018) Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges. Expert Syst Appl 105:233–261. https://doi.org/10.1016/j.eswa.2018.03.056
9. Chen K, Zhang D, Yao L, Guo BIN, Yu Z, Liu Y (2021) Deep learning for sensor-based human activity recognition: overview, challenges and opportunities. ACM Comput Surv 54:1–40
10. Wang J, Chen Y, Hao S, Peng X, Hu L (2019) Deep learning for sensor-based activity recognition: A survey. Pattern Recognit Lett 119:3–11. https://doi.org/10.1016/j.patrec.2018.02.010
11. Kleanthous N, Hussain A, Khan W, Sneddon J, Liatsis P (2022) Deep transfer learning in sheep activity recognition using accelerometer data. Expert Syst Appl 207:117925. https://doi.org/10.1016/j.eswa.2022.117925
12. Wu Y, Liu M, Peng Z, Liu M, Wang M, Peng Y (2022) Recognising cattle behaviour with deep residual bidirectional LSTM model using a wearable movement monitoring collar. Agriculture 12:1237
13. Eerdekens A, Callaert A, Deruyck M, Martens L, Joseph W (2022) Dog's behaviour classification based on wearable sensor accelerometer data. In: 2022 5th Conference on Cloud and Internet of Things. IEEE, pp 226–231

14. Minati L, Li C, Bartels J, Chakraborty P, Li Z, Yoshimura N, Frasca M, Ito H (2023) Accelerometer time series augmentation through externally driving a non-linear dynamical system. Chaos, Solitons and Fractals 168:113100

15. Lin H, Lou J, Xiong L, Shahabi C (2021) SemiFed: Semi-supervised Federated Learning with Consistency and Pseudo-Labeling. arXiv Prepr

16. Durrant A, Markovic M, Matthews D, May D, Enright J, Leontidis G (2022) The role of cross-silo federated learning in facilitating data sharing in the agri-food sector. Comput Electron Agric 193:106648. https://doi.org/10.1016/j.compag.2021.106648

17. Liu Y, Ma X, Shu L, Hancke GP, Abu-Mahfouz AM (2021) From industry 4.0 to agriculture 4.0: current status, enabling technologies, and research challenges. IEEE Trans Ind Informatics 17:4322–4334. https://doi.org/10.1109/TII.2020.3003910

18. Benaissa S, Tuyttens FAM, Plets D, de Pessemier T, Trogh J, Tanghe E, Martens L, Vandaele L, Van Nuffel A, Joseph W, Sonck B (2019) On the use of on-cow accelerometers for the classification of behaviours in dairy barns. Res Vet Sci 125:425–433. https://doi.org/10.1016/j.rvsc.2017.10.005

19. Li G, Huang Y, Chen Z, Chesser GD, Purswell JL, Linhoss J, Zhao Y (2021) Practices and applications of convolutional neural network-based computer vision systems in animal farming: A review. Sensors 21:1–42. https://doi.org/10.3390/s21041492

20. Riekert M, Klein A, Adrion F, Hoffmann C, Gallmann E (2020) Automatically detecting pig position and posture by 2D camera imaging and deep learning. Comput Electron Agric 174:105391. https://doi.org/10.1016/j.compag.2020.105391

21. Sakai K, Oishi K, Miwa M, Kumagai H, Hirooka H (2019) Behavior classification of goats using 9-axis multi sensors: The effect of imbalanced datasets on classification performance. Comput Electron Agric 166:105027. https://doi.org/10.1016/j.compag.2019.105027

22. Xu X, Li W, Duan Q (2021) Transfer learning and SE-ResNet152 networks-based for small-scale unbalanced fish species identification. Comput Electron Agric 180:105878. https://doi.org/10.1016/j.compag.2020.105878

23. López V, Fernández A, García S, Palade V, Herrera F (2013) An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. Inf Sci (Ny) 250:113–141. https://doi.org/10.1016/j.ins.2013.07.007

24. Zhong M, Taylor R, Bates N, Christey D, Basnet H, Flippin J, Palkovitz S, Dodhia R, Lavista Ferres J (2021) Acoustic detection of regionally rare bird species through deep convolutional neural networks. Ecol Inform 64:101333. https://doi.org/10.1016/j.ecoinf.2021.101333

25. Mao A, Giraudet C, Liu K, De I, Nolasco A, Xie Z, Xie Z, Gao Y, Theobald J, Bhatta D, Stewart R, Mcelligott AG (2022) Automated identification of chicken distress vocalizations using deep learning models. J R Soc Interface 19:20210921

26. Wang L, Arablouei R, Alvarenga FAP, Bishop-hurley GJ (2023) Classifying animal behavior from accelerometry data via recurrent neural networks. Comput Electron Agric 206:107647. https://doi.org/10.1016/j.compag.2023.107647

27. Lin S, Xie H, Wang B, Yu K, Chang X, Liang X, Wang G (2022) Knowledge distillation via the target-aware transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp 10915–10924

28. Walton E, Casey C, Mitsch J, Vázquez-Diosdado JA, Yan J, Dottorini T, Ellis KA, Winterlich A, Kaler J (2018) Evaluation of sampling frequency, window size and sensor position for classification of sheep behaviour. R Soc Open Sci 5:171442. https://doi.org/10.1098/rsos.171442

29. Hinton G, Vinyals O, Dean J (2015) Distilling the knowledge in a neural network. arXiv Prepr

30. Khan A, Hammerla N, Mellor S, Plötz T (2016) Optimising sampling rates for accelerometer-based human activity recognition. Pattern Recognit Lett 73:33–40.

https://doi.org/10.1016/j.patrec.2016.01.001

31. Riaboff L, Shalloo L, Smeaton AF, Couvreur S, Madouasse A, Keane MT (2022) Predicting livestock behaviour using accelerometers: A systematic review of processing techniques for ruminant behaviour prediction from raw accelerometer data. Comput Electron Agric 192:106610. https://doi.org/10.1016/j.compag.2021.106610

32. Benaissa S, Tuyttens FAM, Plets D, Cattrysse H, Martens L, Vandaele L, Joseph W, Sonck B (2019) Classification of ingestive-related cow behaviours using RumiWatch halter and neck-mounted accelerometers. Appl Anim Behav Sci 211:9–16. https://doi.org/10.1016/j.applanim.2018.12.003

33. Fan B, Bryant R, Greer A (2022) Behavioral fingerprinting : acceleration sensors for identifying changes in livestock health. Multidiscip Sci J 5:435–454

34. Kleanthous N, Hussain AJ, Khan W, Sneddon J, Al-Shamma'a A, Liatsis P (2022) A survey of machine learning approaches in animal behaviour. Neurocomputing 491:442–463. https://doi.org/10.1016/j.neucom.2021.10.126

35. Bloch V, Frondelius L, Arcidiacono C, Mancino M, Pastell M (2023) Development and analysis of a CNN- and transfer-learning-based classification model for automated dairy cow feeding behavior recognition from accelerometer data. Sensors 23:2611

36. Pan Z, Chen H, Zhong W, Wang A, Zheng C (2023) A CNN-based Animal Behavior Recognition Algorithm for Wearable Devices. IEEE Sens J 23:1–1. https://doi.org/10.1109/jsen.2023.3239015

37. Zhao Z, Ha D, Damle A, White R, Shin S (2022) Improved sensor-based animal behavior classification performance through conditional generative adversarial network. arXiv Prepr

38. Kasnesis P, Doulgerakis V, Uzunidis D, Kogias DG, Funcia SI, González MB, Giannousis C, Patrikakis CZ (2022) Deep learning empowered wearable-based behavior recognition for search and rescue dogs. Sensors 22:1–26. https://doi.org/10.3390/s22030993

39. Hussain A, Ali S, Abdullah, Kim HC (2022) Activity detection for the wellbeing of dogs using wearable sensors based on deep learning. IEEE Access 10:53153–53163. https://doi.org/10.1109/ACCESS.2022.3174813

40. Liseune A, den Poel D Van, Hut PR, van Eerdenburg FJCM, Hostens M (2021) Leveraging sequential information from multivariate behavioral sensor data to predict the moment of calving in dairy cattle using deep learning. Comput Electron Agric 191:106566. https://doi.org/10.1016/j.compag.2021.106566

41. Hosseininoorbin S, Layeghy S, Kusy B, Jurdak R, Bishop-hurley G, Portmann M (2021) Deep learning-based cattle activity classification using joint time-frequency data representation. Comput Electron Agric 187:106241. https://doi.org/10.1016/j.compag.2021.106241

42. Dominguez-Morales JP, Duran-Lopez L, Gutierrez-Galan D, Rios-Navarro A, Linares-Barranco A, Jimenez-Fernandez A (2021) Wildlife monitoring on the edge: A performance evaluation of embedded neural networks on microcontrollers for animal behavior classification. Sensors 21:2975. https://doi.org/10.3390/s21092975

43. Peng Y, Kondo N, Fujiura T, Suzuki T, Ouma S, Wulandari, Yoshioka H, Itoyama E (2020) Dam behavior patterns in Japanese black beef cattle prior to calving: Automated detection using LSTM-RNN. Comput Electron Agric 169:105178. https://doi.org/10.1016/j.compag.2019.105178

44. Peng Y, Kondo N, Fujiura T, Suzuki T, Wulandari, Yoshioka H, Itoyama E (2019) Classification of multiple cattle behavior patterns using a recurrent neural network with long short-term memory and inertial measurement units. Comput Electron Agric 157:247–253. https://doi.org/10.1016/j.compag.2018.12.023

45. Halachmi I, Guarino M, Bewley J, Pastell M (2019) Smart animal agriculture: Application of real-time sensors to improve animal well-being and production. Annu Rev Anim Biosci 7:403–

425. https://doi.org/10.1146/annurev-animal-020518-114851

46. Aquilani C, Confessore A, Bozzi R, Sirtori F, Pugliese C (2022) Review: Precision livestock farming technologies in pasture-based livestock systems. Anim Int J Anim Biosci 16:100429. https://doi.org/10.1016/j.animal.2021.100429

47. Tzanidakis C, Tzamaloukas O, Simitzis P (2023) Precision livestock farming applications ( PLF ) for grazing animals. Agriculture 13:288

48. Arablouei R, Wang Z, Bishop-hurley GJ, Liu J (2023) Multimodal sensor data fusion for in-situ classification of animal behavior using accelerometry and GNSS data. Smart Agric Technol 4:100163. https://doi.org/10.1016/j.atech.2022.100163

49. Benaissa S, Tuyttens FAM, Plets D, Martens L, Vandaele L, Joseph W, Sonck B (2023) Improved cattle behaviour monitoring by combining Ultra-Wideband location and accelerometer data. Animal 17:100730. https://doi.org/10.1016/j.animal.2023.100730

50. Pastell M, Frondelius L, Järvinen M, Backman J (2018) Filtering methods to improve the accuracy of indoor positioning data for dairy cows. Biosyst Eng 169:22–31. https://doi.org/10.1016/j.biosystemseng.2018.01.008

51. Porto SMC, Arcidiacono C, Giummarra A, Anguzza U, Cascone G (2014) Localisation and identification performances of a real-time location system based on ultra wide band technology for monitoring and tracking dairy cow behaviour in a semi-open free-stall barn. Comput Electron Agric 108:221–229. https://doi.org/10.1016/j.compag.2014.08.001

52. Arablouei R, Currie L, Kusy B, Ingham A, Greenwood PL, Bishop-Hurley G (2021) In-situ classification of cattle behavior using accelerometry data. Comput Electron Agric 183:106045. https://doi.org/10.1016/j.compag.2021.106045

53. Coelho Ribeiro LA, Bresolin T, Rosa GJ de M, Rume Casagrande D, Danes M de AC, Dórea JRR (2021) Disentangling data dependency using cross-validation strategies to evaluate prediction quality of cattle grazing activities using machine learning algorithms and wearable sensor data. J Anim Sci 99:1–8. https://doi.org/10.1093/jas/skab206

54. Eerdekens A, Deruyck M, Fontaine J, Martens L, De Poorter E, Joseph W (2020) Automatic equine activity detection by convolutional neural networks using accelerometer data. Comput Electron Agric 168:105139. https://doi.org/10.1016/j.compag.2019.105139

55. Eerdekens A, Deruyck M, Fontaine J, Martens L, Poorter E De, Plets D, Joseph W (2020) Resampling and data augmentation for equines' behaviour classification based on wearable sensor accelerometer data using a convolutional neural network. In: 2020 International Conference on Omni-Layer Intelligent Systems, COINS 2020. pp 1–6

56. Kamminga JW, Le D V, Havinga PJM (2020) Towards deep unsupervised representation learning from accelerometer time series for animal activity recognition. In: Proceedings of the 6th Workshop on Mining and Learning from Time Series

57. Arablouei R, Wang L, Phillips C, Currie L, Yates J, Bishop-hurley G (2023) In-situ animal behavior classification using knowledge distillation and fixed-point quantization. Smart Agric Technol 4:100159. https://doi.org/10.1016/j.atech.2022.100159

58. Arablouei R, Wang L, Currie L, Yates J, Alvarenga FAP, Bishop-Hurley GJ (2023) Animal behavior classification via deep learning on embedded systems. Comput Electron Agric 207:107707. https://doi.org/10.1016/j.compag.2023.107707

59. Mao A, Huang E, Gan H (2022) FedAAR : A novel federated learning framework for animal activity recognition with wearable sensors. Animals 12:2142

60. Hussain A, Begum K, Poupi T, Armand T, Mozumder AI, Ali S, Kim HC, Joo M (2022) Long short-term memory (LSTM)-based dog activity detection using accelerometer and gyroscope. Appl Sci 12:9427

61. Kim J, Moon N (2022) Dog behavior recognition based on multimodal data from a camera and

wearable device. Appl Sci 12:3199. https://doi.org/10.3390/app12063199

62. Pavlovic D, Davison C, Hamilton A, Marko O, Atkinson R, Michie C, Crnojevi V, Andonovic I, Bellekens X, Tachtatzis C (2021) Classification of cattle behaviours using neck-mounted accelerometer-equipped collars and convolutional neural networks. Sensors 21:4050

63. Riaboff L, Couvreur S, Madouasse A, Roig-Pons M, Aubin S, Massabie P, Chauvin A, Bédère N, Plantier G (2020) Use of predicted behavior from accelerometer data combined with GPS data to explore the relationship between dairy cow behavior and pasture characteristics. Sensors (Switzerland) 20:1–33. https://doi.org/10.3390/s20174741

64. Wang J, Chen Y, Hao S, Peng X, Hu L (2019) Deep learning for sensor-based activity recognition: A survey. Pattern Recognit Lett 119:3–11. https://doi.org/10.1016/j.patrec.2018.02.010

65. Mao A, Zhu M, Huang E, Liu K (2022) A teacher-to-student information recovery method toward energy-efficient animal activity recognition at low sampling rates. preprint

66. Shahbazi M, Mohammadi K, Derakhshani SM, Groot Koerkamp PWG (2023) Deep Learning for Laying Hen Activity Recognition Using Wearable Sensors. Agric 13:1–19. https://doi.org/10.3390/agriculture13030738

67. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9:1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

68. Chung J, Gulcehre C, Cho K, Bengio Y (2014) Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling

69. Chambers RD (2020) FilterNet: A many-to-many deep learning architecture for time series classification. Sensors 20:2498

70. Kamminga JW, Bisby HC, Le D V., Meratnia N, Havinga PJM (2017) Generic online animal activity recognition on collar tags. UbiComp/ISWC 2017 - Adjun Proc 2017 ACM Int Jt Conf Pervasive Ubiquitous Comput Proc 2017 ACM Int Symp Wearable Comput 597–606. https://doi.org/10.1145/3123024.3124407

71. Kamminga JW, Le D V., Meijers JP, Bisby H, Meratnia N, Havinga PJM (2018) Robust sensor-orientation-independent feature selection for animal activity recognition on collar tags. In: Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies. pp 1–27

72. Kamminga JW, Janßen LM, Meratnia N, Havinga PJM (2019) Horsing around—A dataset comprising horse movement. Data 4:1–13. https://doi.org/10.3390/data4040131

73. Vehkaoja A, Somppi S, Törnqvist H, Valldeoriola Cardó A, Kumpulainen P, Väätäjä H, Majaranta P, Surakka V, Kujala M V., Vainio O (2022) Description of movement sensor dataset for dog behavior classification. Data Br 40:107822. https://doi.org/10.1016/j.dib.2022.107822

74. Monteiro A, Gonçalves P, Marques MR, Belo AT, Braz F (2022) Sheep Nocturnal Activity Dataset. Data 7:. https://doi.org/10.3390/data7090134

75. Tan J, Wang C, Li B, Li Q, Ouyang W, Yin C, Yan J (2020) Equalization loss for long-tailed object recognition. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp 11659–11668

76. Bocaj E, Uzunidis D, Kasnesis P, Patrikakis CZ (2020) On the benefits of deep convolutional neural networks on animal activity recognition. In: Proceedings of 2020 International Conference on Smart Systems and Technologies, SST 2020. pp 83–88

77. Geng C, Huang S-J, Chen S (2020) Recent advances in open set recognition: A survey. IEEE Trans Pattern Anal Mach Intell 43:3614–3631. https://doi.org/10.1109/tpami.2020.2981604

78. Liu N, Zhang N, Han J (2020) Learning selective self-mutual attention for RGB-D saliency detection. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp 13753–13762

79. Ha S, Choi S (2016) Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors. In: Proceedings of the International Joint Conference on Neural Networks. IEEE, pp 381–388

80. Mustaqeem, Kwon S (2021) Optimal feature selection based speech emotion recognition using two-stream deep convolutional neural network. Int J Intell Syst 36:5116–5135. https://doi.org/10.1002/int.22505

81. Zhang S, Li Z, Yan S, He X, Sun J (2021) Distribution alignment: A unified framework for long-tail visual recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp 2361–2370

82. Khan SH, Hayat M, Bennamoun M, Sohel FA, Togneri R (2018) Cost-sensitive learning of deep feature representations from imbalanced data. IEEE Trans Neural Networks Learn Syst 29:3573–3587. https://doi.org/10.1109/TNNLS.2017.2732482

83. Lin TY, Goyal P, Girshick R, He K, Dollar P (2017) Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp 2980–2988

84. Wang T, Zhu Y, Zhao C, Zeng W, Wang J, Tang M (2021) Adaptive class suppression loss for long-tail object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp 3103–3112

85. Suh S, Lee H, Lukowicz P, Oh Y (2021) CEGAN : Classification enhancement generative adversarial networks for unraveling data imbalance problems. Neural Networks 133:69–86. https://doi.org/10.1016/j.neunet.2020.10.004

86. Gerych W, Agu E, Rundensteiner E (2019) Classifying depression in imbalanced datasets using an autoencoder-based anomaly detection approach. In: Proceedings - 13th IEEE International Conference on Semantic Computing, ICSC 2019. IEEE, pp 124–127

87. Brendan McMahan H, Moore E, Ramage D, Hampson S, Agüera y Arcas B (2017) Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017. pp 1273–1282

88. Huang Y, Chu L, Zhou Z, Wang L, Liu J, Pei J, Zhang Y, Canada HT (2021) Personalized cross-silo federated learning on non-iid data. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp 7865–7873

89. Acar DAE, Zhao Y, Navarro RM, Mattina M, Whatmough PN, Saligrama V (2021) Federated learning based on dynamic regularization. arXiv Prepr

90. Deng Y, Kamani MM, Mahdavi M (2020) Distributionally robust federated averaging. In: Advances in Neural Information Processing Systems. pp 15111–15122

91. Lin T, Kong L, Stich SU, Jaggi M (2020) Ensemble distillation for robust model fusion in federated learning. In: Advances in Neural Information Processing Systems. pp 2351–2363

92. Li T, Sahu AK, Zaheer M, Sanjabi M, Talwalkar A, Smith V (2020) Federated optimization in heterogeneous networks. In: In Conference on Machine Learning and Systems. pp 429–450

93. Li X, Jiang M, Zhang X, Kamp M, Dou Q (2021) FedBN: Federated learning on non-iid features via local batch normalization. arXiv Prepr

94. Karimireddy SP, Kale S, Mohri M, Reddi SJ, Stich SU, Suresh AT (2020) Scaffold: Stochastic controlled averaging for federated learning. In: International Conference on Machine Learning. pp 5132–5143

95. Lee G, Shin Y, Jeong M, Yun S-Y (2021) Preservation of the global knowledge by not-true self knowledge distillation in federated learning. arXiv Prepr

96. Li L, Fan Y, Tse M, Lin KY (2020) A review of applications in federated learning. Comput Ind Eng 149:106854. https://doi.org/10.1016/j.cie.2020.106854

97. Li Q, He B, Song D (2021) Model-contrastive federated learning. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp 10708–10717

98. Li T, Sahu AK, Talwalkar A, Smith V (2020) Federated learning: Challenges, methods, and future directions. IEEE Signal Process Mag 37:50–60. https://doi.org/10.1109/MSP.2020.2975749

99. Bucilă C, Caruana R, Niculescu-Mizil A (2006) Model compression. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. pp 535–541

100. Passalis N, Tzelepi M, Tefas A (2021) Probabilistic knowledge transfer for lightweight deep representation learning. IEEE Trans Neural Networks Learn Syst 32:2030–2039

101. Romero A, Ballas N, Kahou SE, Chassang A, Gatta C, Bengio Y (2015) FitNets: Hints for thin deep nets. 3rd Int Conf Learn Represent ICLR 2015 - Conf Track Proc 1–13

102. Chen P, Liu S, Zhao H, Jia J (2021) Distilling knowledge via knowledge review. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp 5006–5015

103. Liu L, Huang Q, Lin S, Xie H, Wang B, Chang X, Liang X (2021) Exploring inter-channel correlation for diversity-preserved knowledge Distillation. In: Proceedings of the IEEE International Conference on Computer Vision. pp 8251–8260

104. Tian Y, Krishnan D, Isola P (2020) Contrastive representation distillation. arXiv Prepr

105. Park W, Kim D, Lu Y, Cho M (2019) Relational knowledge distillation. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp 3962–3971

106. Zagoruyko S, Komodakis N (2017) Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In: 5th International Conference on Learning Representations

107. Li Z, Hoiem D (2018) Learning without forgetting. IEEE Trans Pattern Anal Mach Intell 40:2935–2947. https://doi.org/10.1109/TPAMI.2017.2773081

108. Hou S, Pan X, Loy CC, Wang Z, Lin D (2018) Lifelong learning via progressive distillation and retrospection. In: Proceedings of the European Conference on Computer Vision. pp 437–452

109. Orbes-arteainst M, Cardoso J, Sørensen L, Pai A (2019) Knowledge distillation for semi-supervised domain adaptation. arXiv Prepr. https://doi.org/10.1007/978-3-030-32695-1

110. Kamminga JW, Meratnia N, Havinga PJM (2019) Dataset: Horse movement data and analysis of its potential for activity recognition. Proc 2nd Work Data Acquis To Anal 22–25. https://doi.org/10.1145/3359427.3361908

111. Martiskainen P, Järvinen M, Skön JP, Tiirikainen J, Kolehmainen M, Mononen J (2009) Cow behaviour pattern recognition using a three-dimensional accelerometer and support vector machines. Appl Anim Behav Sci 119:32–38. https://doi.org/10.1016/j.applanim.2009.03.005

112. Frost AR, Parsons DJ, Stacey KF, Robertson AP, Welch SK, Filmer D, Fothergill A (2003) Progress towards the development of an integrated management system for broiler chicken production. Comput Electron Agric 39:227–240. https://doi.org/10.1016/S0168-1699(03)00082-6

113. Parkes RSV, Weller R, Pfau T, Witte TH (2019) The effect of training on stride duration in a cohort of two-year-old and three-year-old thoroughbred racehorses. Animals 9:1–11. https://doi.org/10.3390/ani9070466

114. van Weeren PR, Pfau T, Rhodin M, Roepstorff L, Serra Bragança F, Weishaupt MA (2017) Do we have to redefine lameness in the era of quantitative gait analysis? Equine Vet J 49:567–569. https://doi.org/10.1111/evj.12715

115. Bosch S, Serra Bragança F, Marin-Perianu M, Marin-Perianu R, van der Zwaag BJ, Voskamp J,

Back W, Van Weeren R, Havinga P (2018) Equimoves: A wireless networked inertial measurement system for objective examination of horse gait. Sensors (Switzerland) 18:1–35. https://doi.org/10.3390/s18030850

116. Rueß D, Rueß J, Hümmer C, Deckers N, Migal V, Kienapfel K, Wieckert A, Barnewitz D, Reulke R (2019) Equine welfare assessment: horse motion evaluation and comparison to manual pain measurements. In: Image and Video Technology: 9th Pacific-Rim Symposium. pp 156–169

117. Kumpulainen P, Cardó AV, Somppi S, Törnqvist H, Väätäjä H, Majaranta P, Gizatdinova Y, Hoog Antink C, Surakka V, Kujala M V., Vainio O, Vehkaoja A (2021) Dog behaviour classification with movement sensors placed on the harness and the collar. Appl Anim Behav Sci 241:105393. https://doi.org/10.1016/j.applanim.2021.105393

118. Tran DN, Nguyen TN, Khanh PCP, Trana DT (2021) An iot-based design using accelerometers in animal behavior recognition systems. IEEE Sens J 22:17515–17528. https://doi.org/10.1109/JSEN.2021.3051194

119. Maisonpierre IN, Sutton MA, Harris P, Menzies-Gow N, Weller R, Pfau T (2019) Accelerometer activity tracking in horses and the effect of pasture management on time budget. Equine Vet J 51:840–845. https://doi.org/10.1111/evj.13130

120. Mustaqeem, Kwon S (2021) MLT-DNet: Speech emotion recognition using 1D dilated CNN based on multi-learning trick approach. Expert Syst Appl 167:114177. https://doi.org/10.1016/j.eswa.2020.114177

121. Johnson JM, Khoshgoftaar TM (2019) Survey on deep learning with class imbalance. J Big Data 6:1–54. https://doi.org/10.1186/s40537-019-0192-5

122. Japkowicz N, Stephen S ju (2002) The class imbalance problem: A systemati study. Intell Data Anal 6:429–449

123. Cui Y, Jia M, Lin TY, Song Y, Belongie S (2019) Class-balanced loss based on effective number of samples. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp 9268–9277

124. Zhang Z, Lin Z, Xu J, Jin W Da, Lu SP, Fan DP (2021) Bilateral attention network for rgb-d dalient object detection. IEEE Trans Image Process 30:1949–1961. https://doi.org/10.1109/TIP.2021.3049959

125. Woo S, Park J, Lee JY, Kweon IS (2018) Cbam: Convolutional block attention module. In: Proceedings of the European conference on computer vision. pp 3–19

126. Mustaqeem, Kwon S (2021) Att-Net: Enhanced emotion recognition system using lightweight self-attention module. Appl Soft Comput 102:107101. https://doi.org/10.1016/j.asoc.2021.107101

127. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit 770–778. https://doi.org/10.1109/CVPR.2016.90

128. Nair V, Hinton GE (2010) Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th international conference on machine learning. pp 807–814

129. Joze HRV, Shaban A, Iuzzolino ML, Koishida K (2020) Mmtm: Multimodal transfer module for cnn fusion. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp 13286–13296

130. Casella E, Khamesi AR, Silvestri S (2020) A framework for the recognition of horse gaits through wearable devices. Pervasive Mob Comput 67:101213. https://doi.org/10.1016/j.pmcj.2020.101213

131. Zeng M, Nguyen LT, Yu B, Mengshoel OJ, Zhu J, Wu P, Zhang J (2014) Convolutional neural networks for human activity recognition using mobile sensors. In: 6th international conference on mobile computing, applications and services. pp 197–205

132. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1:67–82. https://doi.org/10.1109/4235.585893

133. Wei J, Wang Q, Li Z, Wang S, Zhou SK, Cui S (2021) Shallow feature matters for weakly supervised object localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp 5993–6001

134. Maaten L van der, Hinton G (2008) Visualizing data using t-sne. J Mach Learn Res 9:2579–2605. https://doi.org/10.1007/s10479-011-0841-3

135. De Cocq P, Van Weeren PR, Back W (2004) Effects of girth, saddle and weight on movements of the horse. Equine Vet J 36:758–763. https://doi.org/10.2746/0425164044848000

136. Uenishi S, Oishi K, Kojima T, Kitajima K, Yasunaka Y, Sakai K, Sonoda Y, Kumagai H, Hirooka H (2021) A novel accelerometry approach combining information on classified behaviors and quantified physical activity for assessing health status of cattle: a preliminary study. Appl Anim Behav Sci 235:105220. https://doi.org/10.1016/j.applanim.2021.105220

137. Yu B, Lv Y (2022) A survey on federated learning in data mining. Wiley Interdiscip Rev Data Min Knowl Discov 12:1443. https://doi.org/10.1002/widm.1443

138. He Y, Chen Y, Yang X, Zhang Y, Zeng B (2021) Class-wise adaptive self distillation for heterogeneous federated learning. In: Proceedings of the 36th AAAI Conference on Artificial Intelligence

139. Mustaqeem, Kwon S (2020) CLSTM: Deep feature-based speech emotion recognition using the hierarchical convlstm network. Mathematics 8:1–19. https://doi.org/10.3390/math8122133

140. Xiao J, Du C, Duan Z, Guo W (2021) A novel server-side aggregation strategy for federated learning in non-IID situations. In: Proceedings - 2021 20th International Symposium on Parallel and Distributed Computing, ISPDC 2021. IEEE, pp 17–24

141. Reyes J, Di Jorio L, Low-Kam C, Kersten-Oertel M (2021) Precision-weighted federated learning. arXiv Prepr

142. Yeganeh Y, Farshad A, Navab N, Albarqouni S (2020) Inverse distance aggregation for federated learning with non-iid data. Springer International Publishing

143. Tan Y, Long G, Liu L, Zhou T, Lu Q, Jiang J, Zhang C (2021) FedProto : Federated Prototype Learning across Heterogeneous Clients

144. Snell J, Swersky K, Zemel R (2017) Prototypical networks for few-shot learning. In: Advances in neural information processing systems. p 30

145. Andreux M, du Terrail JO, Beguier C, Tramel EW (2020) Siloed Federated Learning for Multi-centric Histopathology Datasets. Springer International Publishing

146. Zhu M, Chen Z, Yuan Y (2023) FedDM: Federated weakly supervised segmentation via annotation calibration and gradient de-conflicting. IEEE Trans Med Imaging PP:1–1. https://doi.org/10.1109/tmi.2023.3235757

147. Smith D, Dutta R, Hellicar A, Bishop-Hurley G, Rawnsley R, Henry D, Hills J, Timms G (2015) Bag of class posteriors, a new multivariate time series classifier applied to animal behaviour identification. Expert Syst Appl 42:3774–3784. https://doi.org/10.1016/j.eswa.2014.11.033

148. Jarchi D, Kaler J, Sanei S (2021) Lameness detection in cows using hierarchical deep learning and synchrosqueezed wavelet transform. IEEE Sens J 21:9349–9358. https://doi.org/10.1109/JSEN.2021.3054718

149. Barwick J, Lamb D, Dobos R, Schneider D, Welch M, Trotter M (2018) Predicting lameness in sheep activity using tri-axial acceleration signals. Animals 8:1–16. https://doi.org/10.3390/ani8010012

150. Haladjian J, Haug J, Nüske S, Bruegge B (2018) A wearable sensor system for lameness detection in dairy cattle. Multimodal Technol Interact 2:27. https://doi.org/10.3390/mti2020027

151. Price E, Langford J, Fawcett TW, Wilson AJ, Croft DP (2022) Classifying the posture and activity of ewes and lambs using accelerometers and machine learning on a commercial flock. Appl Anim Behav Sci 251:105630. https://doi.org/10.1016/j.applanim.2022.105630

152. Gougoulis DA, Kyriazakis I, Fthenakis GC (2010) Diagnostic significance of behaviour changes of sheep: A selected review. Small Rumin Res 92:52–56. https://doi.org/10.1016/j.smallrumres.2010.04.018

153. Hounslow JL, Brewster LR, Lear KO, Guttridge TL, Daly R, Whitney NM, Gleiss AC (2019) Assessing the effects of sampling frequency on behavioural classification of accelerometer data. J Exp Mar Bio Ecol 512:22–30. https://doi.org/10.1016/j.jembe.2018.12.003

154. Shao R, Perera P, Yuen PC, Patel VM (2020) Open-set adversarial defense. In: Proceedings of the european conference on computer vision. pp 682–698

155. Chakravarty P, Cozzi G, Ozgul A, Aminian K (2019) A novel biomechanical approach for animal behaviour recognition using accelerometers. Methods Ecol Evol 10:802–814. https://doi.org/10.1111/2041-210X.13172

156. Cornou C, Lundbye-Christensen S (2008) Classifying sows' activity types from acceleration patterns. An application of the multi-process kalman filter. Appl Anim Behav Sci 111:262–273. https://doi.org/10.1016/j.applanim.2007.06.021

157. Thévenaz P, Blu T, Unser M (2000) Interpolation revisited. IEEE Trans Med Imaging 19:739–758. https://doi.org/10.1109/42.875199

158. Weiss K, Khoshgoftaar TM, Wang DD (2016) A survey of transfer learning. Springer International Publishing

159. Gong Y, Liu L, Yang M, Bourdev L (2014) Compressing deep convolutional networks using vector quantization. arXiv Prepr

160. Blalock D, Ortiz JJG, Jonathan Frankle JG (2020) What is the state of neural network pruning? In: Proceedings of machine learning and systems. pp 129–146

161. De D, Bharti P, Das SK, Chellappan S (2015) Multimodal wearable sensing for fine-grained activity recognition in healthcare. IEEE Internet Comput 19:26–35. https://doi.org/10.1109/MIC.2015.72

# List of Publications

The following is work done during my PhD study.

### Refereed Journal Articles

1. **Mao, A.**, Zhu, M., Huang, E., Yao, X., & Liu, K.* (2023). A teacher-to-student information recovery method toward energy-efficient animal activity recognition at low sampling rates. Computers and Electronics in Agriculture, Accepted.
2. **Mao, A.**, Huang, E., Wang, X., & Liu, K.* (2023). Deep learning-based animal activity recognition with wearable sensors: Overview, challenges, and future directions. Computers and Electronics in Agriculture, 211, 108043.
3. **Mao, A.**, Huang, E., Gan, H., & Liu, K.* (2022). FedAAR: A novel federated learning framework for animal activity recognition with wearable sensors. Animals, 12(16), 2142.
4. **Mao, A.**, Giraudet, C. S., Liu, K.*, De Almeida Nolasco, I., Xie, Z., Xie, Z., Gao, Y., Theobald, J., Bhatta, D., Stewart, R., & McElligott, A. G.* (2022). Automated identification of chicken distress vocalizations using deep learning models. Journal of the Royal Society Interface, 19(191), 20210921.
5. **Mao, A.**, Huang, E., Gan, H., Parkes, R. S., Xu, W., & Liu, K.* (2021). Cross-modality interaction network for equine activity recognition using imbalanced multi-modal data. Sensors, 21(17), 5818.

### Conference Proceedings

6. **Mao, A.**, Huang, E., Zhu, M., & Liu, K.* (2023, May). Robust animal activity recognition using wearable sensors: A correlation distillation-based information recovery method toward data having low sampling rates. In 2nd U.S. Precision Livestock Farming Conference (*USPLF 2023*).
7. **Mao, A.**, Huang, E., Gan, H., & Liu, K.* (2022, August). Uniting farms: Federated learning for sensor-based animal activity recognition. In 10th European Conference on Precision Livestock Farming (*ECPLF 2022*).
8. **Mao, A.**, Huang, E., Xu, W., & Liu, K.* (2021, October). Cross-modality interaction network for equine activity recognition using time-series motion data. In Proceedings of the 2021 International Symposium on Animal Environment and Welfare (*ISAEW 2021*).