Received: 13 June 2023 | Accepted: 19 December 2023

DOI: 10.1111/2041-210X.14294

Methods in Ecology and Evolution
Open Access
BRITISH ECOLOGICAL SOCIETY

RESEARCH ARTICLE

Exploring deep learning techniques for wild animal behaviour classification using animal-borne accelerometers

Ryoma Otsuka[1] | Naoya Yoshimura[1] | Kei Tanigaki[1] | Shiho Koyama[2] | Yuichi Mizutani[2] | Ken Yoda[2] | Takuya Maekawa[1]

[1]Graduate School of Information Science and Technology, Osaka University, Suita, Osaka, Japan

[2]Graduate School of Environmental Studies, Nagoya University, Nagoya, Aichi, Japan

Correspondence
Ryoma Otsuka
Email: ryoma.otsuka87@gmail.com

Takuya Maekawa
Email: maekawa@ist.osaka-u.ac.jp

Funding information
Japan Society for the Promotion of Science, Grant/Award Number: JP21H05293 and JP21H05299

Handling Editor: Edward Codling

Abstract

1. Machine learning-based behaviour classification using acceleration data is a powerful tool in bio-logging research. Deep learning architectures such as convolutional neural networks (CNN), long short-term memory (LSTM) and self-attention mechanism as well as related training techniques have been extensively studied in human activity recognition. However, they have rarely been used in wild animal studies. The main challenges of acceleration-based wild animal behaviour classification include data shortages, class imbalance problems, various types of noise in data due to differences in individual behaviour and where the loggers were attached and complexity in data due to complex animal-specific behaviours, which may have limited the application of deep learning techniques in this area.
2. To overcome these challenges, we explored the effectiveness of techniques for efficient model training: data augmentation, manifold mixup and pre-training of deep learning models with unlabelled data, using datasets from two species of wild seabirds and state-of-the-art deep learning model architectures.
3. Data augmentation improved the overall model performance when one of the various techniques (none, scaling, jittering, permutation, time-warping and rotation) was randomly applied to each data during mini-batch training. Manifold mixup also improved model performance, but not as much as random data augmentation. Pre-training with unlabelled data did not improve model performance. The state-of-the-art deep learning models, including a model consisting of four CNN layers, an LSTM layer and a multi-head attention layer, as well as its modified version with shortcut connection, showed better performance among other comparative models. Using only raw acceleration data as inputs, these models outperformed classic machine learning approaches that used 119 handcrafted features.
4. Our experiments showed that deep learning techniques are promising for acceleration-based behaviour classification of wild animals and highlighted some challenges (e.g. effective use of unlabelled data). There is scope for greater exploration of deep learning techniques in wild animal studies (e.g. advanced data

augmentation, multimodal sensor data use, transfer learning and self-supervised learning). We hope that this study will stimulate the development of deep learning techniques for wild animal behaviour classification using time-series sensor data.

KEYWORDS
acceleration sensor, animal behaviour classification, bio-logging, data augmentation, deep learning, machine learning

1 | INTRODUCTION

1.1 | Behaviour classification of wild animals using time-series sensor data

Knowing when, where and what an animal is doing is fundamental to understanding animal behaviour. Bio-logging is a modern research technique that employs animal-borne data loggers to record a variety of time-series sensor data such as acceleration, temperature, water depth and location data (Fehlmann & King, 2016; Yoda, 2019). Among available sensors, acceleration sensors are commonly used to reconstruct animal behaviours, because many behaviours are characterised by unique patterns of acceleration signals (Yoda et al., 1999). Once the relationship between acceleration signals and behaviours is confirmed through video or direct observation (i.e. labelling or annotation), one can develop a 'behaviour classifier' through supervised learning. Then, it is possible to calculate behavioural time allocation (Yoda et al., 2001) and identify specific behaviours such as prey capture (Watanabe & Takahashi, 2013) from acceleration signals using these classifiers. Numerous techniques have been proposed to classify animal behaviours, including rule-based methods and machine learning.

Recently, the classic machine learning approach, that is, a non-deep learning, machine learning approach that usually requires feature engineering (see Table S1 for explanations of terms in this study), has succeeded in classifying animal behaviour. Previous studies have used various machine learning models with acceleration data to classify the behaviour of various animals, including birds and mammals (Fehlmann et al., 2017; Nathan et al., 2012; Yu et al., 2021). For instance, Nathan et al. (2012) tested the effectiveness of five classic machine learning models for behaviour classification of griffon vultures: linear discriminant analysis (LDA), support vector machine (SVM), decision tree (DT), random forest (RF) and artificial neural network (ANN). Yu et al. (2021) tested XGBoost in addition to LDA, DT, SVM, RF and ANN for five species. Although they mainly focused on seeking a suitable model for onboard behaviour classification, they demonstrated that SVM, RF, ANN and XGBoost generally performed better in terms of the F1-score or overall accuracy. Other methods have been employed, such as the k-nearest neighbour (Sur et al., 2017) and the hidden Markov model (Leos-Barajas et al., 2017).

Only a few studies have leveraged deep learning for wild animal behaviour classification using time-series sensor data. Although not an acceleration-based behaviour classification, Browning et al. (2018) used a multi-layer perceptron to predict diving behaviour in three seabird species using GPS data. Roy et al. (2022) extended their work by using convolutional neural networks (CNNs) and U-Net to predict seabird diving. Recently, Hoffman et al. (2023) applied deep learning models such as CNN and gated recurrent unit to datasets of nine species. As such, there are several examples of deep learning applications on time-series sensor data in recent bio-logging research; however, this area is still in the early stages of development. The effectiveness of more advanced architectures, such as long short-term memory (LSTM) and self-attention mechanism, as well as various training techniques, such as data augmentation, have not yet been extensively tested on acceleration data from wild animals.

1.2 | Behaviour classification techniques for domestic animals and humans

Deep learning-based behaviour classification techniques have been employed extensively in domestic animal and human studies (e.g. Pan et al., 2023; Singh et al., 2021). In the acceleration-based behaviour classification of domestic animals including horses and lactating sows, deep learning models such as CNN have been developed as a technique for automatically monitoring behaviours and obtaining information about animal health and welfare (e.g. Eerdekens et al., 2020; Pan et al., 2023). Although these techniques successfully classified multiple behaviour classes (e.g. six or seven classes), collecting data from domestic animals appeared to be easier than for wild animals.

In human activity recognition (HAR), Ordóñez and Roggen (2016) demonstrated that DeepConvLSTM (DCL), which combines CNN and LSTM, achieved high performance on datasets of daily activity and assembly-line workers' activity. Singh et al. (2021) proposed a model with an additional self-attention layer after the DCL architecture (called DeepConvLSTMSelfAttn (DCLSA) in this study) that could outperform DCL in various human activity datasets. More recently, HAR studies have been conducted in the industrial domain, with a focus on more specific and complex tasks. Xia et al. (2022) proposed attention-based neural networks to identify the skills of high- and low-performing workers. Yoshimura, Maekawa, et al. (2022) proposed a model for recognising complex, ordered and repetitive activities during line production systems and packaging tasks in the logistics domain. As such, the application of deep learning techniques in HAR is more varied and advanced than that in wild animals.

1.3 | Challenges and our approach

The following key challenges may have prevented the use of deep learning models in acceleration-based behaviour classification of wild animals. First, although deep learning models generally benefit from more training data, it is difficult to collect ground truth data for supervised learning, such as annotations acquired from video data, from wild animals. Second, the data are often imbalanced in terms of behaviour class. For example, the proportion of foraging behaviours in our target animals (streaked shearwaters and black-tailed gulls) is much lower than that of flying or stationary behaviour (Figures S1–S3). Third, there may be various types of noise in acceleration data due to differences in individual behaviour and where the loggers were attached. These three problems are also common in domestic animals and humans but may be more severe in wildlife. Fourth, acceleration data have complexity due to difficult animal-specific behaviours, such as those consisting of micro-actions (e.g. prey capture) and those likely requiring consideration of temporal dependencies for classification (e.g. foraging dive of streaked shearwaters, which consists of a sequence of actions such as diving underwater, following a school of fish and ascending to the sea surface (Tanigaki et al., 2024)). In this study, we explored the effectiveness of state-of-the-art deep learning architectures and related techniques for acceleration-based behaviour classification of wild animals, which may overcome the above-mentioned challenges, using datasets from two wild seabird species.

First, we explored the effects of data augmentation and manifold mixup. Data augmentation refers to techniques that transform data to increase their quantity and variation. Manifold mixup (Verma et al., 2019) generates a new training instance (a set of new features and label) by mixing intermediate features and labels of randomly sampled two existing training instances in an intermediate layer (see Section 2.4 for more details). These techniques are considered to improve generalisation performance, robustness to various noises and recognition performance of minor classes, and are thus expected to overcome the above challenges.

Second, we tested the effects of pre-training CNN-based Autoencoder (CNN-AE) with a large amount of unlabelled data, which is expected to be effective when using a small amount of labelled data. The CNN-AE can be either simply trained with labelled data or first pre-trained with unlabelled data and then fine-tuned with labelled data.

Finally, we explored various deep learning model architectures: CNN, LSTM, DCL, DCLSA, ResNet version of DCLSA (DCLSA-RN), Transformer and CNN-AE. Convolution layers in CNN, CNN-AE and DCL-based models are good at extracting local, specific features or patterns. An LSTM layer in LSTM and DCL-based models can incorporate short- and long-term temporal dependencies, which seems essential for time-series sensor data. Multi-head attention layer (Vaswani et al., 2017) in DCLSA, DCLSA-RN and Transformer learns which parts of the data to prioritise, considering global information. Thus, LSTM and multi-head attention layers could overcome the fourth challenge. We expected that this comparison will provide a better understanding of the performance of each of these components and/or their combinations. We also compared these deep learning models with classic machine learning approaches such as XGBoost, which achieved high performance in a previous study but required feature engineering.

2 | MATERIALS AND METHODS

2.1 | Datasets

Since 2018, our research team has developed custom-made bio-loggers with AI that perform real-time behaviour classification using low-power sensors and start camera recording, thus enabling the efficient recording of videos of target behaviours, such as seabird foraging (Korpela et al., 2020). Through this project, we collected acceleration, GPS and water pressure data as well as more than 20 h of video data (excluding those labelled as unknown) from two seabird species in the wild: streaked shearwaters (*Calonectris leucomelas*) and black-tailed gulls (*Larus crassirostris*). Data from 28 streaked shearwaters were collected on Awashima Island, Japan, from 2018 to 2022, and data from 27 black-tailed gulls were collected on Kabushima Island, Japan, in 2018, 2019 and 2022 (Table S2; Figures S1–S3). For streaked shearwaters, all the loggers were attached to the animals' backs (Figure S2), whereas for black-tailed gulls, 18 were attached to the animals' abdomens and the remainder were attached to their backs (Figure S3).

The fieldwork on streaked shearwaters was carried out with the permission of the Animal Experimental Committee of Nagoya University (GSES2018–2022) and the Ministry of the Environment, Japan. The fieldwork on black-tailed gulls was carried out with the permission of the Hachinohe City Board of Education (2018-237, 2019-329, 2022-301) and Aomori Prefecture (2018-4036, 2019-3033, 2022-3050) as well as from the Ministry of the Environment, Japan, to instal the structure (1803201, 1804042, 1903281) with approval from the Nagoya University Animal Experiment Committee (GSES2018, 2019 and 2022).

Using video data, we defined six behaviour classes (stationary, bathing, take-off, cruising flight, foraging dive and dipping) for streaked shearwaters and six behaviour classes (stationary, ground active, bathing, active flight, passive flight and foraging) for black-tailed gulls (Figures S1–S5). See Table S3 for more descriptions of each behaviour.

Acceleration data were recorded at 25 or 31 Hz. Those at 31 Hz were first up-sampled to 1000 Hz using the linear interpolation method and then down-sampled to 25 Hz because 31 Hz is not a multiple of 25 Hz, making it difficult to directly employ down-sampling while preserving the shape of the original signal. The time windows were extracted using a sliding window size of 50 samples (2 s) and an overlap rate of 50%. We labelled the data primarily using video data from animal-borne cameras, but also using GPS and water pressure data when the video footage was not very clear. Labelling was performed in consultation with ecologists who studied each target species. To avoid complexity, windows containing two or more unique behaviour class labels were discarded. In addition, we did not use windows with many missing data. We obtained 42,526 labelled windows from 28 streaked shearwaters and 32,391 from 27 black-tailed gulls. The number of labelled windows for each class was heavily imbalanced (Figures S1–S3). Figures S4 and S5 show examples of typical windows for each behaviour class in streaked shearwaters and black-tailed gulls, respectively. Acceleration values greater than +8G or smaller than −8G were clipped to address measurement errors. We did not perform other data pre-processing such as standardisation because pipelines and hyperparameters of pre-processing heavily rely on domain-specific knowledge and we wanted to eliminate the effect of it on our experiments.

2.2 | Model architectures and hyperparameters

We implemented the CNN, LSTM, DCL, DCLSA, DCLSA-RN, Transformer and CNN-AE, as shown in Figure 1. See the fourth paragraph of Section 1.3 for the reasons why we used these models in this study.

- CNN: CNN has four convolution layers, the number of convolution filters is 128, the kernel size is 5, the stride length is 1, and the amount of padding is 2. Batch normalisation and ReLU layers followed each convolution layer.
- LSTM: LSTM has one LSTM layer and one dropout layer; the number of LSTM hidden units is 128, and the dropout rate is 0.5.
- DCL: The original DCL has two LSTM layers after four convolution layers (Ordóñez & Roggen, 2016), but our DCL has one LSTM layer, following Singh et al. (2021) and Yoshimura, Morales, et al. (2022). Our DCL is a combination of the above CNN and LSTM, and the parameters are the same as above.
- DCLSA: The original DCLSA has an additional self-attention layer after the LSTM layer (Singh et al., 2021), but our DCLSA has a multi-head attention layer with four heads after the above DCL architecture.
- DCLSA-RN: DCLSA-RN is a modified version of DCLSA, with the latter three convolution layers replaced by four residual blocks with shortcut connections (He et al., 2016). The kernel size is 5, and the numbers of convolution filters of the first and second convolution layers in a residual block are 64 and 128, respectively.
- Transformer: Transformer has four transformer encoder blocks, each consisting of layer normalisation, multi-head attention and feedforward neural network layers.
- CNN-AE: CNN-AE mainly consists of three convolution layers as an encoder block and three transposed convolution layers as a decoder block. The kernel size is 5 in all convolution and transposed convolution layers. The number of convolution filters is 128 in the convolution layers and the first two transposed convolution layers, and 3 in the last transposed convolution layer. The time dimension of the data is gradually down-sampled in the encoder block using the max-pooling layer (from 50 to 26, 14 and 8), and up-sampled in the decoder block using the max-unpooling layer (from 8 to 14, 26 and 50).

See the source code (https://github.com/ryoma-otsuka/dl-wabc) and Table S4 for further details on model architectures, hyperparameters and the numbers of parameters. The implementations of the Transformer and CNN-AE were heavily based on those in Qian et al. (2022) but were slightly modified for this study. All deep learning models were implemented using Python (version 3.10.8) and PyTorch (version 1.13.1) on Ubuntu 18.04.6 LTS. The deep learning models were trained using Docker (version 20.10.22), Kubernetes (version 1.26.0) and a GPU cluster (Table S5).

The raw acceleration data of the three axes (*x*, *y* and *z*) were used as inputs to the deep learning models. Note that the 'features' in Figure 1 were fed into the flatten and linear layers to output an estimate per behaviour class for each window, but they were fed into the linear (for adjusting the data shape), dropout, flatten and linear layers for CNN-AE. We then applied a softmax function to obtain the prediction probability of each class and obtained one predicted class label with the maximum probability, resulting in one prediction label per window. Given that our datasets were imbalanced, we used the WeightedRandomSampler in Pytorch to obtain a class balance within each training batch. The batch size was 128. We used the cross-entropy loss as the loss function. We used Adam as the optimiser, set the initial learning rate to 0.001 and the weight decay to 0.0001, and gradually decreased the learning rate using the CosineLRScheduler in the 'timm' library. Unless otherwise stated, the minimum and maximum number of training epochs were 70 and 100, respectively. The patience parameter for early stopping was 10.

2.3 | Evaluation methods

We evaluated model performance by conducting leave-one-ID-out cross-validation (LOIO-CV). In each LOIO-CV fold, only one bird was excluded as a test individual and the model was trained on the remaining individuals. In each fold, the remaining data were divided into training and validation datasets (8:2 random split). The validation dataset was only used for early stopping.

We used the macro and class F1-score as performance metrics because our datasets were imbalanced. The F1-score is a harmonic mean of precision and recall. The precision, recall and F1-score are calculated as below:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}},$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}},$$

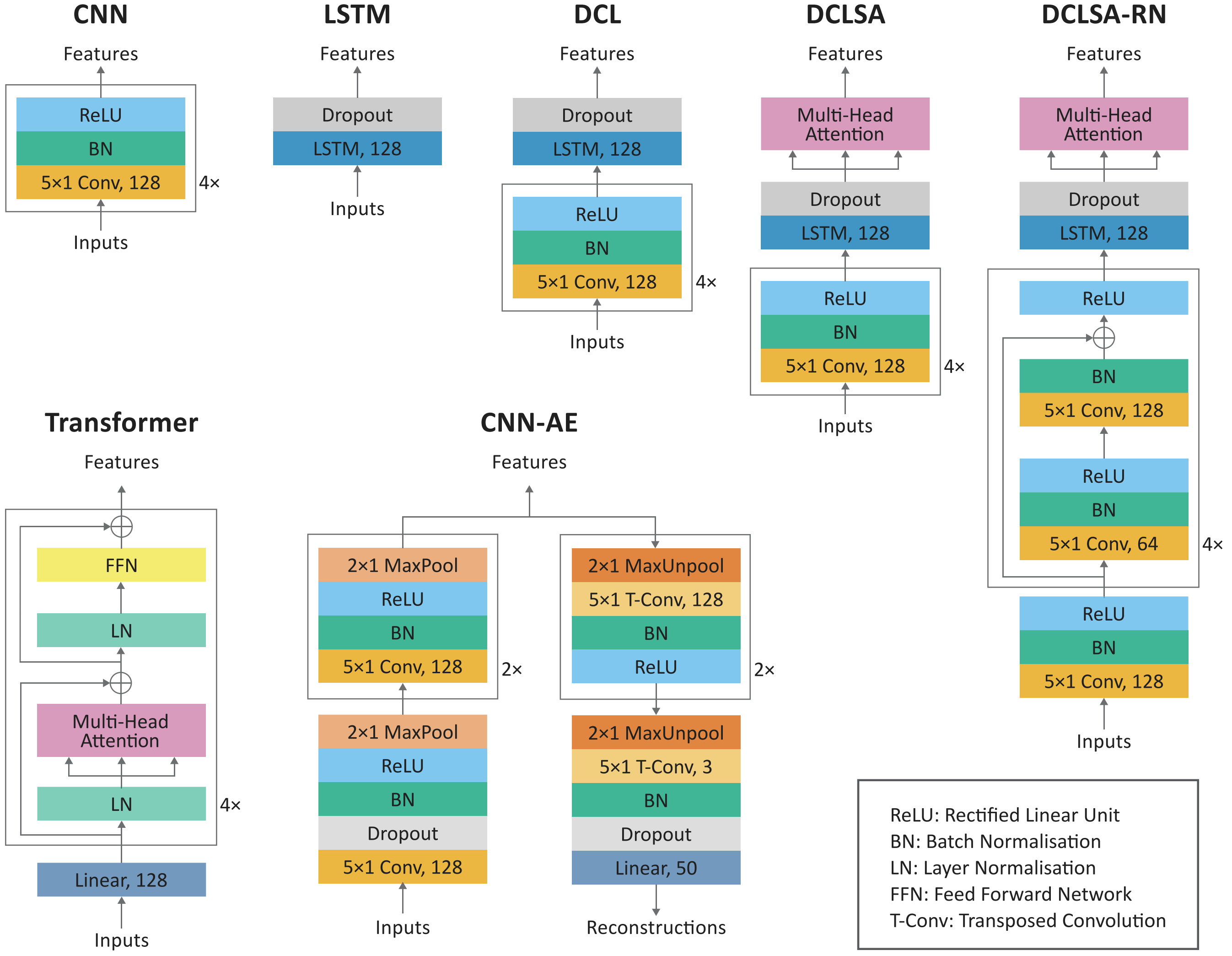

FIGURE 1 Deep learning model architectures: convolutional neural network (CNN), long short-term memory (LSTM), DeepConvLSTM (DCL), DeepConvLSTMSelfAttn (DCLSA), ResNet version of DCLSA (DCLSA-RN), Transformer and CNN-based Autoencoder (CNN-AE). Inputs were raw triaxial acceleration data. The features were fed into the flatten layer and the linear layer with the number of classes as the output dimension (the linear, dropout, flatten and linear layers for CNN-AE).

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}},$$

where TP is the number of true positives, FP is the number of false positives, and FN is the number of false negatives. The class F1-score is an F1-score calculated for each behaviour class, and the macro F1-score is the mean of the class F1-scores for all behaviour classes.

Note that because many individuals do not have data windows from some behaviour classes, F1-scores for such missing classes become zero when we calculate them for each of the individuals. Therefore, we calculated F1-scores by aggregating the prediction results of all the folds. To ensure robustness, we repeated LOIO-CV 10 times by changing the random seeds (seed = 0, 1, ..., 9). The F1-score was presented as the mean and the standard deviation.

2.4 | Experiment 1: Data augmentation and manifold mixup

In the following experiments, we used only DCL or DCLSA and fewer test individuals because our focus was to better understand how and to what extent each data augmentation technique and manifold mixup affected the prediction performance. For both species, we selected six individuals to cover all classes and reflect the differences in year and attachment position (OM1807, OM1901, OM2003, OM2102, OM2212 and OM2213 for streaked shearwaters, and UM1803, UM1807, UM1901, UM1908, UM1913 and UM2203 for black-tailed gulls). We performed LOIO-CV on six test birds and calculated the F1-scores as described above. This was repeated 10 times with 10 random seeds for each of the conditions described below.

Data augmentation is a technique that transforms data to increase its quantity and variation. Data augmentation techniques are expected to help models avoid overfitting, make them robust to various types of noise in acceleration data and improve the classification accuracy of minor behaviour classes. We tested the impacts of the following data augmentation techniques: scaling, jittering, permutation, time-warping (t-warp) and rotation following Um et al. (2017). Scaling samples a scaling factor from a Gaussian distribution (mean = 1.0, standard deviation = 0.2) and multiplies the factor with input data, changing the scale of the acceleration signal. Jittering randomly samples noise signals from a Gaussian distribution (mean = 0, standard deviation = 0.05) and adds the noise to input data. Permutation randomly splits input data into short segments (maximum number of segments = 10) and changes their orders. T-warp stretches and warps the acceleration signal in the temporal dimension (see Supplementary Explanation S1). Rotation applies a rotation matrix to input data with a randomly selected angle $\theta \in [-\pi, \pi]$, around random axes in 3D space. An example visualisation of these data augmentation techniques is shown in Figure 2 and see the source code for more details. We implemented these data augmentation techniques following Qian et al. (2022) but modified the parameters of scaling and jittering for our data. We also implemented random data augmentation which randomly applies one of the six data augmentation types (i.e. none and the five data augmentation techniques) to each window in a training batch. We compared seven data augmentation scenarios (none, scaling, jittering, permutation, t-warp, rotation and random) using DCL and DCLSA.

We also performed a grid search experiment to understand how hyperparameters of data augmentation techniques (e.g. standard deviation parameter for scaling) influence the performance of DCL. See Supplementary Experiment S1 in the supporting information for more details.

Mixup (Zhang et al., 2018) is a data augmentation technique that generates a new training instance by mixing two existing training instances. Manifold mixup (Verma et al., 2019) performs mixup in an intermediate layer. Where $(\mathrm{x_i}, \mathrm{y_i})$ and $(\mathrm{x_j}, \mathrm{y_j})$ are intermediate features and labels of two example instances randomly sampled from a training batch, a set of new features and label $(\hat{x}, \hat{y})$ are generated as below:

$$\hat{x} = \lambda x_i + (1 - \lambda)x_j,$$

$$\hat{y} = \lambda y_i + (1 - \lambda)y_j.$$

The mixing coefficient, $\lambda \in [0, 1]$, is sampled from the following Beta distribution.

$$\lambda \sim \mathrm{Beta}(\alpha, \alpha),$$

where $\alpha \in [0, \infty]$ (mixup alpha hereafter) is a hyperparameter that we explored its impact in this study. The distribution of λ will be skewed near zero or one when mixup alpha is 0.1, while it will be uniform distribution when mixup alpha is 1.0 (Figure S6). Please refer to the original papers (Verma et al., 2019; Zhang et al., 2018) for more details.

We expected that manifold mixup to regularise the model, and smooth the decision boundaries between behaviour classes, and improve the classification accuracy of minor behaviour classes. To test the effects of manifold mixup, we implemented manifold mixup before the LSTM layer in the DCL and compared the following six conditions: no mixup and with mixup (mixup alpha = 0.1, 0.2, 0.5, 1.0 and 2.0) for both species in the same manner as described in the first paragraph of this section. Usually, the reweighted class probabilities are used; however, we applied the argmax function to the reweighted probabilities and subsequently fed the output into the cross-entropy loss function. This was done because the latter approach showed superior performance in our preliminary

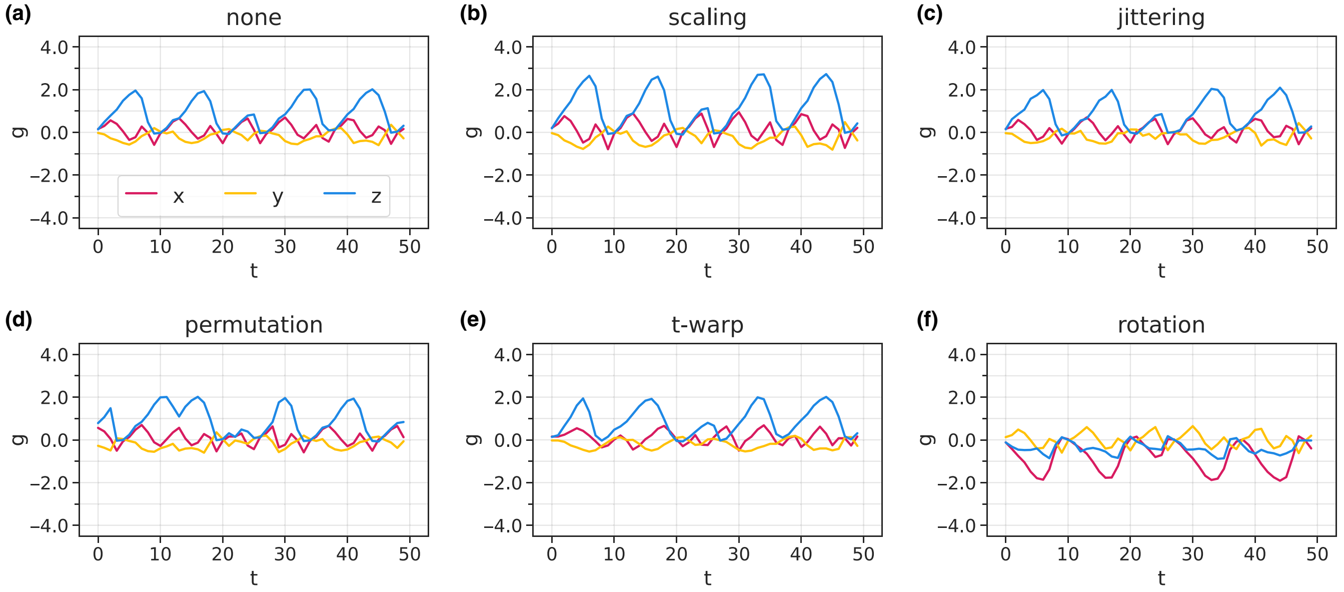

FIGURE 2 Examples of data augmentation types (a) none, (b) scaling, (c) jittering, (d) permutation, (e) t-warp and (f) rotation on an ‘active flight’ window from a black-tailed gull.

experiments. To investigate whether the combination of data augmentation and manifold mixup can improve the model performance, we performed experiments with and without random data augmentation. To examine the impact of manifold mixup position in the DCL architecture on the prediction performance, we also implemented manifold mixup after the LSTM layer without data augmentation.

2.5 | Experiment 2: Pre-training of CNN-AE

When there is much more unlabelled data than labelled data, pre-training with unlabelled data (unsupervised pre-training) may be effective (e.g. Le Paine et al., 2015). We tested the impact of unsupervised pre-training on CNN-AE using 1,546,440 and 1,398,580 instances from 33 streaked shearwaters and 29 black-tailed gulls, respectively (more than 36 and 43 times greater than the number of labelled data). We used the mean squared error to calculate the reconstruction loss during pre-training. We used the same optimiser and scheduler for supervised training. The extracted unlabelled windows were randomly shuffled for each individual. The batch size was 600. The maximum number of epochs was 100 and the patience parameter for early stopping was 10, but the median value of the actual number of epochs for unsupervised pre-training was 22.0 and 25.5 for streaked shearwaters and black-tailed gulls, respectively.

We compared the following four conditions: 'w/o', 'w/', 'w/ soft' and 'w/ hard'. The 'w/o' indicates that the model encoder was trained using only labelled data and cross-entropy loss function without pre-training. The 'w/' indicates that the model was pre-trained with unlabelled data, and then simply fine-tuned. The 'w/ soft' or 'w/ hard' indicates that the learning rate for the encoder parameters during the fine-tuning phase was a smaller value (0.00001) or frozen, respectively. For all conditions in Experiment 2, we applied random data augmentation and did not perform manifold mixup during unsupervised pre-training or supervised training.

2.6 | Experiment 3: Model comparison

We compared the performance of seven deep learning models: CNN, LSTM, DCL, DCLSA, DCLSA-RN, Transformer and CNN-AE w/o (see Sections 2.2 and 2.5), following the evaluation methods described in Section 2.3. In Experiment 3, LOIO-CV was repeated for all individuals (i.e. 28-fold for streaked shearwaters and 27-fold for black-tailed gulls). We applied random data augmentation and did not perform manifold mixup.

To compare deep learning models with classic machine learning models that require feature engineering, we implemented LightGBM (Ke et al., 2017) and XGBoost (Chen & Guestrin, 2016). Tree-based ensemble models, such as Random Forest and XGBoost, often outperform other classic machine learning models such as LDA or DT in various species (Nathan et al., 2012; Yu et al., 2021). LightGBM and XGBoost were implemented using lightgbm (version 3.3.3), xgboost (version 1.7.1) and scikit-learn (version 1.2.1). XGBoost models were trained on GPUs for fast training. The inputs of these models were 119 handcrafted features extracted from raw data. These features were designed based on previous studies (Fehlmann et al., 2017; Nathan et al., 2012; Yu et al., 2021). These features included the statistics (e.g. mean and variance) of the raw data, static components and dynamic components of each axis. They also included statistics of pitch, roll, ODBA, and main frequencies and their amplitudes. Note that calculation methods for some features are not exactly identical to previous studies. See the source code and list of features (Table S6) for further details. We used the synthetic minority over-sampling technique (SMOTE) (Chawla et al., 2002) to obtain class-balanced training data. The parameters for both models are as follows: number of estimators was 10,000, 10 early stopping rounds and a learning rate of 0.01.

We also performed a grid search experiment to understand how hyperparameters associated with model architectures, such as the number of convolution layers or the number of attention heads, influence the model performance of DCLSA and CNN-AE w/o, using a smaller number of test individuals (same as Experiment 1) and three random seeds. See Supplementary Experiment S2 in the supporting information for more details.

3 | RESULTS

3.1 | Experiment 1: Data augmentation and manifold mixup

We first examined the impact of data augmentation techniques on DCL (Figure 3). For streaked shearwaters, permutation and random data augmentation improved the macro F1-score (Figure 3a). Random data augmentation improved the macro F1-score by an average of 4.7% compared with those without data augmentation. Improvements by random data augmentation were observed in the class F1-scores for stationary, bathing, cruising flight, foraging dive and dipping (Figure 3b), while scaling, permutation, t-warp and rotation decreased the class F1-score for take-off, as did random augmentation which included these four types (Figure 3b). Example t-SNE visualisation of features for streaked shearwaters is shown in Figure 3e,f (for one test individual) and Figure S7 (for six test individuals). Similarly, random data augmentation was effective for DCLSA (Figure S8a,b), improving the macro F1-score by an average of 3.3%, but data augmentation types except for jittering decreased the class F1-score of take-off, as did random data augmentation. For both DCL and DCLSA, rotation had a negative effect on the class F1-score for foraging dive (Figure 3b; Figure S8b), indicating that postural information was critical for this behaviour and may be obscured by rotation.

For black-tailed gulls, rotation and random data augmentation improved the macro F1-score (Figure 3c). Rotation may be useful for learning feature representations that are independent of device attachment positions. This is crucial when the dataset contains data from different attachment positions (e.g. abdomen and back).

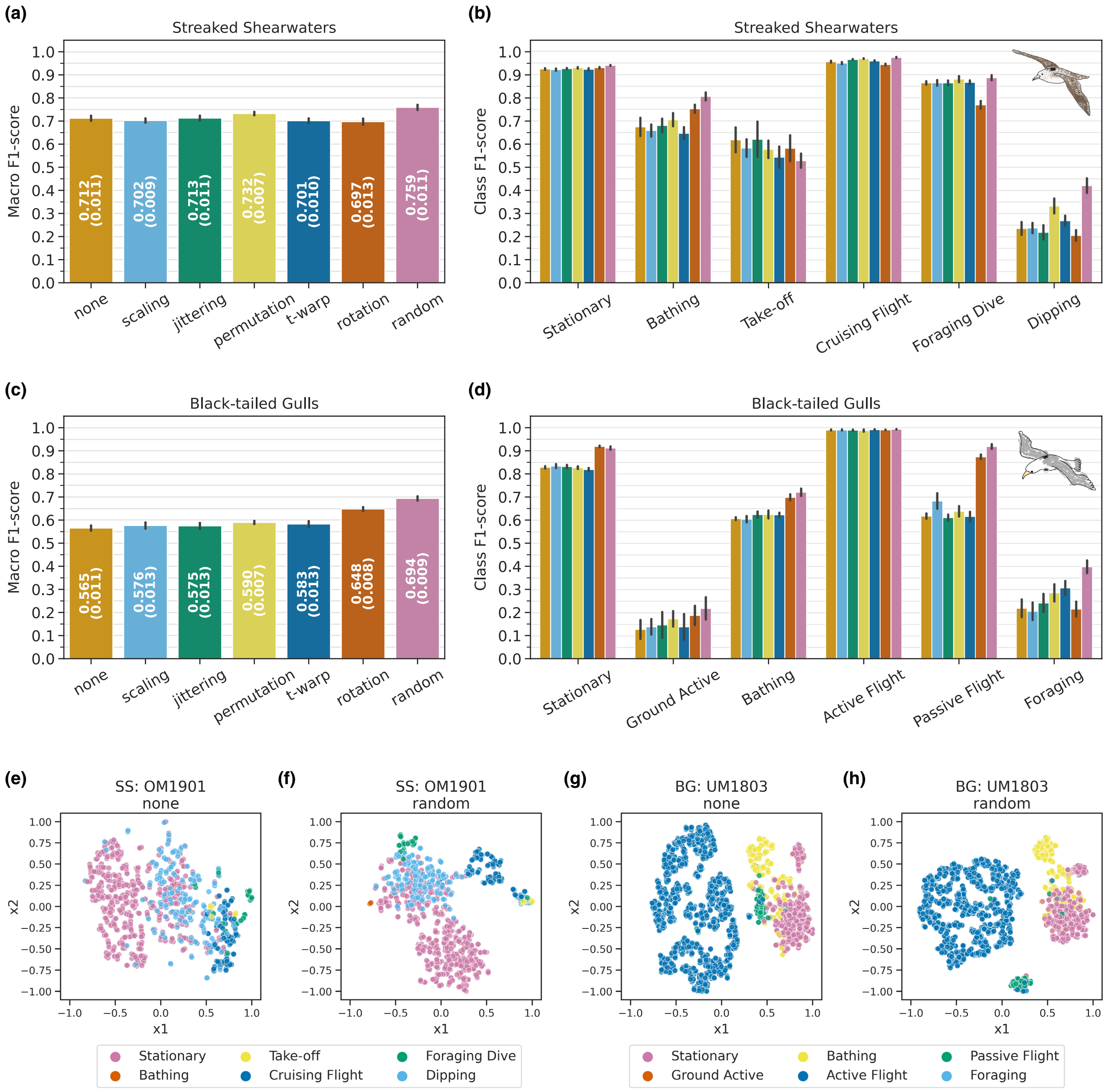

FIGURE 3 Impacts of data augmentation on DeepConvLSTM (DCL). Impacts of data augmentation on DCL for streaked shearwaters (SS) (a, b) and black-tailed gulls (BG) (c, d). A type 'random' indicates a random application of six data augmentation types. Example t-SNE visualisation of features (i.e. features before the output layer) when no or random data augmentation was applied (only when the random seed = 0), for SS (OM1901) (e, f) and BG (UM1803) (g, h). See Figure S7 for example t-SNE visualisation of all test individuals.

Although the impacts of other data augmentation techniques except for rotation were smaller, random data augmentation contributed to the improvement of the macro F1-score by 12.8% in DCL and 12.3% in DCLSA (Figure S8c) on average. Improvements by random data augmentation were observed in the class F1-scores for stationary, ground active, bathing, passive flight and foraging (Figure 3d; Figure S8d). Example t-SNE visualisation of features for black-tailed gulls is shown in Figure 3g,h (for one test individual) and Figure S7 (for six test individuals).

When no data augmentation was applied, DCLSA outperformed DCL by 1.0% and 0.7% in terms of the macro F1-score for streaked shearwaters and black-tailed gulls, respectively. However, DCL achieved performance almost equivalent to or even better than DCLSA when random data augmentation was used (Figure 3; Figure S8).

Experiment S1 showed that data augmentation parameters influence the model performance and the top-ranked parameters were different between the two species except for t-warp and rotation.

In addition, random data augmentation that used the top-ranked parameters slightly outperformed random data augmentation that used the default parameters. The results also showed that there were clear relationships between the parameters of some data augmentation types and the class F1-scores of several specific behaviour classes (e.g. the larger jittering parameters decreased the class F1-score of stationary behaviour). For more detailed results, see Experiment S1.

Figure 4 shows the effect of manifold mixup on DCL. Manifold mixup improved the macro F1-scores by up to 2.5% (mixup alpha = 1.0) and 0.7% (mixup alpha = 0.2) for streaked shearwaters and black-tailed gulls, respectively. However, the combination of manifold mixup and random data augmentation did not further improve the performance. When random data augmentation was combined with manifold mixup, the models outperformed those with manifold mixup alone (Figure 4). These results indicated that the impact of random data augmentation was much higher than that of manifold mixup for our datasets. Manifold mixup after the LSTM layer of DCL also improved the macro F1-scores by up to 2.0% (mixup alpha = 0.1) and 2.3% (mixup alpha = 0.2) for streaked shearwaters and black-tailed gulls, respectively; however, again, the improvements were smaller than those achieved with random data augmentation (Figure S9).

3.2 | Experiment 2: Pre-training of CNN-AE

Pre-training using unlabelled data did not improve model performance for either species. Rather, the condition 'w/o' (CNN-AE was trained with labelled data without pre-training) performed the best, and followed by 'w/', 'w/ soft', 'w/ hard' in decreasing order of performance (Figure 5).

3.3 | Experiment 3: Model comparison

A comparison of the macro and class F1-scores is shown in Figure 6. For streaked shearwaters, CNN, DCL, DCLSA, DCLSA-RN and CNN-AE w/o outperformed LightGBM and XGBoost in terms of the macro F1-score (Figure 6a). For black-tailed gulls, CNN, DCL, DCLSA and DCLSA-RN outperformed LightGBM and XGBoost in terms of the macro F1-score (Figure 6c). As the best model, DCLSA-RN outperformed XGBoost by approximately 4.3% and 1.7% on average

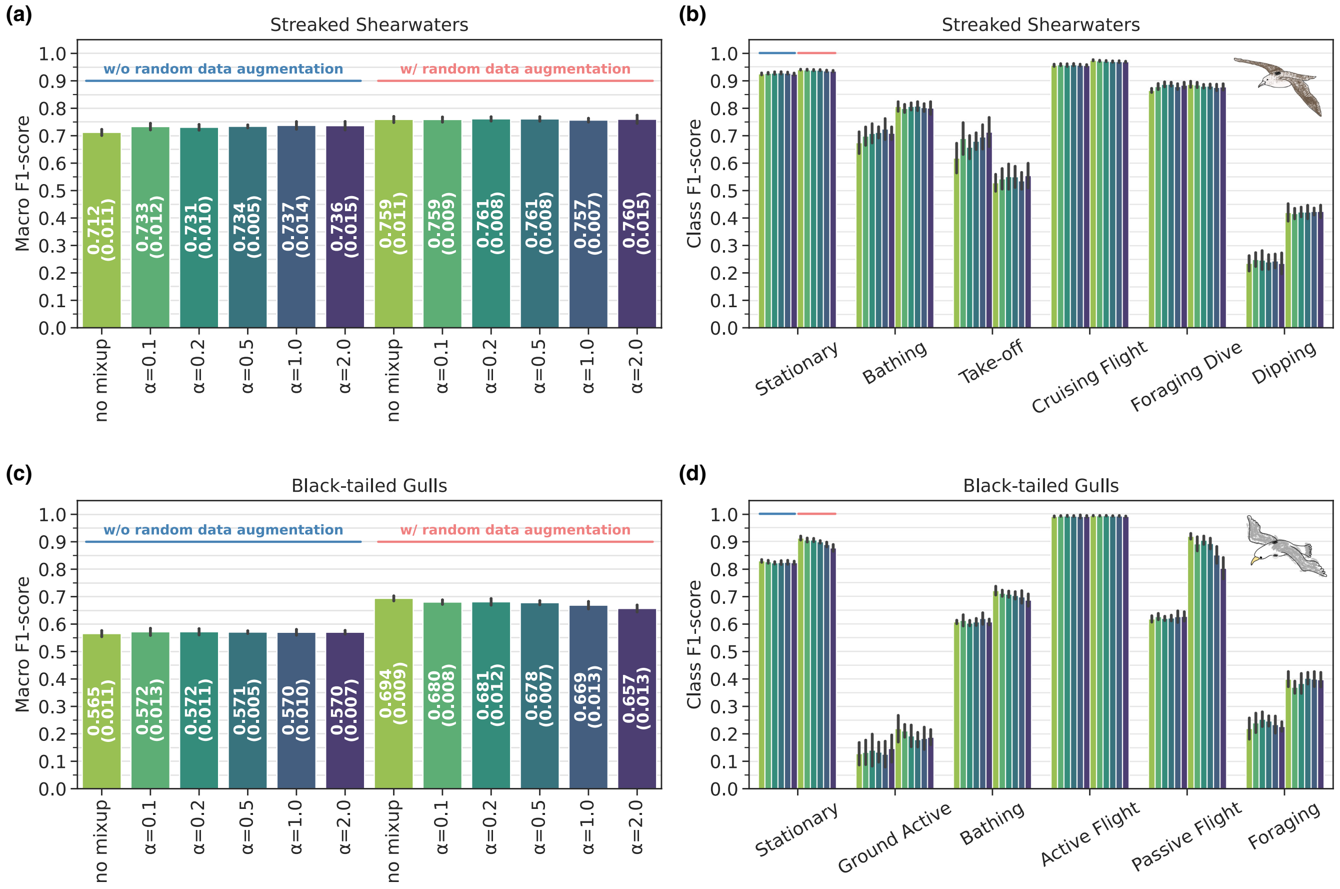

FIGURE 4 Impacts of manifold mixup (no mixup, mixup alpha = 0.1, 0.2, 0.5, 1.0 and 2.0, with and without random data augmentation) on DeepConvLSTM for streaked shearwaters (a, b) and black-tailed gulls (c, d).

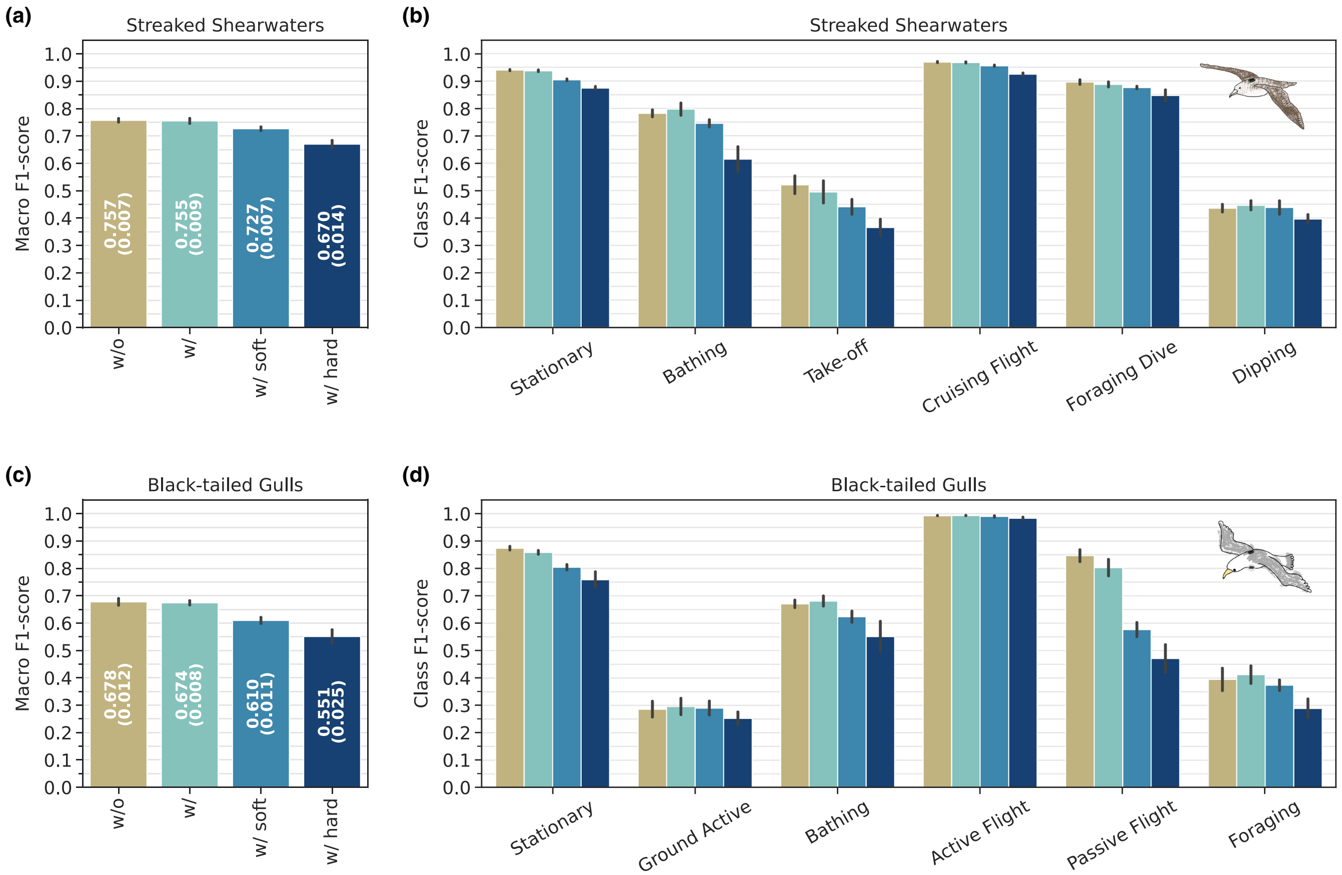

FIGURE 5 Impacts of unsupervised pre-training on CNN-based Autoencoder (CNN-AE) for streaked shearwaters (a, b) and black-tailed gulls (c, d). The following four conditions were compared: 'w/o' (pre-training), 'w/', 'w/ soft' (smaller learning rate for encoder parameters) and 'w/ hard' (with encoder parameters frozen).

in terms of the macro F1-score for streaked shearwaters and black-tailed gulls, respectively.

Looking into each behaviour, the class F1-scores of bathing, foraging dive and dipping for streaked shearwaters, and those of foraging for black-tailed gulls were better in CNN, DCL, DCLSA, DCLSA-RN and CNN-AE w/o than in LightGBM and XGBoost (Figure 6b,d). The confusion matrix of DCLSA-RN for streaked shearwaters (Figure 7a) showed that some cruising flight windows were misclassified as take-off, which reduced the F1-score of take-off. Classifying dipping was the most difficult and dipping windows were often misclassified as stationary windows and vice versa. The confusion matrix of DCLSA-RN for black-tailed gulls (Figure 7b) showed that the classification of ground active and foraging was more difficult than that of the other classes. Ground active windows were often misclassified as stationary windows and vice versa. Foraging windows were misclassified as bathing or stationary windows and vice versa (bathing and stationary windows were also misclassified).

Figures S10 and S11 show comparisons of the confusion matrix of each model for streaked shearwaters and black-tailed gulls, respectively. For the feature importance of XGBoost, see Figure S12. The impact of the number of features and SMOTE on XGBoost is shown in Figures S13 and S14, respectively.

Experiment S2 showed that the better model hyperparameters were not the same across the two species (see Experiment S2 for more results).

4 | DISCUSSION

4.1 | Experiment 1: Data augmentation and manifold mixup

Collecting and labelling large amounts of time-series sensor data is difficult; it is more difficult for humans, domestic animals and wildlife studies, in that order. Data augmentation techniques have been extensively studied for HAR (Um et al., 2017; Wen et al., 2021) and gradually for domestic animals (e.g. Eerdekens et al., 2020; Pan et al., 2023). This study explored and confirmed the effectiveness of data augmentation in wild animal behaviour classification using time-series sensor data.

Experiment 1 indicated that each data augmentation type may have a positive or negative impact on each behaviour, and the impact may also vary depending on architecture; however, applying random data augmentation to each sample during mini-batch

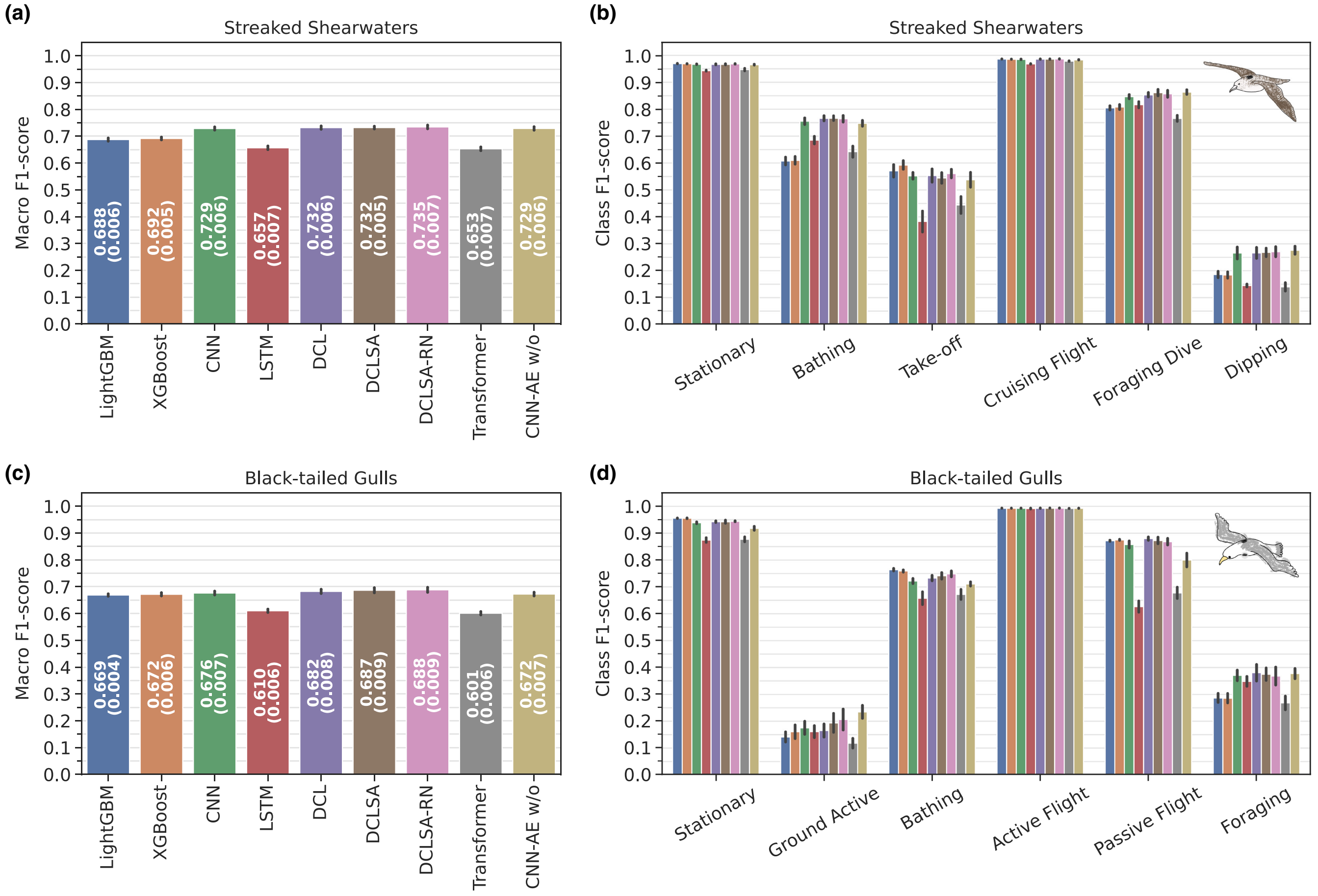

FIGURE 6 Comparison of model performance (mean and standard deviation of macro and class F1-score) for streaked shearwaters (a, b) and black-tailed gulls (c, d).

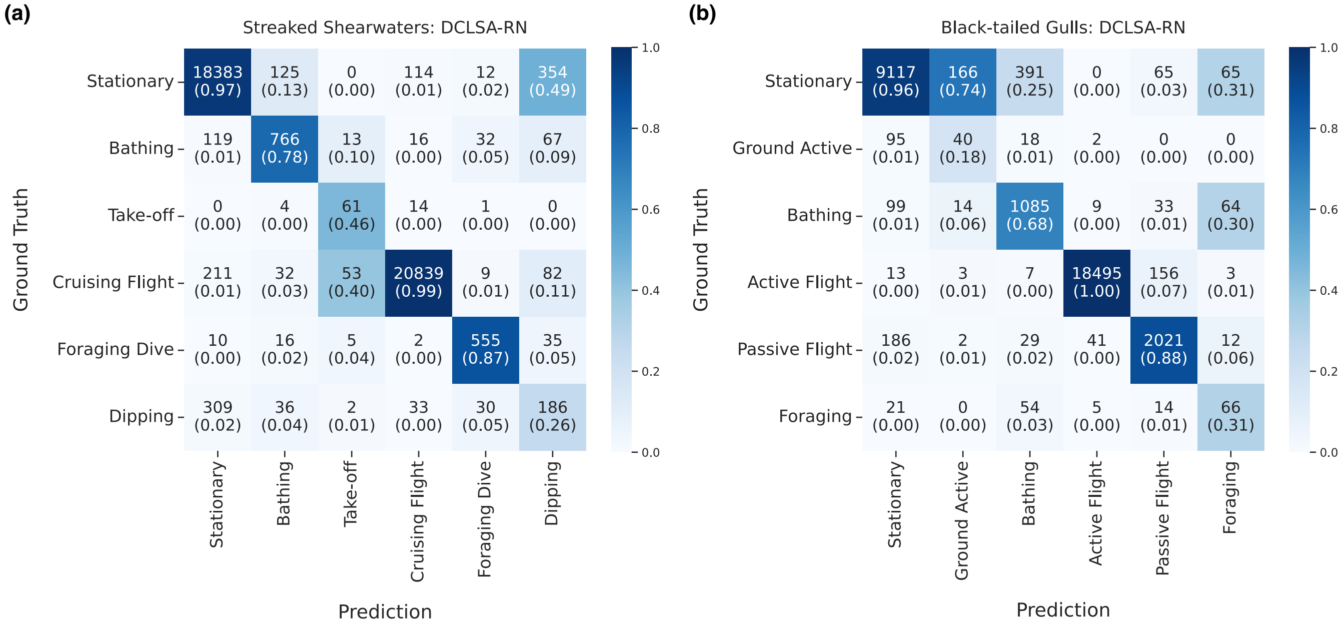

FIGURE 7 Confusion matrix of ResNet version of DeepConvLSTMSelfAttn (DCLSA-RN) for streaked shearwaters (a) and black-tailed gulls (b) when the random seed was 0.

training appears to improve overall performance. Combinations of data augmentation techniques can improve the model performance for HAR (Um et al., 2017). A recent bird-sound recognition study (Lauha et al., 2022) also demonstrated the effectiveness of random combinations of data augmentation techniques, although they were applied to spectrogram images. We believe that random data augmentation is effective against data shortages and imbalance problems in wild animal studies.

In addition to data shortages and class imbalance problems, devices, attachment positions and attachment procedures have an impact on acceleration data in bio-logging studies (Garde et al., 2022). If a classification model is not robust to this noise, it may cause systematic biases that undermine the foundation of the research when biologists or ecologists utilise the models. Similar to the HAR study (Um et al., 2017), Experiment 1 also showed that differences in attachment position could be handled by data augmentation, rotation for black-tailed gulls in particular.

The results of Experiment S1 highlighted the importance of searching the better data augmentation parameters for different datasets, while implying that random data augmentation might be robust to parameter selection. The results also indicated that not only data augmentation types but also their parameter choices may have different effects depending on the nature of target behaviour class. See Experiment S1 for more discussion.

Although the model performance improved by manifold mixup for both species, the overall effects of manifold mixup were smaller than those of random data augmentation. These two techniques were expected to play common roles; however, the random data augmentation was more effective for our dataset, and their combination did not contribute to further improvement. The effects may vary depending on the dataset and model architecture, and manifold mixup is worth trying in different settings.

4.2 | Experiment 2: Pre-training of CNN-AE

Unsupervised pre-training has been generally considered to improve the model performance in image classification (Le Paine et al., 2015). However, some studies have advocated that it does not necessarily improve the generalisation performance of classification models in any case (Alberti et al., 2017; Le Paine et al., 2015). For instance, the effect of pre-training was significant when the ratio of unlabelled to labelled data was large (e.g. 50:1), but the performance was poorer when the ratio was 1:1 (Le Paine et al., 2015). In our case, the amount of our unlabelled data was approximately 36 and 43 times larger than the labelled data for streaked shearwaters and black-tailed gulls, respectively; however, the pre-training of CNN-AE with unlabelled data did not improve performance, rather it degraded performance under some conditions.

One possible reason for this is the extreme imbalance in unlabelled data. Our labelled data were heavily class-imbalanced, but the unlabelled data could be even more imbalanced. This is because labelled data includes data collected by bio-loggers with AI, which can efficiently collect data on target behaviours (Korpela et al., 2020). Besides, we could not use the WeightedRandomSampler of PyTorch in unsupervised pre-training as we did in supervised learning. Therefore, the data in a training batch during pre-training are considered to be extremely imbalanced (e.g. mostly stationary and/or flying). This may also become a major problem when conducting self-supervised learning. In a recent HAR study (Yuan et al., 2022), for example, the acceleration data windows were sampled in proportion to their standard deviation during self-supervised learning. This approach would reduce the frequency of sampling small-amplitude acceleration data, which is prevalent in a large portion of real-world datasets. In our case, for example, reducing the sampling frequency of similar signals (e.g. stationary or flying) that exist in large numbers but are less informative may improve the results.

4.3 | Experiment 3: Model comparison

In Experiment 3, DCL slightly outperformed CNN and clearly outperformed LSTM for both species, indicating that adding an LSTM layer after CNN layers is also effective for wildlife behaviour classification, as shown for human datasets in Ordóñez and Roggen (2016). DCLSA slightly outperformed DCL for black-tailed gulls, which is consistent with Singh et al. (2021), but not for streaked shearwaters. Yet, our data augmentation experiment (Experiment 1) on DCL and DCLSA revealed that adding a multi-head attention layer slightly improved the performance for both species, when no data augmentation was applied (Figure 3; Figure S8). This suggests that, for our datasets, both data augmentation and the additional multi-head attention layer have positive impacts, but the former may have a larger impact. Residual blocks with shortcut connection (He et al., 2016) in the DCLSA-RN may also slightly improve the performance, as shown in Figure 6. Transformer has achieved great success, especially in natural language processing (Vaswani et al., 2017), and is extensively used as the basis for well-known models. Although we used only the encoder block of transformer, it did not achieve a higher performance in this study or when used as a backbone network in contrastive learning in the HAR study (Qian et al., 2022). CNN-AE w/o performed comparably to CNN, probably because the encoder of CNN-AE w/o shares a very similar architecture with CNN, except for the max pooling layers that gradually compress the time dimension.

DCL, DCLSA and DCLSA-RN achieved slightly higher overall performance than the simple CNN, but did not show the great improvement in the class F1-score of complex behaviours, such as foraging of black-tailed gulls, that we had expected. Besides, these three models have more trainable parameters than CNN and the number of parameters is larger in this order (Table S4). When data augmentation is effective (e.g. to the extent that the performance difference between DCL and DCLSA almost disappears), simpler models may be a better choice in terms of the balance between the performance and training time and/or computational cost.

Experiment S2 suggested the importance of performing model hyperparameter tuning for different datasets. However, hyperparameter tuning requires enormous time and computational resources, and see Experiment S2 for more discussion on this point.

Using only raw triaxial acceleration data as inputs, deep learning architectures, such as CNN, DCL, DCLSA and DCLSA-RN, outperformed LightGBM and XGBoost, which used 119 handcrafted features. Note that our feature list covers most features used in the previous studies we referenced, and the number of features is larger than those previous studies (e.g. 38 features in Nathan et al., 2012; 25 in Fehlmann et al., 2017; 78 in Yu et al., 2021). We also used SMOTE, which improved the macro F1-scores (Figure S14). The classic machine learning approach usually requires feature engineering, which often requires specialised knowledge and time. Our results indicate that deep learning may enable end-to-end classification of wildlife behaviour using time-series sensor data.

It should be noted that simply comparing the F1-scores in this study with those of previous studies is meaningless. This is because the target species, number and types of behaviours, data amount, evaluation methods, etc., have an impact on performance metrics. If the target behaviours are basic, such as stationary, walking and running, the macro F1-score tends to be higher, even with a naive approach. In general, the greater the number of target behaviour classes and the greater the degree of class imbalance, the lower the macro F1-score would be. Regarding evaluation methods, some may use only the train/test or train/validation split (i.e. two datasets) rather than the train/validation/test split (i.e. three datasets); the former may tend to return a higher accuracy or F1-score if test or validation data are also used during training (e.g. for early stopping). More importantly, if one randomly splits the time-series sensor data into training, validation and test data (e.g. a 7:2:1 random split), these three datasets will include data segments from the same individuals or the same behavioural sequences. To avoid the above problems, we recommend using LOIO-CV, which is stricter and more robust and thus tends to produce lower scores than the above evaluation methods. However, note that we calculated F1-scores by aggregating the prediction results of all the folds because calculating an F1-score for each individual and behaviour class is not realistic when only a few individuals have completed sets of all target classes.

4.4 | Future directions

Finally, we discuss interesting future directions for the behaviour classification of wild animals using time-series sensor data. Although data augmentation is promising, searching for optimal data augmentation techniques and/or their combinations and parameters is time-consuming and requires considerable computational resources (see also Discussion of Experiment S1). Developing a method specifically for wildlife that automatically finds the optimal data augmentation techniques and their parameters would be interesting, as would other data augmentation approaches such as deep generative models (see Cubuk et al., 2020; Wen et al., 2021). Domain adaptation techniques such as domain adversarial neural networks (Ganin et al., 2016) can be explored to further reduce F1-score variations between individuals. The development of a new model architecture for more specific tasks (Xia et al., 2022; Yoshimura, Maekawa, et al., 2022), the use of a specific loss function to deal with class imbalance (e.g. Park et al., 2021) and the use of multimodal sensor data (e.g. acceleration, gyroscope, magnetometer, GPS and depth) are also exciting approaches.

We trained our deep learning models using relatively large datasets; however, such a situation may be rare in wild animal research. In addition, labelling enormous amounts of sensor data is labour intensive and time-consuming. In data-scarce scenarios, transfer learning and self-supervised learning may be promising, in addition to data augmentation. For example, in transfer learning, a model can be pre-trained on a large dataset of different individuals from different study sites or different but similar species and fine-tuned on the target data. Self-supervised learning, such as contrastive learning (Chen et al., 2020; Qian et al., 2022), uses unlabelled data to train the feature extractor, and then, the classifier or whole network can be fine-tuned with fewer labelled data. Contrastive learning such as SimCLR with ResNet-50 as the backbone network has succeeded in image classification task (Chen et al., 2020) and an exploratory study on contrastive learning has already been conducted in HAR using acceleration data (Qian et al., 2022). These approaches have the potential to be not only effective against data shortages and class imbalance problems but also robust against various types of noise. If established, researchers can easily use behaviour classification techniques for various animals without much effort to collect and label the data.

5 | CONCLUSIONS

Acceleration-based behaviour classification using deep learning models has only been extensively studied in humans and domestic animals and has rarely been applied to wildlife research. Challenges include data shortages, class-imbalanced problems, various types of noise due to differences in individual behaviour and where the loggers were attached, and complexity in acceleration data due to difficult animal-specific behaviours. This study explored the effectiveness of data augmentation and manifold mixup, pre-training of CNN-AE with unlabelled data and state-of-the-art deep learning model architectures to overcome these challenges. We demonstrated that data augmentation is effective and that deep learning models such as DCL, DCLSA and DCLSA-RN are promising for wildlife behaviour classification using time-series sensor data. We believe that deep learning approaches have great potential for development, and we discussed their future directions. These include more advanced approaches for data augmentation, domain adaptation, model architectures and loss functions development, the use of multimodal sensor data, transfer learning and self-supervised learning. We hope that this study will fill the gap between acceleration-based behaviour classification studies of wild animals and humans or domestic animals and stimulate the development of deep learning techniques in behaviour classification using time-series sensor data for wild animals.

AUTHOR CONTRIBUTIONS

Ryoma Otsuka performed the method design, data collection, software implementation and paper writing. Naoya Yoshimura and Kei Tanigaki helped Ryoma Otsuka with the software implementation and data collection. Shiho Koyama and Yuichi Mizutani performed fieldwork and data collection and helped with labelling. Ken Yoda performed data collection and paper writing. Takuya Maekawa directed the study and performed the method design, data collection and paper writing.

ACKNOWLEDGEMENTS

We thank the Hachinohe City mayor, the Aomori Prefectural Government and the Ministry of the Environment, Japan, for providing permission to collect data. We thank the people on Awashima Island and Kabushima Island, Japan, for their help during the fieldwork. We thank Qingxin Xia, Rikuto Tsubouchi, Takuma Yamashita and Wang Yuqiao, for helpful suggestions and comments regarding this study. We thank Kana Yasuda for illustrating the streaked shearwater and the black-tailed gull in the figures. We thank the anonymous reviewers for their valuable and constructive comments.

FUNDING INFORMATION

This work was supported by JSPS KAKENHI Grant Numbers JP21H05293 and JP21H05299.

CONFLICT OF INTEREST STATEMENT

The authors declare no competing interests.

DATA AVAILABILITY STATEMENT

Data are available via the Dryad Digital Repository https://doi.org/10.5061/dryad.2ngf1vhwk (Otsuka et al., 2024). The source code used in this study is available from https://github.com/ryoma-otsuka/dl-wabc.

ORCID

Ryoma Otsuka https://orcid.org/0000-0002-5147-1916
Naoya Yoshimura https://orcid.org/0000-0003-3017-8873
Shiho Koyama https://orcid.org/0000-0003-0801-5963
Yuichi Mizutani https://orcid.org/0000-0002-8521-8759
Ken Yoda https://orcid.org/0000-0002-8346-3291
Takuya Maekawa https://orcid.org/0000-0002-7227-580X

REFERENCES

Alberti, M., Seuret, M., Ingold, R., & Liwicki, M. (2017). A pitfall of unsupervised pre-training. *arXiv preprint* arXiv:1703.04332v4 [cs.CV]. http://arxiv.org/abs/1703.04332

Browning, E., Bolton, M., Owen, E., Shoji, A., Guilford, T., & Freeman, R. (2018). Predicting animal behaviour using deep learning: GPS data alone accurately predict diving in seabirds. *Methods in Ecology and Evolution*, *9*(3), 681–692. https://doi.org/10.1111/2041-210x.12926

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *The Journal of Artificial Intelligence Research*, *16*(1), 321–357. https://doi.org/10.1613/jair.953

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2016)* (pp. 785–794). https://doi.org/10.1145/2939672.2939785

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th international conference on machine learning* (Vol. *119*, pp. 1597–1607). PMLR. https://proceedings.mlr.press/v119/chen20j.html

Cubuk, E. D., Zoph, B., Shlens, J., & Le, Q. V. (2020). Randaugment: Practical automated data augmentation with a reduced search space. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (pp. 3008–3017). https://doi.org/10.1109/cvprw50498.2020.00359

Eerdekens, A., Deruyck, M., Fontaine, J., Martens, L., De Poorter, E., 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)* (pp. 1–6). https://doi.org/10.1109/COINS49042.2020.9191639

Fehlmann, G., & King, A. J. (2016). Bio-logging. *Current Biology*, *26*(18), R830–R831. https://doi.org/10.1016/j.cub.2016.05.033

Fehlmann, G., O'Riain, M. J., Hopkins, P. W., O'Sullivan, J., Holton, M. D., Shepard, E. L. C., & King, A. J. (2017). Identification of behaviours from accelerometer data in a wild social primate. *Animal Biotelemetry*, *5*, 6. https://doi.org/10.1186/s40317-017-0121-3

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., & Lempitsky, V. (2016). Domain-adversarial training of neural networks. *Journal of Machine Learning Research: JMLR*, *17*(59), 1–35. https://jmlr.org/papers/v17/15-239.html

Garde, B., Wilson, R. P., Fell, A., Cole, N., Tatayah, V., Holton, M. D., Rose, K. A. R., Metcalfe, R. S., Robotka, H., Wikelski, M., Tremblay, F., Whelan, S., Elliott, K. H., & Shepard, E. L. C. (2022). Ecological inference using data from accelerometers needs careful protocols. *Methods in Ecology and Evolution*, *13*(4), 813–825. https://doi.org/10.1111/2041-210x.13804

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)* (pp. 770–778). https://doi.org/10.1109/CVPR.2016.90

Hoffman, B., Cusimano, M., Baglione, V., Canestrari, D., Chevallier, D., DeSantis, D. L., Jeantet, L., Ladds, M. A., Maekawa, T., Mata-Silva, V., Moreno-González, V., Trapote, E., Vainio, O., Vehkaoja, A., Yoda, K., Zacarian, K., Friedlaender, A., & Rutz, C. (2023). *A benchmark for computational analysis of animal behavior, using animal-borne tags*. *arXiv preprint* arXiv:2305.10740 [cs.LG]. http://arxiv.org/abs/2305.10740

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)* (pp. 3149–3157). https://proceedings.neurips.cc/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html

Korpela, J., Suzuki, H., Matsumoto, S., Mizutani, Y., Samejima, M., Maekawa, T., Nakai, J., & Yoda, K. (2020). Machine learning enables improved runtime and precision for bio-loggers on seabirds. *Communications Biology*, *3*(1), 633. https://doi.org/10.1038/s42003-020-01356-8

Lauha, P., Somervuo, P., Lehikoinen, P., Geres, L., Richter, T., Seibold, S., & Ovaskainen, O. (2022). Domain-specific neural networks improve automated bird sound recognition already with small amount of local data. *Methods in Ecology and Evolution*, *13*(12), 2799–2810. https://doi.org/10.1111/2041-210x.14003

Le Paine, T., Khorrami, P., Han, W., & Huang, T. S. (2015). *An analysis of unsupervised pre-training in light of recent advances*. *arXiv preprint* arXiv:1412.6597v4 [cs.CV]. https://doi.org/10.48550/arXiv.1412.6597

Leos-Barajas, V., Photopoulou, T., Langrock, R., Patterson, T. A., Watanabe, Y. Y., Murgatroyd, M., & Papastamatiou, Y. P. (2017). Analysis of animal accelerometer data using hidden Markov models. *Methods in Ecology and Evolution*, *8*(2), 161–173. https://doi.org/10.1111/2041-210x.12657

Nathan, R., Spiegel, O., Fortmann-Roe, S., Harel, R., Wikelski, M., & Getz, W. M. (2012). Using tri-axial acceleration data to identify behavioral modes of free-ranging animals: General concepts and tools illustrated for griffon vultures. *Journal of Experimental Biology*, *215*(6), 986–996. https://doi.org/10.1242/jeb.058602

Ordóñez, F. J., & Roggen, D. (2016). Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. *Sensors*, *16*(1), 115. https://doi.org/10.3390/s16010115

Otsuka, R., Yoshimura, N., Tanigaki, K., Koyama, S., Mizutani, Y., Yoda, K., & Maekawa, T. (2024). Data from: Exploring deep learning techniques for wild animal behaviour classification using animal-borne accelerometers. *Dryad Digital Repository*, https://doi.org/10.5061/dryad.2ngf1vhwk

Pan, Z., Chen, H., Zhong, W., Wang, A., & Zheng, C. (2023). A CNN-based animal behavior recognition algorithm for wearable devices. *IEEE Sensors Journal*, *23*(5), 5156–5164. https://doi.org/10.1109/JSEN.2023.3239015

Park, S., Lim, J., Jeon, Y., & Choi, J. Y. (2021). Influence-balanced loss for imbalanced visual classification. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV 2021)* (pp. 715–724). https://doi.org/10.1109/iccv48922.2021.00077

Qian, H., Tian, T., & Miao, C. (2022). What makes good contrastive learning on small-scale wearable-based tasks? In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2022)* (pp. 3761–3771). https://doi.org/10.1145/3534678.3539134

Roy, A., Lanco Bertrand, S., & Fablet, R. (2022). Deep inference of seabird dives from GPS-only records: Performance and generalization properties. *PLoS Computational Biology*, *18*(3), e1009890. https://doi.org/10.1371/journal.pcbi.1009890

Singh, S. P., Sharma, M. K., Lay-Ekuakille, A., Gangwar, D., & Gupta, S. (2021). Deep ConvLSTM with self-attention for human activity decoding using wearable sensors. *IEEE Sensors Journal*, *21*(6), 8575–8582. https://doi.org/10.1109/JSEN.2020.3045135

Sur, M., Suffredini, T., Wessells, S. M., Bloom, P. H., Lanzone, M., Blackshire, S., Sridhar, S., & Katzner, T. (2017). Improved supervised classification of accelerometry data to distinguish behaviors of soaring birds. *PLoS ONE*, *12*(4), e0174785. https://doi.org/10.1371/journal.pone.0174785

Tanigaki, K., Otsuka, R., Li, A., Hatano, Y., Wei, Y., Koyama, S., Yoda, K., & Maekawa, T. (2024). Automatic recording of rare behaviors of wild animals using video bio-loggers with on-board light-weight outlier detector. *PNAS Nexus*, *3*(1), gad447. https://doi.org/10.1093/pnasnexus/pgad447

Um, T. T., Pfister, F. M. J., Pichler, D., Endo, S., Lang, M., Hirche, S., Fietzek, U., & Kulić, D. (2017). Data augmentation of wearable sensor data for Parkinson's disease monitoring using convolutional neural networks. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction (ICMI 2017)* (pp. 216–220). https://doi.org/10.1145/3136755.3136817

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS 2017)*. https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Lopez-Paz, D., & Bengio, Y. (2019). Manifold mixup: Better representations by interpolating hidden states. In *Proceedings of the 36th International Conference on Machine Learning* (Vol. *97*, p. 6438, 6447). PMLR. https://proceedings.mlr.press/v97/verma19a.html

Watanabe, Y. Y., & Takahashi, A. (2013). Linking animal-borne video to accelerometers reveals prey capture variability. *Proceedings of the National Academy of Sciences of the United States of America*, *110*(6), 2199–2204. https://doi.org/10.1073/pnas.1216244110

Wen, Q., Sun, L., Yang, F., Song, X., Gao, J., Wang, X., & Xu, H. (2021). Time series data augmentation for deep learning: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI 2021)* (pp. 4653–4660). https://doi.org/10.24963/ijcai.2021/631

Xia, Q., Wada, A., Yoshii, T., Namioka, Y., & Maekawa, T. (2022). Comparative analysis of high- and low-performing factory workers with attention-based neural networks. In *Mobile and ubiquitous systems: Computing, networking and services* (pp. 469–480). https://doi.org/10.1007/978-3-030-94822-1_26

Yoda, K. (2019). Advances in bio-logging techniques and their application to study navigation in wild seabirds. *Advanced Robotics*, *33*(3–4), 108–117. https://doi.org/10.1080/01691864.2018.1553686

Yoda, K., Naito, Y., Sato, K., Takahashi, A., Nishikawa, J., Ropert-Coudert, Y., Kurita, M., & Le Maho, Y. (2001). A new technique for monitoring the behaviour of free-ranging Adélie penguins. *Journal of Experimental Biology*, *204*(4), 685–690. https://doi.org/10.1242/jeb.204.4.685

Yoda, K., Sato, K., Niizuma, Y., Kurita, M., Bost, C.-A., Le Maho, Y., & Naito, Y. (1999). Precise monitoring of porpoising behaviour of Adélie penguins determined using acceleration data loggers. *Journal of Experimental Biology*, *202*(22), 3121–3126. https://doi.org/10.1242/jeb.202.22.3121

Yoshimura, N., Maekawa, T., Hara, T., Wada, A., & Namioka, Y. (2022). Acceleration-based activity recognition of repetitive works with lightweight ordered-work segmentation network. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, *6*(2), 1–39. https://doi.org/10.1145/3534572

Yoshimura, N., Morales, J., Maekawa, T., & Hara, T. (2022). OpenPack: A large-scale dataset for recognizing packaging works in IoT-enabled logistic environments. *arXiv preprint* arXiv:2212.11152v1 [cs.CV]. http://arxiv.org/abs/2212.11152

Yu, H., Deng, J., Nathan, R., Kröschel, M., Pekarsky, S., Li, G., & Klaassen, M. (2021). An evaluation of machine learning classifiers for next-generation, continuous-ethogram smart trackers. *Movement Ecology*, *9*(1), 15. https://doi.org/10.1186/s40462-021-00245-x

Yuan, H., Chan, S., Creagh, A. P., Tong, C., Clifton, D. A., & Doherty, A. (2022). Self-supervised learning for human activity recognition using 700,000 person-days of wearable data. *arXiv preprint* arXiv:2206.02909 [eess.SP]. http://arxiv.org/abs/2206.02909

Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2018). Mixup: Beyond empirical risk minimization. *International Conference on Learning Representations*. https://openreview.net/forum?id=r1Ddp1-Rb

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

Table S1: Description on technical terms used in this paper.

Table S2: Summary of the datasets.

Table S3: Description of target behaviours of streaked shearwaters (SS) and black-tailed gulls (BG).

Table S4: Number of parameters of deep learning models.

Table S5: A list of GPU nodes in the GPU cluster used for experiments in this study.

Table S6: A list of 119 features used in this study for LightGBM and XGBoost.

Figure S1: Behaviour class label distribution for streaked shearwaters and black-tailed gulls.

Figure S2: Behaviour class label count by individual for streaked shearwaters.

Figure S3: Behaviour class label count by individual for black-tailed gulls.

Figure S4: Visualisation of typical windows of six behaviour classes for streaked shearwaters.

Figure S5: Visualisation of typical windows of six behaviour classes for black-tailed gulls.

Figure S6: Distributions of lambda values sampled from Beta distribution with different mixup alpha values (0.1, 0.2, 0.5, 1.0 and 2.0).

Figure S7: Example t-SNE visualisation of features (i.e. features before the output layer) when no or random data augmentation was applied (only when the random seed = 0) during the training of DeepConvLSTM (DCL) models, for streaked shearwaters (SS) and black-tailed gulls (BG).

Figure S8: Impacts of data augmentation on DeepConvLSTMSelfAttn (DCLSA) models for streaked shearwaters (a, b) and black-tailed gulls (c, d).

Figure S9: Impacts of manifold mixup after the LSTM layer of DeepConvLSTM (DCL) (no mixup, mixup alpha = 0.1, 0.2, 0.5, 1.0 and 2.0, without data augmentation) for streaked shearwaters (a, b) and black-tailed gulls (c, d).

Figure S10: A comparison of confusion matrix of (a) LightGBM, (b) XGBoost, (c) CNN, (d) LSTM, (e) DeepConvLSTM (DCL), (f) DeepConvLSTMSelfAttn (DCLSA), (g) ResNet version of DeepConvLSTMSelfAttn (DCLSA-RN), (h) Transformer and (i) CNN-AE w/o pretraining, for streaked shearwaters.

Figure S11: A comparison of confusion matrix of (a) LightGBM, (b) XGBoost, (c) CNN, (d) LSTM, (e) DeepConvLSTM (DCL), (f) DeepConvLSTMSelfAttn (DCLSA), (g) ResNet version of DeepConvLSTMSelfAttn (DCLSA-RN), (h) Transformer and (i) CNN-AE w/o pretraining, for black-tailed gulls.

Figure S12: Feature importance of top 30 features in XGBoost for streaked shearwaters (a) and black-tailed gulls (b).

Figure S13: Comparison of performance when different numbers of handcrafted features were given as inputs (25, 78 and 119) to XGBoost for streaked shearwaters (a, b) and black-tailed gulls (c, d).

Figure S14: Impacts of Synthetic Minority Over-sampling Technique (SMOTE) on XGBoost with 119 features as inputs for streaked shearwaters (a, b) and black-tailed gulls (c, d).

Table ExS1-1: Impacts of data augmentation (DA) parameters on the macro F1-scores of DCL for streaked shearwaters.

Table ExS1-2: Impacts of data augmentation (DA) parameters on the macro F1-scores of DCL for black-tailed gulls.

Figure ExS1-1: Impacts of scaling parameters (0.1, 0.2, 0.4 and 0.8) on the macro and class F1-scores of DeepConvLSTM for streaked shearwaters (a, b) and black-tailed gulls (c, d).

Figure ExS1-2: Impacts of jittering parameters (0.05, 0.1, 0.2 and 0.3) on the macro and class F1-scores of DeepConvLSTM for streaked shearwaters (a, b) and black-tailed gulls (c, d).

Figure ExS1-3: Impacts of permutation parameters (5, 10 and 15) on the macro and class F1-scores of DeepConvLSTM for streaked shearwaters (a, b) and black-tailed gulls (c, d).

Figure ExS1-4: Impacts of t-warp parameters (0.1, 0.2, 0.4 and 0.8) on the macro and class F1-scores of DeepConvLSTM for streaked shearwaters (a, b) and black-tailed gulls (c, d).

Figure ExS1-5: Impacts of rotation parameters (45, 90 and 180) on the macro and class F1-scores of DeepConvLSTM for streaked shearwaters (a, b) and black-tailed gulls (c, d).

Figure ExS1-6: Streaked shearwaters' individual differences in the mean of the maximum difference for each axis for all windows of each behaviour class.

Figure ExS1-7: Black-tailed gulls' individual differences in the mean of the maximum difference for each axis for all windows of each behaviour class.

Table ExS2-1: Impacts of hyperparameters on the macro F1-scores of DCLSA for streaked shearwaters.

Table ExS2-2: Impacts of hyperparameters on the macro F1-scores of DCLSA for black-tailed gulls.

Table ExS2-3: Impacts of hyperparameters on the macro F1-scores of CNN-AE w/o for streaked shearwaters.

Table ExS2-4: Impacts of hyperparameters on the macro F1-scores of CNN-AE w/o for black-tailed gulls.

How to cite this article: Otsuka, R., Yoshimura, N., Tanigaki, K., Koyama, S., Mizutani, Y., Yoda, K., & Maekawa, T. (2024). Exploring deep learning techniques for wild animal behaviour classification using animal-borne accelerometers. *Methods in Ecology and Evolution*, *15*, 716–731. https://doi.org/10.1111/2041-210X.14294