# Leveraging machine learning and accelerometry to classify animal behaviours with uncertainty

**Medha Agarwal**[*†]
Department of Statistics
University of Washington
Seattle WA, United States
medhaaga@uw.edu

**Kasim Rafiq**[†]
Department of Biology
Center for Ecosystem Sentinels
University of Washington
Seattle WA, United States
Botswana Predator Conservation
Wild Entrust, Maun, Botswana
kasim.rafiq@hotmail.co.uk

**Ronak Mehta**
Department of Statistics
University of Washington
Seattle WA, United States
ronakdm@uw.edu

**Briana Abrahms**[‡]
Department of Biology
Center for Ecosystem Sentinels
University of Washington
Seattle WA, United States
Botswana Predator Conservation
Wild Entrust, Maun, Botswana
abrahms@uw.edu

**Zaid Harchaoui**[‡]
Department of Statistics
University of Washington
Seattle WA, United States
zaid@uw.edu

December 29, 2024

**Abstract**

1. Animal-worn sensors have revolutionised the study of animal behaviour and ecology. Accelerometers, which measure changes in acceleration across planes of movement, are increasingly being used in conjunction with machine learning models to classify animal behaviours across taxa and research questions. However, the widespread adoption of these methods faces challenges from imbalanced training data, unquantified uncertainties in model outputs, shifts in model performance across contexts, and noisy classifications.
2. To address these challenges, we introduce an open-source approach for classifying animal behaviour from raw acceleration data. Our approach integrates machine learning and statistical inference techniques to evaluate and mitigate class imbalances, changes in model performance across ecological settings, and noisy classifications. Importantly, we

[*]Corresponding author
[†]Co-first authors
[‡]Co-senior authors

extend predictions from single behaviour classifications to prediction sets –collections of probable behaviours with a user-specified likelihood of containing the true behaviour – in a framework analogous to the use of confidence intervals in statistical analyses.

3. We highlight the utility of our approach using data collected from a free-ranging large carnivore, African wild dogs (*Lycaon pictus),* in the Okavango Delta, Botswana. We demonstrate significantly improved predictions along with associated uncertainty metrics in African wild dog behaviour classification, particularly for rare and ecologically important behaviours such as feeding, where correct classifications more than doubled following quality checks and data rebalancing introduced in our pipeline.
4. Our approach is applicable across taxa and represents a key step towards advancing the burgeoning use of machine learning to remotely observe around-the-clock behaviours of free-ranging animals. Future work could include the integration of multiple data streams, such as accelerometer, audio, and GPS data, for model training and could be incorporated directly into our pipeline.

Keywords: *accelerometer, behaviour classification, bio-logging, conformal prediction, convolution neural networks, machine learning*

# 1 Introduction

Animal-worn sensors have been pivotal in accelerating our understanding of animal ecology, providing a diversity of data that have advanced fundamental ecological theory and informed conservation actions (Snape et al., 2018; Nickel et al., 2021; VonBank et al., 2023; West et al., 2024). From their origins as tools primarily used to track animal locations and movements, animal-worn sensors have evolved to encompass a wide range of devices capable of monitoring the environments, behaviours, and internal states of animals (Wilmers et al., 2015). Animal-worn accelerometers – sensors that measure changes in acceleration across planes of movement – have been used to estimate energetic expenditure and infer animal behaviours for a wide variety of study systems and questions (see Halsey et al. (2011); Fehlmann et al. (2017)). From enabling the detection of spawning behaviours in large pelagic fish in the open ocean to characterising the hunting and energetics of elusive terrestrial predators (Clarke et al., 2021; Wang et al., 2015), accelerometers have become a valuable tool in ecology and have greatly expanded our capacity to understand ecological phenomena across spatial and temporal scales that were previously unattainable through direct observations alone (Studd et al., 2021).

The utility of accelerometers to capture animal behaviours lies in their ability to capture distinct waveform patterns that correspond to specific movements or postures distinct to different behaviours (Brown et al., 2013). However, in contrast to the direct measurements provided by other sensor modalities, such as GPS or temperature sensors, the relatively abstract nature of accelerometer data can make the interpretation of waveforms challenging. As such, identifying behaviours with accelerometer data often requires pairing raw accelerometer data with known behaviours to create labelled datasets that can be used to learn the specific waveform patterns underlying different behaviours of interest (Brown et al., 2013). Due to the vast volumes of accelerometer data that can be collected and the subtle distinctions between behaviour signatures, manually detecting different behaviours within unseen accelerometer data can be challenging. To overcome this, machine-learning techniques are increasingly being leveraged to learn the accelerometer patterns underlying different behaviours from labelled datasets (Chakravarty et al., 2019; Garde et al., 2021; Otsuka et al., 2024).

Machine learning models for classifying animal behaviour encompass a diversity of techniques from classical machine learning classifiers, such as support vector machines (Martiskainen

et al., 2009) and random forests (Lush et al., 2016), to more sophisticated models for sequential data, including convolutional neural networks, long short-term memory networks, and transformers (Otsuka et al., 2024). Yet current applications of these techniques in ecology have shared several challenges and limitations that can impact model performance and interpretability, including (1) imbalances in the volume of labelled data available for training models (i.e., class imbalance), (2) a lack of methods for statistically quantifying uncertainty in behaviour classifications, (3) shifts in model performance across application contexts (i.e., distribution shifts), and (4) rapidly fluctuating (and ecologically unlikely) classifications of consecutive behaviour segments. Below we detail each of these challenges.

(1) **Class imbalance.** Machine learning models for behaviour classification often require large volumes of labelled accelerometer data for each behaviour of interest for model training. Class imbalance — the unequal distribution of training data between the behaviours of interest — is pervasive in ecological datasets, where the frequency of common or easy-to-observe behaviours, such as resting or moving, can outweigh rarer behaviours, such as feeding or mating (e.g., Otsuka et al. (2024); Clermont et al. (2021)), and can bias model outputs (Johnson and Khoshgoftaar, 2019). However, existing approaches for addressing class imbalances typically aim for equal distributions across behaviour classes through oversampling the least represented behaviours (minority classes) or undersampling the most represented behaviours (majority classes), which can lead to biases in model outputs as a result of excessive resampling Haixiang et al. (2017).

(2) **Uncertainty quantification.** Most machine learning models provide single-label behaviour classifications without quantifying the uncertainties associated with the prediction (Resheff et al., 2014; Nathan et al., 2012). This lack of transparency limits our ability to assess a model's limits and its applicability across datasets. Without clear uncertainty metrics indicating when and where model performance degrades, models can produce misleading results, particularly when applied to new datasets.

(3) **Distribution shifts.** Model performance can also decline due to distribution shifts, where the characteristics of the training dataset differ from those of the broader dataset used for implementation (Kulinski and Inouye, 2023). For example, a model trained on one set of individuals within a population may perform poorly on others due to subtle behavioural differences (Koh et al., 2021). Similarly, temporal changes, such as inter-annual behavioural variations, can degrade model performance when training and prediction periods differ (Ellington et al., 2020). Such distribution shifts are likely widespread within noisy ecological data and further underscore the importance of evaluation to detect and mitigate potential performance drops.

(4) **Temporal context.** Finally, many (though not all) machine learning models classify behaviours based on isolated segments of accelerometer data, often ignoring the continuity or context of preceding behaviours (Riaboff et al., 2022). This lack of temporal context can lead to ecologically implausible behaviour sequences that rapidly fluctuate between distinct behaviours, undermining the biological realism of the classifications. Considering the accelerometry patterns before or after a specific segment of data can help improve prediction performance for that segment. Indeed, in other applications, models that leverage temporal context, such as convolutional neural networks (Wang et al., 2017) and transformers (Zerveas et al., 2021), have become standard in machine learning for time series analysis tasks.

In this paper, we introduce an open-source approach for classifying animal behaviour using raw acceleration data that combines machine learning and statistical inference techniques to address class imbalances, uncertainty quantification, distribution shifts, and temporal dependencies. We start by introducing the concept of creating labelled training datasets

and introduce a flexible rebalancing method that allows users to flexibly tune and evaluate the extent of resampling for their datasets. We then introduce a novel machine learning architecture for uncertainty quantification, wherein we train a convolutional neural network and a conformal model for predicting animal behaviour with explicitly quantified uncertainties. Next, we describe how evaluation setups can be created to assess the presence and mitigate the impacts of distributional shifts within datasets. Finally, we describe a technique for smoothing behaviour predictions across multiple temporal windows, leveraging smoothening techniques from signal processing and computer vision. We demonstrate the utility of our pipeline for predicting animal behaviour using accelerometer data collected from African wild dogs (*Lycaon pictus*) in the Okavango Delta, Botswana. All code for implementing our pipeline can be provided on special request and will published publicly after peer review.

# 2 Methods

## 2.1 Creating Labelled Datasets

Classically, the data for behavioural classification from accelerometers is collected in two modalities—accelerometer readings and behavioural annotations—which are collected through independent means. Acceleration data are collected as continuous streams of input from sensors mounted on animals, whereas behavioural annotations (i.e. labels of known behaviours with associated start and end timestamps), are determined by researchers based on video or other surveillance data (e.g., Fehlmann et al., 2017; Clermont et al., 2021; Chakravarty et al., 2019). These disparate data sources are then carefully aligned temporally to create a labelled dataset of known behaviours and their associated acceleration readings from various animals and time intervals. We then partition these labelled observations randomly into training, validation, and testing datasets. The learning algorithm uses the training set to set model parameters, the validation set is used to select between model configurations, and the testing set is used to assess how well the model performs on new, unseen data (the ultimate use case of the trained behaviour classification model).

Given the timescales and sampling frequencies (which can exceed 100 Hz) of accelerometer deployments within ecology, the continuous stream of acceleration data, which can span from hours to years, is often too large to be loaded into CPU memory or even stored on disk as a single file. Therefore, in our pipeline, we first split continuous acceleration data into granular time segments of 12 hours each (i.e. half-days) to reduce CPU memory needs. Each half-day segment is accompanied by metadata, including details about the individual animal, observation day, and relevant environmental conditions. Our ML pipeline allows users to filter this metadata to retrieve segments that meet specific criteria. For instance, users can request segments from particular individuals for the training set and segments from another, disjoint set of individuals for the testing set in order to investigate whether the model can generalise to unseen individuals. This metadata-driven segmentation approach also supports the creation of train-test splits that reflect realistic distributional shifts between training and deployment, facilitating robust model evaluation (for detailed evaluation setups, see Section 2.4).

The duration of behaviours in a labelled dataset can vary significantly, whereas machine learning models are often trained using datasets with consistent time durations across examples. Thus, we select a fixed window size based on a chosen percentile of behaviour durations from the dataset. For observations exceeding this window duration, we extract multiple windows by splitting the signal into multiple parts. For those shorter than the window, we repeat the signal until it reaches the required duration, similar to circular padding in signal processing,

where the signal is looped at the beginning and end to achieve the target length. Looping (as opposed to padding with silence) ensures that the model does not spuriously use the length of the annotation to identify behaviour and relies primarily on the content of the recording. For more details on padding techniques in signal processing, refer to Schoeters et al. (2020).

Given a sampling frequency of $f$ Hertz and a window duration of $w$ seconds, the resulting tensor for a tri-axial dataset, i.e., an accelerometer data across three spatial axes, has size $T := \lfloor fw \rfloor$ (where $\lfloor x \rfloor$ denotes the truncation of $x$ to the nearest integer) along the temporal axis resulting in tensors of shape $(3, T)$. The first dimension corresponds to the $X, Y$, and $Z$ acceleration axes. Suppose the number of observations is $n$ and the total number of behaviour classes is $K$. We denote the labelled data by $\{(X_i, y_i)\}_{i=1}^{n}$, where $X_i$ represents the windowed raw acceleration data with a shape of $(3, T)$, and $y_i$ is the behaviour label in the set $\{1, ..., K\}$ for $i$th observation. Herein, we will use the general terminology *input* to refer to each accelerometer clip $X_i$ and *class* to each behviour label $y_i$.

## 2.2 Class Rebalancing

Animal behaviour datasets are often highly imbalanced (Homburger et al., 2014), resulting in poor machine learning model performance on minority classes Japkowicz and Stephen (2002), defined as behavioural classes with relatively little data. We mitigate this issue by using flexible class rebalancing through resampling, adjusting the training data's class distribution for improved class balance (Figure 1). Concretely, let $P_K := (p_1, \ldots, p_K)$ denote the empirical proportion of class labels, where $p_K$ is the proportion of training examples that are of behaviour $k$. Ideally, the class distribution is the uniform distribution $U_K := \big(1/K, \ldots, 1/K\big)$. We introduce the balancing parameter $\theta \in [0, 1]$ that controls the degree of resampling, creating the adjusted class distribution:

$$Q_K = \theta U_K + (1 - \theta) P_K \,. \tag{1}$$

Here, $\theta = 0$ retains the original distribution (no rebalancing), and $\theta = 1$ corresponds to perfect class balance (identical sample sizes across classes). Higher $\theta$ values improve minority class representation but risk over-sampling the same observations. Conversely, lower $\theta$ values increase majority class representation, which may cause the model to favour the majority class and underperform on less frequent behaviours. For unbalanced datasets, we can select a non-zero $\theta$ and resample the training data to produce $Q_K$, the adjusted class distribution. In our pipeline, users can specify their preferred $\theta$, and this rebalancing is integrated into the training phase. The optimal $\theta$ should be fine-tuned for each dataset by training the model on rebalanced datasets across a range of equally spaced $\theta$ values and choosing the $\theta$ that gives the best prediction accuracy for the classification (see Figure 1).

## 2.3 Model Architecture and Uncertainty Quantification

Our approach integrates two machine learning models trained separately on disjoint splits of the dataset. These models are essential for augmenting our predicted behaviours with uncertainty quantification. The first model takes a segment of acceleromter readings as input and outputs scores indicating the probability of success of each behavioural class. While typical models would simple return the behaviour class with the highest probability, this approach does not come with a confidence guarantee for the predicted label. The second model takes these probabilities as input and outputs a *prediction set* which accounts for statistical noise and contains the true behaviour with a pre-specified confidence level. The combination of

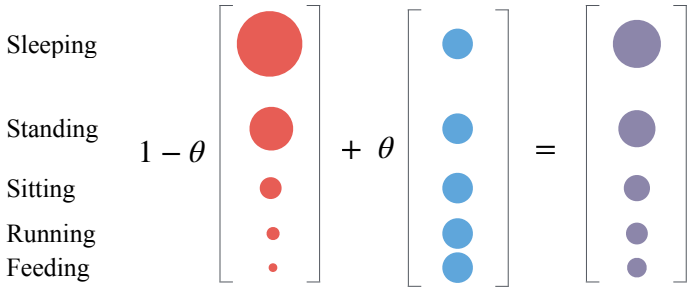



Figure 1: Rebalancing on a sample set of behaviour classes and their empirical distribution. The size of the red (left) dots represents the proportion of each behaviour class in the training dataset. The blue (middle) dots represent the ideal uniform class distribution (each class is equally represented). A linear combination of these two class distributions, parameterized by $\theta$, gives a more balanced class distribution in violet (right).

these two models offers a statistically valid behaviour classification. Thus, this next step in the pipeline involves classifying animal behaviour and performing statistical inference. We employ a one-dimensional convolutional neural network (1D CNN) to extract features from the raw accelerometry data for behaviour classification. Adopting a LeNet-style architecture (LeCun et al., 1998), we use a sequence of convolution layers to extract spatial features from the signal, followed by a fully connected layer for classification based on learned features (Figure 2). In the upcoming paragraphs, we describe technical details of the architecture and feedback signal for the training procedure.

### 2.3.1 Model Architecture

Suppose $x$ denotes the raw accelerometry signal of size $(3, T)$ where the first dimension corresponds to the three acceleration axes and $T$ is the length of the signal. The input signal is passed through $m$ convolution layers. Each convolution layer consists of three steps - 1D convolution, max pooling, and activation. The input to a 1D convolution is of shape $(C_{\text{in}}, T)$, where $C_{\text{in}}$ represents the number of input channels, and the output is of shape $(C_{\text{out}}, T)$, where $C_{\text{out}}$ represents the number of output channels. For accelerometry data, the initial convolution layer starts with $C_{\text{in}} = 3$, so that we interpret each spacial axis as a channel. The number of output channels may not have such an interpretation, so is thought of as a hyperparameter for the model's architecture. Each output channel $i$ (for $i = 1, \ldots, C_{\text{out}}$) is computed using the filter $W_i$. The filter is a $(C_{\text{in}}, k)$ dimensional tensor where $k$ is the kernel (or filter) size of the convolution. A convolution involves sliding the filter over the input data along the temporal axis, generating a new array of values known as the feature map. This process is described mathematically in Appendix A. The feature map highlights the presence and location of patterns or features detected by the filter. Choosing the right kernel size is essential, as it depends on factors like sampling frequency and duration of behaviours in the target species. The kernel size dictates the duration over which features are extracted. For species with slower sampling frequencies, larger kernels are more effective in capturing gradual changes in accelerometer readings. For example, to convolve over $s$ seconds, the kernel size would be $sf$, where $f$ is the sampling frequency. The kernel size is a hyperparameter that can be optimised by training the model on a range of values and selecting the one that yields the best performance on a validation set.

The 1D convolution is adjoined with a max pooling and an activation layer. Max pooling

extracts the maximum value within a specified window of a feature map. This operation reduces the dimensionality of the feature map, which helps in decreasing the computational load and controlling overfitting. Using a window size of 2, we halve the size of the feature map along the temporal axis. The output is then passed through a rectified linear unit (ReLU) activation function, applying $\mathrm{ReLU}(u) = \max(0, u)$ element-wise. We stack m convolution layer to obtain a sequentially deeper model (Figure 2). Standard LeNet style architectures increase the number of output channels with convolution layers to increase model expressiveness. We double the number of output channels with each additional convolution layer resulting in a final output of size $(2^{m-1}\, C_{\text{out}}, T/2^m)$. This output is flattened and fed into a fully connected network to produce a vector $\mathbf{s} = (s_1, \ldots, s_K)$ where recall $K$ is the number of label classes. Finally, the `softmax` function converts this vector into class probabilities:

$$\texttt{softmax}(\mathbf{s})_i = \frac{\exp(s_i)}{\sum_{j=1}^{K} \exp(s_j)}\,. \tag{2}$$

Thus, the final output of the 1D CNN is a probability distribution over the class labels for each input acceleration signal. We denote this multi-class classification model by $\mathcal{M}_1$ throughout the rest of the manuscript.

The 1D CNN architecture is better suited than fully connected neural networks for extracting features from raw time series data from a memory point of view due to weight sharing (LeCun et al., 2015). The number of learnable parameters in each convolution layer of a 1D CNN depends on the filter size, which includes the number of input and output channels and the kernel size, but is independent of the input signal dimension. This is beneficial when working with raw time series data, where input dimensions can be very large depending on the chosen window size for training. Conversely, in fully connected neural networks, the number of parameters grows with the input dimension.

### 2.3.2 Training Objective & Calibration

Now we describe the *loss function* we minimise to train $\mathcal{M}_1$, which provides the necessary feedback for the model to learn the free parameters mentioned above. In the presence of class imbalance, training a 1D CNN becomes challenging, especially as the number of classes increases. Class imbalance causes the model to favour majority classes, leading to minority classes not getting predicted (Ali et al., 2013). To address this, we adopt a one-vs-all classification approach. This method converts the multi-class classification problem to $K$ separate binary classification problems, simplifying the training process. Instead of predicting an animal's behaviour from a set of behaviour classes, the model now determines binary outcomes, e.g., sleeping versus not-sleeping, for each class. Consequently, we train $K$ binary classifiers, one for each class, thereby reducing complexity and improving the model's ability to handle imbalanced data. In this setup, the model's output for each class is treated as the probability of success or failure for that particular class. To achieve this, we minimise the popular *binary cross-entropy loss* for each of the $K$ one-vs-all classifiers. Let $y_{i,k}$ be the binary indicator for whether the $i$th observation belongs to class $k$ and let $\hat{y}_{i,k}$ denote the $\mathcal{M}_1$'s output probability for sample $i$ belonging to class $k$. The binary cross entropy loss for class $k$ is

$$\mathcal{L}_k = -\frac{1}{n}\sum_{i=1}^{n} y_{i,k} \log \hat{y}_{i,k} + (1 - y_{i,k}) \log(1 - \hat{y}_{i,k})\,.$$

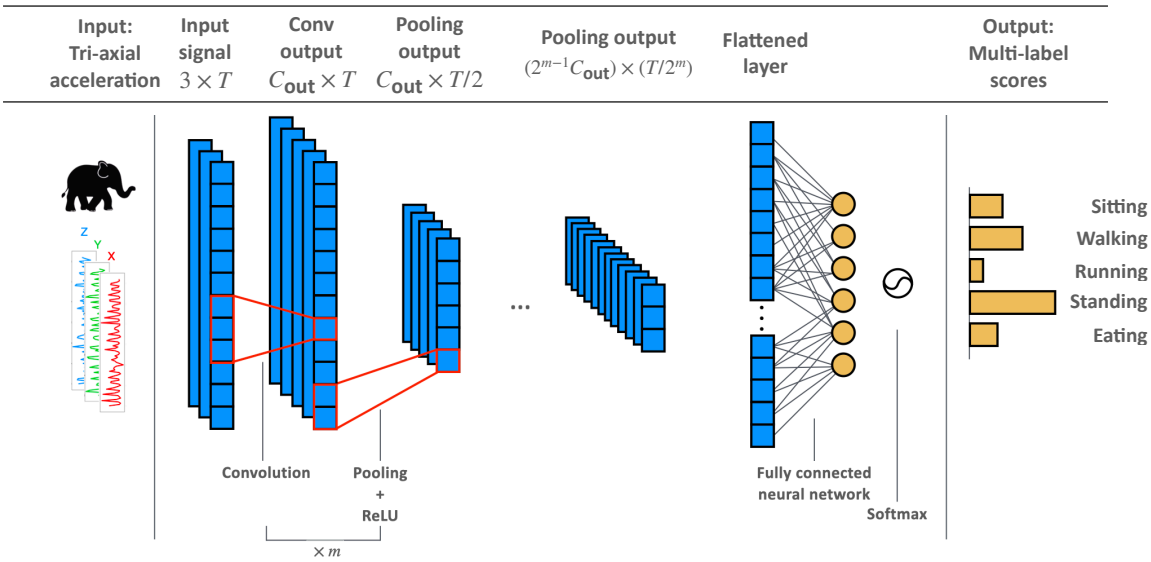



Figure 2: Schematic of a 1D convolution neural network. In the first convolutional layer, a 1D convolution is applied to an input of size $(3, T)$, preserving the temporal dimension while increasing the number of channels to $C_{\text{out}}$, a pooling layer reduces the temporal dimension by selecting maximum values within each window of size 2, followed by ReLU activation; applying $\text{ReLU}(u) = \max(0, u)$ elementwise. This process iterates in subsequent layers: $\ell$th convolution layer for $\ell > 1$ receives a signal of size $((2^{\ell-2}C_{\text{out}}, T/2^{\ell-1}))$ and outputs features of size $((2^{\ell-1}C_{\text{out}}, T/2^{\ell}))$. The output of the last convolution layer is flattened into a 1D tensor, passed through a fully-connected network to yield a vector of size $K$. A softmax function (Eq (2) then converts this vector into a probability distribution, representing the probability of success for each class.

The overall loss is the sum of the binary cross-entropy losses across all classes, i.e. we minimise

$$\mathcal{L} = \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}_k. \tag{3}$$

Minimising the binary cross-entropy loss, $\mathcal{M}_1$ produces class scores that resemble probabilities (in that they are non-negative and add up to one), but are often miscalibrated (Guo et al., 2017). A calibrated model, by definition, produces class scores that closely match the true likelihood of each class being the correct label. Without calibration, even if the model achieves high classification accuracy by selecting the class with the highest score, the prediction lacks probabilistic guarantees. It is crucial to convert these scores into calibrated *probability scores* to accurately report uncertainties associated with our predictions. Several calibration techniques, such as Platt scaling (Platt et al., 1999) and isotonic regression (Zadrozny and Elkan, 2002), address this by mapping the model's scores to calibrated probabilities. Once the model is calibrated, the associated probability score with each class can be used for uncertainty quantification.

Instead of predicting one most likely behaviour with a potentially low probability score, a more statistically-informed approach is to report a set of labels that contain the true label with

high probability. This set is called a *prediction set*. Conformal prediction (Angelopoulos et al., 2023) is a statistical framework that provides prediction sets containing the true class with a pre-specified coverage probability, typically denoted by $1-\alpha$, where $1-\alpha$ is a pre-specified confidence level. This method seeks to balance two goals: ensuring that the prediction set includes the true class with the desired coverage probability, and minimising the size of the prediction set to maintain concise predictions. Achieving this balance naturally introduces trade-offs, as larger sets naturally increase coverage but can become impractical. We use regularised adaptive prediction sets (RAPS) proposed by Angelopoulos et al. (2020), which balances these goals by incorporating a penalty for larger prediction sets while maintaining statistical guarantees. Concretely, the RAPS algorithm is as follows: for any input $x$ (e.g., acceleration reading) with a true label $y$, the input is fed into the calibrated classification model $\mathcal{M}_1$ to obtain probability scores $(s_1, \ldots, s_K)$ which are fed into $\mathcal{M}_2$ to obtain a prediction set $C(x)$ such that the probability of $y$ belonging to $C(x)$ is greater than or equal to $1-\alpha$ for a pre-specified confidence level $1-\alpha$. Note that $\mathcal{M}_2$ is trained on a dataset, called a calibration set, distinct from the one used to train $\mathcal{M}_1$. Once the classification model $\mathcal{M}_1$ and conformal model $\mathcal{M}_2$ are determined, we employ the evaluation metrics specified in the next section to assess their performance.

### 2.4 Evaluation Setups

To assess the trained model's performance, we then evaluate it on a subset of data not used for training $\mathcal{M}_1$ and $\mathcal{M}_2$, called the testing data. In order to meaningfully assess the model's performance after implementation in the real world, the test data should ideally represent future data on which predictions will be made. However, various temporal, environmental, and experimental factors can cause the future data that the model will encounter to differ slightly from the training data, leading to a distribution shift, which typically degrades model performance in real-world applications (Quiñonero-Candela et al., 2022). Distribution shifts are ubiquitous in ecological data (Koh et al., 2021), and thus, it is critical to ascertain whether trained models are robust to distribution shifts using the available data. We achieve this by creating train-test splits that mimic expected distribution shifts by leveraging our pipeline's ability to retrieve data subsets with specific properties using metadata (Section 2.1). For instance, to test robustness to individual-specific shifts, we might train on data from 4 out of 6 individuals and test on data from the remaining 2. Similarly, to evaluate robustness to temporal shifts, we can train on data collected before a specific year and test on data from subsequent years.

After training the predictive and conformal models, $\mathcal{M}_1$ and $\mathcal{M}_2$, we proceed to evaluate their performance using specific metrics for each model. For $\mathcal{M}_1$, we assess performance with three key metrics: precision, recall, and F1-score.

1. Precision measures the proportion of true positives among all predicted positives, indicating how accurate the model's predictions are for a given class. Precision values range from 0 to 1, with 1 being perfect precision.

2. Recall indicates the proportion of actual positives that are correctly identified by the model, reflecting its ability to detect the presence of a class. Recall values range from 0 to 1, with 1 being perfect recall.

3. F1-scores balance precision and recall by taking their harmonic mean, given by the

following formula

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

F1-scores offer a single metric that combines precision and recall to evaluate overall effectiveness. F1-score values range between 0 and 1, with 1 achieved only under perfect precision and recall. This metric is helpful in hyperparameter finetuning.

The evaluation metrics above are used to tune hyperparameters such as the number of CNN layers ($m$), the number of output channels $C_{\text{out}}$, and the kernel size $k$. This tuning process involves training the model across a range of values for each hyperparameter and selecting the value that optimises the evaluation metric. To evaluate the conformal model $\mathcal{M}_2$, which operates on top of the calibrated probability scores output by $\mathcal{M}_1$, we use metrics such as average coverage and the average size of RAPS:

1. Coverage measures the proportion of instances for which the correct label is included in the prediction set provided by $\mathcal{M}_2$. Coverage values range from 0 to 1, with 1 being perfect coverage.

2. Average RAPS size refers to the average size of prediction sets of the test data provided by $\mathcal{M}_2$. Average RAPS size values range from $1 - K$, where $K$ is the total number of classes. Lower values of average RAPS size indicate more precise prediction sets.

### 2.5 Temporally Smoothed Classifications

After training and tuning the prediction model, we use the model to predict animal behaviours from a continuous stream of unseen accelerometry data. The model, trained on fixed window sizes $T$, breaks the acceleration data into chunks of width $T$ and evaluates each chunk. However, predictions based on isolated windows can be erratic due to the lack of context from surrounding windows. Since the duration of each window is often chosen to be smaller than the true duration of behaviours, rapid transitions between classes can be biologically unlikely. To smooth these predictions, we average the predicted scores over a fixed number of windows, $s$, with a hop length $t$. Specifically, for an input signal $x$ of shape $(3, N)$, the signal is segmented into windows of width $T$, and the model produces a sequence of probability scores of shape $(K, \lfloor N/T \rfloor)$. The windows can overlap to ensure that the behaviour scores do not vary too drastically between consecutive windows, and the degree of this overlap can be determined by the user.

To smooth sharp changes in these scores, we average $s$ consecutive scores, calling this set the averaging window. Therefore, each averaging window averages scores over $s \times w$ seconds (recall $w$ is window duration in seconds). The averaging window shifts by $t$ steps, causing consecutive averaged scores to share $s - t$ evaluations. A larger $s$ results in smoother, coarser behaviour predictions, suitable for longer activities like resting. Conversely, a smaller $s$ provides finer, less smooth predictions, ideal for shorter behaviours like eating. The value of $s$ should be chosen based on the typical behaviour duration for the species studied. As an example, suppose we want to identify the occurrences of behaviour $k$ in long sequences of acceleration data spanning months or even years. The typical duration of behaviour $k$ in the concerned wildlife species is known to be $d_k$, and the averaging window size should be a fraction of $d_k/w$. Ideally, the averaged scores would favour behaviour $k$ because it will receive the maximum vote for behaviour $k$ from all $s$ averaging windows. This approach ensures smoother and more reliable behaviour classification over long time horizons.

# 3 African Wild Dogs Study

This section parallels Section 2, focusing on a particular case study on African wild dogs. We describe the data collection process and discuss key ecological and machine learning aspects that are unique to this context.

## 3.1 Data Collection and Annotation

From October 2021 to September 2023, we deployed wildlife tracking collars with inbuilt GPS and accelerometer sensors (GPS PLUS 1C, Vectronic Aerospace, Germany) on five African wild dogs (*Lycaon pictus*) in the Okavango Delta, Botswana (centre $19°31S, 23°37E$). Tracking collars were scheduled to record GPS fixes every three hours and continuous accelerometer data at 16 Hz. Animal-worn microphones (uMoth, Open Acoustic Devices, United Kingdom) were attached to collars with a bespoke attachment mechanism that detached from collars following battery expiry after three weeks (Rafiq et al., 2023a). Microphones were included within collar deployments as the recordings could be used to collect labelled data for the accelerometers. To maximise deployment durations, microphones were scheduled to record audio at 16 kHz in 25-second on and 5-second off sampling cycles between 0600-1000 and 1600-2000 (local time). Recording windows were chosen to correspond with the times that African wild dogs were most active (Rafiq et al., 2023b).

We paired accelerometer and audio data with 124.79 hours of time-stamped video collected from a handheld video camera recording from within an observation vehicle. African wild dogs within the study system were habituated to vehicles due to research and tourism activity within the area, and as such, no obvious changes in behaviour were observed due to the presence of vehicles. All collars had biodegradable drop-off mechanisms or were manually removed and replaced with new collars following battery expiry. Accelerometer and audio data were manually downloaded after collecting tracking collars or detached audio recorders, respectively. We attached collars in collaboration with a Botswana-registered veterinarian using established protocols (Hubel et al., 2016a,b), and our work received ethical approval from the University of Washington (IACUC #4514-01) and Botswana's Department of Wildlife and National Parks (Permit # ENT 8364 XLIX (38)). No ill effects of collar deployments were observed.

We labelled a subset of the acceleration and audio data into one of five biologically relevant behavioural classes – feeding, resting, moving, running, and vigilant – by pairing data with timestamped behaviours observed within our vehicle-collected video. Herein, we refer to these behaviour labels as *video labels*. We selected these behavioural classes based on their relevance to wider ecological questions of interest and as they represented distinct behaviours possible to unambiguously identify within the video. Behaviours that did not fit into these categories or that were unclear from video recordings were excluded.

There was significant class imbalance within our behaviours due to the relative rarity of less observed behaviours, as well as the logistical challenges of directly observing these behaviours in the wild from a vehicle. Consequently, after using the video-labelled audio data to listen to the audio profiles of our behaviours, we used a subset of our audio (for which we had no video) to add additional feeding, moving, running, and vigilance labels and matched them to our acceleration data. We used audio to only add labels for these behaviours because these could be clearly delineated within the audio datasets with high levels of certainty. Herein, we refer to these as *audio labels*. At the end of this step, we had 73.94 hours of labelled acceleration data (both video and audio labels) with labels corresponding to one of the five chosen behavioural classes.

We further discarded labelled behaviours under one-second duration because they were too brief to capture meaningful acceleration patterns, and we removed observations of feeding, moving, and running under eight seconds after visual inspection of waveforms suggested such labels were unreliable. This amounted to removing approximately 2.34% of data and left us with 72.21 hours of labelled accelerometer data for our five behaviours (Table 1), with the most represented behaviour (resting) having $\approx 78.66$ times more extracted windows than the least represented behaviour (running). We used this filtered dataset as our training data. (Table 1). The matched behaviour-acceleration pairs exhibited significant variability in duration due to inherent differences in the durations of behaviours exhibited by animals and the availability of reliable video recordings (see Table 3 in Appendix B for an individual-wise data summary).

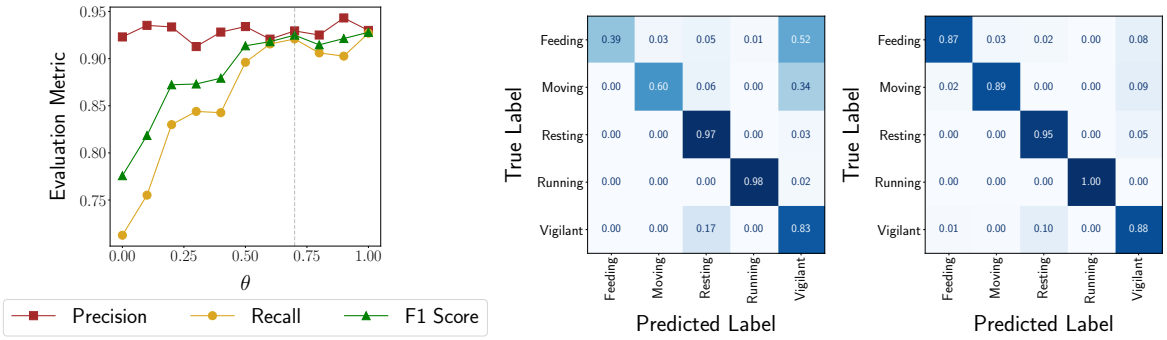
| **Behaviour** | **Video labels** | | **Audio labels** | |
|---|---|---|---|---|
| | Duration [h] | # Extracted windows | Duration [h] | # Extracted windows |
| Feeding | 1.32 | 429 | 0.20 | 60 |
| Moving | 1.67 | 616 | 0.39 | 120 |
| Resting | 51.57 | 15442 | 0.00 | 0 |
| Running | 0.09 | 32 | 0.48 | 153 |
| Vigilant | 16.45 | 6496 | 0.05 | 20 |

Table 1: Behaviour-wise summary of the filtered 72.21 hours of labelled acceleration data. The table shows the total duration (in hours) and number of extracted windows for each behaviour across the video and audio labels.

### 3.2 Training Data

The duration of labelled behaviours varied significantly within and across behaviour classes in the filtered dataset. For instance, among the video labels, the average duration of resting was 85 seconds whereas the average duration of running was 17 seconds. We extracted accelerometer signals of a consistent duration from matched behaviour-accelerometer pairs to train our machine-learning models. When an observation's duration exceeded the chosen window duration, we extracted multiple non-overlapping windows. If the observation was longer than 8 seconds but shorter than the window duration, the signal was repeated until the required window duration was met. We determined the window length as the median duration of all behaviours, resulting in $w = 12.94$ seconds. This resulted in $23,368$ behaviour-acceleration pairs (Table 1), each with acceleration signals of a fixed duration of $w$ seconds (see Figure 6 in Appendix B for representative waveforms for each behaviour class). Resting and vigilant collectively account for approximately 94% of the dataset, making the class distribution unbalanced.

For training the 1D CNN for behaviour classification, we divided our $23,368$ matched acceleration behaviour pairs into train and test sets. By default, we split the dataset into train and test sets in an $80-20\%$ ratio while preserving the distribution of behaviour classes in both sets. We also considered specific distribution shifts that did not lead to such fixed train-test set size ratios because the number of observations satisfying the train and test set criteria differed. For example, in an individual-based train-test split where the test set contained observations from a single individual, with the rest of the individuals in the train set, the size ratio of train and test sets was $70-30\%$. Details on these distribution shifts and associated train-test sets




(a) Tuning $\theta$ (b) $\theta = 0.0$ (c) $\theta = 0.7$

Figure 3: Summary showing how flexible class rebalancing improved model performance for our original training data (comprising 2.09% of Feeding samples). The left plot illustrates how precision, recall, and the F1 score evolved with increasing rebalancing intensity (parameterized by $\theta$). The confusion matrices compare classification results: the first matrix for no rebalancing ($\theta = 0.0$) and the second for optimal rebalancing ($\theta = 0.7$).

are provided in Section 3.5.

### 3.3 Class Imbalance

Our dataset had a high class imbalance, with minority classes - feeding, moving, and running - comprising only 6% of the data. Thus, we applied our class rebalancing technique using $\theta$ values from 0.0 to 1.0 (Figure 3) and used the precision, recall, and F1 score metrics to evaluate model performance following rebalancing. All three evaluation metrics increased with $\theta$, plateauing at $\theta$ values of 0.7. Thus, a $\theta$ value of 0.7 represented the ideal amount of resampling for optimising model performance while minimising the extent of resampling adjustments. Following rebalancing, the proportion of correctly predicted feeding instances in the test set (true positives) increased from 0.39 to 0.87.

After training $\mathcal{M}_1$ on rebalanced training data, we fit the conformal model $\mathcal{M}_2$ to the predictions of $\mathcal{M}_1$ on a held-out validation set with a desired confidence level of 0.95. This validation set was selected to mirror the class distribution of the training set, ensuring consistency in data distribution for both $\mathcal{M}_1$ and $\mathcal{M}_2$. The overall performance of the prediction model $\mathcal{M}_1$ and conformal model $\mathcal{M}_2$ on various train-test splits is reported in the next section based on the evaluation metrics discussed in Section 2.4.

### 3.4 Model Training

After class rebalancing, we trained the prediction model $\mathcal{M}_1$ using our 1D CNN architecture (see Section 2.2), minimising the multi-label loss Eq (3). The hyperparameters of the model were tuned by comparing model performance for different combinations of hyperparameter values on a held-out validation set. Refer to Appendix C.1 for details and recommendations on hyperparameter fine-tuning. The class distribution in the validation was maintained to be the same as that of the remaining training set. We fit the conformal model $\mathcal{M}_2$ on the output probabilities of the prediction model with confidence level $1 - \alpha = 0.95$. Figure 4 illustrates the regularised adaptive prediction sets (RAPS) for a random test sample for each behaviour class along with the associated probability of each class in the prediction set.

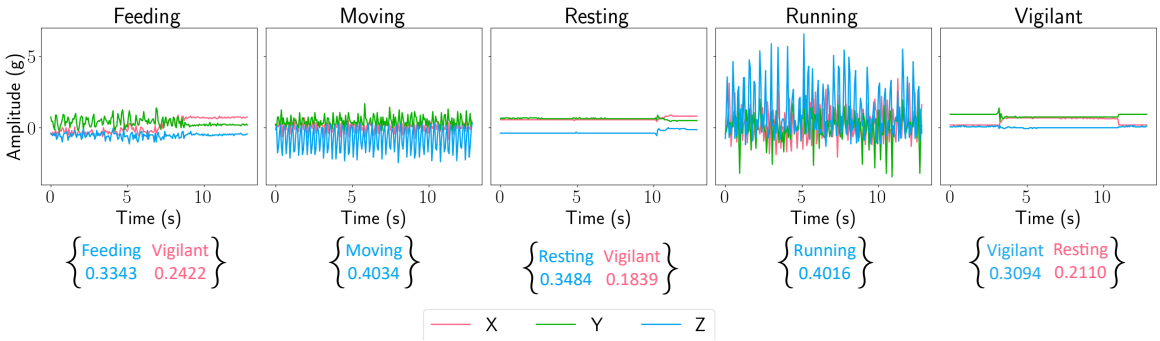



Figure 4: Regularised adaptive prediction sets (RAPS) of a random test sample from each behaviour class. RAPS refers to a prediction set which contains the true class with a probability of $1 - \alpha = 0.95$. The numbers associated with each behaviour in the prediction set are the estimated and calibrated probability of success of that class. Here the most likely behaviour in the prediction set is in blue and the remaining behaviours are in pink. Notice that the model estimates the true class with the highest probability for each behaviour.

Model training was carried out on a CUDA-enabled GPU machine with 12 GB of GPU memory, 64 GB of CPU RAM, and 24 virtual CPU cores. Creating the train-test split and fixed duration windows from the matched behaviour-acceleration pairs took approximately 2 minutes. This resulted in train, validation, and test set sizes of 14,978, 3,745, and 4,645, respectively. For a model configuration with 887,237 trainable parameters, the entire process of training took approximately 3 minutes. See Appendix C.2 for more details on computing.

### 3.5 Testing Model Robustness

To test the model's robustness against distribution shifts that may occur between training time and after implementation in the real world, we mimicked biologically realistic shifts using the metadata associated with our data. We created four experimental setups. Within the 'no split' setup, we randomly divided our data into training and testing sets in an $80 - 20\%$ ratio, maintaining class distributions across each set. Within the 'inter-dog' setup, we trained our models with data from four dogs and reserved data from the fifth dog for testing. Within the 'inter-year' setup, we trained on data from one year (2021) and tested the data on a separate year (2022). Finally, within the 'inter-period' setup, we trained the model on data collected in the morning and tested the model on evening data.

Hyperparameter tuning was conducted independently for each experimental setup (see Appendix C). Model performance remained consistent across different train-test splits, indicating our model's robustness to the distributional shifts we considered (Table 2).

### 3.6 Smoothing Model Classifications

The final step of the pipeline, after model training and selection, is smoothing continuous streams of accelerometer signals. As described in Section 2.5, smoothing is conducted in two steps. First, the model is applied to consecutive windows of acceleration data, each 12.94 seconds long, along the temporal axis to generate a sequence of scores. Second, this sequence of scores is smoothed by averaging the scores within an averaging window of length $s$. Thus, higher values of $s$ represent greater degrees of smoothing and predictions on longer time

| Evaluation Metric | No split | Interdog | Interyear | InterAMPM |
|---|---|---|---|---|
| Train set size | 14978 | 13104 | 9528 | 13712 |
| Validation set size | 3745 | 3277 | 2382 | 3429 |
| Test set size | 4645 | 6987 | 11458 | 6227 |
| Precision (val, test) | (0.93, 0.92) | (0.94, 0.86) | (0.92, 0.84) | (0.91, 0.88) |
| Recall (val, test) | (0.92, 0.92) | (0.93, 0.90) | (0.89, 0.84) | (0.90. 0.88) |
| F1 score (val, test) | (0.92, 0.92) | (0.93, 0.88) | (0.91, 0.83) | (0.90, 0.88) |
| Accuracy (val, test) | (0.93, 0.93) | (0.93, 0.91) | (0.89, 0.80) | (0.87, 0.85) |
| Top-1 coverage (val, test) | (0.88, 0.86) | (0.89, 0.79) | (0.90, 0.80) | (0.88, 0.83) |
| RAPS coverage (val, test) | (0.95, 0.93) | (0.95, 0.89) | (0.92, 0.83) | (0.94, 0.90) |
| RAPS avg size (val, test) | (1.32, 1.32) | (1.21, 1.30) | (1.05, 1.06) | (1.21, 1.23) |

Table 2: Dataset sizes and evaluation metrics (as described in Section 2.4) for the best-tuned classification $\mathcal{M}_1$ and conformal $\mathcal{M}_2$ models for the four distribution shift splits. For the metrics accuracy, precision, recall, and F1 score, the most likely class based on the scores output by $\mathcal{M}_1$ are considered. The RAPS coverage and RAPS average size assess the performance of $\mathcal{M}_2$, which is fitted on the validation set. To compare model performance between the validation (same distribution as the training set) and testing sets, the metrics are presented side by side within the brackets.

windows, i.e., coarser predictions. The averaging window is then moved by $t$ steps to obtain consecutive average scores. We used window sizes $s$, ranging from $s = 25$, suited for the detection of shorter-duration behaviours such as feeding, to s = 100, suited for longer-duration behaviours, such as resting (Figure 5).

# 4 Discussion

In this paper, we have introduced a novel approach for animal behaviour classification, applicable across taxa, that integrates machine learning with statistical inference to address common challenges and limitations of machine learning behaviour classification in ecology. Our approach allows users to evaluate and mitigate common issues in ecological datasets, including class imbalances and distributional shifts. Moreover, our framework extends behaviour classification from single predictions with no uncertainty measures to robust sets of predicted behaviours with probabilities of containing the true behaviour. Additionally, our approach presents a method to simply smooth noisy behavioural classifications.

Our results highlight the utility of flexible class resampling techniques in improving animal behaviour classification accuracy, particularly in datasets with significant class imbalances. By combining undersampling and oversampling during model training, our approach allows researchers to evaluate the performance impacts of different resampling intensities. This framework enables researchers to achieve an optimal class distribution tailored to their study-specific needs, such as optimising precision or recall while minimising the biases introduced by excessive resampling Haixiang et al. (2017). For example, in our heavily imbalanced African wild dog dataset, rebalancing improved model recall by up to 29.2% when the class distribution was resampled to a mildly imbalanced state, with performance gains plateauing beyond this point. Accurate classifications for under-represented behaviours, such as feeding, more than

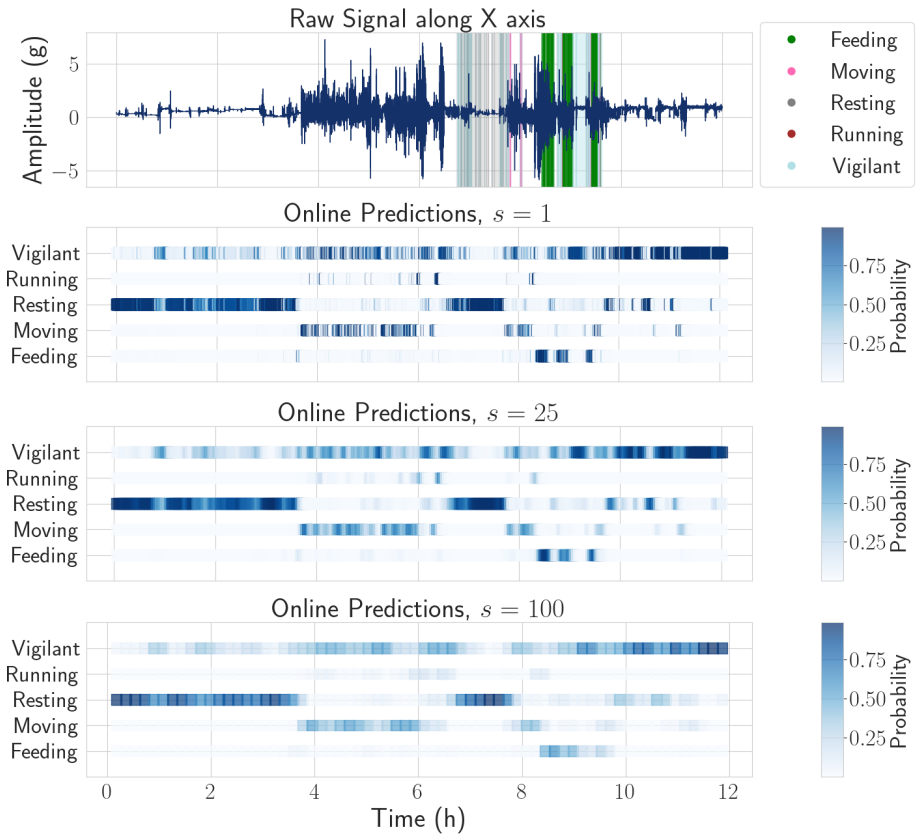



Figure 5: Smoothed classifications for a sample 12-hour African wild dog accelerometer segment. The top plot displays the original accelerometer signal, with shaded panels representing known labelled behaviours. The subsequent three plots illustrate the predicted probabilities of the different behavioural classes with no ($s = 1$), moderate ($s$=25), and high ($s$=100) average window lengths.

doubled as a result.

The severity of class imbalances and optimal resampling thresholds are likely to vary across populations and may be particularly severe for cryptic, wide-ranging, aerial, or marine species, where key behaviours can occur rarely or during periods when direct observation of individuals is not possible. For example, foraging behaviours represented only 0.49-1.46% of the total data for training seabird behaviour classifiers in Otsuka et al., 2024 whilst in arctic foxes foraging (caching behaviours) represented 7.59% of collected data Clermont et al., 2021. In cases where labelled samples are particularly sparse, resampling may yield little additional performance gains and advanced data augmentation techniques may be required to improve model performance, such as artificially introducing noise into existing data or generating completely new data instances with AI (e.g., (Luleci et al., 2022)). In such cases, researchers could expand upon our open-source pipeline and incorporate data augmentation techniques alongside resampling to optimise the extent of augmentation and resampling within a unified framework.

Quantifying and reporting uncertainty is a cornerstone of ecological research, and our pipeline is among the first to explicitly quantify uncertainty in animal behaviour classifications. By extending predictions from single behaviour classifications to prediction sets - collections

of potential behaviours with a user-specified likelihood of containing the true behaviour - we provide a framework analogous to the use of confidence intervals in statistical analyses rather than defaulting to binary outcomes of presence or absence of the behaviour of interest. This provides a mechanism for researchers to propagate uncertainty from behaviour classifications into downstream analyses, such as quantifying how environmental (e.g., weather, human disturbance) or biological (e.g., social, physiological) processes affect animal behaviours.

Uncertainty quantification additionally allows researchers to balance precision and confidence in behaviour classifications according to their study needs. As the user-chosen likelihood increases, so will the size of the prediction set and the confidence that it contains the true behaviour, and vice versa. As such, while higher likelihood thresholds will improve the likelihood of sets containing the true behaviour, they can also complicate downstream analyses by yielding a wider set of predicted behaviours. As a rule of thumb, we suggest likelihoods of 0.90 or 0.95 may be most familiar and suitable within ecology as they mirror the probability values widely used in null hypotheses testing across ecology Castilho and Prado, 2021. We further recommend researchers adopt language of uncertainty when reporting model results, such that they indicate *moderate* or *strong* evidence for the prediction set containing the true behaviour based on 0.90 or 0.95 likelihood thresholds, respectively (see Muff et al., 2022).

Our pipeline also introduces a technique to evaluate model responses to distribution shifts, which can significantly impact model performance in real-world applications. Collecting labelled training data from animals in the wild can be logistically challenging and expensive. As a result, labelled ecological datasets often represent a narrow subset of the conditions in which the models will ultimately be applied. For example, labelled data collection may be carried out over relatively narrow time windows Clermont et al., 2021, be constrained to an easily observable subset of the population Chakravarty et al., 2019, or use captive animals Rast et al., 2020. In our case study, we checked for potential sources of distributional shifts across years of data collection, study individuals, and diurnal periods; the lack of distributional shifts across these data subsets provides greater confidence in the validity of our models. To mitigate the challenges of any distribution shifts, we recommend that ecologists evaluate their model's robustness to distributional shifts both during initial model development and through routine monitoring over time. This longitudinal monitoring is particularly important for long-term studies as plasticity in animal behaviours, updates in sensor technology, and the immigration of new animal phenotypes into populations could lead to distributional shifts that undermine model performance.

We also provide a flexible approach to smoothing out noisy and anomalous behaviour classifications, drawing on techniques widely used in other fields such as signal processing and computer vision (Hamilton, 2020). Our smoothing technique allows users to filter out unlikely behavioural transitions by adding temporal context from surrounding classification windows into the classification process. For example, in our case study system, rapid and repeated transitions between feeding and running, or very short isolated feeding events, are biologically unlikely. Smoothing over a temporal window substantially reduced the frequency of abrupt and unlikely transitions between different predicted behaviour classes.

The specific nuances of the study system and research questions of interest will dictate the intensity of smoothing required, and users should incorporate biological knowledge, such as typical behaviour durations, when selecting window lengths (analogous to smoothing intensity) to avoid filtering out key behaviours of interest. Larger window lengths will increase smoothing, reducing noise but risk potentially biasing against shorter behaviours. In contrast, shorter windows retain more noise but may inflate estimates for shorter duration behaviour categories. To help navigate this trade-off, we provide guidelines for selecting window lengths in the

methods and framework for implementing it in our code. We also recommend conducting sensitivity analyses on window thresholds to evaluate and mitigate any biases that may arise from the smoothing.

To support the adoption of our pipeline, we provide open-source code, a step-by-step worked example, and recommendations for key decisions (e.g., parameter selection) that users will have to make. Recognising that hardware requirements may limit accessibility, we have developed our pipeline to function on a wide range of devices, from personal computers to GPU-equipped workstations. The largest impact on compute times comes from the availability of GPU support on machines. For example, model training (the most compute-intensive process) was 150 times quicker time on GPU-equipped devices (see Appendix C.2 for more details). Ultimately, the compute time required to run our pipeline will vary across model parameters, data volumes, and computing hardware but is likely to be on the order of minutes for studies with similar hardware and data volumes as our case study. For researchers without access to local hardware, the code can be executed seamlessly within a Google Colab notebook, enabling users to run the pipeline directly in their browsers.

## Conclusion

The use of accelerometers to classify animal behaviours is becoming increasingly widespread in animal ecology. Our study introduces a flexible, open-source approach that addresses common challenges, including class imbalances, distribution shifts, and uncertainty quantification. Using this approach, we demonstrated significantly improved predictions along with associated uncertainty metrics in African wild dog behaviour classification, particularly for rare and ecologically significant behaviours such as feeding, where correct classifications more than doubled following resampling. Looking forward, the integration of multiple sensor modalities, such as accelerometer, gyroscope, and GPS data, during model training offers intriguing opportunities for further improving model performance and expanding the range of detectable animal behaviours *in situ* Castillo et al., 2014; Batpurev et al., 2021; Mao et al., 2021, and could be readily built into our pipeline. Our approach represents a key step towards advancing the burgeoning use of machine learning to remotely observe around-the-clock behaviours of free-ranging animals in their natural environments.

## Acknowledgements

We thank the Botswana Ministry of Environment, Wildlife, and Tourism for permission to conduct this research under research permit 'ENT 8364 XLIX (38)' to B.A. We thank the many researchers and collaborators associated with Botswana Predator Conservation for their invaluable contributions in the field and volunteers within the Abrahms lab for helping to annotate data. This work was funded by NSF DMS-2023166, CCF-2019844, DMS-2134012, NSF IOS (#2337405), the University of Washington Royalty Research Fund, eScience Institute, and Washington Research Foundation.

## References

Ali, A., Shamsuddin, S. M., and Ralescu, A. L. (2013). Classification with class imbalance problem. *Int. J. Advance Soft Compu. Appl*, 5(3):176–204.

Angelopoulos, A., Bates, S., Malik, J., and Jordan, M. I. (2020). Uncertainty sets for image classifiers using conformal prediction. *arXiv preprint arXiv:2009.14193*.

Angelopoulos, A. N., Bates, S., et al. (2023). Conformal prediction: A gentle introduction. *Foundations and Trends® in Machine Learning*, 16(4):494–591.

Batpurev, T., Shibata, T., Matsumoto, J., and Nishijo, H. (2021). Automatic identification of mice social behavior through multi-modal latent space clustering. In *2021 Joint 10th International Conference on Informatics, Electronics & Vision (ICIEV) and 2021 5th International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, pages 1–8. IEEE.

Brown, D. D., Kays, R., Wikelski, M., Wilson, R., and Klimley, A. P. (2013). Observing the unwatchable through acceleration logging of animal behavior. *Animal Biotelemetry*, 1:1–16.

Castilho, L. B. and Prado, P. I. (2021). Towards a pragmatic use of statistics in ecology. *PeerJ*, 9:e12090.

Castillo, J. C., Carneiro, D., Serrano-Cuerda, J., Novais, P., Fernández-Caballero, A., and Neves, J. (2014). A multi-modal approach for activity classification and fall detection. *International Journal of Systems Science*, 45(4):810–824.

Chakravarty, P., Cozzi, G., Ozgul, A., and Aminian, K. (2019). A novel biomechanical approach for animal behaviour recognition using accelerometers. *Methods in Ecology and Evolution*, 10(6):802–814.

Clarke, T. M., Whitmarsh, S. K., Hounslow, J. L., Gleiss, A. C., Payne, N. L., and Huveneers, C. (2021). Using tri-axial accelerometer loggers to identify spawning behaviours of large pelagic fish. *Movement ecology*, 9(1):26.

Clermont, J., Woodward-Gagné, S., and Berteaux, D. (2021). Digging into the behaviour of an active hunting predator: arctic fox prey caching events revealed by accelerometry. *Movement Ecology*, 9:1–12.

Ellington, E. H., Muntz, E. M., and Gehrt, S. D. (2020). Seasonal and daily shifts in behavior and resource selection: how a carnivore navigates costly landscapes. *Oecologia*, 194(1):87–100.

Fehlmann, G., O'Riain, M. J., Hopkins, P. W., O'Sullivan, J., Holton, M. D., Shepard, E. L., and King, A. J. (2017). Identification of behaviours from accelerometer data in a wild social primate. *Animal Biotelemetry*, 5:1–11.

Garde, B., Wilson, R. P., Fell, A., Cole, N., Tatayah, V., Holton, M. D., Rose, K. A., Metcalfe, R. S., Robotka, H., Wikelski, M., et al. (2021). Ecological inference using data from accelerometers needs careful protocols. *bioRxiv*, pages 2021–07.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.

Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., and Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. *Expert systems with applications*, 73:220–239.

Halsey, L. G., Shepard, E. L., and Wilson, R. P. (2011). Assessing the development and application of the accelerometry technique for estimating energy expenditure. *Comparative Biochemistry and Physiology Part A: Molecular & Integrative Physiology*, 158(3):305–314.

Hamilton, J. D. (2020). *Time series analysis*. Princeton university press.

Homburger, H., Schneider, M. K., Hilfiker, S., and Lüscher, A. (2014). Inferring behavioral states of grazing livestock from high-frequency position data alone. *PloS one*, 9(12):e114522.

Hubel, T. Y., Myatt, J. P., Jordan, N. R., Dewhirst, O. P., McNutt, J. W., and Wilson, A. M. (2016a). Additive opportunistic capture explains group hunting benefits in african wild dogs. *Nature communications*, 7(1):11033.

Hubel, T. Y., Myatt, J. P., Jordan, N. R., Dewhirst, O. P., McNutt, J. W., and Wilson, A. M. (2016b). Energy cost and return for hunting in african wild dogs and cheetahs. *Nature Communications*, 7(1):11034.

Japkowicz, N. and Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449.

Johnson, J. M. and Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. *Journal of big data*, 6(1):1–54.

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., et al. (2021). Wilds: A benchmark of in-the-wild distribution shifts. In *International conference on machine learning*, pages 5637–5664. PMLR.

Kulinski, S. and Inouye, D. I. (2023). Towards explaining distribution shifts. In *International Conference on Machine Learning*, pages 17931–17952. PMLR.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Luleci, F., Catbas, F., and Avci, O. (2022). Generative adversarial networks for labeled acceleration data augmentation for structural damage detection. *Journal of Civil Structural Health Monitoring*, 13:181–198.

Lush, L., Ellwood, S., Markham, A., Ward, A., and Wheeler, P. (2016). Use of tri-axial accelerometers to assess terrestrial mammal behaviour in the wild. *Journal of Zoology*, 298(4):257–265.

Mao, A., Huang, E., Gan, H., Parkes, R. S., Xu, W., and Liu, K. (2021). Cross-modality interaction network for equine activity recognition using imbalanced multi-modal data. *Sensors*, 21(17):5818.

Martiskainen, P., Järvinen, M., Skön, J.-P., Tiirikainen, J., Kolehmainen, M., and Mononen, J. (2009). Cow behaviour pattern recognition using a three-dimensional accelerometer and support vector machines. *Applied animal behaviour science*, 119(1-2):32–38.

Muff, S., Nilsen, E. B., O'Hara, R. B., and Nater, C. R. (2022). Rewriting results sections in the language of evidence. *Trends in ecology & evolution*, 37(3):203–210.

Nathan, R., Spiegel, O., Fortmann-Roe, S., Harel, R., Wikelski, M., and 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.

Nickel, B. A., Suraci, J. P., Nisi, A. C., and Wilmers, C. C. (2021). Energetics and fear of humans constrain the spatial ecology of pumas. *Proceedings of the National Academy of Sciences*, 118(5):e2004592118.

Otsuka, R., Yoshimura, N., Tanigaki, K., Koyama, S., Mizutani, Y., Yoda, K., and Maekawa, T. (2024). Exploring deep learning techniques for wild animal behaviour classification using animal-borne accelerometers. *Methods in Ecology and Evolution*.

Platt, J. et al. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.

Quiñonero-Candela, J., Sugiyama, M., Schwaighofer, A., and Lawrence, N. D. (2022). *Dataset shift in machine learning*. Mit Press.

Rafiq, K., Appleby, R., Davies, A., and Abrahms, B. (2023a). Sensordrop: A system to remotely detach individual sensors from wildlife tracking collars. *Ecology and Evolution*, 13(7):e10220.

Rafiq, K., Jordan, N. R., Golabek, K., McNutt, J. W., Wilson, A., and Abrahms, B. (2023b). Increasing ambient temperatures trigger shifts in activity patterns and temporal partitioning in a large carnivore guild. *Proceedings of the Royal Society B*, 290(2010):20231938.

Rast, W., Kimmig, S. E., Giese, L., and Berger, A. (2020). Machine learning goes wild: Using data from captive individuals to infer wildlife behaviours. *PloS one*, 15(5):e0227317.

Resheff, Y. S., Rotics, S., Harel, R., Spiegel, O., and Nathan, R. (2014). Accelerater: a web application for supervised learning of behavioral modes from acceleration measurements. *Movement ecology*, 2:1–7.

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

Schoeters, S., Dewulf, W., Kruth, J.-P., Haitjema, H., and Boeckmans, B. (2020). Description and validation of a circular padding method for linear roughness measurements of short data lengths. *MethodsX*, 7:101122.

Snape, R. T., Bradshaw, P. J., Broderick, A. C., Fuller, W. J., Stokes, K. L., and Godley, B. J. (2018). Off-the-shelf gps technology to inform marine protected areas for marine turtles. *Biological Conservation*, 227:301–309.

Studd, E., Derbyshire, R., Menzies, A., Simms, J., Humphries, M., Murray, D., et al. (2021). The purr-fect catch: Using accelerometers and audio recorders to document kill rates and hunting behaviour of a small prey specialist.

VonBank, J. A., Schafer, T. L., Cunningham, S. A., Weegman, M. D., Link, P. T., Kraai, K. J., Wikle, C. K., Collins, D. P., Cao, L., and Ballard, B. M. (2023). Joint use of location and acceleration data reveals influences on transitions among habitats in wintering birds. *Scientific Reports*, 13(1):2132.

Wang, Y., Nickel, B., Rutishauser, M., Bryce, C. M., Williams, T. M., Elkaim, G., and Wilmers, C. C. (2015). Movement, resting, and attack behaviors of wild pumas are revealed by tri-axial accelerometer measurements. *Movement ecology*, 3:1–12.

Wang, Z., Yan, W., and Oates, T. (2017). Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*, pages 1578–1585. IEEE.

West, L., Rafiq, K., Converse, S. J., Wilson, A. M., Jordan, N. R., Golabek, K. A., McNutt, J. W., and Abrahms, B. (2024). Droughts reshape apex predator space use and intraguild overlap. *Journal of Animal Ecology*.

Wilmers, C. C., Nickel, B., Bryce, C. M., Smith, J. A., Wheat, R. E., and Yovovich, V. (2015). The golden age of bio-logging: How animal-borne sensors are advancing the frontiers of ecology. *Ecology*, 96(7):1741–1753.

Zadrozny, B. and Elkan, C. (2002). Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699.

Zerveas, G., Jayaraman, S., Patel, D., Bhamidipaty, A., and Eickhoff, C. (2021). A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 2114–2124.

# APPENDIX

## A Convolution Neural Networks

Let $y$ represent the output from the 1D convolution, $y_i[t]$ denote the $t$-th element of the $i$th output channel, and $x[t_1 : t_2]$ denotes the segment of $x$ along the second axis starting from $t_1$ and ending at $t_2$. Then,

$$y_i[t] = W_i \odot x[t : t + k - 1] + b_i \,,$$

where $\odot$ performs elementwise multiplication of two matrices, followed by the summation of their elements. Consequently, the output from such 1D convolution is of shape $(C_{\text{out}}, T)$.

## B Data Summary

The following is an individual-wise summary of the 72.21 hours of labelled acceleration data used for model training.

| Dogs | Start date | End date | No. of days | Duration (h) | # Extracted windows |
|---|---|---|---|---|---|
| Ash | 2021-10-25 | 2021-11-13 | 11 | 13.71 | 4555 |
| Fossey | 2022-07-14 | 2022-07-20 | 5 | 6.95 | 2275 |
| Green | 2021-09-03 | 2021-09-22 | 13 | 21.42 | 6987 |
| Jessie | 2022-08-01 | 2022-08-16 | 15 | 29.04 | 9183 |
| Palus | 2021-10-27 | 2021-11-12 | 2 | 1.09 | 368 |

Table 3: Individual-wise summary of labeled acceleration data including start and end date of matched data, number of unique days, duration of matched data, and total number of extracted windows of duration $w$ seconds.

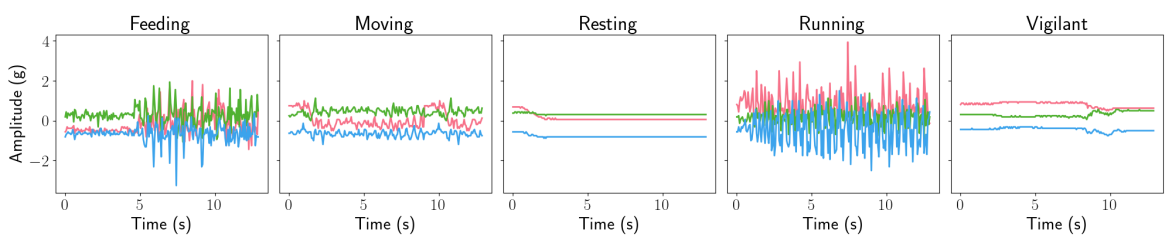


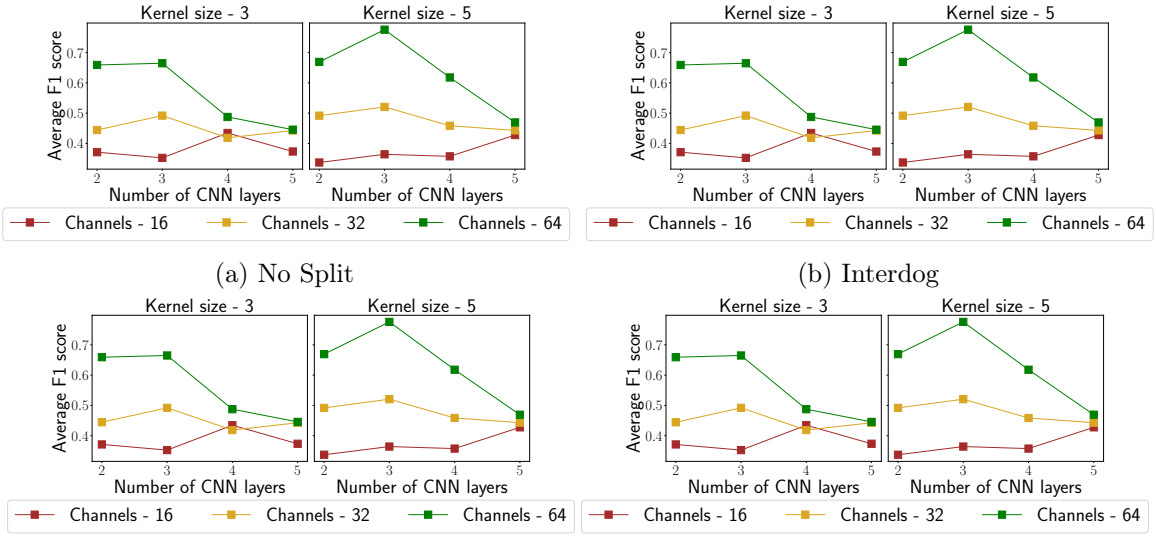

Figure 6: Accelerometry data trace plot for a randomly selected observation of each behaviour class, with a consistent window duration of $w = 12.94$ seconds.

## C Training Details

### C.1 Hyperparameter Finetuning

For the prediction model $M_1$, three key hyperparameters are tuned: the number of convolutional layers, the number of output channels in the first layer, and the kernel size. Increasing any of these parameters raises the model's trainable parameters, thus enhancing its complexity

and expressivity. Figure 7 displays the average F1 score on the validation set across different combinations of these hyperparameters for all four experimental setups. For the “no split" experiment, the best performance is offered by the model with 3 convolution layers, 64 output channels in the first convolution layer, and kernel size 5. For all four experiments, we choose 64 output channels from the first layer and kernel size 5. The number of convolution layers is chosen appropriately for each experiment.




(a) No Split (b) Interdog

(c) Interyear (d) InterAMPM

Figure 7: Hyperparameter tuning for four experiments by comparing the precision, recall, and F1 score on the validation set for different choices of the number of CNN layers, the number of output channels in the first CNN layer, and the kernel sizes.

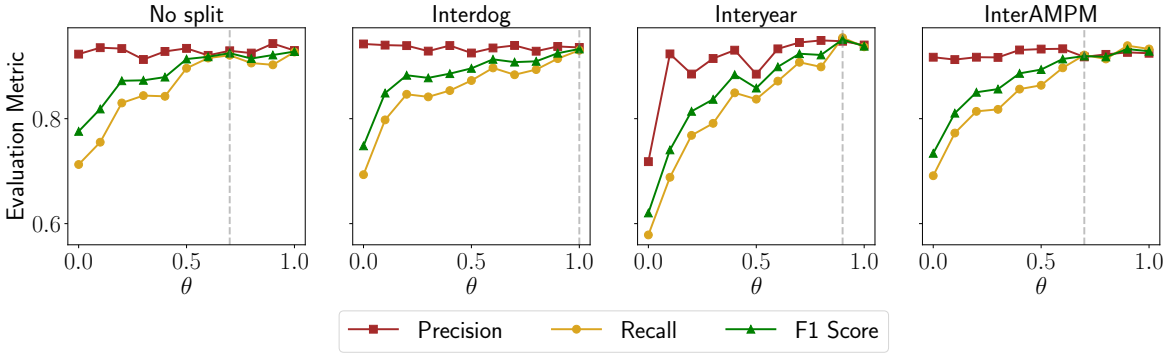



Figure 8: Tuning the rebalancing parameter for four experiments by comparing the precision, recall, and F1 score on the validation set for different choices $\theta$.

## C.2 Compute Details

All experiments for the African wild dog case study were conducted on a GPU-equipped machine with CUDA support, featuring 12 GB of GPU memory, 64 GB of CPU RAM, and 24 virtual CPU cores. To assess the performance of our pipeline on machines without GPU

support, we also ran the pipeline on a workstation with 32 virtual CPU cores and 125 GB of RAM. Additionally, the pipeline was tested on a personal Apple MacBook with an 8-core CPU and 16 GB of RAM. The time-consuming steps of the pipeline can be divided into three main processes. The first step, metadata creation, involves reading all half-day CSV files and extracting information such as year, individual identification, and the average temperature for each half-day. This step is CPU-bound and takes approximately 148 seconds to process a total of 4,276 half-day CSV files. The second step, creating training and testing splits using user-defined filters, takes around 70 seconds across all machines. The third step, creating fixed-duration windows from the filtered data, consistently takes about 25 seconds across all platforms. GPU support proves to be particularly advantageous during the final step - model training, where efficient parallelism for tensor computations accelerates the process. The machine learning model is built in the PyTorch framework. The model with 5 CNN layers, 32 output channels, and kernel size 5 has 887,237 trainable parameters. For a training set size of 14,978 samples and this model configuration with 887,237 trainable parameters, one epoch of training on the GPU-equipped machine takes an average of 2 seconds. In comparison, training on the workstation takes approximately 25 seconds per epoch, while the personal MacBook requires around 330 seconds per epoch.