Received: 3 May 2022 | Accepted: 17 April 2023

DOI: 10.1111/2041-210X.14131

Methods in Ecology and Evolution

RESEARCH ARTICLE

Open-source workflow approaches to passive acoustic monitoring of bats

Signe M. M. Brinkløv[1] | Jamie Macaulay[2] | Christian Bergler[3] | Jakob Tougaard[4] | Kristian Beedholm[2] | Morten Elmeros[1] | Peter Teglberg Madsen[2]

[1]Department of Ecoscience—Wildlife Ecology, Aarhus University, Aarhus C, Denmark

[2]Department of Biology—Zoophysiology, Aarhus University, Aarhus C, Denmark

[3]Department of Computer Science—Pattern Recognition Lab, Friedrich-Alexander-University Erlangen-Nuremberg, Erlangen, Germany

[4]Department of Ecoscience—Marine Mammal Research, Aarhus University, Aarhus C, Denmark

Correspondence
Signe M. M. Brinkløv
Email: brinklov@ecos.au.dk

Funding information
Carlsbergfondet; Indenpendent Research Fund Denmark

Handling Editor: Natalie Cooper

Abstract

1. The affordability, storage and power capacity of compact modern recording hardware have evolved passive acoustic monitoring (PAM) of animals and soundscapes into a non-invasive, cost-effective tool for research and ecological management particularly useful for bats and toothed whales that orient and forage using ultrasonic echolocation. The use of PAM at large scales hinges on effective automated detectors and species classifiers which, combined with distance sampling approaches, have enabled species abundance estimation of toothed whales. But standardized, user-friendly and open access automated detection and classification workflows are in demand for this key conservation metric to be realized for bats.
2. We used the PAMGuard toolbox including its new deep learning classification module to test the performance of four open-source workflows for automated analyses of acoustic datasets from bats. Each workflow used a different initial detection algorithm followed by the same deep learning classification algorithm and was evaluated against the performance of an expert manual analyst.
3. Workflow performance depended strongly on the signal-to-noise ratio and detection algorithm used: the full deep learning workflow had the best classification accuracy (≤67%) but was computationally too slow for practical large-scale bat PAM. Workflows using PAMGuard's detection module or triggers onboard an SM4BAT or AudioMoth accurately classified up to 47%, 59% and 34%, respectively, of calls to species. Not all workflows included noise sampling critical to estimating changes in detection probability over time, a vital parameter for abundance estimation. The workflow using PAMGuard's detection module was 40 times faster than the full deep learning workflow and missed as few calls (recall for both ~0.6), thus balancing computational speed and performance.
4. We show that complete acoustic detection and classification workflows for bat PAM data can be efficiently automated using open-source software such as PAMGuard and exemplify how detection choices, whether pre- or

Signe M. M. Brinkløv and Jamie Macaulay shared first authors.

post-deployment, hardware or software-driven, affect the performance of deep learning classification and the downstream ecological information that can be extracted from acoustic recordings. In particular, understanding and quantifying detection/classification accuracy and the probability of detection are key to avoid introducing biases that may ultimately affect the quality of data for ecological management.

KEYWORDS
automated, bat, classification, deep learning, detection, echolocation, open source, passive acoustic monitoring

1 | INTRODUCTION

Bats represent over a fifth of the global mammalian biodiversity (Burgin et al., 2018), provide key ecosystem services (Ancillotto et al., 2017; Ghanem & Voigt, 2012) and are prospective bio-indicators for habitat quality and impacts of climate change (Jones et al., 2009; Russo et al., 2021). Susceptible to impact by their longevity, slow reproductive rates and sensitivity to ecological pressures including habitat degradation and increased insect mortality (Sánchez-Bayo & Wyckhuys, 2019), many bat species are endangered or declining in numbers, directly or indirectly due to human activity (Voigt & Kingston, 2016). The need to monitor bat populations is consequently urgent and globally acknowledged to inform their future management and conservation.

The affordability, capacity and endurance of modern recorders have substantiated their use as a non-invasive method for passive acoustic monitoring (PAM) of vocalizing animals in the wild, including echolocating toothed whales and bats that actively sample their surroundings with high-amplitude ultrasound (McCordic et al., 2021; Sugai et al., 2020). However, with the upscaling of high-capacity recorders into multiple-device networks follows a need for efficient data processing and interpretation. While the performance of automated detection and classification algorithms is integral to large-scale PAM, the adoption of sophisticated and automated acoustic processing methods is only practical if they are accessible to a wide range of technical abilities through user-friendly software. Freeware options such as BatScope (Obrist & Boesch, 2018), BatClassify (https://bitbucket.org/chrisscott/batclassify) and Tadarida (Bas et al., 2017) exist for bat classification, but most workflows still rely on a patchwork of commercial hardware and/or proprietary software (e.g. Kaleidoscope Pro [wildlifeacoustics.com], SonoChiro [sonochiro.biotope.fr], SonoBat [sonobat.com]), lack the accessibility and transparency needed for the broad user community or require coding experience. Despite consensus that a successful bat PAM program needs full transparency and standardized protocols for consistency (Mac Aodha et al., 2018; Russo & Voigt, 2016; Rydell et al., 2017), we are unaware of any such efficient, commonly adopted and accessible acoustic workflow for terrestrial bioacoustics in general and echolocating bats in particular. In contrast, extensive software infrastructure exists for large-scale PAM of echolocating odontocetes (Gibb et al., 2019), such as the open-source PAMGuard toolbox (www.pamguard.org, Gillespie et al., 2008).

We posit that the quality of downstream ecological information extracted from acoustic data depends critically on the type of automated analysis workflow used. Accordingly, we outline and test four extensible analysis workflows for bat PAM, based on the open-source AudioMoth and the proprietary SM4BAT FS recorders. We review and show empirically how choices made before and at the detection stage impact the overall performance of acoustic analyses workflows integrating a deep learning species classification algorithm. We do so while demonstrating the potential of PAMGuard as an application for accessible and automated acoustic analyses workflows for bat PAM and consider the final accuracy of species classification and implications for the probability of detecting bat calls for each workflow. Lastly, we use our results to discuss recommendations for acoustic workflows for bat PAM.

1.1 | Call variation and signal-to-noise ratio

Any PAM workflow faces a two-level challenge including detection: extracting signals of interest, and classification: identification of detected signals to species or species complex, all from the intricate and dynamic soundscape of acoustic field recordings. Two main factors affect the difficulty and effectiveness of these detection and classification tasks. The first factor is the signal-to-noise ratio (SNR). Noise, whether ambient or self-noise of the recording system, impacts signal extraction by distorting spectral and temporal properties of received signals. Yet, noise is an inherent condition of PAM, as low amplitude and hence lower SNR calls tend to dominate in acoustic monitoring (Figure 1) because most of the volume monitored by a receiver is near its outer detection periphery.

The second factor affecting detection and classification is call variability. The ultrasonic calls recorded from echolocating bats are inherently affected by their directionality, the frequency-dependent absorption in air and interference from echoes, creating a strong dependence on both the bat-receiver angle and range (Goerlitz, 2018; Pedersen et al., 2022). In concert, these factors introduce complexity and variation into recorded acoustic data (Jakobsen et al., 2013; Madsen & Wahlberg, 2007), which is further confounded by context-dependent intra- and interspecific call variation.

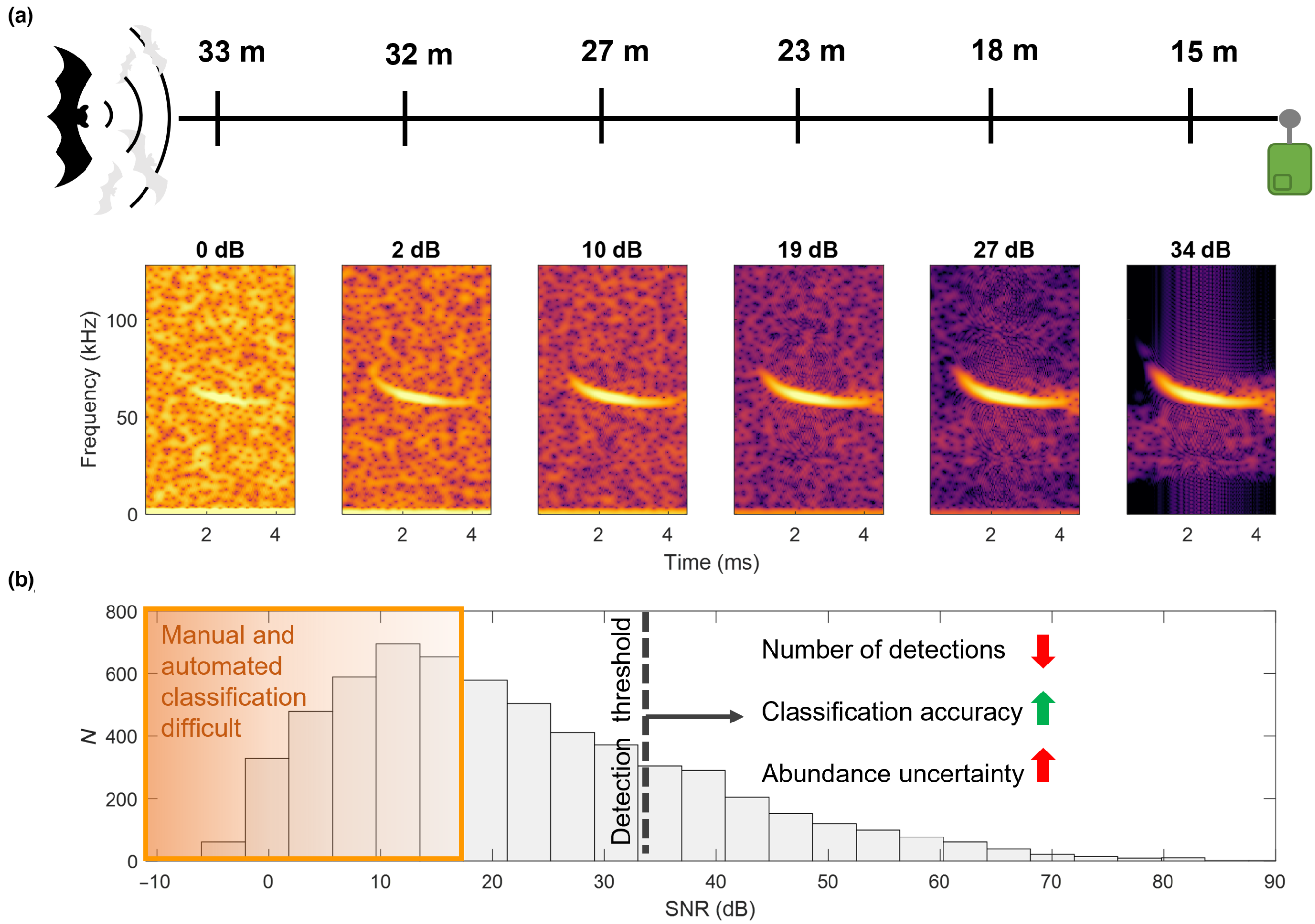

FIGURE 1 (a) The signal-to-noise ratio (SNR) of the same bat call received on-axis by a recorder (green box) with added noise simulating different bat recorder ranges (assuming a source level at 10 cm of 121 dB re 20 µPa [rms] and an ambient noise level of 20 dB re 20 µPa above 10 kHz, Merchant et al., 2015). See Supporting Information S3 for details on SNR estimation. (b) Typical SNR distribution of detected calls (*N*) with most produced at greater ranges from the recorder and, for geometric reasons, of lower SNR. With decreasing SNR, detection success drops as calls fade into background noise. Increasing the SNR threshold increases the number of correct detections for any detector, but at the expense of less detections overall, consequently increasing the uncertainty of abundance estimates.

Lower SNR calls also associate with distinct patterns of behaviour, including stealth echolocation (Goerlitz et al., 2010), buzzing in the final stage of prey capture (Stidsholt et al., 2021) and communication signals (Knörnschild et al., 2017) that may contribute important ecological information.

1.2 | Probability of detection and SNR

The overall probability of detecting a call in a recording depends on the SNR (Darras et al., 2020). The average probability of detection ($\hat{P}$) across all calls within the recorder detection range allows estimation of the fraction of calls missed by the PAM system and, in turn, the true number of calls inside a specified area (Buckland et al., 2001). If the average call rate is then known from independent data, the number of detected calls per time unit can be converted into absolute density (number per km^2) of bats (Marques et al., 2013). While behavioural and ecological parameters that contribute to $\hat{P}$ for a given species in each habitat must be measured by other means, $\hat{P}$ usually scales inversely with noise (whether ambient or system noise is dominant); increasing noise → decreasing SNR of received calls → decrease in $\hat{P}$. This means that $\hat{P}$ is intrinsically linked to the overall soundscape in the habitat unless the self-noise of the recording chain is consistently above the ambient noise in the relevant frequency band.

1.3 | Automatic detection and classification

A skilled manual analyst offers excellent visual pattern recognition for extraction of lower SNR calls from PAM data and can deal with inconsistencies like unexpected noise sources and atypical call structure. By itself, however, the manual approach is slow, expensive and unfeasible in a large-scale PAM framework. Automated algorithms, in contrast, require extensive training to deal with data complexity and variability, but add speed and consistency. Hybrid approaches allow results output by automated algorithms to be rapidly user-visualized and validated, exploiting the best of both (López-Baucells et al., 2019).

An automated workflow archiving and running on continuous, full-bandwidth ultrasound recordings without information loss is rarely feasible as data management becomes prohibitively time and space consuming for larger scale surveys, even if duty cycling and compression algorithms (Johnson et al., 2013) are used. Bat calls are short (often <25 ms) compared to the mean call interval and time between bat encounters, so only a small fraction of a continuous recording contains bat calls, even in densely populated habitats. Consequently, analysis of bat PAM data is typically split into two stages: *detection*; often run onboard the recording device as a trigger algorithm prompted by default or user settings to save only signals or sequences of potential interest and reduce the data volume passed to the second stage; the *classification* of saved detections for species identification.

Thus, *detection* is an important stage for large-scale bat monitoring as it supports data management. An efficient detection algorithm should extend effective recording time, reduce data volumes and transfer time, yet maintain data integrity, to yield high-quality non-biased outputs. This can be achieved by running the detector with a high sensitivity to avoid missing emitted calls[1] but potentially creating many false positives, a trade-off which should be considered in the context of purpose. Further offline analyses by automated *classification* algorithms are therefore required to identify false-positive *detections* and provide species identification from raw waveform information saved for each detection. Crucially, noise affecting the probability of detecting target sounds should be stored or quantified continuously to allow a running quantification of $\hat{P}$.

Classification algorithms are inevitably more complex than detection algorithms and frequently employ machine learning to achieve high accuracy. Random forest and convolutional neural networks (CNNs) have shown promising ability to detect and/or classify bat calls (Bas et al., 2017; López-Baucells et al., 2019; Roemer et al., 2021; Walters et al., 2012). However, no common workflow is currently widely adopted, and the robustness of machine learning approaches applied across different workflows is sparsely documented. Hence, the performance of classification models may sometimes be significantly affected by type of hardware and onboard trigger algorithms, noise fluctuations and/or other parameters not included in training data. Accurate automated classification algorithms can be vital for effective PAM studies, especially for monitoring rare species (Caillat et al., 2013). Thus, understanding how they perform across different hardware and environmental domains is a critical step in creating practical and widely applicable acoustic workflows.

2 | MATERIALS AND METHODS

2.1 | Hardware, recording sites and data

We used two common recorders to collect acoustic data at woodland edges in Denmark (coordinates and recording specifications in Table S1): the proprietary SM4BAT FS (Wildlife Acoustics Inc.) and the open-source AudioMoth (Hill et al., 2018, openacousticdevices.info).

To test how efficiently each workflow detected low SNR calls and classified species, we compared triggered and continuous recordings per recorder type by pairwise deployment of two SM4BAT (16-bit recorder with external microphone) and two AudioMoths (12-bit recorder with integrated MEMs microphone), respectively. Within each recorder pair, one was in trigger mode while the other recorded continuously, with otherwise identical settings (Table S1).

To illustrate the importance of concurrent noise sampling to estimate how noise affects call detection probability over time, we also analysed an extended 40-night dataset of continuous (sunset–sunrise), full-bandwidth recordings from an SM4BAT.

2.2 | Acoustic analyses workflows

To explore how the choice of detection or trigger algorithm and hardware affects both the detection and subsequent classification of bat echolocation calls, we tested four acoustic analysis workflows each based on a different detection stage (Figure 2). Two ran *offline* on continuous recordings, using a deep learning and PAMGuard's dynamic threshold detection algorithm respectively. The other two workflows used the detection algorithms integrated onboard the AudioMoth (static trigger), and the SM4BAT FS (adaptive trigger). Detections from each workflow were passed to the same deep learning classification stage for species identification and workflow performance evaluated against manually annotated data from an hour of continuous and triggered recordings from both AudioMoth and SM4BAT recorder pairs. Manual detection and classification were performed by an expert bat bioacoustics analyst (SB) based on visual inspection of raw waveforms, spectrograms and spectra in Raven Lite 2.0.1 (K. Lisa Yang Center for Conservation Bioacoustics, 2016).

2.2.1 | Full deep learning workflow

The full deep learning workflow used a deep learning *detection* model based on ANIMAL-SPOT (Bergler et al., 2022), a ResNet-18 CNN for binary and multi-species classification. We trained the binary CNN (bat or no bat) on manually annotated noise clips (n = 17,964) and call clips (n = 6933) from five focal species for which we had reliable audio files: *Pipistrellus pygmaeus* (n = 1570), *Nyctalus noctula* (n = 1052), *Eptesicus serotinus* (n = 1502), *Myotis nattereri* (n = 1528) and *Myotis daubentonii* (n = 1280). All audio clips were recorded by SM4BAT recorders (Table S1).

This workflow processed sequential overlapping segments of raw acoustic data (5 ms window, 2.5 ms overlap), converting each into a noise reduced spectrogram image passed to the CNN and assigned a probability of containing a bat call. Any segment assigned a probability >0.7 was considered a bat detection. Immediately surrounding segments above the prediction threshold were merged

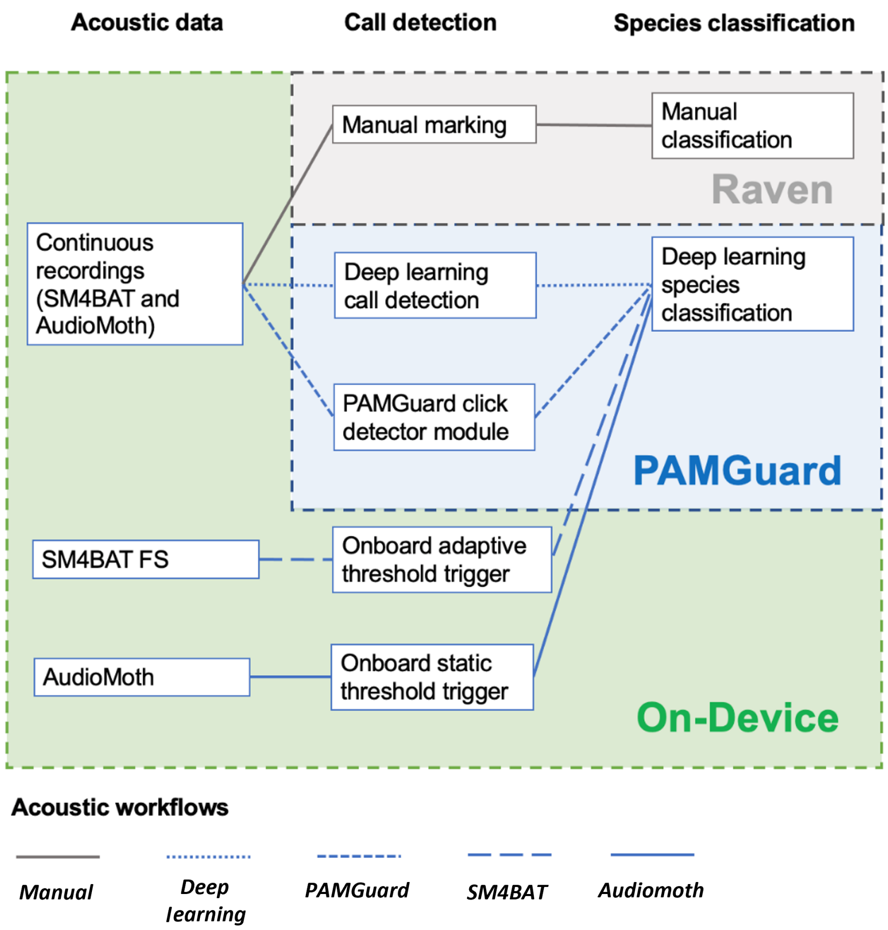

FIGURE 2 The four automated workflows tested and compared against manually annotated data. Data were sampled at 384 kHz (SM4BAT) and 256 kHz (the highest sample rate available for the AudioMoth at the time of data collection) and all resampled to 256 kHz before entering the deep learning classification step. Each workflow used the same deep learning classification module in PAMGuard but different detection approaches; Full deep learning workflow—deep learning detector on continuous recordings; PAMGuard workflow—PAMGuard click detector module on continuous recordings; SM4BAT workflow—on-device adaptive threshold trigger; AudioMoth workflow—on-device static threshold trigger.

into one detection and passed to a downstream CNN *multi-species* classifier in ANIMAL-SPOT for species identification. Supporting Information S4.2 provides details on the full deep learning workflow, including training parameters.

2.2.2 | PAMGuard click detector workflow

The PAMGuard workflow processed continuous recordings using PAMGuard's click detector module (Supporting Information S1 includes user guide and settings); a simple energy detector, which saves any transients within a specified frequency band with zero-to-peak amplitude at a set above noise threshold of 7 dB (equivalent to a bat call zero-to-peak level being 13 dB above background/self-noise: see Supporting Information S5.2), to reduce the number of missed calls. Each detection was split into 5 ms segments, passed to the CNN classifier. Calls <5 ms were zero-padded and each 5 ms chunk of calls >5 ms was classified.

2.2.3 | AudioMoth workflow

The AudioMoth workflow imported AudioMoth triggered detections into PAMGuard and segmented them into 5 ms chunks, passed to the deep learning classifier. The AudioMoth (firmware v1.4.4) trigger algorithm uses a static detection threshold at a defined amplitude level set here at 0.00323 (normalized amplitude = −50 dB relative to full-scale of the recording), filters and splits data into 32 kB chunks, saving only those above the threshold level to the μSD card. The AudioMoth also stores an ~64 ms data chunk every 10 s.

2.2.4 | SM4BAT workflow

The SM4BAT workflow ran in PAMGuard on SM4BAT triggered detections. The detection algorithm (firmware v2.2.7) used by this proprietary hardware is unpublished but quoted as using an adaptive threshold that tracks background noise across frequency bands above a specified minimum frequency and per default saves detections with SNR (as opposed to absolute received level) above a trigger threshold of 12 dB (https://www.wildlifeacoustics.com/user-guides).

2.3 | Automated species classification

We ran the same classification stage in all workflows, using ANIMAL-SPOT to train a multi-class bat species model (Bergler et al., 2022) on the same data as the binary deep learning detector but supplemented with manually annotated AudioMoth ($n = 1786$)

and SM4BAT (n = 11,150) random noise clips. Training data were augmented by adding simulated clipping and aliasing (Supporting Information S4.1). Adding AudioMoth noise clips and augmenting training significantly improved accuracy compared to an early version of the classifier based only on SM4BAT data. While adding manually annotated calls from an AudioMoth may have further improved accuracy, we intentionally did not do so to demonstrate the *practical* implementation of deep learning using different recorders and in different environments, that is, it is far more feasible to retrain a classifier using random noise clips than to manually annotate many additional bat calls.

2.4 | Long-term dataset and estimation of detection probability

Common goals of automated PAM workflows are to extract ecological information of importance to the monitoring objective, for example, presence/absence, activity indices, call types reflecting specific behaviours, spatio-temporal trends and abundance estimates. For many of these parameters, the average probability of detecting bat calls ($\hat{P}$) is vital to reduce spatial and temporal bias in acoustic data. Over a single deployment, $\hat{P}$ will scale with changes in background/system noise (see Section 1.2); we quantified the potential magnitude of these changes in $\hat{P}$ over the course of a 40-day extended dataset to demonstrate the potential downstream implications of the ambient/system noise logging capabilities of each workflow.

The average probability of detecting bat calls ($\hat{P}$) over time could not be calculated without information about distance to the calling bats.[2] Instead, we estimated $\hat{P}$ using a Monte Carlo simulation (Frasier et al., 2016; Küsel et al., 2011), randomly distributing N simulated bats in horizontal space around a recorder with the height, orientation, source level and beam profile of each bat parameterized from predefined distributions derived from literature (see Table S2). The received level on the recorder was determined per simulated bat assuming spherical spreading propagation with appropriate absorption. The received SNR was then calculated as the received level minus a predefined noise level; a simulated bat was considered detected or not by sampling a probability density function of SNR (based on results in Figures 4 and 6). Overall $\hat{P}$ was then calculated as the number of simulated calls successfully detected, divided by the total number of simulated bats, N. Each simulation was repeated 20 times and mean $\hat{P}$ calculated. We ran simulations to estimate $\hat{P}$ as a function of detection thresholds ($\hat{P}(n)$) between 33 and 73 dB re 20 μPa pp (assuming background/self-noise limits of 20–60 dB re 20 μPa rms + 13 dB detector threshold within the 23–90 kHz detector trigger band; see Supporting Information S5.2). These were the minimum and maximum expected detection thresholds based on the noise levels in the extended continuous dataset and settings of the PAMGuard click detector module (Supporting Information S1).

The extended dataset of continuous recordings was analysed using the PAMGuard workflow to balance speed and accuracy. The PAMGuard click detector module continually measures ambient/system noise in the detector filter band. This was converted to an equivalent detection threshold by adding the threshold of the click detector (see Supporting Information S5.2). The detection threshold was then calculated every second and mapped to the simulated $\hat{P}(n)$, providing a continuous time series of the probability of detecting a bat call. Combined with the number of detected calls, we could then estimate call density (calls per km^2 per hour).

2.5 | Software

We used PAMGuard as the primary analysis tool for each workflow and primed it for terrestrial analysis with new data display options and a comprehensive deep learning module for species classification (https://github.com/PAMGuard/PAMGuard/tree/main/src/rawDeepLearningClassifier). All workflows can be implemented fully in PAMGuard, or the output exported for further analyses. We used the PG-MATLAB library (http://www.pamguard.org/48_MATLABRcode.html) to extract data from PAMGuard for further analysis and plotting, but the same features are available through R (open-source, PAMGuard-R library, https://github.com/TaikiSan21/PamBinaries).

3 | RESULTS

We evaluated the detection performance of each workflow by comparing the number of automatic detections to manual annotation of an hour-long continuous recording from the AudioMoth recorder pair and the SM4BAT recorder pair. Performance was quantified by calculating recall (the number of true automatic detections divided by the number of manually annotated calls) and precision (the number of true automatic detections divided by the total number of automatic detections) for each workflow using discrete detections (dd) and total duration of detections (td; see Supporting Information S6).

3.1 | AudioMoth detection performance

For the hour-long AudioMoth recording analysed (Figure 3), the PAMGuard workflow had the highest false-positive call detection rate (Recall dd = 0.60, td = 0.18, Precision dd = 0.08, td = 0.11). The deep learning detector had a substantially lower false alarm rate and a slightly improved recall rate (Figure 4) but needed 20 times longer processing time even with the use of a graphics processing unit (GPU; Recall dd = 0.63, td = 0.17, Precision dd = 0.22, td = 0.32). The AudioMoth missed many lower SNR calls, resulting in the lowest overall recall (Recall dd = 0.33, td = 0.16, Precision dd = 0.71, td = 0.13).

3.2 | SM4BAT detection performance

The SM4BAT dataset was recorded on a different date and location than the AudioMoth dataset, with fewer bat calls overall (Figure 5)

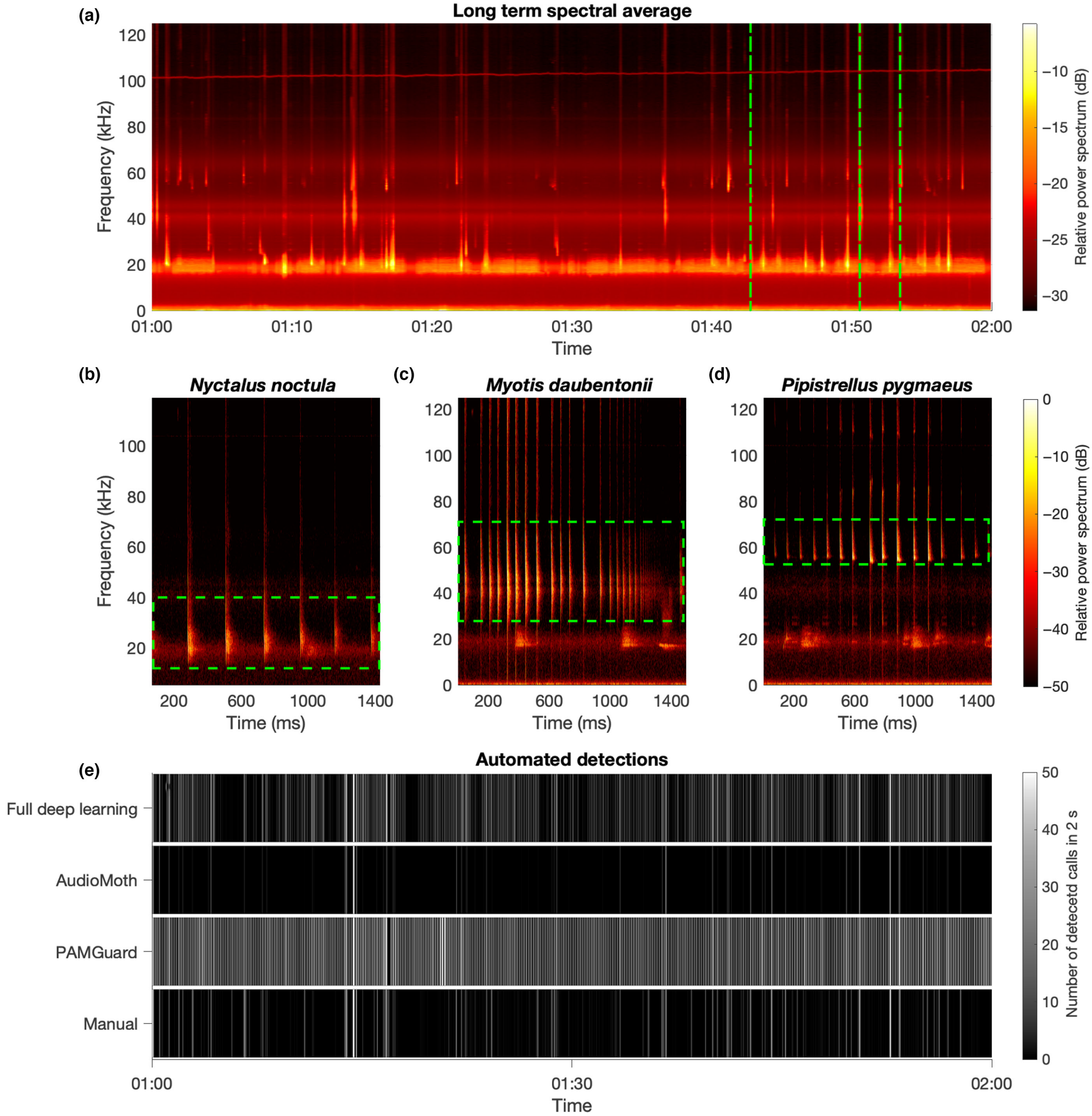

FIGURE 3 (a) Long-term spectrogram (Hann window, FFT size: 1024 samples, no overlap) of a continuous 1-h AudioMoth recording. Green lines: call sequences from three species, enlarged in spectrograms (Hann window, FFT size: 512 samples, 128 sample overlap) below (b–d). Green boxes: frequency range of the target calls among artefacts and other species' calls. (e) Ribbon plots showing detections from the full deep learning, AudioMoth and PAMGuard workflow plus manual annotations. Each vertical line on the ribbon plots represents the total number of detections per 2-s period.

and a higher proportion of low SNR calls (compare Figures 4 and 6).

For this recorder, the background/self-noise was generally lower and more consistent, resulting in a lower false-positive rate for all detectors. The detection performance of the SM4BAT workflow was less consistent above 10 dB SNR (Recall dd = 0.53, td = 0.54, Precision dd = 34.3, td = 0.05) than the PAMGuard (Recall dd = 0.56, td = 0.31, Precision dd = 0.12, td = 0.12) and deep learning (Recall dd = 0.60, td = 0.37, Precision dd = 0.81, td = 0.73) workflows, but all three had similar overall detection (dd) recall rates (Figure 6). Note that the SM4BAT default trigger results in comparatively long (3 s) recordings, each including multiple calls, causing precision to be >1 on a single detection basis, but very low on a time duration basis because continuous recordings occurred between multiple calls.

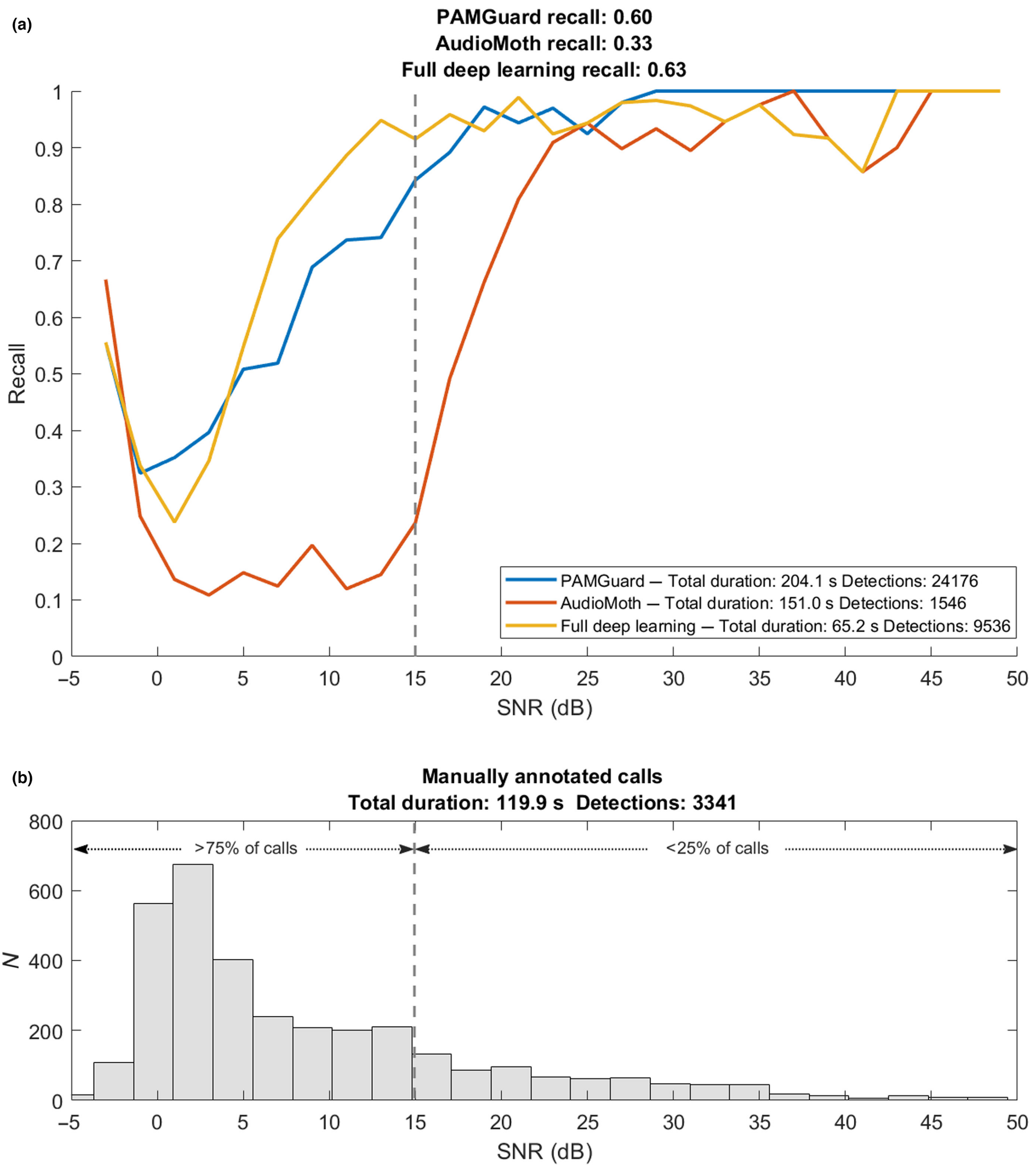

FIGURE 4 Recall versus SNR (a) and SNR distribution of manually annotated calls (b) for the 1-h AudioMoth recording in Figure 3 (see Supporting Information S3 for details on SNR estimation). At SNR>15 dB, the PAMGuard and deep learning detection algorithms perform well (recall>0.8). At lower SNR (<15 dB), where most calls occur (>75%), all three detectors decline in performance. Total duration of raw data and number of detections saved by each algorithm.

3.3 | Call Classification

The deep learning *detector* balanced high recall with a moderate false-positive rate (Figures 4 and 6). However, the aim of the detection stage is simply to record all sounds of interest allowing a secondary *classification* stage to remove false positives and assign to species level. The classification performance of each workflow depended on both the initial detection stage, including the size and

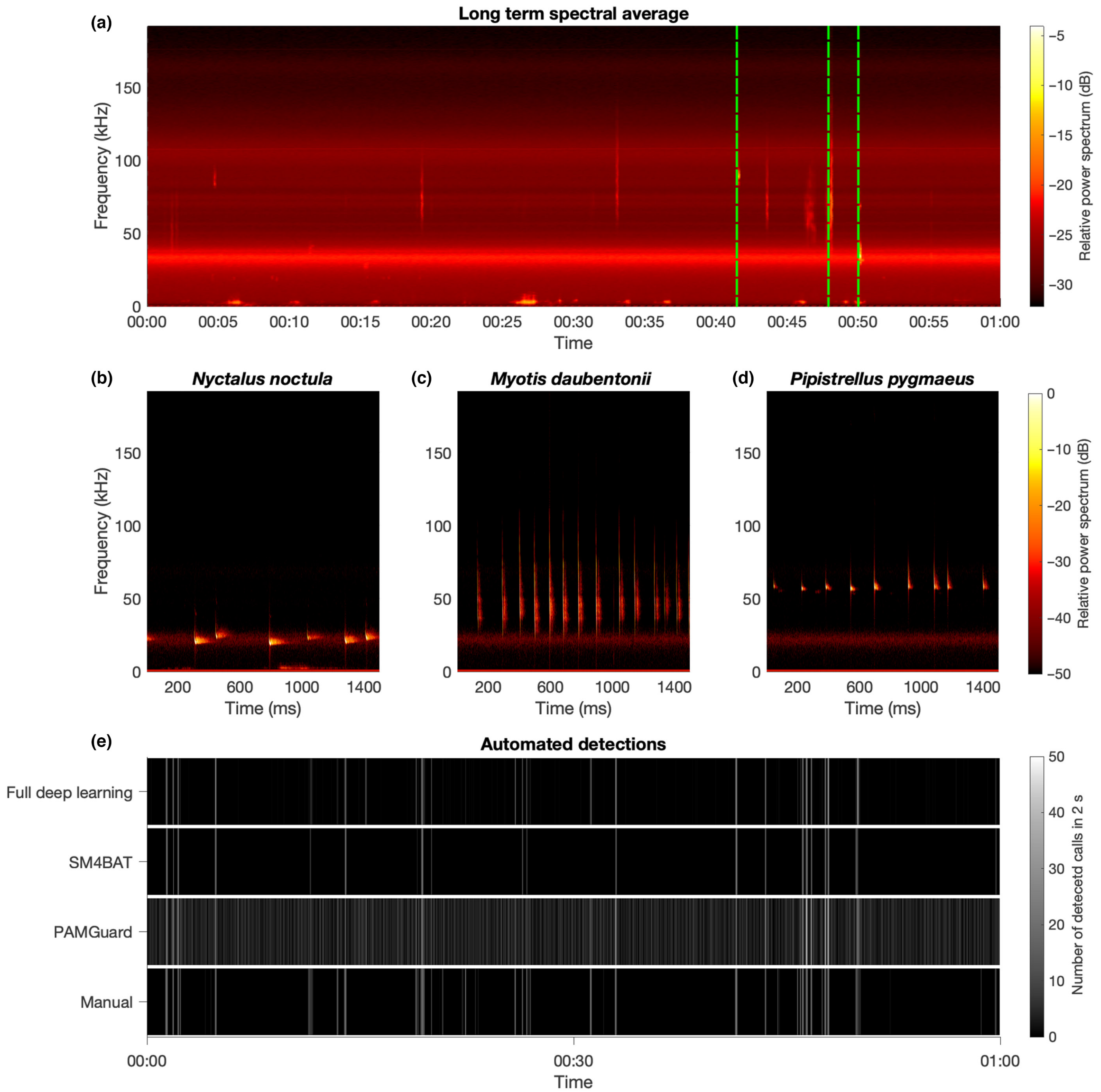

FIGURE 5 (a) Long-term spectrogram (Hann window, FFT size: 1024 samples, no overlap) of a continuous 1-h SM4BAT recording. Green lines: call sequences from three species, enlarged in spectrograms (Hann window, FFT size: 512 samples, 128 sample overlap) below (b–d). (e) Ribbon plots showing detections from the full deep learning, SM4BAT and PAMGuard workflow plus manual annotations. Each vertical line on the ribbon plots represents the total number of detections per 2-s period.

SNR of the raw waveform snippets saved for upstream processing, and on the classifier itself. Compared against the manual annotations from an AudioMoth and SM4BAT, the classifier, respectively, discarded 22% and 25% overall of the manually identified calls as noise (Figure 7), while correctly identifying 82% and 93% of the remaining calls. The most common call misclassification was of *N. noctula* as *E. serotinus* (12% and 40%), species known to emit similar calls that may overlap in frequency. Notably, most misclassified calls had a lower SNR. If only calls with SNR >15 dB (above which most detectors perform well, Figures 4 and 6) were considered, the classification performance significantly improved with ~0.5% of calls misclassified as noise and (except for *N. noctula*) 97% of calls correctly identified (see Supporting Information S7).

With the classifier integrated, the full deep learning workflow outperformed the three other workflows, accurately classifying 39%–67% of calls per species, except for *N. noctula* in the SM4BAT

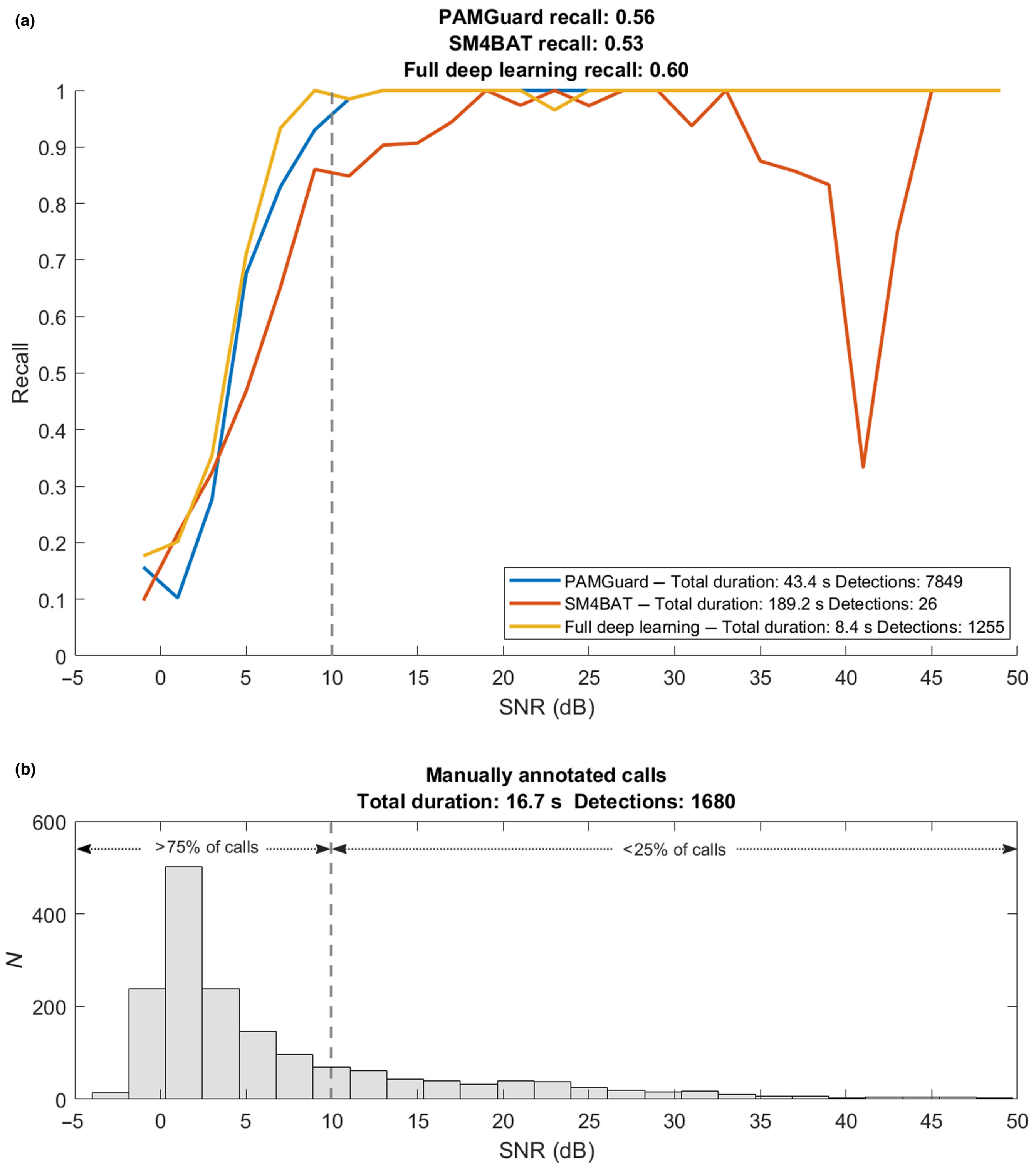

FIGURE 6 Recall as a function of SNR (a) and SNR distribution of manually annotated calls (b) for the 1-h SM4BAT recording in Figure 5. At SNRs above 10 dB (<25% of detections), the PAMGuard and deep learning detection algorithms performed consistently well (recall >0.9). Total duration of raw data and number of detections saved by each algorithm.

dataset, for which classification accuracy across workflows was ≤20% (Figure 8). The PAMGuard and SM4BAT workflows had classification accuracies of 20%–47% and 11%–59%, respectively, with the AudioMoth workflow lagging (classification accuracy 14%–34%) as expected from its detection performance.

3.4 | Long-term dataset

The PAMGuard workflow was used to analyse the long-term dataset and assess the impact of background noise levels on call detection probability ($\hat{P}$). PAMGuard analysed the 423-h dataset of continuous

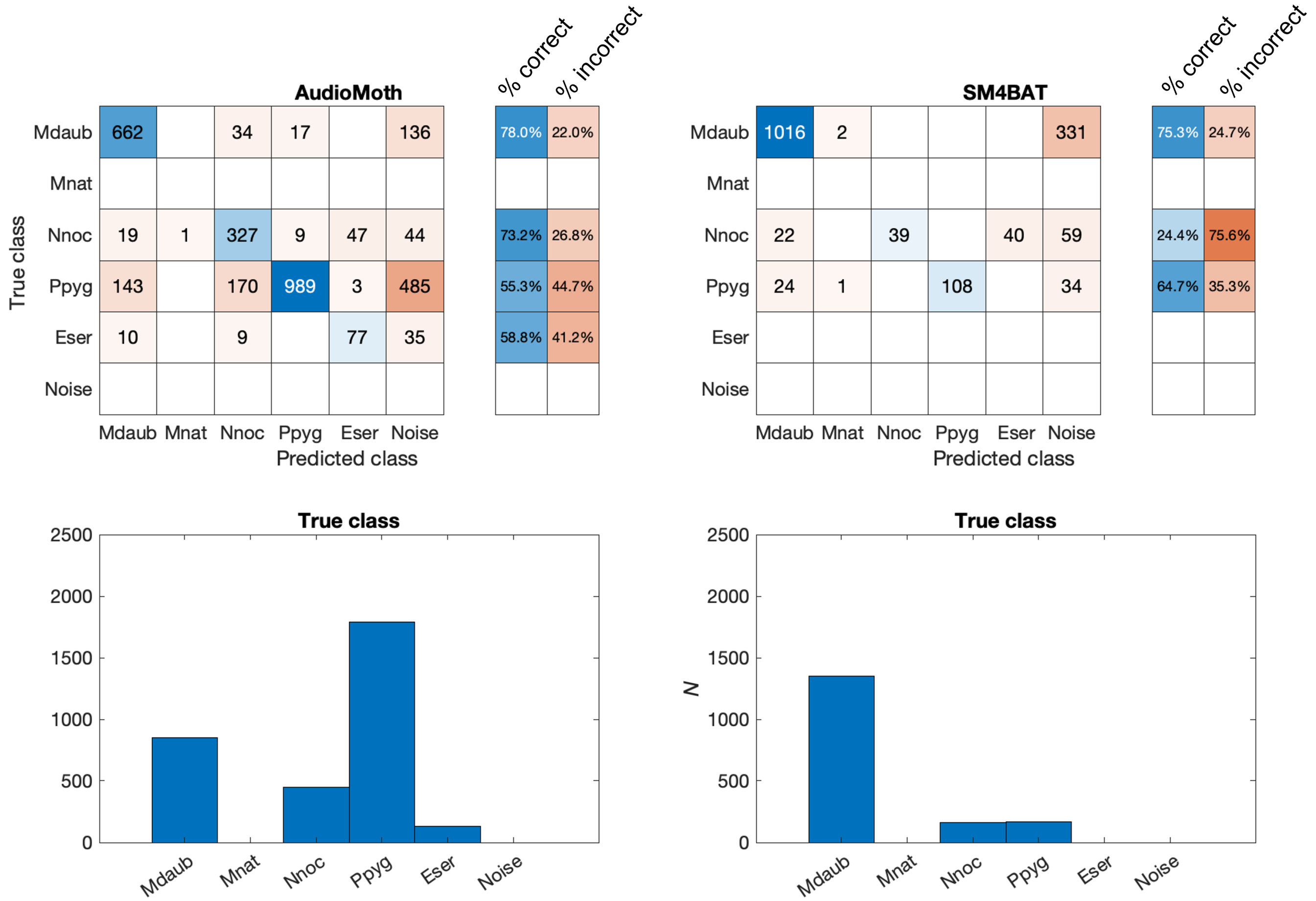

FIGURE 7 Deep learning classifier performance on manually annotated data from an AudioMoth (left) and SM4BAT (right). Top: confusion matrices stating true versus predicted class per species. Blue and red shading: correct and incorrect classification respectively. Numbers increase with colour shading. Right-hand column summaries indicate percent overall of correctly (left) and incorrectly (right) classified detections per species. Blank rows reflect no manual identifications of Mnat in the AudioMoth dataset and of Mnat and Eser in the SM4BAT dataset. Bottom: bar plots of total number of manually annotated calls per species. Eser, *Eptesicus serotinus*; Mdaub, *Myotis daubentonii*; Mnat, *Myotis nattereri*; Nnoc, *Nyctalus noctula*; Ppyg, *Pipistrellus pygmaeus*.

recording in 35.3 h (detection: 14.3 h, classification: 21 h), ×12 faster than real-time.

Figure 9, which summarizes estimates of $\hat{P}$ and call density of *P. pygmaeus* over time, demonstrates a major impact of noise levels on call detection probability; $\hat{P}$ changes between ~0.0025 and 0.015 (over ×6 variation) over the entire recording period. See Supporting Information S5.2 for more information on how $\hat{P}$ maps to noise.

4 | DISCUSSION

Each step of an automated workflow for acoustic analyses can be approached in myriad ways, with the optimal detection and classification choices depending on the research questions asked. Here, we show that species classification accuracy and the quality of downstream ecological information extracted from acoustic data by automated workflows incorporating a deep learning classifier depend critically on the recorder type and pre-deployment setup choices.

The initial *detection* stage increases monitoring time, reduces the data load, post-processing time and storage requirements and is key to both the species classification accuracy and call detection probability. No perfect detector, able to identify and save raw acoustic clips of every call, exists as performance of any system is ultimately limited by the SNR set by ambient noise or electronic self-noise, distance, propagation conditions, orientation and source level of a calling bat. Detector and settings choices pose an inevitable trade-off between false positives (non-bat detection) and missed true detections, representing a question of managing, rather than avoiding, errors. Lowering the detection threshold reduces the proportion of missed calls but yields more false detections. Particularly for low duty cycle bats, a *detection* stage running at a high false-positive rate still greatly reduces data volumes while minimizing missed calls. How detection performs across changing noise and soundscape conditions is also vital. Thus, a workflow must continuously record and quantify noise to estimate the relative or absolute probability of detection and allow recordings to be compared across space and time without bias.

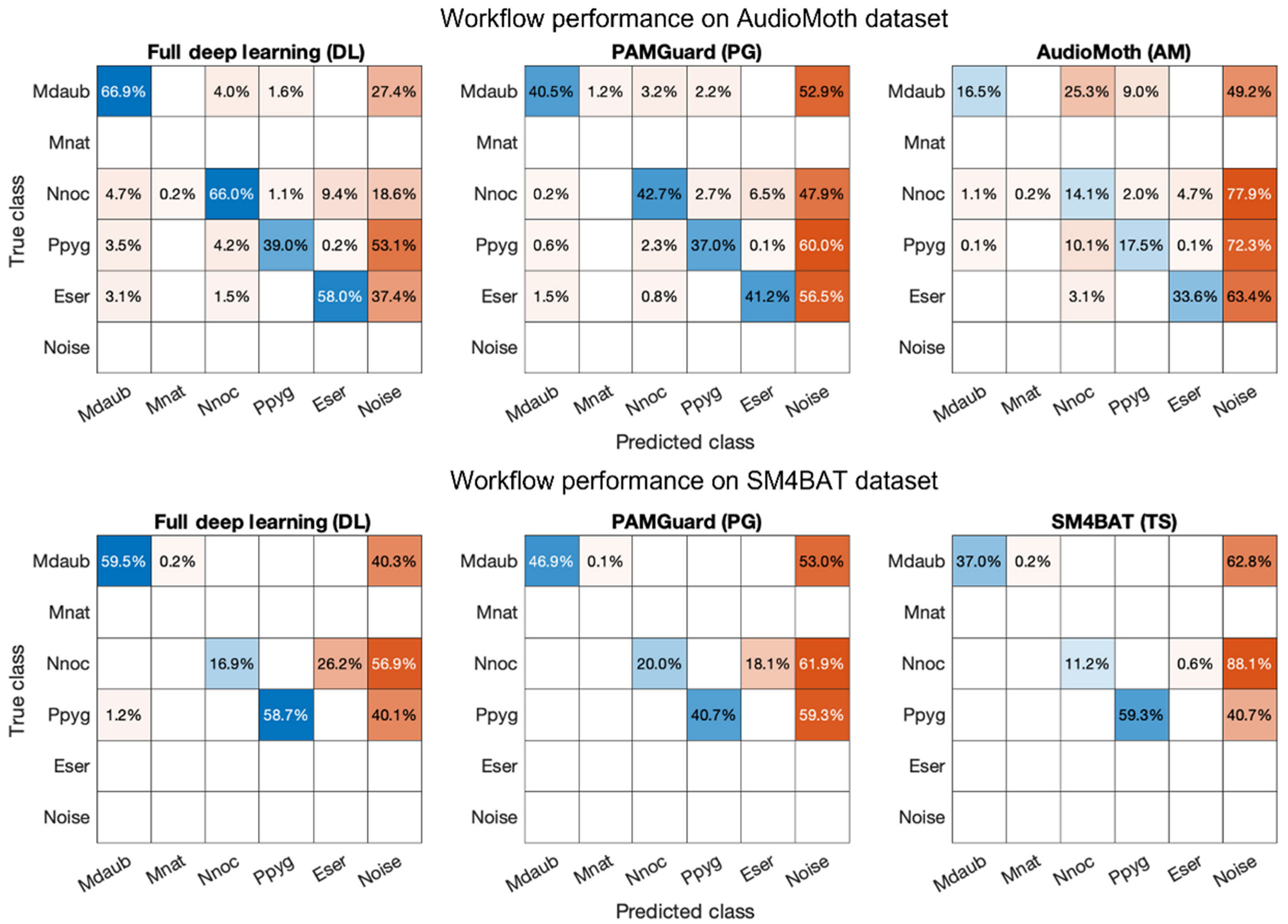

FIGURE 8 Confusion matrices of the final species classification stating the true versus predicted class per species from each automated workflow. Blue and red shading: % correct and incorrect classification respectively. Numbers increase with colour shading. Top panel: performance of the full deep Learning, PAMGuard and AudioMoth workflows on the AudioMoth dataset. Bottom panel: performance of the full deep learning, PAMGuard and SM4BAT workflows on the SM4BAT dataset. The full deep learning workflow performed best overall, followed by the PAMGuard workflow. Eser, *Eptesicus serotinus*; Mdaub, *Myotis daubentonii*; Mnat, *Myotis nattereri*; Nnoc, *Nyctalus noctula*; Ppyg, *Pipistrellus pygmaeus*.

Post detection, automated *classification* then must trade-off false positives with the number of missed calls depending on the study objectives. Relaxed settings are permissible if the target species is common and species prone to misclassification as the target species are absent or rare in the monitored area, that is, most calls are likely from the target species. The opposite applies for rare species; even low false-positive rates can lead to substantial overestimations of their abundance (Caillat et al., 2013), necessitating more restrictive settings (e.g. a higher detection threshold—see Supporting Information S7) at the expense of a lower overall detection probability.

The quality of downstream ecological information extracted from acoustic recordings is therefore intrinsically linked to the choice of workflow. Understanding *changes* in the probability of *detecting* a call is key to many acoustic surveys because it reduces bias in data collected across time and space. However, modelling changes in the probability of detection over a recording period usually requires *continuous* measurements of ambient/system noise and quantification of how detection recall relates to SNR. Species classification performance is then also dependent on the output detection stage, not just the type of classifier used. Chasing lower SNR calls might result in more detected calls, but at the expense of more unstable automated detection and subsequent classification performance between workflows (Figures 4 and 6). An automated acoustic workflow therefore requires a balance between performance and the lower bound SNR of detected calls in addition to quantification of detector performance with SNR, classifier performance and continuous noise sampling so that relative differences in workflow performance across noise conditions and between target species do not introduce bias into downstream data.

Each of the four workflows we compared was based on a different detection algorithm, including two on-board triggers (SM4BAT: adaptive trigger, AudioMoth: static trigger) and two post-processing detectors (deep learning and PAMGuard detection module). We found clear differences in performance: The

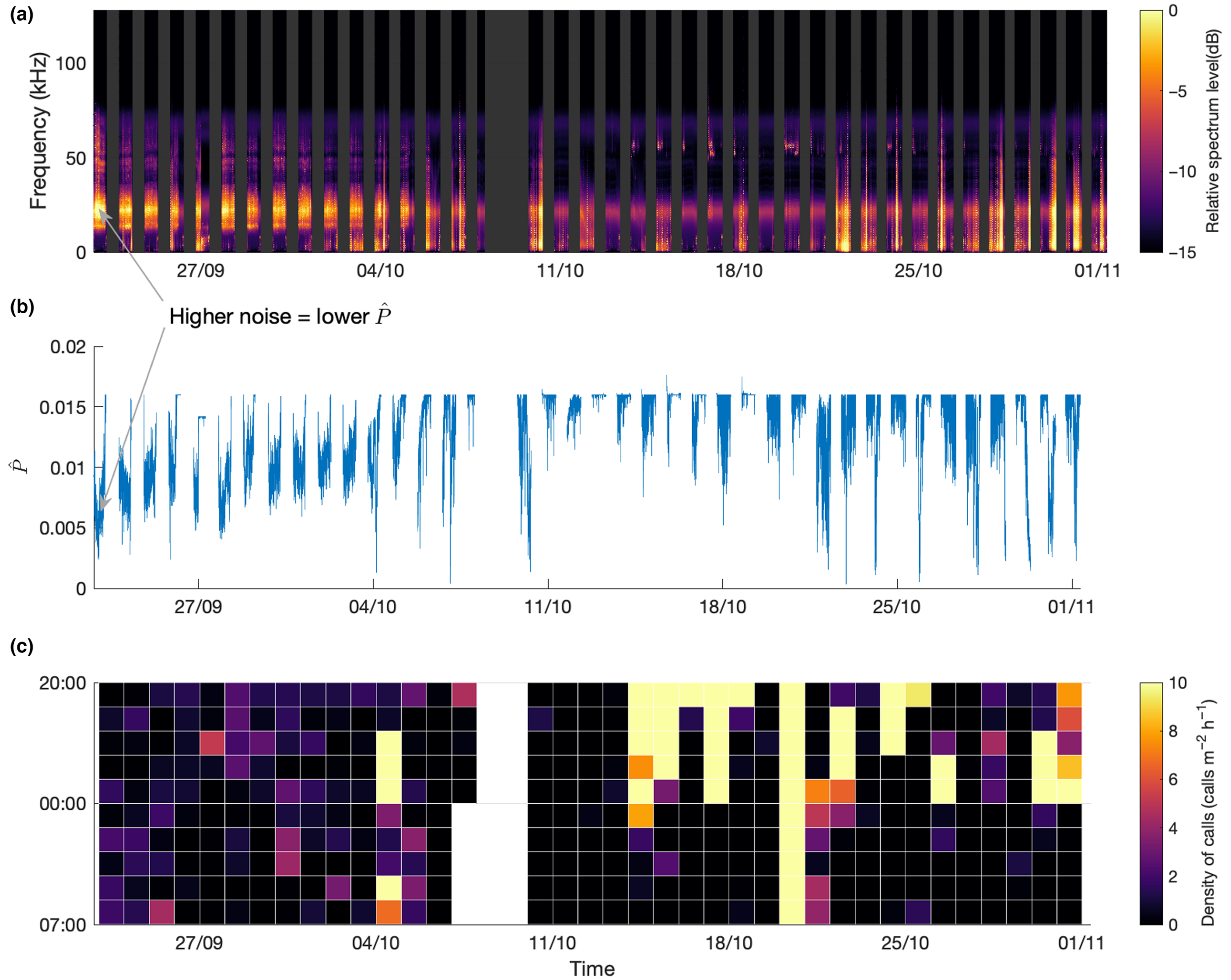

FIGURE 9 Visualization of the dataset recorded continuously every night for 40 days. (a) Long-term spectrogram showing changes in background noise and bat call levels. (b) The probability of detection ($\hat{P}$) of a single bat call within a 40-m recorder radius. (c) Calendar plot of the density of calls classified as *Pipistrellus pygmaeus*, calculated by dividing the number of calls every hour by $\hat{P}$. $\hat{P}$ is highly variable and can change by up to ×6 depending on noise levels, consequently, altering ecological inferences from acoustic data.

full deep learning workflow performed with best accuracy but at the slowest processing speed. The SM4BAT and AudioMoth workflows emphasized data reduction at the detection stage, causing particularly the AudioMoth to miss many detections, and the PAMGuard workflow balanced the two, yielding many false-positive detections, yet with better classification accuracy than the two workflows depending on onboard triggers (Figures 3–6). These differences reflect the relative emphasis on data reduction and integrity and demonstrate how detection settings influence the performance of the entire acoustic workflow, including final call classification (Figure 8). While manual analyses in our case outperformed all automated workflows, annotation was roughly ×10 slower than real time. Notably, most misclassified calls were of lower SNR and classified as noise, resulting in a lower overall detection probability of calls which can be compensated for in subsequent analysis.

4.1 | SM4BAT workflow

The SM4BAT workflow had similar detection and classification accuracy to the other workflows tested (Figures 5, 6 and 8). The proprietary detection algorithm records a 3-s data window for transients above a default 12 dB adaptive threshold even if no other transients pass the threshold within this window. The algorithm continuously measures but, to our knowledge, does not log in-band ambient/system noise although this could be done onboard the device using minimal additional storage and processing power. Consequently, it is impossible to directly estimate changes in the probability of detecting a bat call ($\hat{P}$), which in our example dataset (Figure 9) is significantly affected by ambient noise. In addition, most bat calls are short (<100 ms), suggesting the use of a recording window much shorter than the SM4BAT default (3 s) to save memory. The long windows capture ensuing low SNR calls but create an unfortunate dependency between the recorded calls within each window, that is, the probability of detecting a low SNR call is based on whether other calls of higher SNR immediately precede or follow it. Consequently, noise cannot be continuously quantified and related to each detected call, making it difficult to relate detection probability to SNR. Encouragingly, the SM4BAT compares favourably to other workflows and using non-default settings and/or alterations to the detection firmware could solve the issues above, but we still emphasize the value of implementing open-source algorithms to improve transparency, user understanding and reduce the risk of errors.

4.2 | AudioMoth workflow

The AudioMoth onboard static trigger workflow had the lowest recall (0.33, Figure 4) and subsequent inferior classification performance (Figure 8) but presented certain advantages over the SM4BAT workflow. The AudioMoth trigger uses a static threshold that does not adapt to background noise levels and records any 64 ms sound window (on the order of bat call intervals) in which a single bin passes a user-specified number of amplitude bins (here: 1024). Thus, the probability of detection equals the probability that a single call passes above threshold and because it is static, $\hat{P}$ does not change with noise levels if the detection threshold is set high enough. This simplifies data interpretation, but the threshold must be set significantly above the expected noise level to prevent continuous recordings. Furthermore, the AudioMoth uses an analogue-to-digital converter of much lower dynamic range (12-bit) than the standard 16-bit, leading to a relatively insensitive detector and low recall (Figures 3 and 4). This effectively means that a triggered AudioMoth monitors a lower habitat volume (Darras et al., 2020), but allows estimation of detection probability. Importantly, the ambient/system noise can be extracted, and the SNR quantified for each detection from the parts of each 64 ms snippet without bat calls.

4.3 | Full deep learning workflow

Our full deep learning detection and classification workflow was the most accurate in final species classification (Figure 8) but over ×40 slower than real time using a standard CPU and just under real-time speeds using a mid-range graphical processing unit, making it presently impractical for bat PAM (greater than ×10 real time is usually practical for post-processing large datasets). The short duration, high-frequency calls of bats require a small segment size to adequately represent call features on a spectrogram, resulting in hundreds of segments to process per second. Longer vocalizations (e.g. killer whale whistles, Bergler et al., 2019) yield far fewer segments per second resulting in workflow speeds orders of magnitude faster. However, with ongoing optimisation of CNN networks and the rapidly increasing speed of hardware, we predict that the full deep learning workflow on continuous recordings has practical potential for future large-scale bat monitoring.

4.4 | PAMGuard workflow

The PAMGuard workflow used PAMGuard's dynamic minimum threshold detector module with bat call-specific input parameters to detect transients at a high false-positive rate (Supporting Information S1). At the detection stage, PAMGuard recall was almost identical to that of the deep learning workflow (Figures 4 and 6). Overall, the PAMGuard workflow also performed well, but misclassified more calls as noise than the deep learning workflow, and with reduced recall for *N. noctula* (Figure 8). To improve classification accuracy of this workflow further without compromising processing speed, the PAMGuard detector could be run initially to significantly reduce data yet maintain the high false-positive rate to ensure data integrity, followed by an additional, deep learning, detection stage before deep learning classification.

Despite having a higher false-positive rate than the deep learning workflow, the PAMGuard detection stage was ×25 *faster* than real-time analysis and subsequent species classification comparable to real time (a 1-h recording took 1 h to classify) on a standard CPU and ×12 faster than real time with a consumer graphics card. Additionally, the PAMGuard workflow continuously sampled noise, consequently offering a practical approach for large-scale analysis, especially if lossless compression algorithms (Johnson et al., 2013) are used on the recording device. For upscaled acoustic surveys, the PAMGuard (or similar) algorithm running at a high false-positive rate could be integrated on a low power processor on-board the recorder to minimize information loss while maintaining significant data reduction. Combined with computationally efficient methods, many device types could continuously measure ambient noise, for example, octave band levels (ANSI S1.11-2004; American National Standards Institute, 2004), to optimize long-term PAM.

4.5 | Importance of ambient noise logging

While speed and classification accuracy are key in automated analysis, so is ambient noise logging to allow changes in detection probability ($\hat{P}$) to be quantified, as many monitoring metrics extracted from acoustic data, such as abundance estimates, temporal trends, etc., scale directly with $\hat{P}(n)$. If we had assumed a static $\hat{P}$ rather than calculating ambient/system noise for the dataset in Figure 9, genuine changes in detection rates and thereby estimates of bat abundance and/or activity would be confounded with changes in detection probability due to fluctuating noise in the recording, highlighting the importance of retaining this information. In our Monte Carlo simulation of call detection in the extended dataset (Figure 9), the detection probability changed with ambient noise level between ~0.0025 and 0.015 within a 40-m radius of the recorder, meaning that 0.25%–1.5% of all calls emitted within this space were detected, and reflecting a sixfold variation in call detection probability. Our absolute values of $\hat{P}(n)$ were simulated based on a range of assumptions but provide a ballpark estimate of typical changes in call detection probability due to fluctuating noise levels (Darras et al., 2020). Empirical data on actual call rates are now obtainable with miniature acoustic tags that record echolocation activity on-board animals in the wild (Stidsholt et al., 2021).

While rarely reported above 20 kHz (Darras et al., 2020; Merchant et al., 2015), ambient noise is likely low in the ultrasonic range away from turbulent water, rain and biological sound sources such as insects or vocalizing rodents. Importantly, however, we show that noise levels, whether ambient or within the recording system, may vary considerably over time.

4.6 | Importance of software

Most detected calls had low SNR because most animals were detected within the larger area covered at the outer detection range of the recorder; thus, our results reflect *realistic* workflow performance, rather than emphasizing high SNR calls to boost performance. Even the best-performing deep learning workflow showed a precipitous fall in detection performance below 10–15 dB SNR compared to our manual analyst (Figures 4 and 6), but manual analyses by itself work in a timeframe that is increasingly unrealistic for large-scale acoustic monitoring.

Our primary focus was not on optimizing the classification algorithm (Figure 7) and our deep learning models were trained only with SM4BAT data (plus AudioMoth noise clips for the classifier). Although our results indicate (Figures 3–9) that the models are robust when applied to AudioMoth data, we analysed AudioMoth recordings separated in space and time and differing in bat activity from the SM4BAT recordings and the transferability of deep learning between devices, locations and times should be a focus of future research. A general classifier may improve by adding training data from more species, with more intraspecific variation, noise and/or calls of varying quality and data from different recorders.

A fully automated workflow has the advantage of speed and although it requires higher call SNR and thus provides less data for subsequent analysis, this loss in statistical power can be compensated for by increasing the number of recorders, and/or increasing recording time (Buckland et al., 2001). Software like PAMGuard provides visual and navigation interfaces integrated with signal processing and automated detection and classification tools without prerequisite coding. It is an available open source and can be used to exploit rapid automated analysis of large-scale datasets in flexible workflows combined with the accuracy and ability of a manual analyst to recognize patterns and handle inconsistencies. We predict that such a software-assisted human-in-the-loop approach (López-Baucells et al., 2019) will be the most reliable for acoustic detection and classification of large-scale bat PAM data for the foreseeable future, especially for rare species and species difficult to classify correctly.

5 | CONCLUSION

The performance of the four tested workflows varied significantly, stressing the importance of knowing in detail the frequency response, directionality, clip level and self-noise of recording systems and the detection algorithms, that is, trigger settings, used to make informed choices (Perea & Tena, 2020). We advocate that all parts of an analysis workflow should be open source, that onboard trigger algorithms are thoroughly documented, that appropriate system/ambient noise is logged, and that the source code is fully disclosed to promote understanding of the performance and caveats of automated detection and classification algorithms. This, in turn, helps to quantify important metrics such as the probability of detection and allows better biological inferences to be made from acoustic data. Open source further allows fast and qualified feedback to developers on improvements to those algorithms and promotes user interaction and knowledge exchange.

Here, we demonstrate PAMGuard as an effective open-source tool for automated bat call extraction, implementable within a variety of analysis workflows and show why these should carefully consider detection and hardware choices prior to deployment. Helped by sufficiently documented and tested hardware and software, such choices allow PAM data to be used for quantitative measures of animal distribution (Tena & Tellería, 2021) and abundance, rather than occurrence or activity indices. In marine PAM, this approach has recently opened a whole new field within monitoring of wild animals and could likewise provide the bat research community with an unprecedented tool for processing and yielding high-quality data over extended time and space, conferring significant benefits for bat conservation and management.

AUTHOR CONTRIBUTIONS

Jamie Macaulay, Signe M. M. Brinkløv, Jakob Tougaard and Peter Teglberg Madsen conceived the ideas and designed the methodology; Signe M. M. Brinkløv collected the data; Jamie Macaulay and Christian Bergler built the deep learning-based binary detection and multi-class classification model, assisted by Kristian Beedholm. Jamie Macaulay coded the PAMGuard module; Jamie Macauley and Signe M. M. Brinkløv analysed the data; Jamie Macaulay drafted the figures; Jamie Macauley and Signe M. M. Brinkløv led the writing of the manuscript with input from Peter Teglberg Madsen and Jakob Tougaard. All authors contributed critically to the drafts and gave final approval for publication.

ACKNOWLEDGEMENTS

The work was funded by grants to PTM from Carlsberg Semper Ardens Research Projects and the Independent Research Fund Denmark. We also thank Bikubenfonden (Svanninge Bjerge) and Naturstyrelsen (National Park Mols Bjerge) for access to deployment sites.

CONFLICT OF INTEREST STATEMENT

JM develops modules for PAMGuard as part of his research work and interests. The authors declare no conflicts of interest.

PEER REVIEW

The peer review history for this article is available at https://www.webofscience.com/api/gateway/wos/peer-review/10.1111/2041-210X.14131.

DATA AVAILABILITY STATEMENT

Data available via the Dryad Digital Repository https://doi.org/10.5061/dryad.4xgxd25fh (Brinkløv et al., 2023). Relevant code can be found in Github (PAMGuard: https://github.com/PAMGuard, ANIMAL-SPOT: https://github.com/ChristianBergler/ANIMAL-SPOT).

ORCID

Signe M. M. Brinkløv https://orcid.org/0000-0002-7289-3536
Jamie Macaulay https://orcid.org/0000-0003-1309-4889

ENDNOTES

[1] The proportion of calls missed by the detector is sometimes referred to as the false-negative rate, a term which we prefer to avoid as it is often unknown and poorly defined for field recordings, because it is difficult to know how many calls were emitted by the bats. To evaluate automated workflow performance, a false-negative rate can be estimated through comparison with exhaustive manual annotation of recordings, but this approach still misses many calls (Figure 1b: leftmost orange rectangle).

[2] If the distance between bat and recorder can be accurately estimated for each call, $\hat{P}$ can be determined through application of tools from radial distance sampling (Buckland et al., 2001).

REFERENCES

American National Standards Institute. (2004). *ANSI S1.11-2004: American national standard specification for octave-band and fractional-octave-band analog and digital filters*. Acoustical Society of America.

Ancillotto, L., Ariano, A., Nardone, V., Budinski, I., Rydell, J., & Russo, D. (2017). Effects of free-ranging cattle and landscape complexity on bat foraging: Implications for bat conservation and livestock management. *Agriculture, Ecosystems & Environment*, *241*, 54–61. https://doi.org/10.1016/j.agee.2017.03.001

Bas, Y., Bas, D., & Julien, J.-F. (2017). Tadarida: A Toolbox for Animal Detection on Acoustic Recordings. *Journal of Open Research Software*, *5*(1), 6. https://doi.org/10.5334/jors.154

Bergler, C., Schröter, H., Cheng, R. X., Barth, V., Weber, M., Nöth, E., Hofer, H., & Maier, A. (2019). ORCA-SPOT: An automatic killer whale sound detection toolkit using deep learning. *Scientific Reports*, *9*, 10997. https://doi.org/10.1038/s41598-019-47335-w

Bergler, C., Smeele, S. Q., Tyndel, S. A., Barnhill, A., Ortiz, S. T., Kalan, A. K., Cheng, R. X., Brinkløv, S., Osiecka, A. N., Tougaard, J., Jakobsen, F., Wahlberg, M., Nöth, E., Maier, A., & Klump, B. C. (2022). ANIMAL-SPOT enables animal-independent signal detection and classification using deep learning. *Scientific Reports*, *12*, 21966. https://doi.org/10.1038/s41598-022-26429-y

Brinkløv, S. M. M., Macaulay, J., Bergler, C., Tougaard, J., Beedholm, K., Elmeros, M., & Madsen, P. T. M. (2023). Data from: Open-source workflow approaches to passive acoustic monitoring of bats. *Methods in Ecology and Evolution*. https://doi.org/10.5061/dryad.4xgxd25fh

Buckland, S. T., Anderson, D., Burnham, K., Laake, J., Borchers, D. L., & Thomas, L. (2001). *Introduction to distance sampling: Estimating abundance of biological populations*. Oxford University Press.

Burgin, C. J., Colella, J. P., Kahn, P. L., & Upham, N. S. (2018). How many species of mammals are there? *Journal of Mammalogy*, *99*(1), 1–14. https://doi.org/10.1093/jmammal/gyx147

Caillat, M., Thomas, L., & Gillespie, D. (2013). The effects of acoustic misclassification on cetacean species abundance estimation. *The Journal of the Acoustical Society of America*, *134*, 2469–2476. https://doi.org/10.1121/1.4816569

Darras, K. F. A., Deppe, F., Fabian, Y., Kartono, A. P., Angulo, A., Kolbrek, B., Mulyani, Y. A., & Prawiradilaga, D. M. (2020). High microphone signal-to-noise ratio enhances acoustic sampling of wildlife. *PeerJ*, *8*, e9955. https://doi.org/10.7717/peerj.9955

Frasier, K. E., Wiggins, S. M., Harris, D., Marques, T. A., Thomas, L., & Hildebrand, J. A. (2016). Delphinid echolocation click detection probability on near-seafloor sensors. *The Journal of the Acoustical Society of America*, *140*, 1918–1930. https://doi.org/10.1121/1.4962279

Ghanem, S. J., & Voigt, C. C. (2012). Chapter 7—Increasing awareness of ecosystem services provided by bats. In H. J. Brockmann, T. J. Roper, M. Naguib, J. C. Mitani, & L. W. Simmons (Eds.), *Advances in the study of behavior* (pp. 279–302). Academic Press. https://doi.org/10.1016/B978-0-12-394288-3.00007-1

Gibb, R., Browning, E., Glover-Kapfer, P., & Jones, K. E. (2019). Emerging opportunities and challenges for passive acoustics in ecological assessment and monitoring. *Methods in Ecology and Evolution*, *10*, 169–185. https://doi.org/10.1111/2041-210X.13101

Gillespie, D., Mellinger, D. K., Gordon, J., McLaren, D., Redmond, P., McHugh, R., Trinder, P. W., Deng, X. Y., & Thode, A. (2008). PAMGUARD: Semiautomated, open source software for real- time acoustic detection and localisation of Cetaceans. *Proceedings of the Institute of Acoustics*, *30*(5), 54–62.

Goerlitz, H. R. (2018). Weather conditions determine attenuation and speed of sound: Environmental limitations for monitoring and analyzing bat echolocation. *Ecology and Evolution*, *8*, 5090–5100. https://doi.org/10.1002/ece3.4088

Goerlitz, H. R., ter Hofstede, H. M., Zeale, M. R. K., Jones, G., & Holderied, M. W. (2010). An aerial-hawking bat uses stealth echolocation to counter moth hearing. *Current Biology*, *20*(17), 1568–1572. https://doi.org/10.1016/j.cub.2010.07.046

Hill, A. P., Prince, P., Piña Covarrubias, E., Doncaster, C. P., Snaddon, J. L., & Rogers, A. (2018). AudioMoth: Evaluation of a smart open acoustic device for monitoring biodiversity and the environment. *Methods in Ecology and Evolution*, *9*, 1199–1211. https://doi.org/10.1111/2041-210X.12955

Jakobsen, L., Brinkløv, S., & Surlykke, A. (2013). Intensity and directionality of bat echolocation signals. *Frontiers in Physiology*, *4*, 89. https://doi.org/10.3389/fphys.2013.00089

Johnson, M., Partan, J., & Hurst, T. (2013). Low complexity lossless compression of underwater sound recordings. *The Journal of the Acoustical Society of America*, *133*(3), 1387–1398. https://doi.org/10.1121/1.4776206

Jones, G., Jacobs, D., Kunz, T., Willig, M., & Racey, P. (2009). Carpe noctem: The importance of bats as bioindicators. *Endangered Species Research*, *8*, 93–115. https://doi.org/10.3354/esr00182

Knörnschild, M., Blüml, S., Steidl, P., Eckenweber, M., & Nagy, M. (2017). Bat songs as acoustic beacons—Male territorial songs attract dispersing females. *Scientific Reports*, *7*, 13918. https://doi.org/10.1038/s41598-017-14434-5

Küsel, E. T., Mellinger, D. K., Thomas, L., Marques, T. A., Moretti, D., & Ward, J. (2011). Cetacean population density estimation from single fixed sensors using passive acoustics. *The Journal of the Acoustical Society of America*, *129*, 3610–3622. https://doi.org/10.1121/1.3583504

López-Baucells, A., Torrent, L., Rocha, R., Bobrowiec, P. E. D., Palmeirim, M. J., & Meyer, C. F. J. (2019). Stronger together: Combining automated classifiers with manual post-validation optimizes the workload vs reliability trade-off of species identification in bat acoustic surveys. *Ecological Informatics*, *49*, 45–53. https://doi.org/10.1016/j.ecoinf.2018.11.004

Mac Aodha, O., Gibb, R., Barlow, K. E., Browning, E., Firman, M., Freeman, R., Harder, B., Kinsey, L., Mead, G. R., Newson, S. E., Pandourski, I., Parsons, S., Russ, J., Szodoray-Paradi, A., Szodoray-Paradi, F., Tilova, E., Girolami, M., Brostow, G., & Jones, K. E. (2018). Bat detective—Deep learning tools for bat acoustic signal detection. *PLoS Computational Biology*, *14*, e1005995. https://doi.org/10.1371/journal.pcbi.1005995

Madsen, P. T., & Wahlberg, M. (2007). Recording and quantification of ultrasonic echolocation clicks from free-ranging toothed whales. *Deep Sea Research Part I: Oceanographic Research Papers*, *54*, 1421–1444. https://doi.org/10.1016/j.dsr.2007.04.020

Marques, T. A., Thomas, L., Martin, S. W., Mellinger, D. K., Ward, J. A., Moretti, D. J., Harris, D., & Tyack, P. L. (2013). Estimating animal population density using passive acoustics. *Biological Reviews*, *88*, 287–309. https://doi.org/10.1111/brv.12001

McCordic, J. A., DeAngelis, A. I., Kline, L. R., McBride, C., Rodgers, G. G., Rowell, T. J., Smith, J., Stanley, J. A., Stokoe, A., & Van Parijs, S. M. (2021). Biological sound sources drive soundscape characteristics of two Australian marine parks. *Frontiers in Marine Science*, *8*, 1199. https://doi.org/10.3389/fmars.2021.669412

Merchant, N. D., Fristrup, K. M., Johnson, M. P., Tyack, P. L., Witt, M. J., Blondel, P., & Parks, S. E. (2015). Measuring acoustic habitats. *Methods in Ecology and Evolution*, *6*, 257–265. https://doi.org/10.1111/2041-210X.12330

Obrist, M. K., & Boesch, R. (2018). BatScope manages acoustic recordings, analyses calls, and classifies bat species automatically. *Canadian Journal of Zoology*, *96*, 939–954. https://doi.org/10.1139/cjz-2017-0103

Pedersen, M. B., Uebel, A. S., Beedholm, K., Foskolos, I., Stidsholt, L., & Madsen, P. T. (2022). Echolocating Daubenton's bats call louder, but show no spectral jamming avoidance in response to bands of masking noise during a landing task. *Journal of Experimental Biology*, *225*(7), jeb243917. https://doi.org/10.1242/jeb.243917

Perea, S., & Tena, E. (2020). Different bat detectors and processing software… same results? *Journal of Bat Research & Conservation*, *13*(1), 4–8. https://doi.org/10.14709/BarbJ.13.1.2020.01

Roemer, C., Julien, J.-F., & Bas, Y. (2021). An automatic classifier of bat sonotypes around the world. *Methods in Ecology and Evolution*, *12*, 2432–2444. https://doi.org/10.1111/2041-210X.13721

Russo, D., Salinas-Ramos, V. B., Cistrone, L., Smeraldo, S., Bosso, L., & Ancillotto, L. (2021). Do we need to use bats as bioindicators? *Biology*, *10*(8), 693. https://doi.org/10.3390/biology10080693

Russo, D., & Voigt, C. C. (2016). The use of automated identification of bat echolocation calls in acoustic monitoring: A cautionary note for a sound analysis. *Ecological Indicators*, *66*, 598–602. https://doi.org/10.1016/j.ecolind.2016.02.036

Rydell, J., Nyman, S., Eklöf, J., Jones, G., & Russo, D. (2017). Testing the performances of automated identification of bat echolocation calls: A request for prudence. *Ecological Indicators*, *78*, 416–420. https://doi.org/10.1016/j.ecolind.2017.03.023

Sánchez-Bayo, F., & Wyckhuys, K. A. G. (2019). Worldwide decline of the entomofauna: A review of its drivers. *Biological Conservation*, *232*, 8–27. https://doi.org/10.1016/j.biocon.2019.01.020

Stidsholt, L., Greif, S., Goerlitz, H. R., Beedholm, K., Macaulay, J., Johnson, M., & Madsen, P. T. (2021). Hunting bats adjust their echolocation to receive weak prey echoes for clutter reduction. *Science Advances*, *7*. https://doi.org/10.1126/sciadv.abf1367

Sugai, L. S. M., Desjonquères, C., Silva, T. S. F., & Llusia, D. (2020). A roadmap for survey designs in terrestrial acoustic monitoring. *Remote Sensing in Ecology and Conservation*, *6*, 220–235. https://doi.org/10.1002/rse2.131

Tena, E., & Tellería, J. L. (2021). Modelling the distribution of bat activity areas for conservation in a Mediterranean mountain range. *Animal Conservation*, *25*, 65–76. https://doi.org/10.1111/acv.12719

Voigt, C., & Kingston, T. (2016). *Bats in the anthropocene: Conservation of bats in a changing world*. Springer.

Walters, C. L., Freeman, R., Collen, A., Dietz, C., Brock Fenton, M., Jones, G., Obrist, M. K., Puechmaille, S. J., Sattler, T., Siemers, B. M., Parsons, S., & Jones, K. E. (2012). A continental-scale tool for acoustic identification of European bats. *Journal of Applied Ecology*, *49*, 1064–1074. https://doi.org/10.1111/j.1365-2664.2012.02182.x

K. Lisa Yang Center for Conservation Bioacoustics. (2016). *Raven lite: Interactive sound analysis software (V2.0.1)*. The Cornell Lab of Ornithology.

SUPPORTING INFORMATION

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

Appendix S1. Supporting information.

How to cite this article: Brinkløv, S. M. M., Macaulay, J., Bergler, C., Tougaard, J., Beedholm, K., Elmeros, M., & Madsen, P. T. (2023). Open-source workflow approaches to passive acoustic monitoring of bats. *Methods in Ecology and Evolution*, *00*, 1–17. https://doi.org/10.1111/2041-210X.14131