Assessing deep learning protocols for optimizing time series-based species distribution models

***Austin M. Smith[1]** , Andrew M. Kramer[1]*

[1]Department of Integrative Biology, University of South Florida, 4202 E. Fowler Avenue, Tampa, FL 33620, USA

UNIVERSITY of SOUTH FLORIDA

INTRODUCTION:

Species distribution models (SDMs) are widely used to gain ecological understanding and guide conservation decisions. SDMs have been developed with a wide variety of machine learning algorithms, which have improved considerably with the standardization of modelling workflows and thorough inter-model comparisons of predictive accuracy being performed (e.g., Valavi et al., 2022). However, one property current models have in common is the use of predictors that strongly simplify the temporal variability of driving factors. This is despite most factors driving species distributions (e.g., climate, land-use) being temporally dynamic. Thus, recent studies (Capinha et al, 2021; Smith et al. 2022) have turned to deep learning to process raw, ordered temporal data to assess environmental patterns and to effectively predict suitable habitat conditions. Unlike for conventional methods, no such analysis has been done to determine the best working protocols for time series-based deep learning SDMs. This study aims to determine a consistent workflow for such cases and evaluated models on the following parameters: average epoch computation time; number of training iterations (i.e., epochs); architecture-type; and performance statistics.

METHODS:

Species records included 14 bird species from Ontario, Canada, compiled by the the National Center for Ecological Analysis and Synthesis (NCEAS; Elith et al., 2020). Climate and elevation data were extracted from Worldcim (Fick & Hijmans, 2017). Models were generated using the automated machine learning (AutoML; He et al., 2021) assemblage Python package McFly (Van Kuppevelt et al., 2020). Ten candidates for each available architecture (Table 1) were trained for five epochs. Four performance metrics (Table 1) were used to evaluate model performance. A series of tests based on architecture-performance metric combinations were performed to select best candidate models (Figure 1). The final model for each species was chosen based on the cumulative performance metric (ACC + ROC + PR) on an independent presence-absence data set

Table 1: List of model components evaluated in this study

Architectures	Performance metrics	Systematic metrics
• Convolutional Neural Network (CNN) • Inception Time • Long Short-Term Memory (LSTM) • Residual Network (ResNet)	• Minimized cost of loss function (Loss) • Classification accuracy (ACC) • Area under the receiver operating curve (ROC) • Area under the precision-recall curve (PR)	• Computation time • Number of training/tuning iterations (Epochs)

RESULTS:

Table 2: Summary of best models for each species

Species	No. Records	Architecture	No. epochs	Epoch duration (sec)	Measure	Loss	ACC	ROC	PR	Total score
can02	740	CNN	6	16.766	ACC	0.911	0.816	0.851	0.841	2.508
can03	165	CNN	4	26.378	ACC	0.411	0.952	0.957	0.946	2.855
can05	138	LSTM	11	49.696	Loss	0.24	0.942	0.956	0.952	2.85
can07	221	InceptionTime	5	27.562	ROC	0.533	0.977	0.985	0.984	2.946
can08	322	ResNet	5	30.104	ACC	0.65	0.822	0.83	0.813	2.465
can09	119	LSTM	6	55.27	ROC	0.384	0.989	0.993	0.993	2.975
can10	234	CNN	4	9.433	PR	0.392	0.914	0.933	0.931	2.778
can11	478	CNN	10	20.268	Loss	1.661	0.69	0.69	0.654	2.034
can12	312	LSTM	21	85.588	Loss	0.026	1	1	1	3
can15	721	CNN	11	10.963	PR	0.798	0.792	0.847	0.843	2.482
can17	313	LSTM	34	47.102	Loss	0.496	0.811	0.883	0.873	2.567
can18	612	InceptionTime	5	37.622	Loss	0.419	0.919	0.958	0.957	2.834
can19	109	LSTM	30	30.226	Loss	0.034	0.998	0.999	0.999	2.996
can20	380	LSTM	4	53.288	PR	0.354	0.939	0.944	0.925	2.808

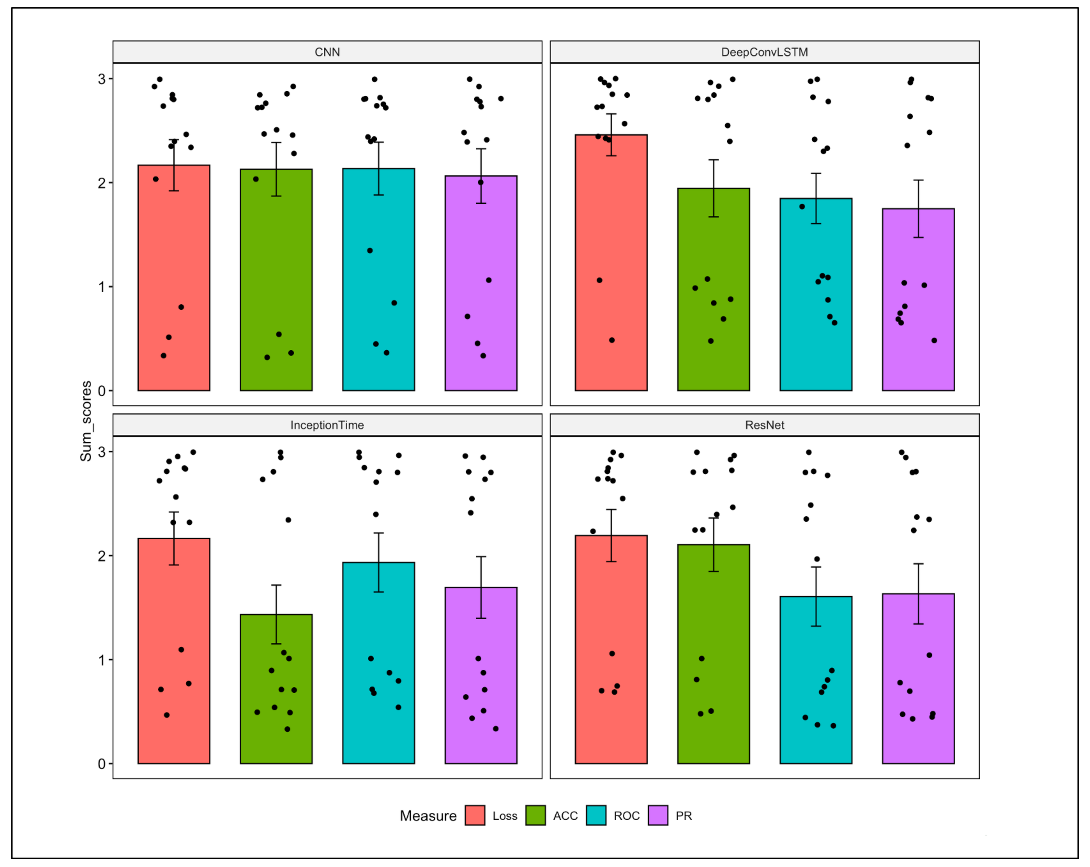

Figure 2: Summary of cumulative prediction metrics (ACC + ROC + PR) for all best models, organized by architecture-metric combinations. Each bar represents the average score for models optimized by the given metric. Note, a score of 3 indicates perfect prediction across each prediction metrics. CNN models more frequently produce higher scores across all metrics. All architectures performed best when focusing on Loss values, followed by ACC. All models had polarizing results when optimized on ROC or PR.

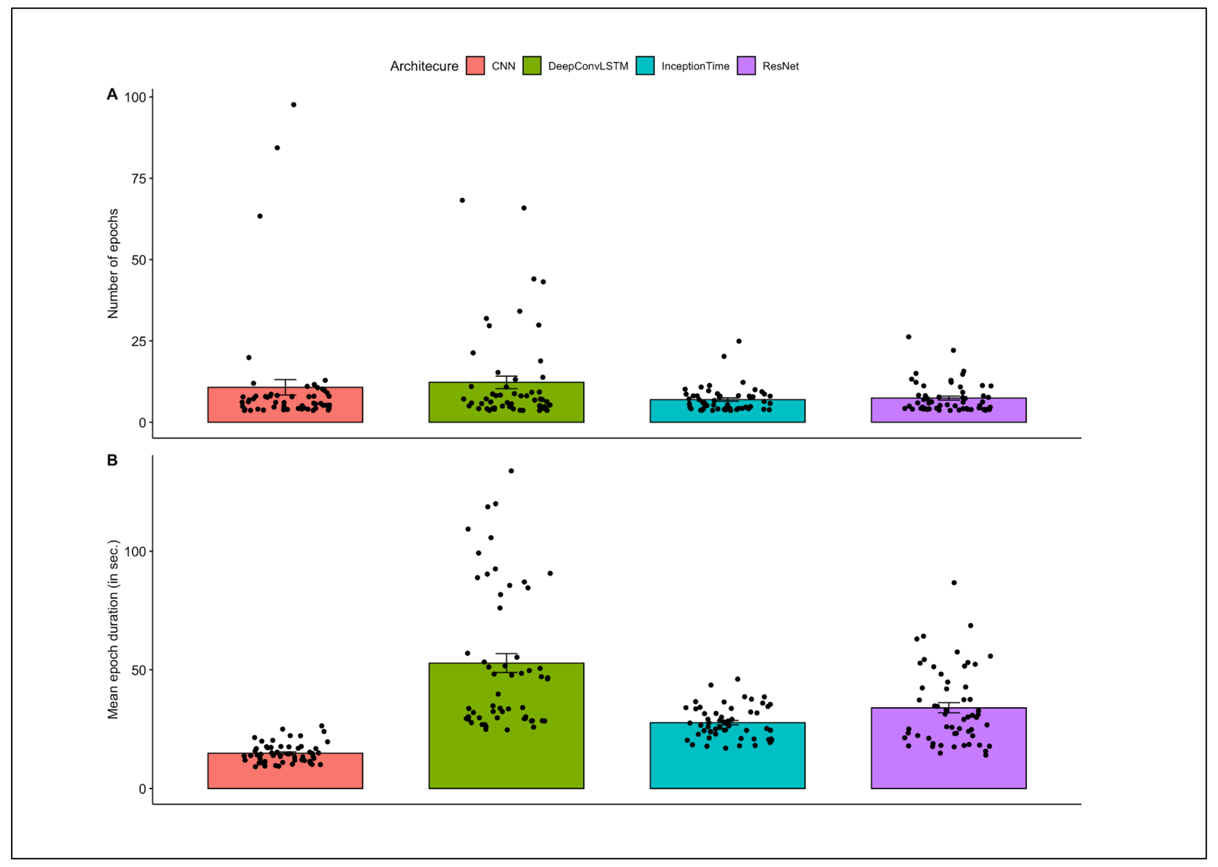

Figure 3: Summary of modeling computation time for each type of architecture. A) The average number of epochs performed before early stopping was activated. Mean value across all models was ~ 9 epochs. Inception time and Resnet models were quicker to converge while both CNN and LSTM needed extensive number of additional tuning iterations. B) Average length of time for each epoch to complete. CNN models were the quickest followed by Inception time. LSTM models were least efficient, followed by ResNet models.

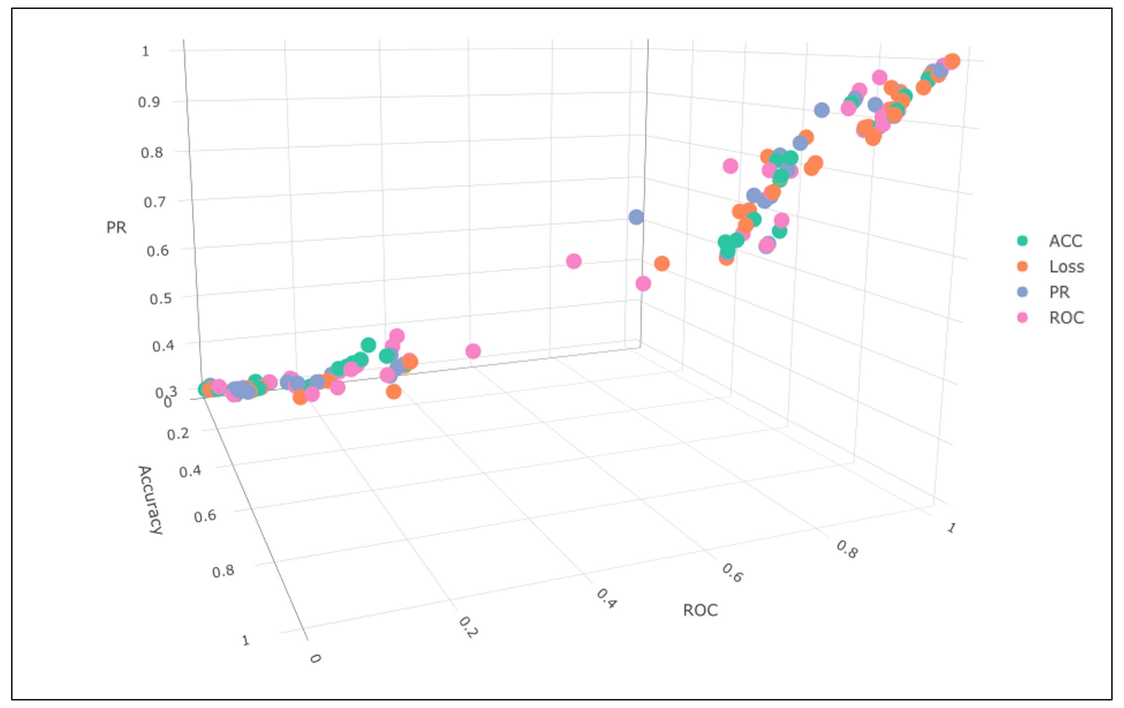

Figure 4: Results from final models showing the relationship between three prediction statistics. Each point is is defined by the modeling metric that was used to tune the model. Results indicate a positive correlation between all three metrics . Models focusing on the Loss value more frequently scored best across all three metrics.

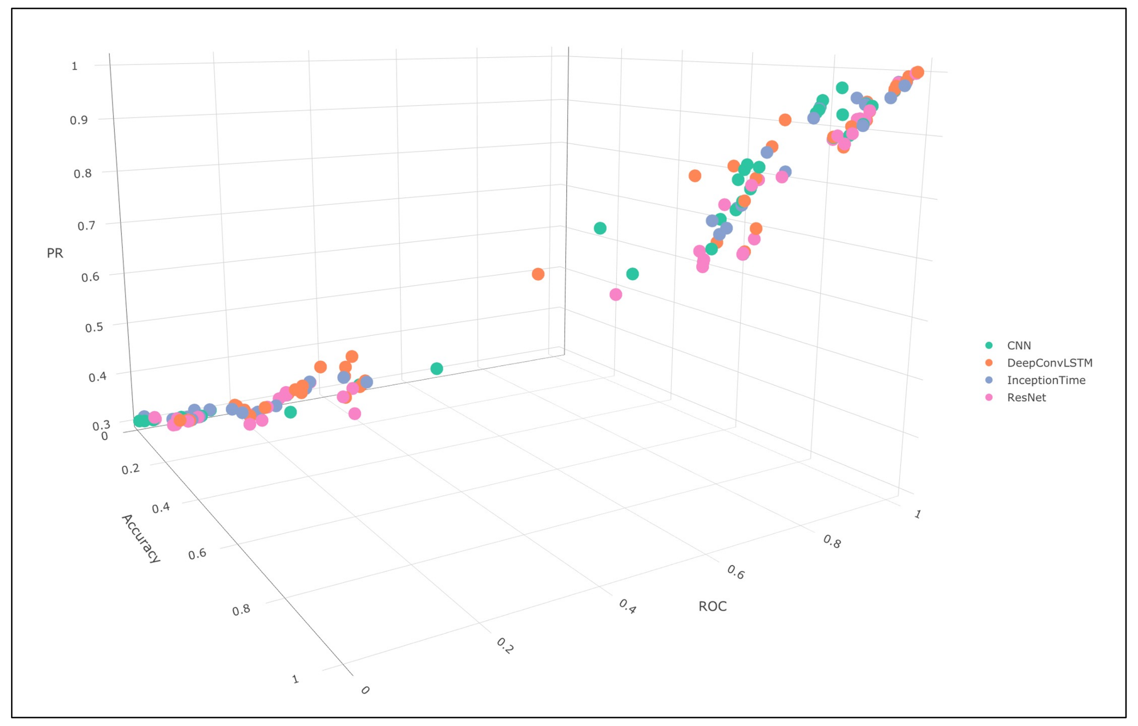

Figure 3: Results from final models showing the relationship between three prediction statistics. Each point is is defined by the model architecture. Results indicate a positive correlation between all three metrics . A best architecture could not be defined as varying results .

Data collection and modeling workflow

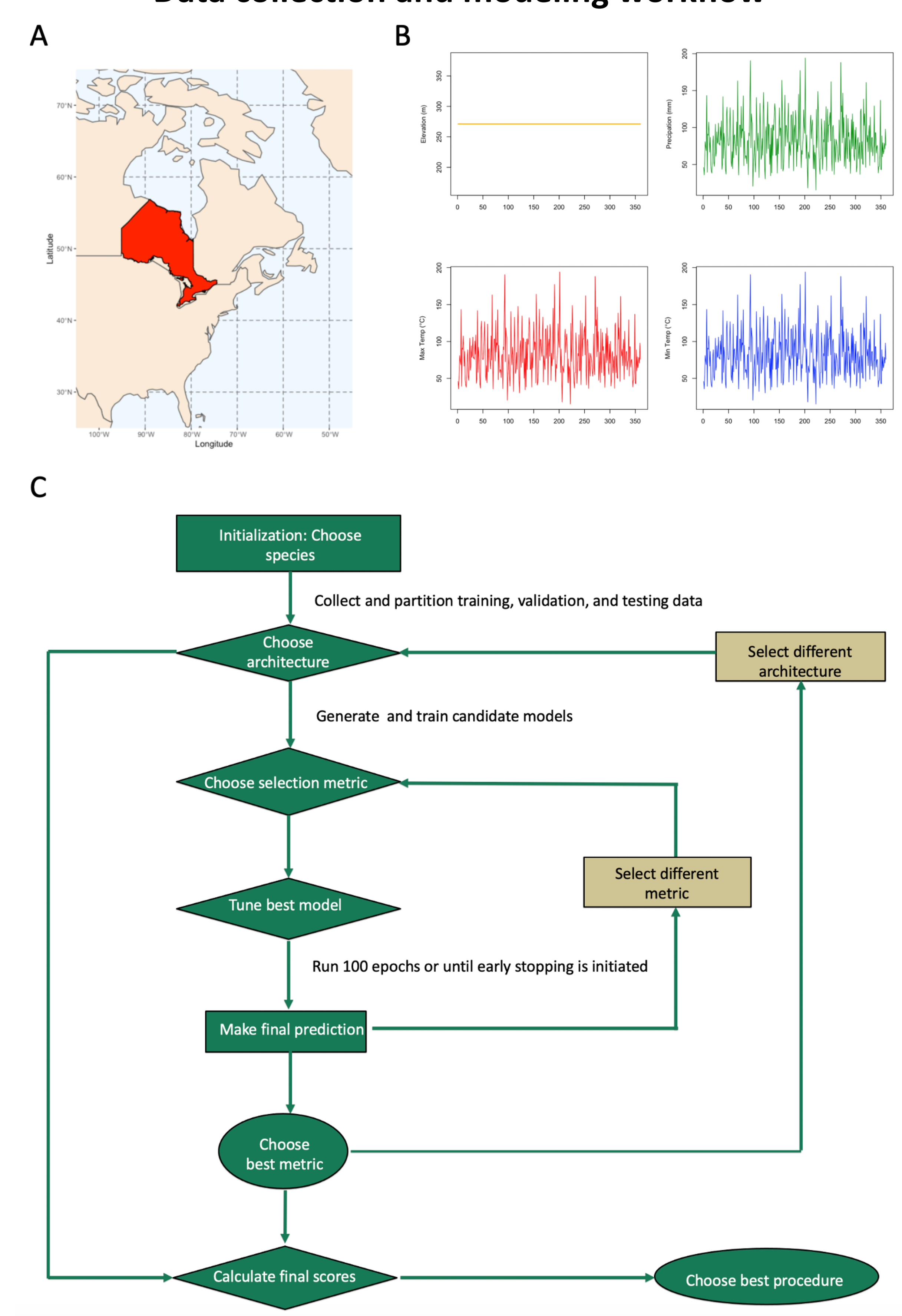

Figure 1: Workflow of modeling process. A) Region where species records occur. B) Example of environmental data collected for a species record. C) Conceptual framework for model construction and analyzing various protocols.

WORK CITED:

- Capinha, C., Ceia-Hasse, A., Kramer, A. M., & Meijer, C. (2021). Deep learning for supervised classification of temporal data in ecology. *Ecological Informatics*, *61*, 101252.
- Elith, J., Graham, C., Valavi, R., Abegg, M., Bruce, C., Ferrier, S., ... & Zimmermann, N. E. (2020). Presence-only and presence-absence data for comparing species distribution modeling methods. *Biodiversity informatics*, *15*(2), 69-80.
- Fick, S. E., & Hijmans, R. J. (2017). WorldClim 2: new 1-km spatial resolution climate surfaces for global land areas. *International journal of climatology*, *37*(12), 4302-4315.
- He, X., Zhao, K., & Chu, X. (2021). AutoML: A survey of the state-of-the-art. *Knowledge-Based Systems*, *212*, 106622.
- Smith, A. M., Capinha, C., & Kramer, A. M. (2022). Predicting species distributions with environmental time series data and deep learning. *bioRxiv*, 2022-10.
- Valavi, R., Guillera-Arroita, G., Lahoz-Monfort, J. J., & Elith, J. (2022). Predictive performance of presence-only species distribution models: a benchmark study with reproducible code. *Ecological Monographs*, *92*(1), e01486.
- Van Kuppevelt, D., Meijer, C., Huber, F., van der Ploeg, A., Georgievska, S., & van Hees, V. T. (2020). Mcfly: Automated deep learning on time series. *SoftwareX*, *12*, 100548.

Main finding:

This study found no clear relationship between performance metrics, model architecture, or overall performance, suggesting that the best protocol may be species or study specific. However, AutoML provides an accessible and efficient way to compare a variety of models.