

UC Berkeley

UC Berkeley Electronic Theses and Dissertations

Title

Applications of Neural Network Models for Conservation Decision-Making and Ecological Forecasting

Permalink

<https://escholarship.org/uc/item/96x9s82d>

Author

Lapeyrolerie, Marcus

Publication Date

2024

Peer reviewed|Thesis/dissertation

Applications of Neural Network Models for Conservation Decision-Making and
Ecological Forecasting

By

Marcus Francois Lapeyrolerie

A dissertation submitted in the partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Environmental Science, Policy, and Management

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Carl Boettiger, Chair
Professor Benjamin Wong Blonder
Professor Fernando Perez

Summer 2024

Abstract

Applications of Neural Network Models for Conservation Decision-Making and Ecological Forecasting

by

Marcus Lapeyrolerie

Doctor of Philosophy in Environmental Science, Policy, and Management

University of California, Berkeley

Professor Carl Boettiger, Chair

Improving the ability to manage and forecast ecological processes would enhance conservation practices and further our understanding of the natural world. Ecology is transitioning from a discipline that was once data poor to a discipline that is increasingly data rich. With the aggregation of data into large repositories, significant investments in long-term ecological monitoring networks, and the development of richly detailed process-based simulators, ecologists need new tools to support the analysis of extensive data sets. Recently, scientists outside of ecology have used neural network models to solve formerly intractable problems characterized by large data sets. Ecologists have started to use neural networks to make progress on challenging questions, but the majority of this work has been limited to automated monitoring. In this dissertation, I explore applications of neural network models for conservation decision-making and ecological forecasting. Chapter 2 presents how concepts and methods taken from the field of reinforcement learning can be used to solve decision-making problems in conservation. Chapter 3 investigates the ability of neural network models to forecast critical transitions observable in ecological systems. And, lastly, in Chapter 4, I compare the forecasting performance of neural network models on water quality data taken from the National Ecological Observatory Network. Together, these chapters demonstrate that neural networks have the capacity to provide novel insights on ecological processes.

This dissertation is dedicated to my late grandmother, Mary Charles Evans.

Contents

1	Preface	1
2	Deep Reinforcement Learning for Conservation Decisions	5
2.1	Introduction	5
2.2	Materials and Methods	7
2.3	Results	11
2.4	Discussion	16
3	Limits to Ecological Forecasting: Estimating Uncertainty for Critical Transitions with Deep Learning	21
3.1	Introduction	21
3.2	Materials and Methods	22
3.3	Results	29
3.4	Discussion	35
4	A Comparison of Neural Network Models for Water Quality Forecasting	39
4.1	Introduction	39
4.2	Materials and Methods	40
4.3	Results	44
4.4	Discussion	45
5	Conclusion	53
	References	55

Chapter 1

Preface

In the current era of global change, we are faced with a multitude of environmental crises. Biological diversity is declining at an alarming rate in what has been called the sixth mass extinction event (Ceballos, Ehrlich, and Dirzo 2017). Various ecological communities and components of the Earth system have critical thresholds that could lead to significant societal impacts if crossed (Dietz et al. 2021). And freshwater ecosystems, which provide critical services to humanity, have been acutely affected by anthropogenic causes (Tickner et al. 2020). To better understand and manage the environmental problems that we face, there is an urgency to develop tools that can handle the obfuscating complexity characteristic of ecological systems. This dissertation focuses on the exploration of computational methods for conservation decision-making and ecological forecasting, disciplines where classical methods have generally lacked scalability and often become ineffective without unrealistic approximations.

Neural networks present a way forward as they can model complex patterns directly from the data without suffering from an inability to scale. A neural network is a computational model that is inspired by how the human brain functions. Neural networks consist of interconnected layers of nodes (neurons) wherein each node applies a non-linear function to the inputs from the preceding layer. There are numerous types of neural networks that have been designed for a variety of specific applications – e.g., some neural networks have been designed exclusively for time series, while others have been designed to work across data types like textual, audio and visual data. For this dissertation, it is not critical to know all the fine-grained differences between the neural networks that are presented. Instead, it is important to understand that neural networks are used to approximate mathematical functions. For decision-making problems, neural networks are often used to approximate the policy function which maps the observed state of the system to an action to be performed by the manager/agent; and, for time-series forecasting, neural networks are often used to map a sequence of historical values to a sequence of future values. Through the universal approximator theorem, it has been established that a neural network with a single layer of arbitrary size can approximate any continuous non-linear function (Hornik, Stinchcombe, and White 1989). While it is not feasible to use an arbitrarily wide network in practice as this network could have an exorbitant number of parameters, it has been well established that neural networks with multiple layers (which are referred to as deep neural networks) are able to model a wide range of functions with a tractable amount of parameters (Liang and Srikant 2016). There are other methods that one can use for function approximation in place of neural networks, such as Gaussian Processes, but the performance of these methods struggle to match the

accuracy and scalability achieved with neural networks (Grande, Walsh, and How 2014).

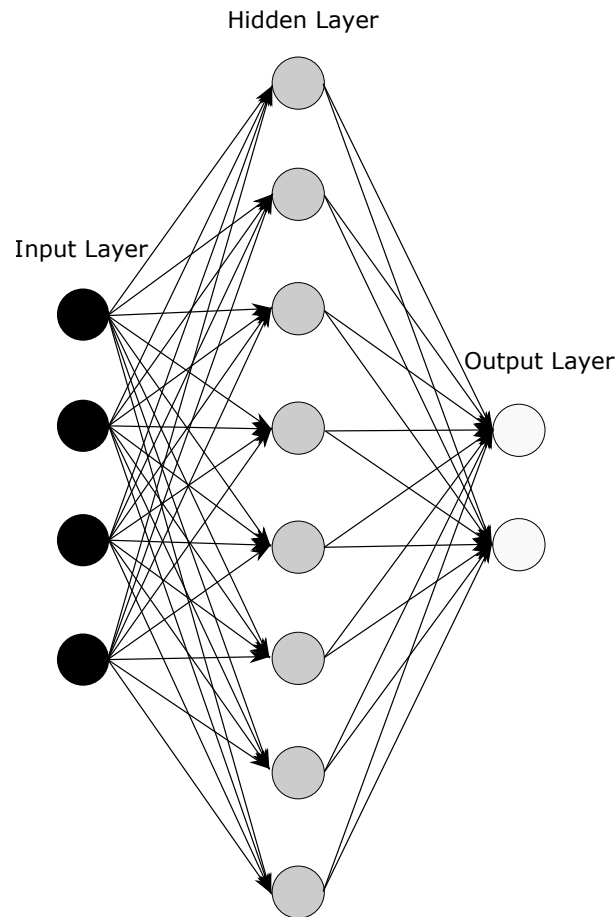


Figure 1.1: Feedforward neural network. The feedforward neural network is a commonly used architecture that illustrates how neural networks work in general. Feedforward networks operate in a unidirectional manner wherein the input passes to the hidden layer before reaching the output layer. At each node in the hidden layer, a non-linear function is applied to the sum of inputs from the preceding layer. During training, the parameters of the neural network are adjusted so that the output of the model approximates the expected output provided by the data. The network shown in this figure is a shallow neural network as it has 1 hidden layer. A neural network that has more than 1 hidden layer is described as being "deep".

A connecting theme of this dissertation is the focus on model-free, also referred to as non-mechanistic, methods. All the algorithms used in this dissertation are model-free: they do not employ a model that describes the data generating process. These algorithms, instead, make predictions solely by drawing inferences from the data. This model-free approach is a departure from the historically common inclination in ecological forecasting towards methods that rely on understanding the mechanism of the target system. For instance, in popular textbooks on ecological forecasting, the general forecasting paradigm that is presented is to write out a set of equations that describes the data generating

process, estimate this model’s parameters from the data, and then use the estimated parameters to generate a forecast (M. Dietze 2017). The orientation towards mechanistic models has partially been a result of there not being enough large data sets in ecology in the 20th century to accommodate model-free methods; yet, as the rates of data generation and aggregation have increased rapidly in ecology in the 21st century, model-free methods offer some significant advantages (Farley et al. 2018). With so much ecological data to be analyzed now, constructing a mechanistic model that can accommodate all the patterns that are realizable in the data is an extremely challenging task. A model that can learn directly from the data without being misled by assumptions offers an appealing alternative¹.

Along with the advantages that neural networks present, there are significant drawbacks. Neural networks suffer from a variety of practical issues like the tendency to overfit, instability with respect to hyperparameters, high computational costs, requirements of specialized hardware, and more. Beyond these technical concerns, neural networks also present social issues with their development: a significant amount of the foundational research in machine learning is carried out by private companies, which raises ethical concerns over whether this work will serve the public’s interest. Yet, whether the model is mechanistic or non-mechanistic, a neural network or otherwise, the model will always be wrong. This dissertation is an exploration of where non-mechanistic neural network models can both succeed and fail to lend insight to ecologists – in the often quoted words of George Box, “All models are wrong, but some are useful”.

Another commonly perceived disadvantage of neural networks is their focus on prediction at the exclusion of interpretability. The neural networks used in this dissertation are “black boxes”: it is not possible to explain what the neural networks identify in the input data that results in the neural networks’ outputs. While ecologists have historically favored methods with strong mechanistic bases, neural networks present a contrastive paradigm where there is little emphasis placed on understanding the dynamics of the underlying process. In this comparison, it is important to state that the abilities of a model to predict and be explainable are independent of each other. For certain problems, neural networks can provide helpful insights that would be practically unattainable otherwise. Yet, in other contexts, such as safety critical problems, making decisions solely on the basis of uninterpretable models could lead to catastrophic outcomes (Rudin 2019). This dissertation does not support the perspective that neural network models should replace all other methods used in conservation decision-making and ecological forecasting. Instead, this dissertation advocates for neural networks being used in a diverse suite of methodologies that ecologists and conservation managers can rely on to inform their decisions.

This dissertation has the primary aim of investigating how neural networks can be employed for conservation decision-making and ecological forecasting. Accordingly, throughout Chapters 2, 3 and 4, there will be commonalities regarding the presented methods and the focus towards prediction, but the systems of interest will be disparate across the chapters.

Chapter 2 focuses on how neural networks can be used for conservation decision-making. In this chapter, I present the mathematical formalism of Markov Decision Pro-

¹There are nuances in the comparison model-free and model-based methods that I omit for clarity. One distinction that is worth mentioning is the concept of hybrid models. For example, in the water quality forecasting literature, there is a body of recent work on process-guided neural networks, which use a mechanistic model to direct what would otherwise be a model-free neural network (Read et al. 2019). This dissertation focuses on purely model-free methods, but hybrid models can exhibit performance advantages over more purely mechanistic and model-free methods (Read et al. 2019).

cesses, and show how decision-making problems in conservation can be posed as Markov Decision Processes. Throughout this chapter, I provide an introduction to reinforcement learning, which is the sub-field of machine learning that has the goal of solving Markov Decision Processes. I present two decision-making problems taken from the conservation literature, and evaluate how neural network models compare against optimal and approximate solutions.

Chapter 3 explores the forecasting performance of neural networks on critical transitions. Critical transitions pose extremely challenging forecasting problems, which necessitate informative uncertainty estimation rather than point forecasts. In this chapter, I use neural networks to forecast time series that were generated from models that describe critical transitions. I compare the neural network-based methods against other forecasting methods that ecologists commonly use.

In Chapter 4, I compare the forecasting performance of neural network models on water quality data taken from the National Ecological Observatory Network (NEON). In this chapter, I use neural networks to forecast time series data that track dissolved oxygen concentration, water temperature, and chlorophyll-a concentration at 34 sites across North America. The neural network models are compared to a selection of alternative methods including a historical daily mean model and a naive persistence model.

In conclusion, I offer a short reflection and some thoughts for how this work could be extended.

Chapter 2

Deep Reinforcement Learning for Conservation Decisions

This chapter was previously published, see Lapeyrolierie et al., 2022. It is included here with permission from the co-authors.

2.1 Introduction

Advances in both available data and computing power are opening the door for machine learning (ML) to play a greater role in addressing some of our planet’s most pressing environmental problems. But will ML approaches really help us tackle our most pressing environmental problems? From the growing frequency and intensity of wildfire (Moritz et al. 2014), to over-exploited fisheries (Worm et al. 2006) and declining biodiversity (Dirzo et al. 2014), to emergent zoonotic pandemics (Dobson et al. 2020), the diversity and scope of environmental problems are unprecedented. Applications of ML in ecology have to-date illustrated the promise of two methods: *supervised learning* (M. B. Joseph 2020) and *unsupervised learning* (Valletta et al. 2017). However, the fields of ecology and conservation have largely overlooked the third and possibly most promising approach in the ML triad: *reinforcement learning* (RL). Three features distinguish RL from other ML methods in ways that are particularly well suited to addressing issues of global ecological change:

- 1) RL is explicitly focused on the task of selecting actions in an uncertain and changing environment to maximize some objective,
- 2) RL does not require massive amounts of representative sampled historical data,
- 3) RL approaches easily integrate with existing ecological models and simulations, which may be our best guide to understanding and predicting future possibilities.

Despite relevance to decision making under uncertainty that could make RL uniquely well suited for ecological control, RL has only been applied to this field in a few cases (Xu et al. 2021; Silvestro et al. 2022; Treloar et al. 2020). To date, the problems considered by RL research have largely been drawn from examples in robotic movement and games like Go and Starcraft (OpenAI et al. 2019; Silver et al. 2018; Vinyals et al. 2019). Complex environmental problems share many similarities to these tasks and games: the need to plan many moves ahead given a large number of possible outcomes, to account for uncertainty and to respond with contingency to the unexpected. RL agents typically develop strategies by interacting with simulators, a practice that should not be

unsettling to ecologists since learning from simulators is common across ecology. Rich, processes-based simulations such as the SORTIE model in forest management (Pacala et al. 1996), Ecopath with Ecosim in fisheries management (Steenbeek et al. 2016), or climate change policy models (Nordhaus 1992) are already used to explore scenarios and inform ecosystem management. Decision-theoretic approaches based on optimal control techniques can only find the best strategy in the simplest of ecological models; the so called “curse of dimensionality” makes problems with a large number of states or actions intractable by conventional methods (Wilson et al. 2006; Marescot et al. 2013; Ferrer-Mestres et al. 2021; Chades et al. 2021). Neural-network-based RL techniques, referred to as *deep RL*, have proven particularly effective in problems involving complex, high-dimensional spaces that have previously proven intractable to classical methods.

While deep RL may have the potential to open up such intractable problems, it also risks making those problems tractable only for stakeholders with access to extensive computational resources and expertise. It is notable that the landmark advances cited above have been solved not by academic teams but by specialized research teams of international technology firms such as Alphabet. Precise estimates of computational resources used in that research are difficult to establish, but previous estimates benchmarked against commercially available cloud computing platforms place the training of a single model at over \$35 million (Huang 2018; Hernandez and Brown 2020; Silver et al. 2017), and many realistic ecological problems will involve even greater complexity than these landmark examples (Silver et al. 2017, 2018; OpenAI et al. 2019). While the history of improved efficiency in computing technology has shown a remarkable ability to reduce such barriers, it has simultaneously moved the leading edge of those capabilities farther beyond reach of traditional ecological research. We believe that ecologists must seek to better understand the design, capabilities and limitations of these algorithms while keeping in mind that the application of RL to conservation will surely require the ambitious collaboration, resources and expertise on par with the scale of the immense environmental and ecological problems we face.

In this chapter, we draw on examples from fisheries management and ecological tipping points to illustrate how deep RL techniques can successfully discover optimal solutions to previously solved management scenarios and discover highly effective solutions to unsolved problems. We focus on examining the potential and limitations of deep RL through the lens of simple, classical models. Over a century of theory and practice in ecology has demonstrated that simple models can provide meaningful insights which improve management outcomes (Getz et al. 2018). As Richard Levins successfully established in his classic paper on the principles of model building (Levins 1966), model complexity must not be mistaken for model realism. Levins espoused simple mechanistic models which satisfy the goals of being both *realistic* and *general*. More complex models such as those used in fisheries to guide the management of specific stocks typically sacrifice *generality* for *precision*. Such simple, realistic and general models are still the bedrock of most theory and practice today (for instance, the notion of maximum sustainable yield, MSY, in fisheries, or R_0 in epidemiology, remain important concepts in management). These models provide an ideal first benchmark for evaluating the performance of emerging methods of deep RL for several reasons: Firstly, for some cases the optimal solution is already known, providing a clear standard-of-comparison to evaluate RL performance. Prior work sometimes overlooks this essential step, assuming that whatever behavior an RL agent produces is sufficiently optimal (Mnih et al. 2015). As our evaluations will illustrate, such an assumption can be quickly misleading. Second, these models are already widely studied and will be familiar to many readers: Schaefer (1954) is a staple of

fisheries management textbooks and practice, with over 2800 citations, while Robert M. May (1977) has become a canonical model of thresholds and tipping points which still continues to dominate how many ecologists think about these phenomena (Scheffer et al. 2015). Many readers can thus benefit from existing knowledge and intuition about the behavior and implications of these models in interpreting the performance of deep RL, something that would not be possible with a more complex model. Third, these models include or can easily be extended to contexts for which the optimal management policy is unknown or inaccessible to classical methods. Our implementations of these models have been published to the python-based PyPi code archive and include many such variations which represent open problems for RL. We include carefully annotated code which should allow readers to both reproduce and extend this analysis.

This chapter does not intend to validate deep RL as a method that should be used to directly inform decision-making on current conservation problems. Rather, we seek to provide ecologists with a greater understanding of both potentials and pitfalls of this emerging approach. We have selected familiar example problems to provide ecologists with a greater background and intuition to understand these techniques and engage in the collaborative development of deep RL-based methods, while also highlighting challenges which ecological problems pose to existing techniques. Validating deep RL for current conservation problems is beyond the scope of any one paper: this will necessitate examining a range of more “precise” models which will require more computational resources than is available to most researchers and extensive collaboration between large teams of ecologists and computer scientists.

2.2 Materials and Methods

All applications of RL can be divided into two components: an *environment* and an *agent*. The *environment* is typically a computer simulation, though it is possible to use the real world as the RL environment (Ha et al. 2020). The *agent*, which is often a computer program, continuously interacts with the environment. At each time step, the agent *observes* the current *state* of the environment then performs an *action*¹. As a result of this action, the environment transitions to a new state and transmits a numerical *reward* signal to the agent. The goal of the agent is to learn how to maximize its expected cumulative reward. The agent learns how to achieve this objective during a period called *training*. In training, the agent *explores* the available actions. Once the agent comes across a highly rewarding sequence of observations and actions, the agent will reinforce this behavior so that it is more likely for the agent to *exploit* the same high reward trajectory in the future. Throughout this process, the agent’s behavior is codified into what is called a *policy*, which describes what action an agent should take for a given observation.

2.2.1 RL Environments

An environment is a mathematical function, computer program, or real world experience that takes an agent’s proposed *action* as input and returns an *observation* of the environment’s current *state* and an associated *reward* as output. In contrast to classical

¹The terms observation and state are used nearly interchangeably in describing RL, so it is worth clarifying the distinction. An observation is the depiction of the environment that is given to the agent at each time step, but the state is the true underlying description of the environment. When the term observation is used, this usually means that the observation does not provide an accurate portrayal of the environment’s state. Yet, in cases when the observation and state are in agreement, the term observation is typically not used at all.

approaches (Marescot et al. 2013; Chades et al. 2021), there are few restrictions on what comprises a state or action. States and actions may be continuous or discrete, completely or partially observed, single or multidimensional. The main focus of building an RL environment, however, is on the environment’s transition dynamics and reward function. The designer of the environment can make the environment follow any transition and reward function provided that both are functions of the current state and action. The ability to tailor the actions, states, transition dynamics and reward function allows RL environments to model a broad range of decision making problems. For example, we can set the transitions to be deterministic or stochastic. We could map any countable set of actions to a discrete action space. We can also specify the reward function to be *sparse*, whereby a positive reward can only be received after a long sequence of actions, e.g. the end point in a maze. In other environments, an agent may have to learn to forgo immediate rewards (or even accept an initial negative reward) in order to maximize the net discounted reward as we illustrate in examples here.

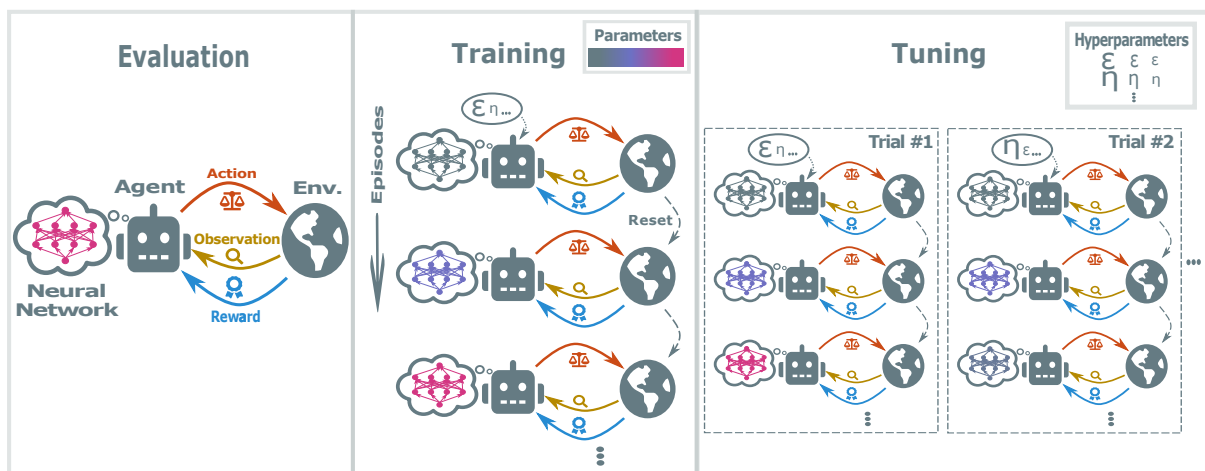


Figure 2.1: Deep Reinforcement Learning: A deep RL *agent* uses a *neural network* to select an *action* in response to an *observation* of the *environment*, and receives a *reward* from the environment as a result. During *training*, the agent tries to maximize its cumulative reward by interacting with the environment and learning from experience. In the RL loop, the agent performs an action, then the environment returns a reward and an observation of the environment’s state. The agent-environment loop continues until the environment reaches a terminal state, after which the environment will reset, causing a new *episode* to begin. Across training episodes, the agent will continually update the *parameters* in its neural network, so that the agent will select better actions. Before training starts, the researcher must input a set of *hyperparameters* to the agent; hyperparameters direct the learning process and thus affect the outcome of training. A researcher finds the best set of hyperparameters during *tuning*. Hyperparameter tuning consists of iterative *trials*, in which the agent is trained with different sets of hyperparameters. At the end of a trial, the agent is evaluated to see which set of hyperparameters results in the highest cumulative reward. An agent is *evaluated* by recording the cumulative reward over one episode, or the mean reward over multiple episodes. Within evaluation, the agent does not update its neural network; instead, the agent uses a trained neural network to select actions.

The OpenAI **gym** software framework was created to address the lack of standardization of RL environments and the need for better benchmark environments to advance RL research (Brockman et al. 2016). The **gym** framework defines a standard interface and methods by which a developer can describe an arbitrary environment in a computer program. This interface allows for the application of software agents that can interact and learn in that environment without knowing anything about the environment’s internal details. Using the **gym** framework, we turn existing ecological models into valid environmental simulators that can be used with any RL agent.

Abbreviation	Algorithm Name	Model
PlaNet	Deep Planning Network (Hafner et al. 2019)	Model-based
I2A	Imagination-Augmented Agents (Weber et al. 2018)	Model-based
MBPO	Model-based Policy Optimization (Janner et al. 2019)	Model-based
DQN	Deep Q Networks (Mnih et al. 2015)	Model-free
A2C	Advantage Actor Critic (Mnih et al. 2016)	Model-free
A3C	Asynchronous A2C (Babaeizadeh et al. 2017)	Model-free
TRPO	Trust Region Policy Optimization (Schulman, Levine, et al. 2017)	Model-free
PPO	Proximal Policy Optimization (Schulman, Wolski, et al. 2017)	Model-free
DDPG	Deep Deterministic Policy Gradient (Lillicrap et al. 2019)	Model-free
TD3	Twin Delayed DDPG (Fujimoto, Hoof, and Meger 2018)	Model-free
SAC	Soft Actor Critic (Haarnoja et al. 2018)	Model-free
IMPALA	Importance Weighted Actor Learner (Espeholt et al. 2018)	Model-free

Table 2.1: Survey of common deep RL algorithms.

2.2.2 Deep RL Agents

To optimize the RL objective, agents either take a *model-free* or *model-based* approach. The distinction is that *model-free* algorithms do not attempt to learn or use a predictive model of the environment; yet, *model-based* algorithms employ a predictive model of the environment to achieve the RL objective. A trade-off between these approaches is that when it is possible to quickly learn a model of the environment or the model is already known, model-based algorithms tend to require much less interaction with the environment to learn good-performing policies (Janner et al. 2019; Sutton and Barto 2018). Yet, frequently, learning a model of the environment is very difficult, and in these cases, model-free algorithms tend to outperform (Janner et al. 2019).

Neural networks become useful in RL when the environment has a large observation-action space², which happens frequently with realistic decision-making problems. Whenever there is a need for an agent to approximate some function, typically a function to represent the policy and/or to model the transition dynamics, neural networks can be used in this capacity due to their property of being general function approximators (Hornik, Stinchcombe, and White 1989). Although there are other function approximators that can be used in RL, e.g. Gaussian processes (Grande, Walsh, and How 2014), neural networks have excelled in this role because of their ability to learn complex, non-linear, high dimensional functions and their ability to adapt given new information (Arulkumaran et al. 2017). There is a multitude of deep RL algorithms since there are many design choices that can be made in constructing a deep RL agent. In Table 2.1, we present some of the more common deep RL algorithms which serve as good reference points for the current state of deep RL.

Training a deep RL agent involves allowing the agent to interact with the environment for potentially thousands to millions of time steps. During training, the deep RL agent continually updates its neural network parameters so that it will converge to an optimal policy. The amount of time needed for an agent to learn high reward yielding behavior cannot be predetermined and depends on a host of factors including the complexity of the environment, the complexity of the agent, and more. Yet, overall, it has been well established that deep RL agents tend to be very sample inefficient (Gu et al. 2017), so it is recommended to provide a generous training budget for these agents.

The deep RL agent controls the learning process with parameters called *hyperparamete-*

²Conventionally, an observation-action space is considered to be large when it is non-tabular, i.e. cannot be represented in a computationally tractable table.

ters. Examples of hyperparameters include the step size used for gradient ascent and the interval to interact with the environment before updating the policy. In contrast, a weight or bias in an agent’s neural network is simply called a *parameter*. Parameters are learned by the agent, but the hyperparameters must be specified by the RL practitioner. Since the optimal hyperparameters vary across environments and can not be predetermined (Henderson et al. 2019), it is necessary to find a good-performing set of hyperparameters in a process called hyperparameter tuning which uses standard multi-dimensional optimization methods.

2.2.3 RL Objective

The reinforcement learning environment is typically formalized as a discrete-time partially observable Markov decision process (POMDP). A POMDP is a tuple that consists of the following:

- \mathcal{S} : a set of states called the state space
- \mathcal{A} : a set of actions called the action space
- Ω : a set of observations called the observation space
- $E(o_t|s_t)$: an emission distribution, which accounts for an agent’s observation being different from the environment’s state
- $T(s_{t+1}|s_t, a_t)$: a state transition operator which describes the dynamics of the system
- $r(s_t, a_t)$: a reward function
- $d_0(s_0)$: an initial state distribution
- $\gamma \in (0, 1]$: a discount factor which describes how much the agent will value rewards to be received in the distant future versus the immediate future (Colin Whitcomb Clark 2010)

The agent interacts with the environment in an iterative loop, whereby the agent only has access to the observation space, action space and the discounted reward signal, $\gamma^t r(s_t, a_t)$. As the agent interacts with the environment by selecting actions according to its policy, $\pi(a_t|o_t)$ ³, the agent creates a trajectory, $\tau = (s_0, o_0, a_0, \dots, s_{H-1}, o_{H-1}, a_{H-1}, s_H)$. From these definitions, we can provide an agent’s trajectory distribution for a given policy as,

$$p_\pi(\tau) = d_0(s_0) \prod_{t=0}^{H-1} \pi(a_t|o_t) E(o_t|s_t) T(s_{t+1}|s_t, a_t).$$

The goal of reinforcement learning is for the agent to find an optimal policy distribution, $\pi^*(a_t|o_t)$, that maximizes the expected return, $J(\pi)$:

$$\pi^* = \operatorname{argmax}_\pi \mathbb{E}_{\tau \sim p_\pi(\tau)} \left[\sum_{t=0}^{H-1} \gamma^t r(s_t, a_t) \right] = \operatorname{argmax}_\pi J(\pi).$$

Although there are RL-based methods for infinite horizon problems, i.e. when $H = \infty$, we will only present episodic or finite horizon POMDPs in this chapter.

³The policy can also be conditioned on a history of observations, (o_0, \dots, o_t) .

2.3 Results

We provide two examples that illustrate the application and potential of deep RL to ecological and conservation problems, highlighting both the potential and the inherent challenges. Annotated code for these examples may be found at <https://github.com/boettiger-lab/rl-intro>. All algorithms were run on an NVIDIA Quadro RTX 8000 GPU. The training budget for the fishing scenario was 300K timesteps (3K runs, taking about 25 minutes). The training budget for the tipping point example was 3M timesteps (6K runs, taking around 3 hours). Software details and hyperparameters are provided in the associated GitHub repo. Hyperparameter tuning typically required 100s of training runs using both Optuna, a python-based hyperparameter optimization module, and manual adjustments.

2.3.1 Sustainable harvest

The first example focuses on the important but well-studied problem of setting harvest quotas in fisheries management. This provides a natural benchmark for deep RL approaches, since we can compare the RL solution to the mathematical optimum directly. Determining fishing quotas is both a critical ecological issue (Worm et al. 2006, 2009; Costello et al. 2016), and a textbook example that has long informed the management of renewable resources within fisheries and beyond (Colin W. Clark 1990).

Given a population growth model that predicts the total biomass of a fish stock in the following year as a function of the current biomass, it is straightforward to determine what biomass corresponds to the maximum growth rate of the stock, or B_{MSY} , the biomass at Maximum Sustainable Yield (MSY) (Schaefer 1954). When the population growth rate is stochastic, the problem is slightly harder to solve, as the harvest quota must constantly adjust to the ups and downs of stochastic growth, but it is still possible to show the optimal strategy merely seeks to maintain the stock at B_{MSY} , adjusted for any discounting of future yields (Reed 1979).

For illustrative purposes, we consider the simplest version of the dynamic optimal harvest problem as outlined by Colin W. Clark (1973) (for the deterministic case) and Reed (1979) (under stochastic recruitment). The manager seeks to optimize the net present value (discounted cumulative catch) of a fishery, observing the stock size each year and setting an appropriate harvest quota in response. In the classical approach, the best model of the fish population dynamics must first be estimated from data, potentially with posterior distributions over parameter estimates reflecting any uncertainty. From this model, the optimal harvest policy – that is, the function which returns the optimal quota for each possible observed stock size – can be determined either by analytic (Reed 1979) or numerical (Marescot et al. 2013) methods, depending on the complexity of the model. In contrast, a model-free deep RL algorithm makes no assumption about the precise functional form or parameter values underlying the dynamics – it is in principle agnostic to the details of the simulation.

We illustrate the deep RL approach using the model-free algorithm known as Twin Delayed Deep Deterministic Policy Gradient or more simply, TD3 (Fujimoto, Hoof, and Meger 2018). We compare the resulting management, policy, and reward under the RL agent to that achieved by the optimal management solution [Fig 2]. Despite having no knowledge of the underlying model, the RL agent learns enough to achieve nearly optimal performance.

The cumulative reward (utility) realized across 100 stochastic replicates is indistin-

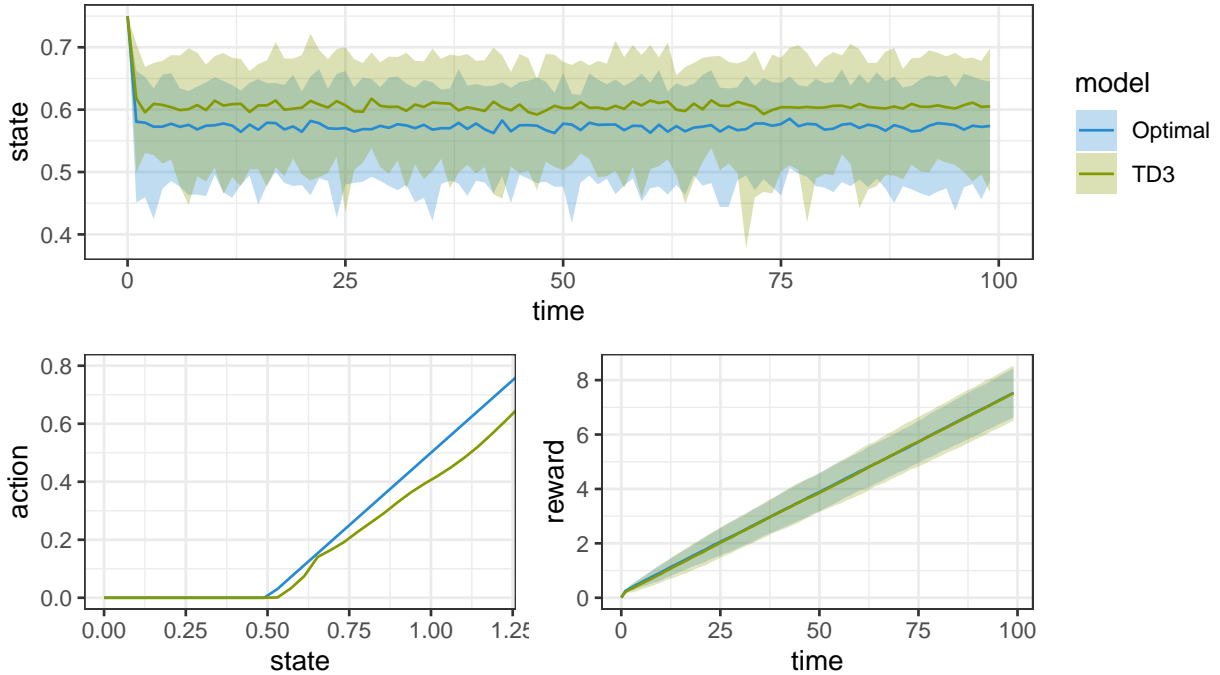


Figure 2.2: Fisheries management using neural network agents trained with RL algorithm TD3 compared to optimal management. Top panel: mean fish population size over time across 100 replicates. Shaded region shows the 95% confidence interval over simulations. Lower left: The optimal solution is policy of constant escapement. Below the target escapement of 0.5, no harvest occurs, while any stock above that level is immediately harvested back down. The TD3 agent adopts a policy that ceases any harvest below this level, while allowing a somewhat higher escapement than optimal. TD3 achieves a nearly-optimal mean utility.

guishable from that of the optimal policy [Fig 2.2]. Nevertheless, comparing the mean state over replicate simulations reveals some differences in the RL strategy, wherein the stock is maintained at a slightly higher-than-optimal biomass. Because our state space and action space are sufficiently low-dimensional in this example, we are also able to visualize the policy function directly, and compare to the optimal policy [Fig 2.2]. This confirms that quotas tend to be slightly lower than optimal, most notably at larger stock sizes. These features highlight a common challenge in the design and training of RL algorithms. RL cares only about improving the realized cumulative reward, and may sometimes achieve this in unexpected ways. Because these simulations rarely reach stock sizes at or above carrying capacity, i.e. larger stock sizes are under-explored, these larger stock sizes show a greater deviation from the optimal policy than we observe at more frequently visited lower stock sizes. This observation brings up a point that is well worth discussing which is how to best identify and resolve under-explored scenarios. Usually, RL practitioners identify under-explored scenarios by either doing extensive testing or visualizing the policy, then tweaking the hyperparameters relevant to exploration in hopes of improving the result.

How could an RL agent be applied to empirical data? One solution would be to train an agent on a simulation environment that approximates the fishery of interest then query the policy of the agent to find a quota for the observed stock. To illustrate this, we examine the quota that would be recommended by our newly trained RL agent, above, against historical harvest levels of Argentine hake based on stock assessments from 1986 - 2014 (RAM Legacy Stock Assessment Database 2020). Hake stocks showed a marked decline throughout this period, while harvests decreased only in proportion [Fig 2.3]. In

contrast, our RL agent would have recommended significantly lower quotas over most of the same interval, including the closure of the fishery as stocks were sufficiently depleted – a stark contrast to the management policy evidenced in the historical catch. Note that we have no way of knowing for sure if the RL quotas would have led to recovery nor do we know the optimal harvest rates, because we can never know the “true model” of the Argentine hake dynamics. We can confirm that the fishery closures seen in the RL agent’s solution are considered optimal under the assumptions of constant escapement theory (Reed 1979) whenever the stock is below the biomass of maximum sustainable yield (B_{MSY}), and that most fisheries models of this stock (RAM Legacy Stock Assessment Database 2020) would suggest that the populations observed in the latter two decades of the data are below that threshold.

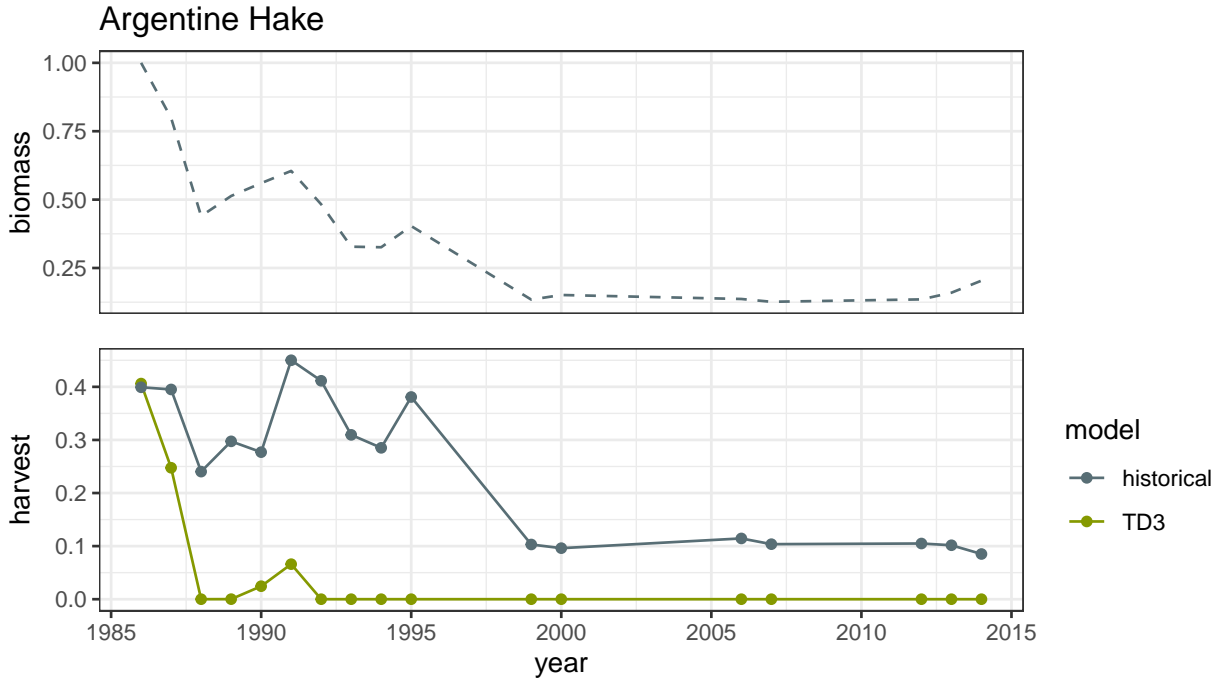


Figure 2.3: Setting fisheries harvest quotas using Deep RL. Argentine Hake fish stocks show a marked decline between 1986 and 2014 (upper panel). Historical harvests (lower panel) declined only slowly in response to consistently falling stocks, suggesting overfishing. In contrast, RL-based quotas would have been set considerably lower than observed harvests in each year of the data. As decline persists, the RL-based management would have closed the fishery to future harvest until the stock recovered.

This approach is not as different from conventional strategies as it may seem. In a conventional approach, ecological models are first estimated from empirical data, (stock assessments in the fisheries case). Quotas can then be set based directly on these model estimates, or by comparing alternative candidate “harvest control rules” (policies) against model-based simulations of stock dynamics. This latter approach, known in fisheries as Management Strategy Evaluation [MSE; Punt et al. (2016)] is already closely analogous to the RL process. Instead of researchers evaluating a handful of control rules, the RL agent proposes and evaluates a plethora of possible control rules autonomously.

2.3.2 Ecological Tipping Points

Our second example focuses on a case for which we do not have an existing, provably optimal policy to compare against. We consider the generic problem of an ecosystem facing slowly deteriorating environmental conditions which move the dynamics closer towards a tipping point [Fig 2.4]. This model of a critical transition has been posited

widely in ecological systems, from the simple consumer-resource model of (Robert M. May 1977) on which our dynamics are based, to microbial dynamics (Dai et al. 2012), lake ecosystem communities (Stephen R. Carpenter et al. 2011) to planetary ecosystems (Barnosky et al. 2012). On top of these ecological dynamics we introduce an explicit ecosystem service model quantifying the value of a more desirable ‘high’ state relative to the ‘low’ state. For simplicity, we assume a proportional benefit b associated with the ecosystem state $X(t)$. Thus when the ecosystem is near the ‘high’ equilibrium \hat{X}_H , the corresponding ecosystem benefit $b\hat{X}_H$ is higher than at the low equilibrium, $b\hat{x}_L$, consistent with the intuitive description of ecosystem tipping points (Barnosky et al. 2012).

We also enumerate the possible actions which a manager may take in response to environmental degradation. In the absence of any management response, we assume the environment deteriorates at a fixed rate α , which can be thought of as the incremental increase in global mean temperature or similar anthropogenic forcing term. Management can slow or even reverse this trend by choosing an opposing action A_t . We assume that large actions are proportionally more costly than small actions, consistent with the expectation of diminishing returns: taking the cost associated with an action A_t as equal to cA_t^2 . Many alterations of these basic assumptions are also possible: our `gym_conservation` implements a range of different scenarios with user-configurable settings. In each case, the manager observes the current state of the system each year and must then select the policy response that year.

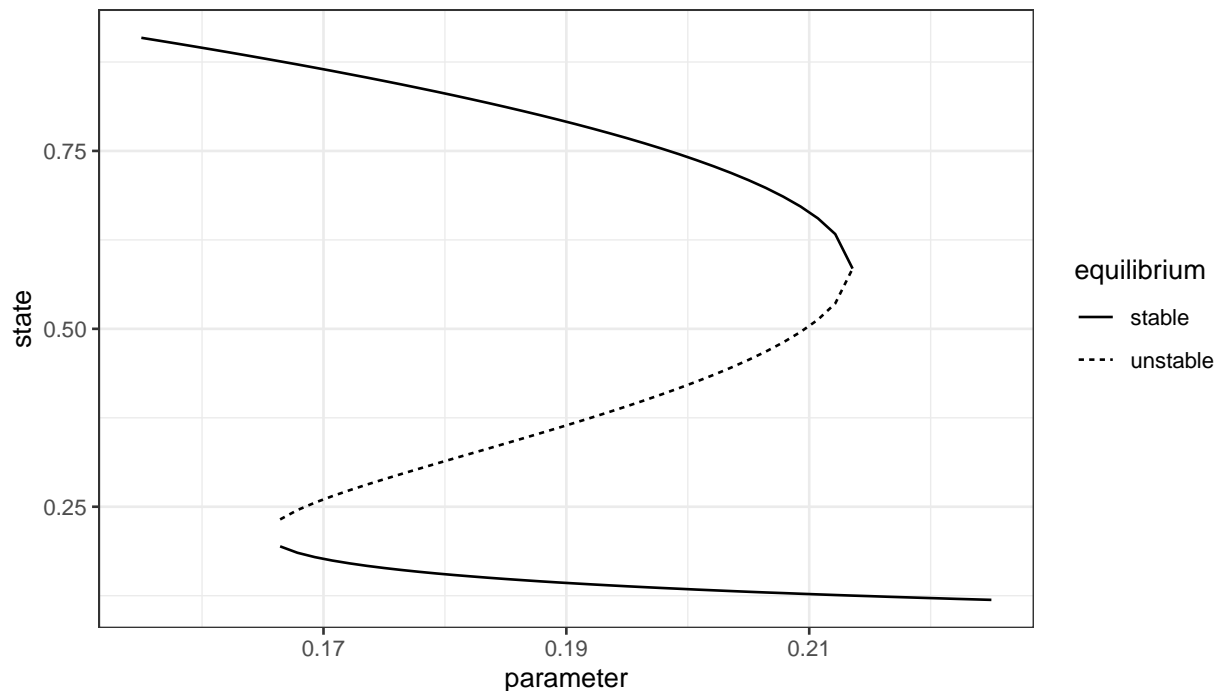


Figure 2.4: Bifurcation diagram for tipping point scenario. The ecosystem begins in the desirable ‘high’ state under an environmental parameter (e.g. global mean temperature, arbitrary units) of 0.19. In the absence of conservation action, the environment worsens (e.g. rising mean temperature) as the parameter increases. This results in only a slow degradation of the stable state, until the parameter crosses the tipping point threshold at about 0.215, where the upper stable branch is annihilated in a fold bifurcation and the system rapidly transitions to lower stable branch, around state of 0.1. Recovery to the upper branch requires a much greater conservation investment, reducing the parameter all the way to 0.165 where the reverse bifurcation will carry it back to the upper stable branch.

Because this problem involves a parameter whose value changes over time (the slowly

deteriorating environment), the resulting ecosystem dynamics are not autonomous. This precludes our ability to solve for the optimal management policy using classical theory such as for Markov Decision Processes (MDP, (Marescot et al. 2013)), typically used to solve sequential decision-making problems. However, it is often argued that simple rules can achieve nearly optimal management of ecological conservation objectives in many cases (Meir, Andelman, and Possingham 2004; Wilson et al. 2006; L. N. Joseph, Maloney, and Possingham 2009). A common conservation strategy employs a fixed response level rather than a dynamic policy which is toggled up or down each year: for example, declaring certain regions as protected areas in perpetuity. An intuitive strategy faced with an ecosystem tipping point would be ‘perfect conservation’, in which the management response is perfectly calibrated to counter-balance any further decline. While the precise rate of such decline may not be known in practice (and will not be known to RL algorithms before-hand either), it is easy to implement such a policy in simulation for comparative purposes. We compare this rule-of-thumb to a policy found by training an agent using the TD3 algorithm.

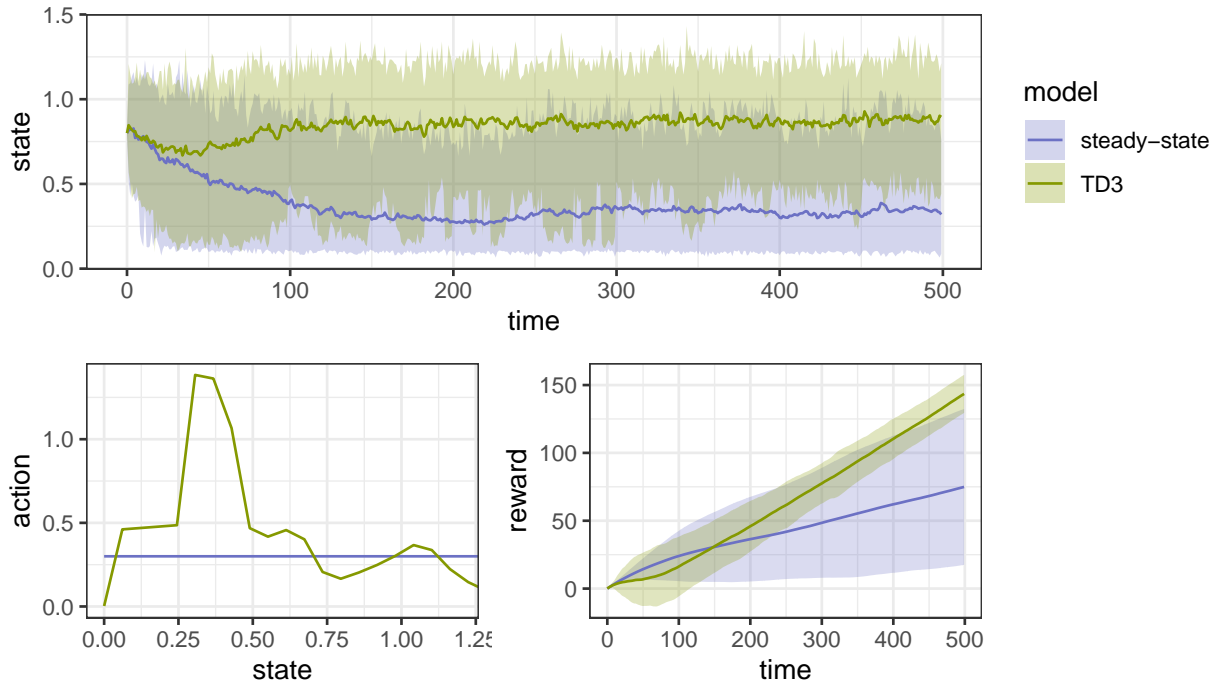


Figure 2.5: Ecosystem dynamics under management using the steady-state rule-of-thumb strategy compared to management using a neural network trained using the TD3 RL algorithm. Top panel: mean and 95% confidence interval of ecosystem state over 100 replicate simulations. As more replicates cross the tipping point threshold under steady-state strategy, the mean slowly decreases, while the TD3 agent preserves most replicates safely above the tipping point. Lower left: the policy function learned using TD3 relative to the policy function under the steady state. Lower right: mean rewards under TD3 management eventually exceed those expected under the steady-state strategy as a large initial investment in conservation eventually pays off.

The TD3-trained agent proves far more successful in preventing chance transitions across the tipping point, consistently achieving a higher cumulative ecosystem service value across replicates than the steady-state strategy.

Examining the replicate management trajectories and corresponding rewards [Fig 2.5] reveal that the RL agent incurs significantly higher costs in the initial phases of the simulation, dipping well below the mean steady-state reward initially before exceeding it in the long run. This initial investment then begins to pay off – by about the 200th

time step the RL agent has surpassed the performance of the steady-state strategy. The policy plot provides more intuition for the RL agent’s strategy: at very high state values, the RL agent opts for no conservation action – so far from the tipping point, no response is required. Near the tipping point, the RL agent steeply ramps up the conservation effort, and retains this effort even as the system falls below the critical threshold, where a sufficiently aggressive response can tip the system back into recovery. For a system at or very close to the zero-state, the RL agent gives up, opting for no action. Recall that the quadratic scaling of cost makes the rapid response of the TD3 agent much more costly to achieve the same net environmental improvement divided into smaller increments over a longer timeline. However, our RL agent has discovered that the extra investment for a rapid response is well justified as the risk of crossing a tipping point increases.

A close examination of the trajectories of individual simulations which cross the tipping point under either management strategy further highlights the difference between these two approaches. Under the steady-state strategy, the system remains poised too close to the tipping point: stochastic noise eventually drives most replicates across the threshold, where the steady-state strategy is too weak to bring them back once they collapse. As replicate after replicate stochastically crashes, the mean state and mean reward bend increasingly downwards. In contrast, the RL agent edges the system slightly farther away from the tipping point, decreasing but not eliminating the odds of a chance transition. In the few replicates that experience a critical transition anyway, the RL agent usually responds with sufficient commitment to ensure their recovery. Only 3 out of 100 replicates degrade far enough for the RL agent to give up the high cost of trying to rescue them. The RL agent’s use of a more dynamic strategy out-performs the steady-state strategy. Numerous kinks visible in the RL policy function also suggest that this solution is not yet optimal. Such quirks are likely to be common features of RL solutions – long as they have minimal impact on realized rewards. Further tuning of hyper-parameters, increased training, alterations or alternatives to the training algorithm would likely be able to further improve upon this performance.

2.3.3 Additional Environments

Ecology holds many open problems for deep RL. To extend the examples presented here to reflect greater biological complexity or more realistic decision scenarios, the transition, emission and/or reward functions of the environment can be modified. We provide an initial library of example environments at <https://boettiger-lab.github.io/conservation-gym>. Some environments in this library include a wildfire `gym` that poses the problem of wildfire suppression with a cellular automata model, an epidemic `gym` that examines timing of interventions to curb disease spread, as well as more complex variations of the fishing and conservation environments presented above.

2.4 Discussion

Ecological challenges facing the planet today are complex, and their outcomes are both uncertain and consequential. Even our best models and best research will never provide a crystal ball to the future, only better elucidate possible scenarios. Consequently, that research must also confront the challenge of making robust, resilient decisions in a changing world. The science of ecological management and quantitative decision-making has a long history (e.g. Schaefer 1954; Walters and Hilborn 1978) and remains an active area of research (Wilson et al. 2006; Fischer et al. 2009; Polasky et al. 2011a). However, the

limitations of classical methods such as optimal control frequently constrain applications to relatively simplified models (Wilson et al. 2006), ignoring elements such as spatial or temporal heterogeneity and autocorrelation, stochasticity, imperfect observations, age or state structure, and other sources of complexity that are both pervasive and influential on ecological dynamics (Hastings and Gross 2012). Complexity comes not only from the ecological processes but also the available actions. Deep RL agents have proven remarkably effective in handling such complexity, particularly when leveraging immense computing resources increasingly available through advances in hardware and software (Matthews 2018).

This chapter does not set the precedent as the first application of RL to ecology. There have been a number of studies applying RL to behavioral ecology, typically with multi-agent environments (Wang, Cheng, and Wang 2020; Frankenhuis, Panchanathan, and Barto 2019; Perolat et al. 2017). Yet, it is important to distinguish the aim of these behavioral studies from the aim of applying RL to conservation management. In previous behavioral ecology studies, RL algorithms as a substitute for animal learning mechanisms (Wang, Cheng, and Wang 2020; Perolat et al. 2017). When applying deep RL to conservation management, we do not make the assumption that an RL algorithm learns analogously to how an animal learns. We instead propose that RL be used as a tool to search for solutions to decision-making problems.

The examples presented here only scrape the surface of possible RL applications to conservation problems. The examples we have focused on are intentionally quite simple, though it is worth remembering that these very same simple models have a long history of relevance and application in both research and policy contexts. Despite their simplicity, the optimal strategy is not always obvious before hand, however intuitive it may appear in retrospect. In the case of the ecosystem tipping point scenario, the optimal strategy is unknown, and the approximate solution found by our RL implementation could almost certainly be improved upon. In these simple examples in which the simulation implements a single model, training is analogous to classical methods which take the model as given (Marescot et al. 2013). But classical approaches can be difficult to generalize when the underlying model is unknown. In contrast, the process of training an RL algorithm on a more complex problem is no different than training on a simple one: we only need access to a simulation which can generate plausible future states in response to possible actions. This flexibility of RL could allow us to attain better decision-making insight for our most realistic ecological models like those used for the management of forests and wildfire (Pacala et al. 1996; Moritz et al. 2014), disease (Dobson et al. 2020), marine ecosystems (Steenbeek et al. 2016), or global climate change (Nordhaus 1992).

The rapidly expanding class of model-free RL algorithms is particularly appealing given the ubiquitous presence of model uncertainty in ecological dynamics. Rarely do we know the underlying functional forms for ecological processes. Methods which must first assume something about the structure or functional form of a process before estimating the corresponding parameter can only ever be as good as those structural assumptions. Frequently, available ecological data are insufficient to distinguish between possible alternative models (Knape and Valpine 2012), or the correct model may be non-identifiable with any amount of data. Model-free RL approaches offer a powerful solution for this thorny issue. Model-free algorithms have proven successful at learning effective policies even when the underlying model is difficult or impossible to learn (Pong et al. 2020), as long as simulations of possible mechanisms are available.

Successfully applying RL to complex ecological problems is no easy task. Even on rel-

Issue	Description
Generalization	Agents struggle to adapt to tasks not seen in training (Kirk et al. 2022).
Reproducibility	It can be very challenging to replicate results due to a host of reasons like differences in computational hardware (Henderson et al. 2019).
Lack of Transparency	Deep RL users cannot interpret why agents select actions (Castelvecchi 2016).
Hyperparameter Instability	Agent performance can vary significantly over slight alterations in hyperparameters, like initialization seed (Henderson et al. 2019).
Reward Misspecification	Agents commonly learn undesirable behavior that still maximizes the RL objective (Hadfield-Menell et al. 2020).
High Capital Demands	Landmark successes like AlphaGo and AlphaStar have required very large teams of researchers and large amounts of computational power (Silver et al. 2017; Vinyals et al. 2019).
Sample Inefficiency	Current algorithms require large amounts of interaction with the environment to achieve reward maximization (Haarnoja et al. 2018).

Table 2.2: Practical issues with deep RL.

actively uncomplicated environments, training an RL agent can be more challenging than expected due to an entanglement of reasons, see Table 2, like hyperparameter instability and poor exploration that can be very difficult to resolve (Henderson et al. 2019; Berger-Tal et al. 2014). It is also worth acknowledging that deep RL algorithms, particularly model-free algorithms, have poor sample efficiency, which could limit deep RL from being effective on environments that are slow to run (Haarnoja et al. 2018). Thus, as Sections 5.1 and 5.2 illustrate, it is important to begin with simple problems, including those for which an optimal strategy is already known. Such examples provide important benchmarks to calibrate the performance, tuning and training requirements of RL. Once RL agents have mastered the basics, the examples can be easily extended into more complex problems by changing the environment. Yet, even in the case that an agent performs well on a realistic problem, there will be a range of open questions in using deep RL to inform decision-making. Since deep neural networks lack transparency (Castelvecchi 2016), can we be confident that the agent will generalize its past experience to new situations – especially when we cannot readily visualize the policy? To gain such confidence, it will be necessary to do extensive testing on previously unseen contexts (Kazak et al. 2019), but even then, it can be difficult to verify that the agent will perform as expected. Given that there have been many examples of reward misspecification leading to undesirable behavior (Hadfield-Menell et al. 2020), what if we have selected an objective that unexpectedly causes damaging behavior? Reward misspecification is not unique to RL and has long been a central problem in ecological management and decision-making (Gregory et al. 2012; Conroy and Peterson 2013), but it is important to make clear that RL does not resolve this issue. A greater role of algorithms in conservation decision-making also raises questions about ethics and power, particularly when those algorithms are opaque or proprietary (Scoville et al. 2021; Chapman et al. 2021).

Yet, a more immediate barrier to the use of deep RL in conservation is deep RL’s hardware requirements. Depending on the complexity of the RL environment and agent, the equipment necessary to train an agent can vary widely. The examples shown above were selected so they can be replicated on a personal computer, but more realistic prob-

lems will likely require specialized computational resources. For instance, one of the most notable achievements in RL, Alphastar, required 33 TPUs, processors that are specialized for deep learning, for more than 40 days (Vinyals et al. 2019). Fully detailed conservation decision-making problems will likely require comparable specialized algorithms and hardware that ecologists do not generally have access to. For deep RL to be an effective tool for conservation, there will need to be large investments of time and money, and extensive collaboration across computer science and ecology.

Deep RL is still a very young field, where despite several landmark successes, potential far outstrips practice. Recent advances in the field have proven the potential of the approach to solve complex problems (Silver et al. 2016, 2017, 2018; Mnih et al. 2015), but typically leveraging large teams with decades of experience in ML and millions of dollars worth of computing power (Silver et al. 2017). Successes have so far been concentrated in applications to games and robotics, not scientific and policy domains, though this is beginning to change (Popova, Isayev, and Tropsha 2018; Zhou, Li, and Zare 2017). Iterative improvements to well-posed public challenges have proven immensely effective in the computer science community in tackling difficult problems, which allow many teams with diverse expertise not only to compete but to learn from each other (Villarreal, Taylor, and Tucci 2013; Deng et al. 2009). By working to develop similarly well-posed challenges as clear benchmarks, ecology and environmental science researchers may be able to replicate that collaborative, iterative success in cracking hard problems.

Chapter 3

Limits to Ecological Forecasting: Estimating Uncertainty for Critical Transitions with Deep Learning

This chapter was previously published, see Lapeyrolerie and Boettiger, 2023. It is included here with permission from the co-authors.

Forecasting and decision-making are inherently interconnected. At the core of many solutions for decision-making problems, including all the deep RL models mentioned previously, is a forward prediction problem: the agent often has the central task of predicting the expected cumulative rewards for following a given policy. Thus, improving our ability to forecast ecological processes will likely improve our ability to make better decisions for conservation problems. Chapter 3 continues the exploration of neural network models for issues in ecology and conservation but changes focus from decision-making to time series forecasting.

3.1 Introduction

Forecasting plays an important and rapidly growing role in both testing our fundamental understanding of ecological processes, and informing ecological applications and conservation decision-making (M. C. Dietze et al. 2018; Schindler, Armstrong, and Reed 2015). Meanwhile, recent advances in machine learning have rapidly improved the prevalence and accuracy of short term forecasts in many fields (Kao et al. 2020; Lyu et al. 2020; Du et al. 2020). Will these emerging methods improve the capacity for forecasts in ecological systems as well? Ecological dynamics are notoriously complex, with uncertainty and non-linearity playing critical roles (Boettiger 2018a; Hallett et al. 2004; Ovaskainen and Meerson 2010). These challenges are nowhere more evident than in *critical transitions*, sudden shifts in the states or patterns of ecosystem dynamics that are more important and more difficult to predict than gradual changes. Here, we examine several of the best-known examples of critical transitions in ecological systems. We evaluate the most promising machine learning methods for probabilistic forecasts relative to traditional statistical and mechanistic approaches applied to several classic models in ecology.

In this chapter, we focus on the task of producing quantitative, probabilistic forecasts reflecting the possible distribution of future states, as frequently called for in ecological research (J. S. Clark et al. 2001; M. C. Dietze et al. 2018). Such forecasting tasks may

arise whenever a manager is interested in knowing the future states of a system: such as setting future catch quotas for a fishery or adjusting eradication effort for an invasive species. It is important to distinguish this objective from the extensive previous literature on “early warning signs” of critical transitions, as reviewed in Scheffer et al. (2009), which has sought to answer only a categorical question: is the system approaching a critical transition? Recent work such as Bury et al. (2021) has introduced ML methods to consider classification of this transition in four possible categories (Hopf, saddle-node, transcritical, or no bifurcation) rather than two (bifurcation or not). These are important results with considerable promise (Lapeyrolerie and Boettiger 2021), but which nevertheless address a very different question using very different methods. Early warning signals only predict “a big change may be coming soon” – they do not try to forecast when or how big. As we shall see, there is good reason to focus on that more modest, qualitative objective when faced with systems that might produce critical transitions. Here, we examine the more ambitious questions of forecasting when and how much change: or more precisely, of making probabilistic forecasts of all future states over a given time horizon.

3.2 Materials and Methods

We will focus the analysis on several different forecasting scenarios based around two classic models in population ecology: Robert May’s consumer-resource model (Robert M. May 1977), and the Nicholson-Bailey parasitoid-host model (A. J. Nicholson and Bailey 1935). Though these models may appear simple when measured against high-dimensional and parameter rich models found in some management contexts such as fisheries, they can exhibit rich nonlinear dynamics and provide greater capacity to generalize (Levins 1966; Getz et al. 2018). These textbook models have been well studied and form the basis of half a century of research in ecology, including much recent work on topics such as resilience and tipping points which has had important theoretical and practical management outcomes (Folke et al. 2004; Fischer et al. 2009; Polasky et al. 2011a). May’s model exhibits alternative stable states. In this one-dimensional model, transitions between these states can occur due to intrinsic stochasticity, external forcing, or the gradual environmental change that results in a catastrophic saddle-node bifurcation and generates hysteresis. The Nicholson-Bailey model is a two species model which contains a supercritical Hopf bifurcation, a non-catastrophic bifurcation which either creates or destroys a limit cycle – a stable oscillatory pattern.

Assessing the accuracy of forecasting methods in the face of such bifurcation dynamics is a particularly important question for ecological systems and global environmental change problems. Bifurcations represent the kind of non-linear responses complex systems can make as the result of slowly changing parameters. This can create a particularly challenging forecasting task when such transitions have not been previously observed in the same system, requiring the forecast to anticipate dynamics for which there are no analogs in the historical data. Forecast skill under such no-analog conditions may be particularly relevant to ecological forecasting in the context of global change (Williams and Jackson 2007).

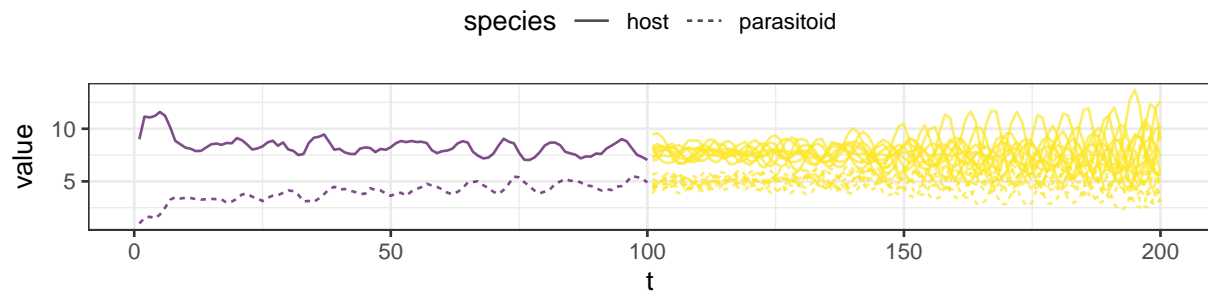
We provide fully reproducible coded examples in R and Python for performing, scoring, and visualizing each of the forecasts considered here. After significant time spent considering alternative frameworks, we have emphasized those which best met our requirements for performance, ease-of-use, flexibility, and support for the latest probabilistic machine learning models for forecasting. Most of our forecasts use the `darts` framework, a sophisticated and well documented Python library with support for a wide range

of methods. Our model-based MCMC forecasts use the `greta` framework, a R library that uses Python-based `TensorFlow Probability` to achieve better performance. While Python-based frameworks currently have the edge in performance and access modern ML algorithms, they lag behind in attention to statistical issues such as the computation of strictly proper skill scores.

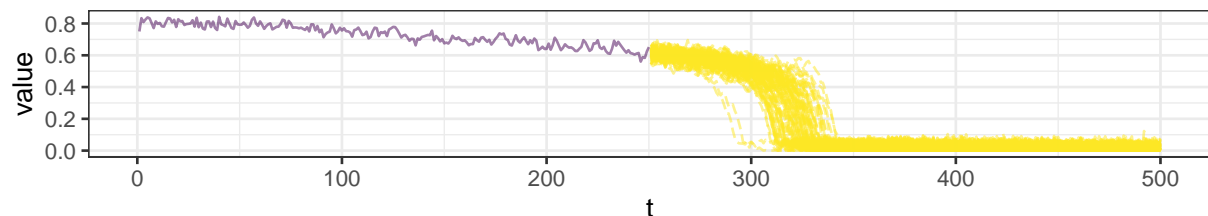
Our examples of scoring and visualization will rely on a collection of R packages, in particular, `scoringRules` for the efficient calculation of Continuous Ranked Probability Score (CRPS) and logarithmic probability (Logs) scores for forecast ensembles (Gneiting and Raftery 2007). Following popular conventions, we express both skill scores in error-orientation, that is, larger values indicate worse skill (higher degree of error).

We expect greater convergence between methods available in R and Python in the future, as already illustrated in the example of `greta`. Complete code for all examples presented here can be found at https://github.com/boettiger-lab/mee_tipping_point_forecasting.

A. Hopf bifurcation



B. Saddle-node bifurcation



C. Stochastic transition

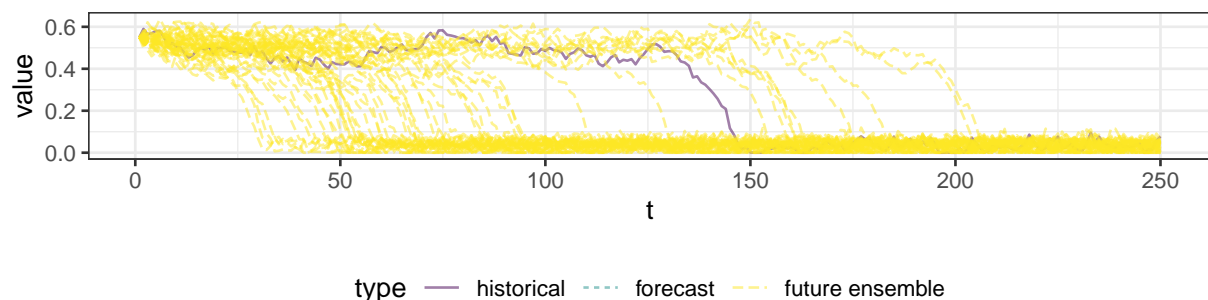


Figure 3.1: Forecast scenarios. A. The Hopf bifurcation: a stable node develops into a limit cycle which gradually grows larger in this predator-prey model. B. The saddle-node bifurcation in a single species. C. The stochastic transition in a single species. Plots show historical data used to train the algorithm in purple, and replicate simulations of the true dynamics (‘future ensemble’) in yellow. Note how the characteristic time for the critical transition varies across the transitions. We will examine forecasts of various models (Fig 3.2) which will each produce probabilistic forecast distributions (blue, Fig 3.3-5) seeking to match the true future ensemble (yellow) as closely as possible.

3.2.1 Scenario 1: Hopf Bifurcation

The Nicholson-Bailey model describes a predator-prey dynamic for the relationship of a host species and an obligate parasitoid, originally used to model the population dynamics of blowflies (*Lucilia cuprina*) (A. J. Nicholson and Bailey 1935; A. Nicholson 1954a, 1954b). We consider the form which includes density dependence in the host species, and we allow for environmental stochasticity,

$$H_{t+1} = H_t \exp \left(r \left(1 - \frac{H}{K_t} \right) - cP_t + \eta_{H,t} \right) \quad (3.1)$$

$$P_{t+1} = H_t \exp \left(r \left(1 - \frac{H}{K_t} \right) \right) (1 - \exp(-cP_t + \eta_{P,t})) \quad (3.2)$$

$$K_{t+1} = K_t + \delta \quad (3.3)$$

Where H_t is the population density of the host species at time t , (in arbitrary units) and P_t is the population density of the parasitoid. The time step is defined by the generation time of the parasitoid, which is about two weeks in the case of Nicholson’s blowflies (A. Nicholson 1954a). Following Dakos et al. (2012), we further allow the carrying capacity of the host, K to slowly increase at a linear rate, which drives a supercritical Hopf bifurcation as K becomes sufficiently large. In a Hopf bifurcation, a stable node starts an oscillatory pattern which grows in amplitude as the bifurcation parameter continues to increase. In this model, the Hopf bifurcation is dubbed ‘supercritical’ as it creates a stable limit cycle instead of an unstable one. This example illustrates one of the many kinds of challenges which nonlinear phenomena pose to forecasting: the “historical” data prior to the bifurcation never exhibit the cyclical dynamics of growing amplitude that will emerge after the bifurcation occurs. If we had used a purely deterministic model, the dynamics would be constrained to a single stable point, corresponding to a slowly changing steady-state population size of host and parasitoid populations. However, stochasticity in this case acts as a source of some additional information about the dynamics, as the noise excites quasi-cycles which are visible in the irregular oscillations that appear significantly prior to the emergence of true limit cycles which follow the bifurcation (Boettiger 2018b). Examples use the following parameters: $H_0 = 9$, $P_0 = 1$, $r = 0.75$, $c = 0.1$, $K_0 = 14$, $\delta = 0.08$, $\sigma_H = 0.02$, $\sigma_P = 0.02$.

3.2.2 Scenario 2: The saddle-node bifurcation

A yet more difficult forecasting scenario is created by the saddle-node bifurcation. May’s consumer-resource model is an one-dimensional model describing the growth of a ‘resource’ population (e.g. herbivore) which is grazed by a consumer (Robert M. May and Anderson 1979). As in the Nicholson-Bailey model, in the absence of that predation, the resource population density grows under a density-dependent pattern described by a logistic function. The resource population is also grazed by a consumer at a rate given by a Holling type III s-curve (typically used to model handling time). For a certain range of parameter choices, this model supports alternative stable state dynamics, and has been identified and employed in explaining alternative stable state dynamics in a broad range of ecological and socio-ecological systems (Scheffer et al. 2001b).

$$N_{t+1} = N_t + rN_t \left(1 - \frac{N_t}{K}\right) - \frac{h_t N_t^2}{s^2 + N_t^2} + \eta_t \quad (3.4)$$

$$h_{t+1} = h_t + \alpha \quad (3.5)$$

$$\eta_t \sim \mathcal{N}(0, \sigma) \quad (3.6)$$

If the environment slowly alters one of the parameters (say, the encounter efficiency, h_t , in our formulation), one of the stable nodes moves closer and closer to the unstable saddle point, leading to a bifurcation that destroys the stable state, leaving the system to suddenly transition to the alternative stable state. Saddle-node bifurcations (also known as fold bifurcations) also create a phenomenon known as hysteresis, where it is not sufficient to restore the environment to the previous parameter values to recover the previous state. Unlike the supercritical Hopf bifurcation which exhibits a continuous transition from a stable node to a small limit cycle that then grows, the saddle-node transition is a discontinuous or so-called ‘catastrophic’ bifurcation. Due both to this sudden, catastrophic nature of the transition and the difficulty in reversing the shift after it has occurred, saddle-node bifurcations have been the subject of intense study.

Tipping point dynamics have long been identified as an important but difficult challenge for forecasting (e.g. Scheffer et al. 2001a; Folke et al. 2004). Much effort in the ecological literature so far has focused on identifying any ‘early warning signs’ that a catastrophic bifurcation might occur at all (Scheffer et al. 2009, 2012) rather than more ambitious attempts to provide quantitative probabilistic forecasts of the likely distribution of waiting times before such a transition occurs. Tipping points resulting from saddle-node bifurcations have been demonstrated in examples ranging from laboratory microcosms (Dai et al. 2012; Dai, Korolev, and Gore 2015) to whole-ecosystem experiments (S. R. Carpenter et al. 2011), and postulated as a model for global change (Barnosky et al. 2012). Examples use the following parameters: $r = 1$, $K = 1$, $s = 0.1$, $h_0 = 0.15$, $\alpha = 0.000375$, $\sigma = 0.02$, $N_0 = 0.75$.

3.2.3 Scenario 3: The stochastic transition

Perhaps the most difficult of all events to predict are those in which large transitions are predominately driven by a random component. An example of such a transition event is possible to observe in May’s consumer-resource model, in which a stochastic term occasionally results in a transition between alternative stable states. In such cases, no forecast can precisely predict when a transition will occur, but it is nonetheless possible to deduce the correct distribution of waiting times knowing the correct model. In the case of small noise, transitions are Poisson distributed, such that the distribution of waiting times is roughly exponential (e.g. Kampen 1992), though post-hoc the trajectories of such transitions can be mistaken for saddle-node transitions (Boettiger and Hastings 2012). To consider such cases, we will again use May’s alternative stable state model, though this time leaving all parameters fixed.

In this context, predicting the probability of a transition in the future based solely on observations prior to a transition occurring is essentially impossible without additional information constraining the model estimate, as such data is equally consistent with infinitely many models or parameter choices which share the same local linearization about the stable point. Unlike the saddle-node bifurcation, there is no slowly warping potential basin which can be detected to inform estimates. Thus, in this scenario, rather than

considering the problem of predicting the future evolution of a single time series based only on its historical values, we consider an alternative framing of the task: we imagine our forecaster has access to historical data from one or more comparable systems which includes a previous stochastic transition event. Based on this data, our forecaster seeks to identify the distribution of expected transition times for analogous systems starting from the same initial condition. This parallels actual practice in which researchers would draw on previous examples of stochastic transitions in a system - lake-ecosystem shifts, disease emergence, changing fire regimes, (Scheffer et al. 2001a; Folke et al. 2004). (Note that such stochastic transitions between alternative stable states can also create oscillatory-like dynamics when stochasticity is sufficiently high enough to drive repeated transitions from one attractor to the other and back again. In such cases, it might be reasonable to estimate a strictly forward-looking forecast of a single system, predicting the distribution of these transitions.) Model definition is the same as May’s model for the saddle node with fixed parameter h , values: $r = 1$, $K = 1$, $s = 0.1$, $h_0 = h = 0.26$, $\alpha = 0$, $\sigma = 0.02$, $N_0 = 0.55$.

3.2.4 Selecting timescales

In each scenario, $t=0$ is the start time of the training data, while the length of training data and forecast horizon (with ensembles sampled from the true distribution) are illustrated in Fig 3.1. For the Hopf bifurcation, forecasts begin at $t=100$ and extend to $t=200$; for the saddle node, forecasts begin at $t=250$ and extend to $t=500$; and, for the stochastic transition, both training data and forecasting tasks begin at 0 and extend to $t=250$. While much attention is often paid to the number of data points in training or testing data, it is essential to realize that these are only meaningful relative to the specific process in question. Thus, in each case, we have selected these time intervals to focus on the dynamical process in question, which unfolds at a different rate and tempo in each scenario. For instance, if the stochastic scenario was restricted to the much shorter timescale used in the Hopf case, few replicate simulations would experience a transition at all. If length of the stochastic transition timeseries was made much longer, most of the timeseries would be spent post-transition. Likewise, if the forecast horizon for the Hopf scenario was extended much further into the future under the current parameterization, the system would experience a homoclinic bifurcation at which the population collapses to 0. Using different length timescales allows us to consider the three different forecasting tasks illustrated in Fig 3.1 that focus around predicting the critical behavior, rather than predicting long periods of relative stasis. These three critical transitions are fundamentally different processes, there is no perfect apples-to-apples parameterization for each that allows the transition to unfold in a way that gives precisely the same time windows.

3.2.5 Method group 1: Markov Chain Monte Carlo

As a reference case, we consider forecasts produced by MCMC estimates of model parameters, *given the true model*. This represents an idealized case where the nature of the underlying process is known precisely. Uncertainty comes from parameter estimates and intrinsic stochasticity specified in the model, but does not reflect any uncertainty in our knowledge of the model structure. Alternative model structures, even when capable of producing the same nonlinear phenomena (i.e. the same bifurcations) will give very different forecasts. Even alternative prior distributions of the parameters will generally yield alternate forecasts, as likelihood ridges are common to nonlinear models. Thus, this

case represents a theoretical upper bound for the performance of forecasts by techniques which do not make such strong assumptions about the underlying processes.

3.2.6 Method group 2: Statistical models (ARIMA)

We present forecasts produced by ARIMA models as the model-free analogs to the forecasts made using parameter estimation with MCMC. Since ARIMA models make the assumption that the future will resemble the past via ARIMA’s auto-regressive and moving average components (Hyndman and Athanasopoulos 2018), these models are not well-suited for problems with complex bifurcation dynamics. Thus, ARIMA-based forecasts should be treated as a lower bound for the performance of non-mechanistic models. In contrast to inference with MCMC, uncertainty with ARIMA models is estimated directly from the learned parameters (Hyndman and Athanasopoulos 2018). Since ARIMA is a commonly encountered method, we will refer readers to Hyndman and Athanasopoulos (2018) for further discussion.

3.2.7 Method group 3: Machine Learning models

Over the past decade, deep learning has become very popular for a broad range of challenging time series prediction problems (Makridakis, Spiliotis, and Assimakopoulos 2018). Deep learning models are often used to make point forecasts, but for their application to ecological time series, it will often be necessary to use multi-step, probabilistic forecasts. For all the deep learning models in this study, we use the same general process. Each machine learning model is trained on one time series drawn from the three scenarios described previously. For the Hopf and saddle node cases, these time series consist of the period leading up to the bifurcation. A critical transition is, however, included in the training set for the stochastic transition case. Each model is trained to learn the parameters of a Laplace distribution for every time step in the forecast horizon. To produce a forecast, we input a time series into a model, then we draw samples from the distributions that were learned during training.

A major nuisance with deep learning methods is their instability to hyperparameters and initialization seeds (Madhyastha and Jain 2019). We found that for the same set of hyperparameters, we could produce starkly different forecasts if we trained the same model with different initialization seeds. One explanation for this instability is that machine learning models often get stuck on the local optima of loss surfaces (Madhyastha and Jain 2019). Another likely cause is that machine learning models commonly overfit the training data (Mehta et al. 2019). Across deep learning, overfitting is a fundamental issue, arising from neural networks being highly overparameterized (Dar, Muthukumar, and Baraniuk 2021). With so many parameters, deep learning models tend to have high variance and thus overfit the training data, a consequence of the bias-variance trade-off common across statistics and machine learning (Mehta et al. 2019). One frequently used method to reduce overfitting is K-fold cross validation (Raschka 2020), but this approach cannot be effectively employed when there is one or few time series in the training set. To remedy the instability problem, we use an ensemble-based method, wherein each ML forecast is the union of forecasts from 5 individual models that were trained with different initialization seeds. We found this simple ensemble technique to be an effective way to improve generalizability in the limited data regime.

Recently, it has become established that using memory or attention-based neural networks, and an encoder-decoder architecture is crucial for improving forecasting performance on time series data (Kao et al. 2020; Lyu et al. 2020; Du et al. 2020). Herein

we will provide some background on what these machine learning methods are and their benefits.

3.2.7.1 Recurrent Neural Networks

Recurrent neural networks (RNN's) are the predominant memory-based deep learning method. Recurrent neural networks differ from feed-forward neural networks in that a recurrent neural network provides feedback to itself between time steps (Sherstinsky 2020). By providing self-feedback, recurrent neural networks are able to retain information from previous time steps and thus learn temporal dependencies. However, a standard recurrent neural network is unwieldy to train because of the vanishing and exploding gradient problem (Pascanu, Mikolov, and Bengio 2013), so there have been specialized neural network architectures designed to avoid these gradient problems. Long Short-term Memory (LSTM) and Gated Recurrent Units (GRU) Networks are considered to be the state of the art recurrent neural networks that address exploding and vanishing gradients (Chung et al. 2014). These methods avoid gradient problems by regulating the self-feedback via gates which perform operations on the feedback signal – see Chung et al. (2014) for more details. While GRU's and LSTM's commonly outperform standard RNN's, it is difficult to anticipate whether GRU's or LSTM's will be best suited for any time series problem (Chung et al. 2014), so we investigate both methods.

3.2.7.2 Transformers

The Transformer is a state of the art ML architecture that is able to model long and short term dependencies on sequence to sequence tasks (Vaswani et al. 2017). Transformers use a mechanism called self-attention which interrelates different positions of the input sequence in order to find an informative representation of the input sequence (Vaswani et al. 2017). For example, if given a sentence, a transformer could learn the contextual relationship between a subject and a direct object, but a recurrent neural network would process all the words as one phrase. Because of self-attention, Transformers do not need to process data sequentially and thus can be parallelized, offering significant computational advantages (Vaswani et al. 2017). The Transformer is likely to be a foundational method for future AI research (Bommasani et al. 2021), so we considered it critical to investigate Transformers in this study.

3.2.7.3 Encoder-Decoders

Encoder-decoder architectures have been shown empirically to excel on sequence to sequence tasks (Aitken et al. 2021). Encoder-decoders work by processing the input sequence into a fixed-length vector then decoding this fixed-length vector to an output sequence. It is thought that by encoding the input sequence to a vector, encoder-decoders find informative representations of the input sequence that make the prediction task much easier (Sutskever, Vinyals, and Le 2014). Note that it is possible to use any type of neural network as the encoder and the decoder, but it is most common to use recurrent neural networks or networks with attention mechanisms (Aitken et al. 2021). Of the models that we present, Block RNN's are a direct example of an encoder-decoder-based model since a Block RNN employs a RNN as an encoder and a separate RNN as a decoder. Transformers also have an encoder and decoder component.

3.2.8 Forecast skill: strictly proper scores

To compare forecasts, we focus exclusively on metrics of forecast skill which satisfy the property from Gneiting and Raftery (2007) of a strictly proper score. This ensures the very desirable behavior that no probabilistic forecast $Q(x, t)$ can have a score as high as the score of the true process $P(x, t)$ on average. In other words, while it is possible for any of the models considered to *overfit* the data against which they are trained, i.e. have a higher likelihood than the true process, it is not possible for these models to overfit the data against which they are scored. It is worth noting that this property applies specifically to probabilistic forecasts and not point forecasts. Not all common metrics often used to compare forecasts are strictly proper – such as the average root-mean-square error or the average absolute error. Concerns about over-fitting arise in most types of model estimation and are a particularly acute concern to machine learning methods due to the bias-variance trade-off (Mehta et al. 2019). This makes the use of strictly proper scoring especially relevant in assessing machine learning predictions.

Not even all strictly proper scores will agree on the same relative ranking between forecasts. We will focus on two of the most common such skill metrics, CRPS score and log probability score (negative log likelihood) (e.g. see Gneiting and Raftery 2007; Gneiting and Katzfuss 2014). We define these scores explicitly in Equations 7-8, where F and f respectively correspond to the cumulative distribution function and probability density function of the forecast; and, y denotes an observation. Of the two metrics, the log probability score puts a much a greater penalty on unexpected observations than CRPS, and may be more suitable when the occurrence of unexpected events incurs a particularly high cost. Note that while the minus log-likelihood can be negative for sufficiently high probability densities, we use a fixed scalar shift of logs score to ensure the log skill score is strictly positive, which facilitates visualization without impacting relative rankings.

$$\text{LogS}(F, y) = -\log f(y) \tag{3.7}$$

$$\text{CRPS}(F, y) = \int (F(z) - \mathbb{1}\{y \leq z\})^2 dz \tag{3.8}$$

3.3 Results

We examine forecast skill for each of the six forecasting methods (MCMC, ARIMA, block-RNN, GRU, LSTM, and Transformer) in each of our three scenarios (Hopf bifurcation, saddle-node bifurcation, and stochastic transition). In addition to these cases, we also consider an “ensemble model”, generated by drawing from the distribution of all models except the MCMC model – throughout our figures, this ensemble model is denoted by “ml_ensemble”. Such ensemble techniques can better reflect uncertainty than relying on any single method (Gneiting and Raftery 2005). For simplicity, we consider the unweighted case, where each model is represented equally in the ensemble. Using model-based simulations allows us to examine performance against multiple (n=100) replicates of the “true” process, which further helps identify differences that may occur solely due to chance. By taking the true model structure as given, MCMC methods can be used to determine a theoretical limit of forecasting skill. Note that in both bifurcation scenarios, future dynamics will visit states never previously observed in the historical data that was used to train each of the methods (e.g. very small population sizes). This no-analog aspect of forecasting bifurcation dynamics means that even with many sample points in the

training data *and* perfect knowledge of the true model structure, posterior distributions of parameter values are still influenced by the choice of priors.

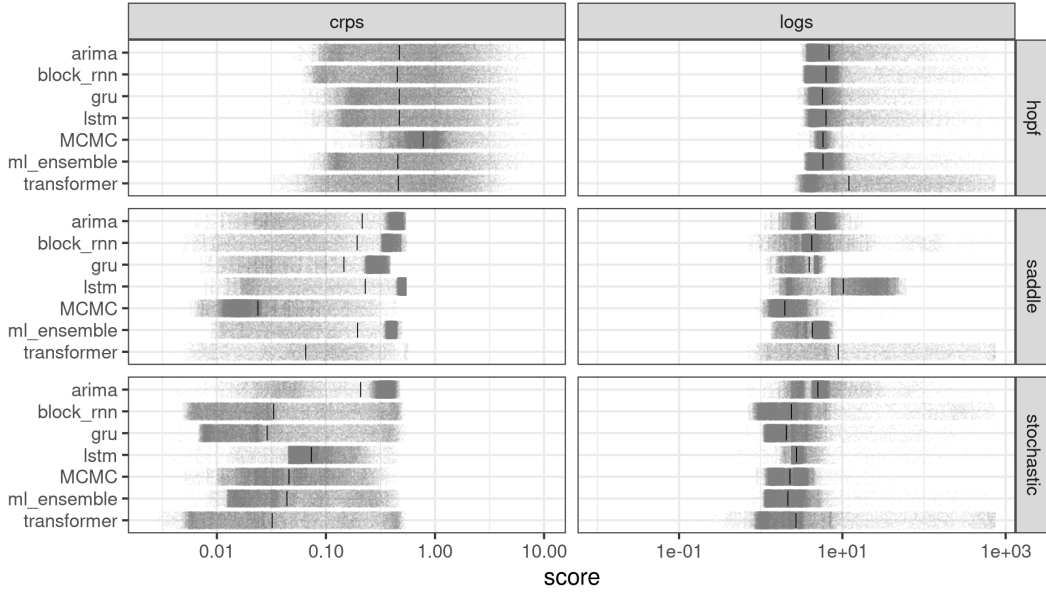


Figure 3.2: Overall distribution of skill scores across models, including an ensemble of methods. Smaller scores are better (indicating smaller errors). Black bars indicate means. The points indicate all individual predictions over time and replicate 'true' simulations of the given scenario.

Overall forecasting skill scores for each model across all three scenarios are summarized in Fig 3.2. Average scores (black lines) hide wide variation in forecast skill. Generally, ML performance tends to be bracketed between MCMC (essentially the theoretical optimum), and the statistical ARIMA model, though sometimes performing worse than ARIMA or better than MCMC. Under scenarios with alternative stable states (saddle and stochastic), the distribution of scores is often bimodal for ML models, though not MCMC. The ML ensemble model often performs as well as the best ML model on average. Note that a wide prediction of uncertainty does not mean a wide range in the score skill – for instance the ensemble model which has the widest array of outcomes often has a relatively tight distribution of score, especially in logs skill. This reflects the relative contributions of accuracy and uncertainty as components in the forecasts. Most ML scores are comparable to MCMC skill except for the scenario of the saddle-node bifurcation, where all other models are much worse. To get a deeper understanding of these general patterns, we now turn to examine each of the forecast distributions themselves in comparison to the future ensemble produced by the true generative process model, Fig 3.3-5.

Forecasts of the Hopf bifurcation (Fig 3.3) are roughly comparable across the phenomenological models (ARIMA and machine learning models). All models are trained using 100 time points drawn from the period of time prior to the onset of the Hopf bifurcation, which leads to a stable limit cycle that gradually grows in magnitude. Most models predict a roughly constant mean with a spread roughly equal to that created by the stochastic oscillations around the stable node as seen in the training data prior to the bifurcation. Notably, the GRU model picks up the oscillatory nature of the dynamics, despite the fact that no true oscillations were yet present in the training data. However, like the other ML models, it fails to predict the growing amplitude of those oscillations. Having access to the true model structure, the MCMC model alone predicts the transition into a pattern of oscillations which grows over time, though it tends to overestimate the

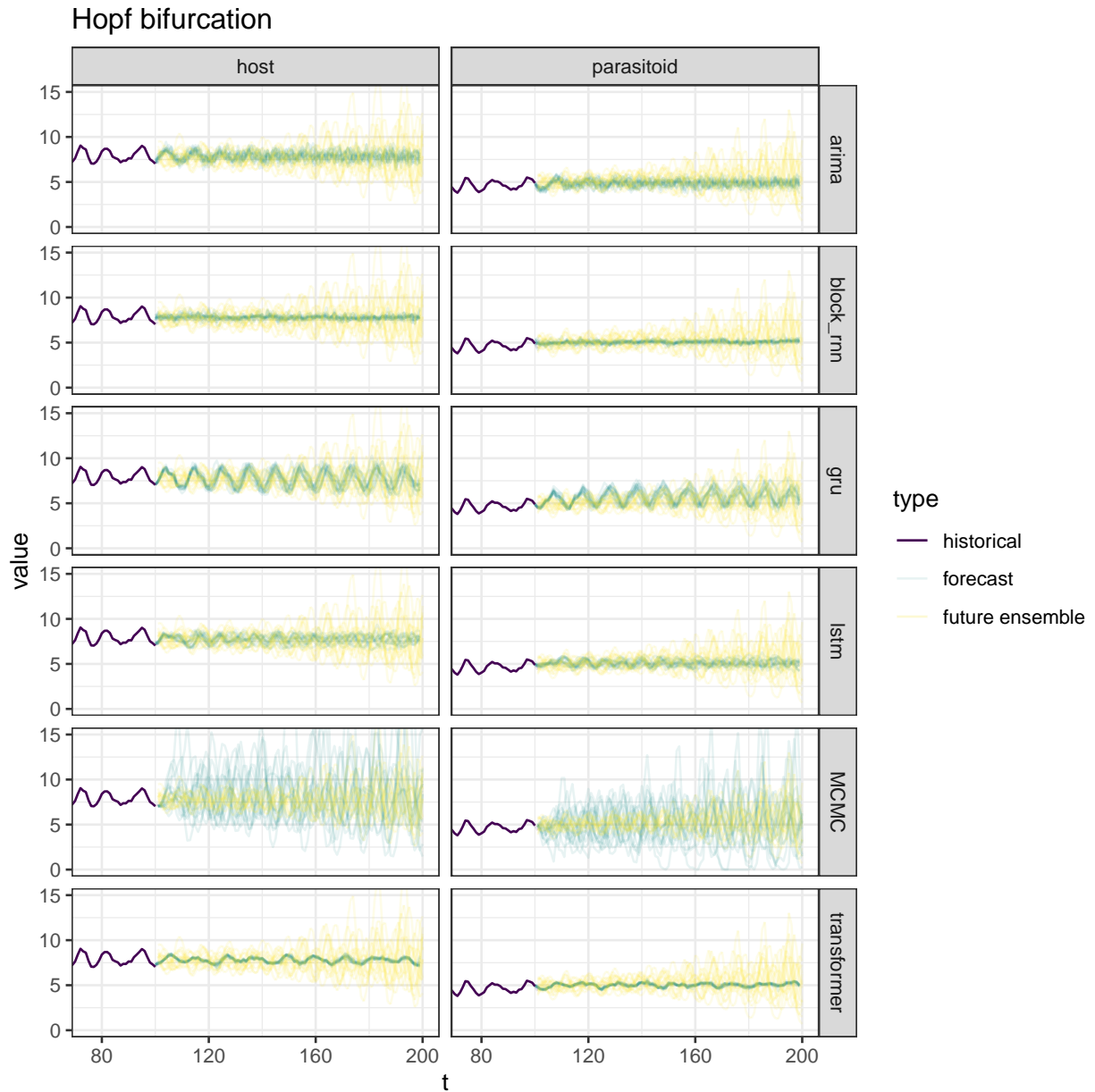


Figure 3.3: Forecasts of the Hopf bifurcation under each model, compared to 15 realizations of the true model. The bifurcation occurs soon after the forecasting period begins, leading to progressively larger oscillations. Prior to the bifurcation, pseudo-cycles are visible in the training data due to stochastic excitations. Following the bifurcation, stochasticity blurs the oscillatory pattern across replicate simulations. Only the last 25 time points of the training data are shown.

amplitude of those oscillations initially. Despite this, all methods score comparably in CRPS score (Fig 3.2) with most ML methods actually out-performing the MCMC score on average (Fig 3.5), albeit with much greater variation in individual scores. A clearer picture can be seen by looking at these skill scores over time (Fig 3.6-7), which show that MCMC is initially performing worse (over-predicting variance) but as oscillations grow further, it starts outperforming the more stationary forecasts of the ML models.

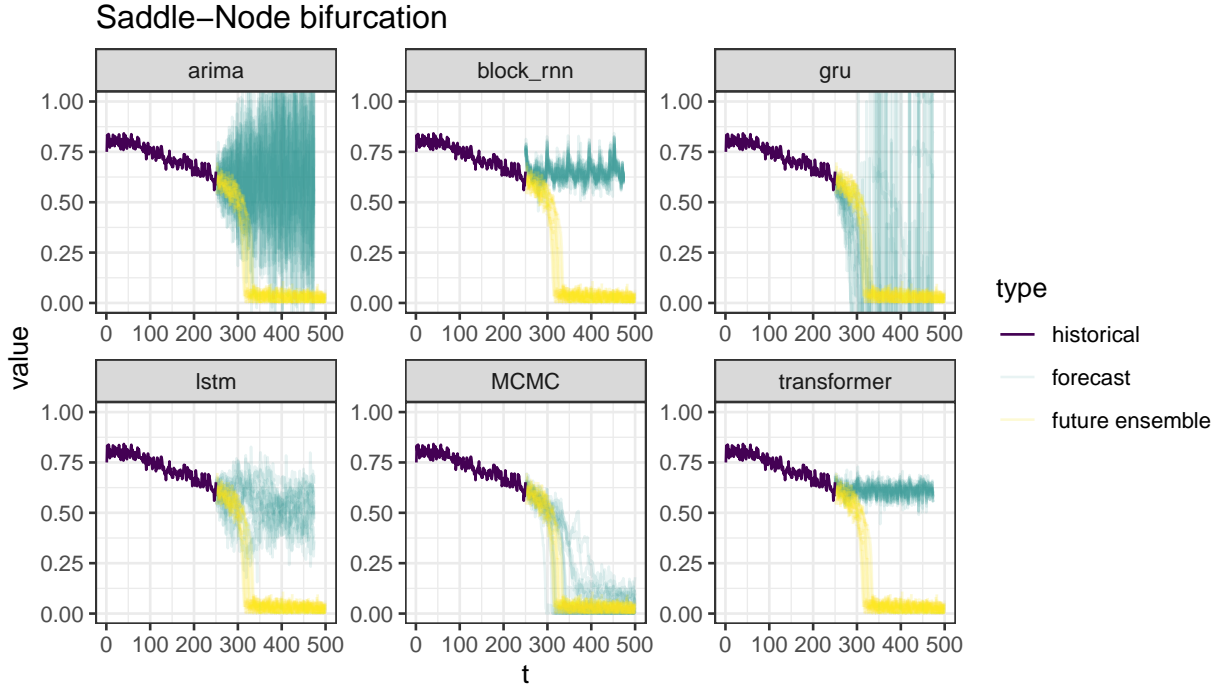


Figure 3.4: Forecasts of the saddle-node bifurcation under each model, compared to 15 realizations of the true model. Training data precedes the bifurcation, making accurate prediction without knowledge of the underlying model very difficult.

The saddle node bifurcation proves even more difficult for most methods (Fig 3.4). Only the MCMC model anticipates the sharp transition to an alternative state. Even accurate estimation of the MCMC requires slightly informative priors, though still broad enough to reflect a wide range of possible outcomes. Two ML methods – Block RNN’s and Transformers – resemble a naive prediction extrapolating the last observed state, failing to reflect the slow downward trend of the training data. LSTM’s indicate greater uncertainty, while GRU’s show very large variability which spans the alternative stable state range. With additional tuning, better performance may be possible for these ML models. The selected ARIMA model reflects wide uncertainty that is nevertheless not broad enough to span the alternative stable state. Consequentially, the MCMC estimate easily outperforms the ML models (Fig 3.2).

Machine learning methods do markedly better on the stochastic transition scenario than in the two bifurcation scenarios (Fig 3.5). This occurs because the training data includes the transition phenomenon of interest. All ML models accurately capture the dynamics of a sharp transition between alternative stable states – a dynamic the statistical ARIMA model entirely fails to reflect. Stochastic transition events should be approximately exponentially distributed, as seen in the wide range of waiting times for transitions to occur in replicates of the true ‘observed’ process (Fig 3.5). Transformer and Block RNN distribution times are much more concentrated, while again GRU and especially the LSTM do a better job reflecting the uncertainty in range of transition

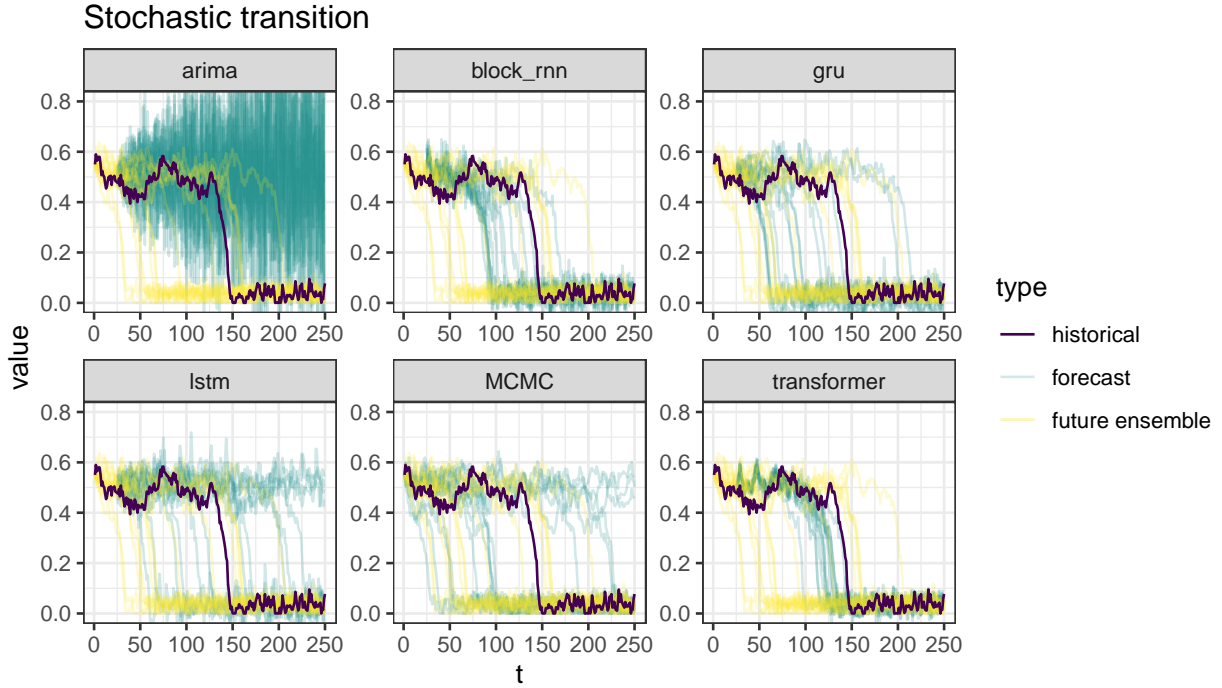


Figure 3.5: Forecasts of the stochastic transition under each model, compared to 15 realizations of the true model. In contrast to the other challenges, this case considers the prediction of replicate systems starting from the same initial condition, rather than forecasting the future evolution of the model after the stochastic transition has already occurred.

times.

Examining patterns in the scores over time (Fig 3.6-7) provides a more nuanced understanding of the forecast dynamics than aggregate scores alone (Fig 3.2). In the Hopf bifurcation, CRPS scores get worse over time across all methods, including the MCMC forecasts. In the saddle node bifurcation and stochastic transition, the same pattern holds somewhat more dramatically for non-MCMC forecasts, while MCMC scores are at their worst around the middle of the forecast horizon. Comparing CRPS scores to logs score also emphasizes the relative role of uncertainty: for instance, the MCMC scores for the Hopf bifurcation get steadily worse under CRPS but not under logs score. A relatively sharp transition can be seen under both MCMC scores on the Hopf bifurcation once the magnitude of the oscillations exceeds the variance created by mere stochasticity: the MCMC model no longer over-estimates the spread of the data, while the ML models now underestimate that variation. CRPS scores for stochastic transitions exhibit a distinct two-branch pattern, with scores for a given replicate being either very high (poor skill) or very low, reflecting whether the individual ‘true’ replicate matches the mean state predicted by the forecast. Logs skill score may be a better measure in this context, where correctly capturing the uncertainty in the forecast means that this bi-modal structure in scores can be avoided entirely, e.g. by the MCMC predictions. The forecast-skill-over time plots illustrate different reasons for the bi-modal distribution in skill seen for the saddle-node and stochastic transition scenarios in Fig 3.2 respectively: in the case of the saddle node, the two modes are distinguished by time-horizon; short term forecasts are relatively accurate, and longer term forecasts (i.e. after the catastrophic transition) are poor. In the case of the stochastic transition model, the two modes are not structured by horizon but by replicate, with some replicates having transitioned and others still in the original state.

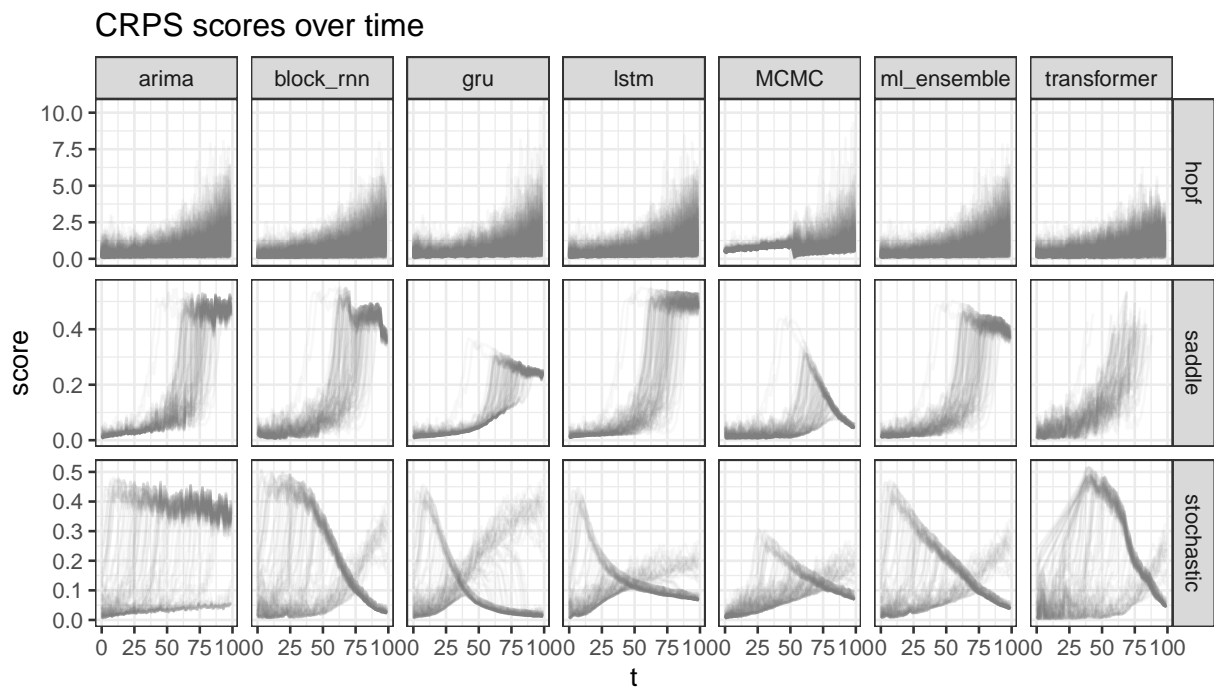


Figure 3.6: CRPS scores over time for each scenario. Each line represents the CRPS scores against a replicate of time series observations from the ‘true model’. The general pattern across these plots is that forecast skill gets worse over time – gradually in the case of a Hopf bifurcation or suddenly in response to the saddle-node bifurcation. In the stochastic transition case, the scores tend to diverge in two branches, where high values indicate periods of time when the forecast predicts the wrong equilibrium state and the lower branch indicates predictions of the correct one. The time axis in this plot and in Figure 3.7 refers to the time from the beginning of the forecast horizon, not the time from the beginning of the time series as in other plots.

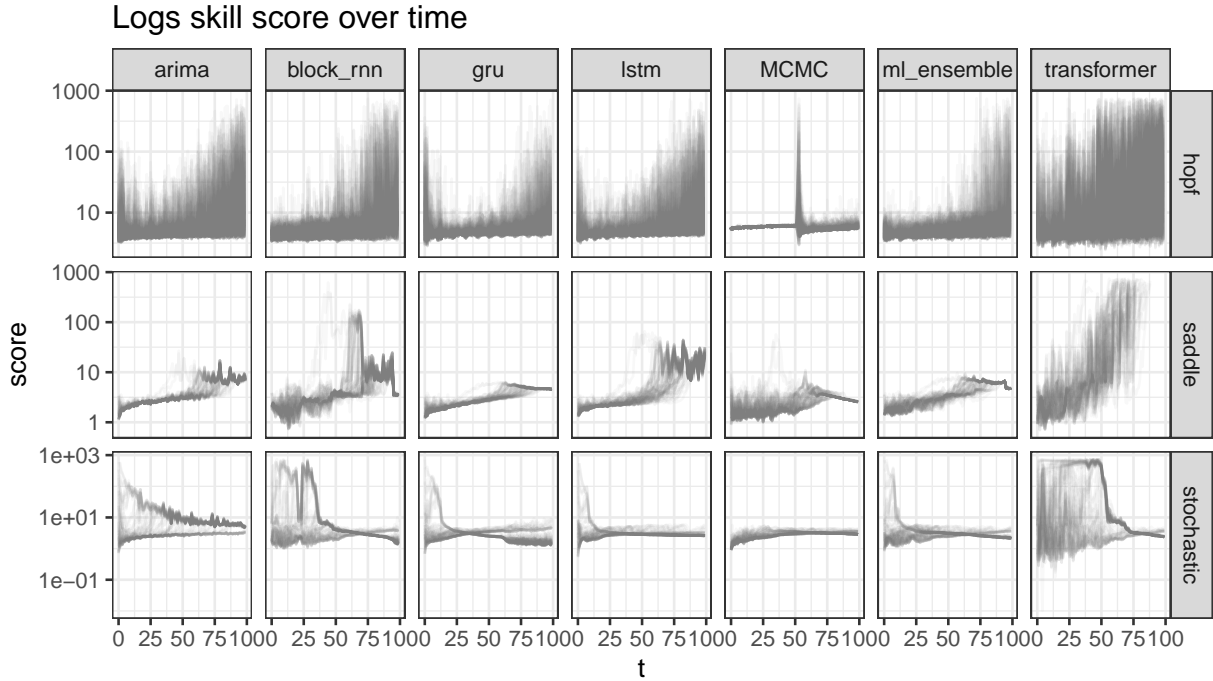


Figure 3.7: Logs skill score over time. Forecasts which underestimate uncertainty do substantially worse in logs score than in CRPS score. Comparing this panel to those in Figure 3.6 highlights scenarios that most often underestimate uncertainty. Generally, MCMC performs better relative to other models under this metric than it does under CRPS, reflecting the bias-variance tradeoff taken when using biased estimators in machine learning. The time axis in this plot and in Figure 3.6 refers to the time from the beginning of the forecast horizon, not the time from the beginning of the time series as in other plots.

3.4 Discussion

Ecological systems have long been acknowledged as complex, due not only to the immense span of dimension and scale such processes involve, but also the frequency of emergent and non-linear phenomena such as stochastic resonance, including bifurcations, tipping points, and hysteresis examined here. Calls for increased forecasting efforts from ecologists frequently reference the role of changing climate and other anthropogenic change, which raise the challenge of prediction in no-analog environments, anticipating ecosystem responses to conditions that have not been previously observed (J. S. Clark et al. 2001; M. C. Dietze et al. 2018). This motivates the question, “What methods will be most reliable in the face of unobserved conditions?”

In this chapter, we carry out an initial exploration on how deep learning methods can perform on predicting critical transition events. We compare the ability of several cutting edge machine learning approaches against statistical and process-based models, and show that deep learning methods are generally able to strike a middle ground between what we consider as acceptable and ideal case forecasting methods, ARIMA and MCMC-based parameter estimation respectively. Although most ML-based forecasting applications focus on point predictions, we have emphasized examples that can provide estimates of uncertainty. When the ML models are able to observe transition phenomena, as in the stochastic case, they performed comparably to MCMC-based forecasting with respect to CRPS and log probability score but under-performed MCMC when there were no transition events in the training sets as in the Hopf and saddle-node examples. An ensemble forecast combining the predictions of all four ML methods generally scores as well or better than any one of the ML methods alone. Yet, examining summary

statistics, CRPS and log probability scores obscures finer detailed components of the forecasts. For instance, forecast skill varies with the length of the forecast horizon in a non-monotonic fashion. This is the result of multiple factors: for some dynamics, such as those involving tipping points, the long term behavior can be easier to predict than transient transitions. Both predicted uncertainty and forecast skill can be better on longer horizons than on shorter ones, as in the MCMC predictions of tipping point dynamics. It is also important to remember that probabilistic forecast skill scores do not only measure how close observations are to expectation, but also reflect the predicted uncertainty: therefore, over-confidence about predictive accuracy can result in worse scores than scores from forecasts that are less accurate on average but correctly reflect a greater degree of uncertainty. The ability to better reflect uncertainty rather than better average predictions explains much of the performance of the ensemble model.

The success of these ML models on the stochastic transition case is particularly notable. All methods are given only a single previous replicate of a stochastic transition (Fig 3.5) on which to base their estimates. This is typical of ecological scenarios where data is so often limited. While even one observation of a transition is more than the methods have in our other forecasts, this still presents a significant challenge to model estimation. Unlike the MCMC case, the ML models have no prior expectation of a model structure that contains sharp transitions – we might have expected these models to perform little differently than the ARIMA model. Given this single replicate, all four ML models successfully capture the phenomenological pattern of a sharp shift between two stable states – this is behavior that the structurally simpler family of ARIMA models cannot express. This provides a clear illustration of the much broader array of phenomenological behaviors that can be accurately modeled with ML models compared to classical statistical models. In this way, the ML models can be seen as imposing even fewer assumptions on the phenomenological behavior of the system than the ARIMA model. In contrast, the MCMC performance benefits from very strong process-based assumptions, which happen to match the ‘true’ model in this case and thus provide a comparison of the theoretical optimal performance.

The MCMC case illustrates some of the hard limits to ecological forecasting of critical transitions. Our MCMC forecast assumes that the data-generating process is known, so the forecaster need only infer the posterior distribution of model parameters. This is a much stronger assumption than that made by the ML models, though this assumption can potentially be justified on the basis of a mechanistic understanding of the processes involved. It is important not to confound the MCMC example here with the use of MCMC in process based models of real systems. In the real world, this is never the case: all models are at best approximations of the underlying processes (Oreskes, Shrader-Frechette, and Belitz 1994). Despite this advantage, even the MCMC forecasts differ from the distribution of the true process. Because the available data come from only a small region of the dynamical state space, they are consistent with many possible parameterizations of the same model structure – which creates likelihood ridges and non-identifiability of specific parameter values. Using more simplified versions of the dynamical processes in question, such as the canonical form of a bifurcation, can mitigate this issue in some cases. Even when such non-identifiability issues cannot be avoided entirely, they can usually be diagnosed by examining the degree of mixing in MCMC sampling and comparing posterior to prior distributions.

When examining the performance of the ML models, it is clear that there is no single method that excels in all scenarios. Neither is there one class of ML methods that outperforms the others – a fact we found surprising given the reported dominance of

encoder-decoders in the field of sequence-to-sequence deep learning (Aitken et al. 2021). These observations underscore the point that ML is a very empirically-driven field in which there are few guarantees on performance. Furthermore, due to the black-box-ness of deep learning and other reasons like instability to initialization seed, it is often impossible to provide an explanation for why certain methods over-perform or fail to meet expectations.

Overall, ML models and the more traditional ARIMA model fail to predict the qualitative shift in dynamic behavior that occurs in the critical transition scenarios (Hopf and saddle node). This is not surprising, as the training data provide no prior example of such behavior (e.g. growing oscillations or a sudden shift). Nevertheless, this should be an important reminder of a central difficulty in ecological forecasting. Note that in such scenarios, near-term forecasts (M. C. Dietze et al. 2018) may be very accurate right up to the transition event before becoming widely wrong. Nor can the possibility of such non-linear behavior be easily dismissed in ecological models – the examples considered here have been bedrock of ecological modeling and management practices for over half a century (Folke et al. 2004), and if anything are only too simple, representing a small slice of possible dynamical behavior of more complicated models.

It may be natural to ask whether this performance would be remedied if the ML models were trained on data which includes prior examples of supercritical Hopf or saddle node bifurcations. This question is not as easy to answer as it may seem, because of the difficulty in defining the corresponding forecasting scenario. The scenarios we have considered are true, pure forecasts: the training data comes from a single realization of a specific generative process, and the task is to predict the future states of that system before they occur. Would it be possible to train a predictive algorithm on ‘analogous’ examples of critical transitions? For instance, could data from other lakes, which may have experienced a critical transition such as an eutrophication event in the past, be used to train machine learning models to predict such events in some focal lake in the future (Scheffer et al. 2001a)? Perhaps, but it depends on what we mean by an ‘analogous’ system. Even if the underlying mechanism was accurately captured by the same model, say, the saddle-node model of Robert M. May (1977) we consider here, it is likely that most of the individual model parameters would be quite different, even after accounting for re-scaling or non-dimensionalization of the model (Hastings 1996). Rarely do ecologists have access to completely controlled replicates for fitting or training models. The ability for ML models to successfully generalize from training in such cases remains an open problem and a promising subject of further investigation.

There are a number of questions that we have left unanswered that we hope will be addressed in future work. In this chapter, we have explored a small number of machine learning and statistical models that can be used for forecasting, so comprehensive conclusions can not be drawn on whether statistical or machine learning-based approaches are better suited for critical transition forecasting problems. Neither can we claim that ML methods will translate well to all sudden transition event forecasting problems in reality, since working with real data will introduce additional difficulties like how to deal with missing data, sparse data and observation errors.

Furthermore, our analysis has focused on the task of making a single forecast prior to the occurrence of a critical transition. Forecasting is ideally a more iterative process of data assimilation, where forecasts are updated with respect to additional observations, rather than projecting 100s of time steps into the future (M. C. Dietze et al. 2018). Updating a forecast after a critical transition has already occurred may be of little use in

the context of hysteresis, such as under the saddle node or stochastic transition – recognizing the alternative stable state only after the system is stuck in that basin will often be considered ‘too late’. Assimilation may be more applicable to the Hopf bifurcation, where additional observations of slowly growing oscillations may lead to more accurate forecasts. Such models may even accurately predict the homoclinic bifurcation that occurs when the limit cycle grows too large, eventually hitting a saddle point of zero population size for the host species. We leave these cases to future exploration rather than attempting to explore all such variations in a single narrative.

Ecological forecasting is invariably difficult, even in the idealized cases of ample measurement data and clearly identified structural models. This chapter is not intended to give a complete answer to whether deep learning is the best suited method for tipping point forecasting problems as this will take numerous studies to resolve; instead, this chapter aims to be an early exploration on whether deep learning methods should be considered as viable tools for this extremely challenging class of prediction problems. Given the difficulty of forecasting never-before-observed behavior, as illustrated by the Hopf and saddle-node bifurcation scenarios, there is good reason for research to focus more on the kind of qualitative predictions long emphasized in the literature on early warning signals and resilience (Scheffer et al. 2012). Recently, ML techniques developed for classification rather than the ML methods used in regression and forecasting models considered here have demonstrated a more nuanced ability to reliably detect different classes of critical transitions in time-series data (Bury et al. 2021; Lapeyrolerie and Boettiger 2021). Rather than seeking to provide managers with quantitative, probabilistic forecasts reflecting a broad uncertainty in possible outcomes, this literature has sought to emphasize only a more qualitative form of prediction, such as establishing whether a system is either “resilient” or “approaching a critical transition.” Decision sciences have long emphasized the importance of reconciling the qualitative predictions of resilience thinking with quantitative forecasts of future states (Fischer et al. 2009; Polasky et al. 2011b). Such approaches could be valuable in concert with probabilistic forecasts considered here, providing a possible mechanism to identify when the probabilistic forecast is least reliable.

Chapter 4

A Comparison of Neural Network Models for Water Quality Forecasting

This chapter will be submitted with Carl Boettiger as a co-author. It is included here with permission from the co-author.

Chapter 4 continues the exploration of neural network models for ecological time series forecasting. In Chapter 3, I posed a challenging prediction problem, where I tested how well neural networks could forecast novel tipping point dynamics in a limited data regime. In Chapter 4, I evaluate how neural networks perform on a water quality data, where there is much more information provided to support inference-making. These two chapters provide examination on the performance of neural networks on ecological time series forecasting problems that are characterized by limited and large data regimes.

4.1 Introduction

The ability to accurately forecast water quality variables has become increasingly important in the era of global change. Freshwater ecosystems have been disproportionately impacted by anthropogenic activities, a trend that is expected to continue throughout the 21st century (Albert et al. 2021). Since water quality variables influence many biological and physical processes in freshwater ecosystems, preemptive water quality forecasts could allow managers to evade situations with far-reaching negative consequences (Ouellet-Proulx, St-Hilaire, and Boucher 2017; Stajkowski et al. 2020; Chen et al. 2024). This chapter gives particular consideration to the variables of dissolved oxygen, water temperature and chlorophyll-a (chl_a), which are key indicators of the health of freshwater ecosystems: water temperature influences how fast aquatic organisms can grow and where they can be found (Caissie 2006); if dissolved oxygen levels become too low (a condition called hypoxia), there can be massive mortalities of fishes and marine mammals (Pollock, Clarke, and Dubé 2007); and, chl_a is an indicator for the amount of algae present in a body of water, which makes it critical to monitor as algae blooms can produce toxins and cause hypoxia (Catherine et al. 2013). Limnologists have historically used statistical, process-based and machine learning models to forecast water quality metrics (Maier and Dandy 2000); but, in recent years, researchers have shown that machine learning models are particularly well suited to take advantage of recent advancements in computer science and the rising availability of water quality data (Hanson et al. 2020; Zwart et al. 2023).

A shortcoming of past limnological forecasting studies that support the use of machine learning is that they tend to focus on a narrow selection of sites that are not representative of freshwater systems across a broad geographic scale. This chapter has the primary aim of making a comprehensive comparison of state-of-the-art machine learning methods by evaluating forecasts at 34 different sites across North America at different times of the year.

The time series that are used in this chapter come from the National Ecological Observatory Network’s (NEON) Ecological Forecasting Challenge, a competition where teams can submit forecasts for data that is collected and made publicly accessible by NEON (Thomas, Boettiger, et al. 2023). A common finding across the challenge is that a day of year historical mean model (also referred to as the climatology model) commonly produced top scoring forecasts (Wheeler et al. 2024; Thomas, McClure, et al. 2023). For instance, in a model comparison that examined the forecasts for phenology, Wheeler et al. (2024) found that the climatology model outperformed all except one of the submitted models; and, the best performing model only marginally outperformed the climatology model. Thus, a primary consideration in this chapter is comparing the performance of the machine learning models against the historical null model.

There are many promising neural network architectures that have not yet been evaluated for water quality forecasting. Most applications of neural networks to limnological time series have focused on the long-short-term-memory (LSTM) network, a model that was published in 1997 (Hochreiter and Schmidhuber 1997). Over the nearly 30 years since LSTM’s were introduced, researchers in computer science have built upon these earlier neural network architectures, introducing new models that achieve state of the art performance (Lim et al. 2019; Bai, Kolter, and Koltun 2018; Oreshkin et al. 2019). Many recent studies that use neural networks for water quality forecasting have not explored more contemporary neural network architectures. We address this gap in the literature by comparing 8 neural network models including LSTM’s as well as more recently developed neural network models.

4.2 Materials and Methods

We evaluated the forecasting performance of 12 different forecasting models on water temperature, dissolved oxygen and chl_a time series recorded by in-situ sensors at 34 freshwater sites across North America. In Figure 4.1, we display where these sites are located in the United States and Puerto Rico. These sites consist of Lakes, Non-wadeable Rivers and Wadeable Streams, subtypes classified by NEON. There is variability across sites in which target variables were observed. Additionally, maintenance issues led to gaps in the data which also varied across locations. Provided the lack of time series and large gaps at certain sites, we evaluated the forecast performance for water temperature, dissolved oxygen and chl_a at 33, 32 and 9 sites respectively.

The imputation of missing data proved to be critical to the performance of the ML models. After experimenting with data filling methods that resulted in poor model performance, we developed an imputation method inspired by the climatology model. If the gap size was less than 5 days, then the gap was filled using a Gaussian Process Filter. For gaps over 5 days, missing data was estimated using the daily historical mean when this statistic is available; and, when there were no data collected for a day of the year, either the monthly median, quarterly median, previous observation or global median was used in this order of preference. The intuition behind this method for missing data imputation is

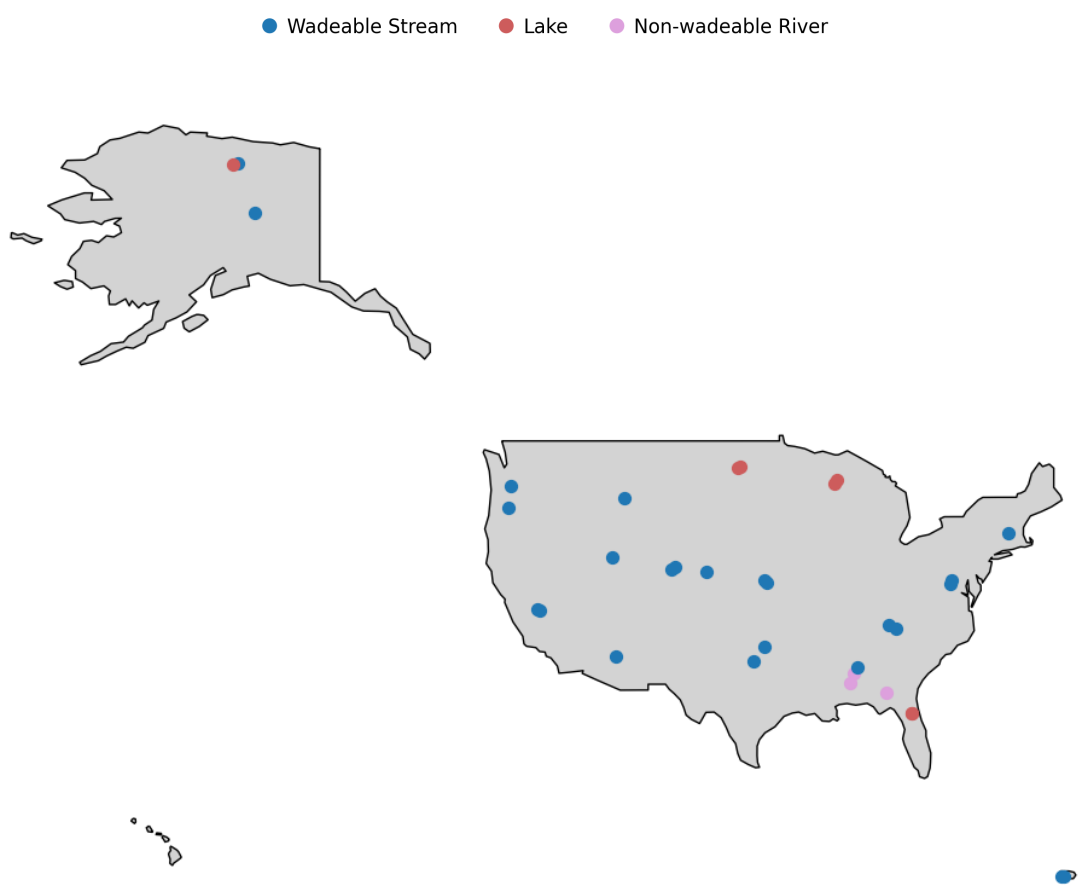


Figure 4.1: Map of site locations across the United States and Puerto Rico.

that this method biases the neural networks towards the climatology model. Since it has been established that a climatology model produces top performing forecasts throughout the NEON forecasting challenge, backfilling with the climatology model seems likely to induce improved accuracy for the neural network models (Wheeler et al. 2024; Thomas, McClure, et al. 2023).

We compared the performance of the neural network models to the climatology and naive persistence model. The climatology model generates forecasts by finding the daily mean and standard deviation and draws samples from a Gaussian distribution with these parameters. The naive persistence model finds the last observed value from the target time series and predicts this value for each day in the forecast horizon. For each model, we computed the Continuous Ranked Probability Score (CRPS) and the Root Mean Square Error (RMSE). To gauge how models performed in relation to the climatology model, we computed the Continuous Ranked Probability Skill Score (CRPSS) which we defined as

$$CRPSS_{model} = 1 - \frac{CRPS_{model}}{CRPS_{clim}}. \quad (4.1)$$

Similarly, using the naive persistence model as a reference, we computed the RMSE Skill Score,

$$RMSE - SS_{model} = 1 - \frac{RMSE_{model}}{RMSE_{naive}}. \quad (4.2)$$

If a target model outperforms the reference model, then the target model will have a positive skill score. If the target model performs worse than the reference model, the skill score will be negative.

4.2.1 Method Group 1: Statistical Models (Theta)

To compare the performance of neural network models with a representative statistical model, we evaluated forecasts generated by the Theta model. Provided a seasonally adjusted univariate time series, the Theta model creates forecasts by modifying the second differences of the data (Assimakopoulos and Nikolopoulos 2000). The magnitude of local curvature modifications is given by the Theta coefficient,

$$\nabla^2 Z_t(\theta) = \theta \nabla^2 X_t, \quad (4.3)$$

where ∇ is the difference operator, Z_t is defined as a theta line and X_t is the original data. When $\theta < 1$, the second differences are reduced from the data, yielding a Theta line that amplifies the long term trends of the data. For instance, if $\theta = 0$, the theta line will be a linear regression of the data. And if $\theta > 1$, the short term behavior of the data will be magnified in the Theta line. The Theta model generates a forecast by extrapolating the linear combination of two or more Theta lines. Although the Theta model is relatively simple, it has performed remarkably well in prominent forecasting competitions that include hard to predict data (Makridakis and Hibon 2000; Makridakis, Spiliotis, and Assimakopoulos 2020). Since the Theta model was originally presented in 2000, new variations of the Theta model have been developed that outperform the original Theta model (Fiorucci et al. 2016). Throughout this study, we use the StatsForecast AutoTheta model which selects the best performing model from a range of Theta model variants.

4.2.2 Method Group 2: Neural Network Models

All the machine learning algorithms that are used in this chapter are based on neural network architectures. Neural networks have the property of being universal function approximators, so it is theoretically possible that all of these models could exactly approximate the data generating process (Hanin 2019). Yet, it is often the case that neural networks greatly underperform their function approximation capabilities in practice (Adcock and Dexter 2020). This performance gap can be due to a variety of reasons including overfitting and insufficient hyperparameter tuning (Adcock and Dexter 2020). The discrepancy between theory and practice in function approximation with neural networks motivates research on how machine learning models perform in specific domains.

The 8 machine learning models that we compare take a variety of approaches with the design of their neural networks. We will not go into detail on how these different models work, but for those interested to learn more, we list the models and their references in Table 4.1. An important concept to understand is that the neural network models considered in this study learn directly from the data and are not instilled with domain knowledge. This non-mechanistic basis is at once very powerful as it does not restrict the models with misleading assumptions, but neural networks are also limiting in that they are not readily interpretable and often require more data than knowledge-guided methods to perform well (Karpatne, Jia, and Kumar 2024; Read et al. 2019). For the management of critical resources like water, the lack of interpretability could be a deterrent to the adoption of NN methods, and there may not be enough data for some water systems to accommodate ML approaches (Zhi et al. 2024).

The neural network models are configured in this study to provide estimates for predictive uncertainty. The neural network architectures that we use are deterministic, so they are not able to produce probabilistic forecasts intrinsically. We work around this limitation by performing quantile regression whereby the neural networks are trained to output quantiles at each time step in the forecast window. Probabilistic forecasts are then generated by drawing samples according to these quantiles.

For the 8 neural network models that we investigate, there are varying design choices made regarding the type of covariates that can be used. For this study, we employed 2 groups of models: one group used past covariates and the other used future covariates. For the models that accept past covariates (which included TCN, BlockRNN, NLinear, DLinear, Transformer and NBEATS), we used the other target variables recorded at that site as well as air temperature as covariates. For the models that only accept future covariates (which included TFT and RNN), we used the day of the year as the sole covariate. All the time series were split into training and validation sets, whereby the models were trained on time series from 2020 to 2023 and validated at 12 30-day non-overlapping intervals in 2023.

We provide fully reproducible code for fitting, scoring and visualizing the forecasts. All the machine learning forecasts were generated using the `darts` python library (Herzen et al. 2021). `darts` is similar to other libraries like `scikit-learn` in Python or `tidymodels` in R in that the library allows users to employ a variety of time series forecasting models without having to implement them. The github repo for this study can be found at <https://github.com/boettiger-lab/neon4cast-darts-ml>.

Abbreviation	Algorithm Name
RNN	Recurrent Neural Network (Hochreiter and Schmidhuber 1997)
BlockRNN	Block Recurrent Neural Network (Du et al. 2020)
TCN	Temporal Convolutional Network (Bai, Kolter, and Koltun 2018)
NLinear	NLinear (Zeng et al. 2022)
DLinear	DLinear (Zeng et al. 2022)
TFT	Temporal Fusion (Lim et al. 2019)
Transformer	Transformer (Vaswani et al. 2017)

Table 4.1: The neural network-based time series forecasting models used in this chapter.

4.3 Results

We examine the forecast skill of 8 neural network models (RNN, BlockRNN, NBEATS, NLinear, DLinear, TFT, TCN and Transformer), 1 statistical model (AutoTheta) and 2 null models (naive persistence and climatology) on time series taken from the NEON Forecasting Challenge’s Aquatics Theme. Following recent work that has established that multi-model ensembles offer advantages over individual models for water temperature forecasting (Olsson et al. 2024), we were inspired to investigate multi-model ensembles in this comparative study. So, in addition to the ML, empirical and null models, we created an ensemble model that aggregates the forecasts taken from all the neural network models. For this neural network ensemble, all the NN models are represented equally, hence its name “Naive Ensemble”.

In Table 4.2, we present the mean CRPS and RMSE scores for the respective target variables. For dissolved oxygen (DO) and water temperature (WT), the models perform similarly: the neural network models outperform the AutoTheta, naive persistence and climatology model with few exceptions; and the naive ensemble model is the best performing model with respect to both CRPS and RMSE. Yet, with chla, there are different patterns in model performance: while the neural network models still generally outperform the reference models in CRPS, the Naive Ensemble model is no longer the top performing model in either CRPS or RMSE. Instead, BlockRNN and the naive persistence model attain the best performance with respect to CRPS and RMSE, respectively.

By examining some of the individual forecasts, as shown in Figure 4.6, it is possible to gain intuition for why the neural network models perform well across the target variables. The AutoTheta model produces forecasts that resemble linear regressions based upon recently observed values. These relatively simple forecasts perform well for many evaluation intervals, but there are a few cases when AutoTheta fails catastrophically, negatively impacting the model’s overall performance. For instance, after a peak of dissolved oxygen in the winter of 2024, the AutoTheta model forecasts that DO will continue to increase which is opposed to the trend that DO peaks in the winter and declines through the spring. Similarly with chla, the AutoTheta model wrongly extrapolates that a spike in chla will lead to higher chla concentrations instead of an immediate reversion to the non-bloom state.

Conversely, the neural network models are able to learn from historical trends. For instance, at the beginning of 2024, the neural network models have learned from the training data that DO peaks in the winter and declines through the spring. With chla, however, we see that the neural network models fail to predict any of the spikes in concentration which originate from algae blooms; instead, the neural networks take a conservative approach, only predicting that the chla concentration will remain at non-

bloom levels throughout the year.

These general patterns in model performance can be observed in the skill score plots displayed in Figures 4.2, 4.3, 4.4. The AutoTheta model tends to have longer tails, indicating that the AutoTheta forecasts are more likely to perform very well or very badly. Meanwhile, the neural network models generally have fewer forecasts that underperform relative to AutoTheta as well as fewer forecasts that are outlier outperformers. The distributions of NN skill scores are more skewed towards outperforming than AutoTheta’s skill scores are. In these plots, we display scores according to water body type, but we did not see any significant differences in performance across these categories. However, this may be due to an underreporting of scores for lakes and non-wadeable rivers relative to the number of scores found for wadeable streams.

When examining the performance of the models within a forecast horizon, additional nuances emerge, confusing the perspective that neural network models are the best performing model class. In Figure 4.5, the AutoTheta model is consistently the best performing model for short-time horizons ($t < 5$) across target variables, but by the end of the 30-day horizon, the AutoTheta model is the worst performing model universally. Meanwhile the neural network models underperform AutoTheta during the early stages of the forecast window, but their forecast skill declines less rapidly than the skill score of AutoTheta. So while, the neural networks generally outperform AutoTheta and the null models according to coarse scoring metrics like mean CRPS and RMSE, AutoTheta is the best performer in short-time horizons.

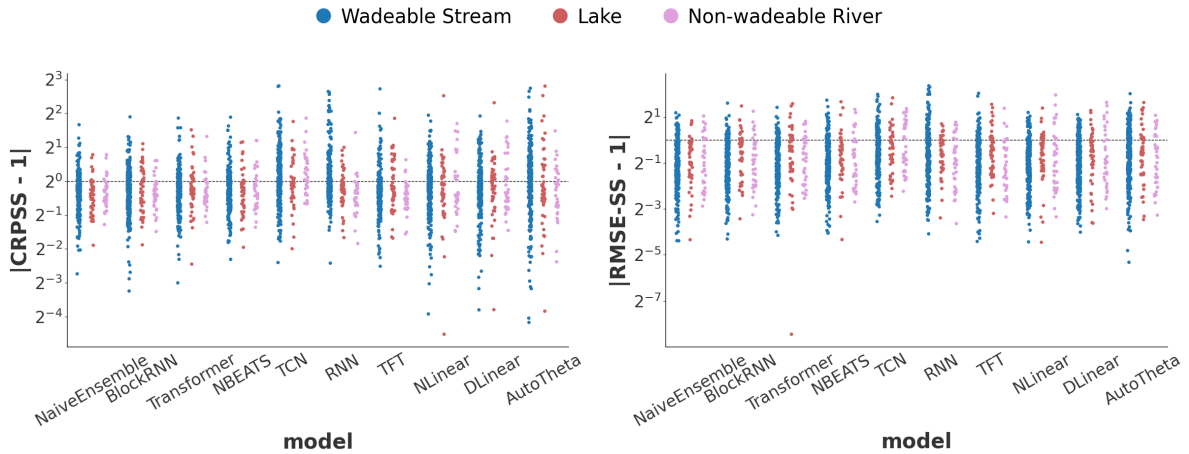


Figure 4.2: Skill score distributions for dissolved oxygen. The CRPS plot measures the CRPS skill relative to the climatology model. The RMSE-SS plot measures the RMSE skill relative to the naive persistence model. Each point is the skill score that has been aggregated over each 30-day forecast. A guideline is plotted at the threshold for underperformance: a score above this line denotes that a forecast is underperforming the reference model; a score below this line denotes that a forecast is outperforming the reference model. The AutoTheta model exhibits longer tails relative to most of the neural network models, indicating that AutoTheta has a larger amount of forecasts that do either very well or very poorly. The neural network models tend to have shorter tails in the underperforming region.

4.4 Discussion

In the hands of decision-makers in water resource management, models that can accurately forecast water quality variables would enable more proactive management strategies. For instance, water resource managers could use these forecasts to release water

(a) Dissolved Oxygen (mg L^{-1})		
Model	Mean CRPS	Mean RMSE
Climatology	0.62	0.92
Naive Persistence		1.55
AutoTheta	0.67	1.02
BlockRNN	0.52	0.80
DLinear	0.52	0.81
NBEATS	0.51	0.80
NLinear	0.52	0.80
NaiveEnsemble	0.47	0.74
RNN	0.60	0.92
TCN	0.62	0.98
TFT	0.52	0.80
Transformer	0.51	0.80

(b) Water Temperature ($^{\circ}\text{C}$)		
Model	Mean CRPS	Mean RMSE
Climatology	1.46	2.19
Naive Persistence		6.05
AutoTheta	1.55	2.35
BlockRNN	1.45	2.26
DLinear	1.35	2.08
NBEATS	1.36	2.10
NLinear	1.33	2.06
NaiveEnsemble	1.18	1.80
RNN	1.32	2.03
TCN	1.89	3.02
TFT	1.19	1.86
Transformer	1.32	2.05

(c) Chlorophyll-a (mg L^{-1})		
Model	Mean CRPS	Mean RMSE
Climatology	4.83	7.50
Naive Persistence		4.86
AutoTheta	4.20	6.31
BlockRNN	3.14	5.16
DLinear	3.47	5.48
NBEATS	3.78	5.95
NLinear	3.67	5.63
NaiveEnsemble	3.44	5.50
RNN	4.15	6.40
TCN	3.72	5.77
TFT	3.67	5.87
Transformer	4.02	6.35

Table 4.2: Mean CRPS and RMSE for dissolved oxygen, water temperature and chl_a forecasts. The neural network-based models generally outperform the statistical benchmark model, AutoTheta, as well as the null models, naive persistence and climatology, across the target variables. For dissolved oxygen and water temperature, the NN ensemble model is the best performing model with respect to CRPS and RMSE. With chl_a, BlockRNN attains the best CRPS score, and naive persistence attains the best RMSE.

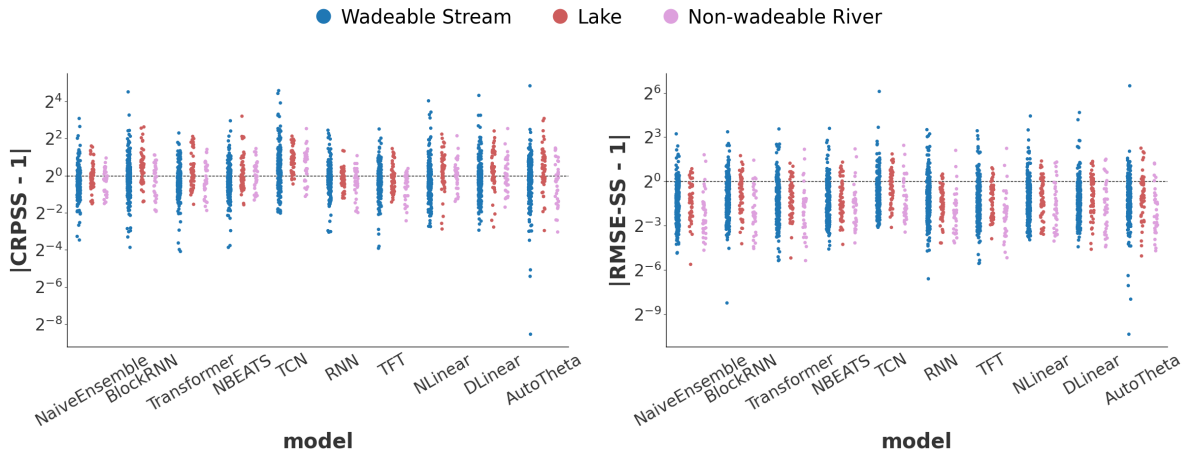


Figure 4.3: Skill score distributions for water temperature. A guideline is plotted at the threshold for underperformance: a score above this line denotes that a forecast is underperforming the reference model; a score below this line denotes that a forecast is outperforming the reference model. AutoTheta does not have the long tails as seen in the skill score distributions for dissolved oxygen and chla, instead the neural network models have heavier tails in the outperforming region. This indicates that the neural network models are producing high accuracy forecasts more frequently than the AutoTheta model.

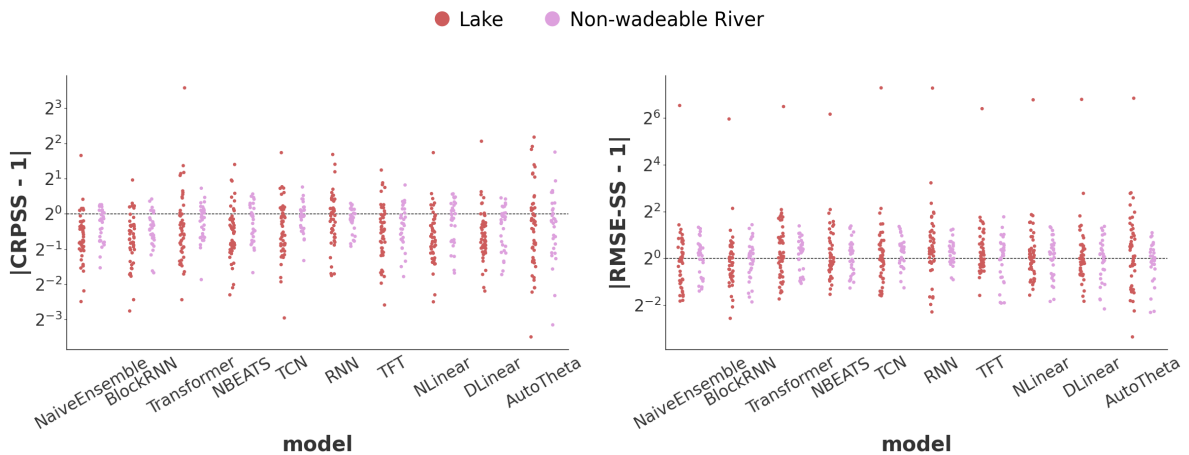


Figure 4.4: Skill score distributions for chla. A guideline is plotted at the threshold for underperformance: a score above this line denotes that a forecast is underperforming the reference model; a score below this line denotes that a forecast is outperforming the reference model. The neural network models have shorter tails in the underperforming region relative to AutoTheta, indicating that the neural network models are producing a smaller amount of inaccurate forecasts. The naive persistence null model attained the lowest RMSE on chla, and this is apparent in the RMSE-SS plot as the models have skill score distributions that are generally centered above the underperformance threshold.

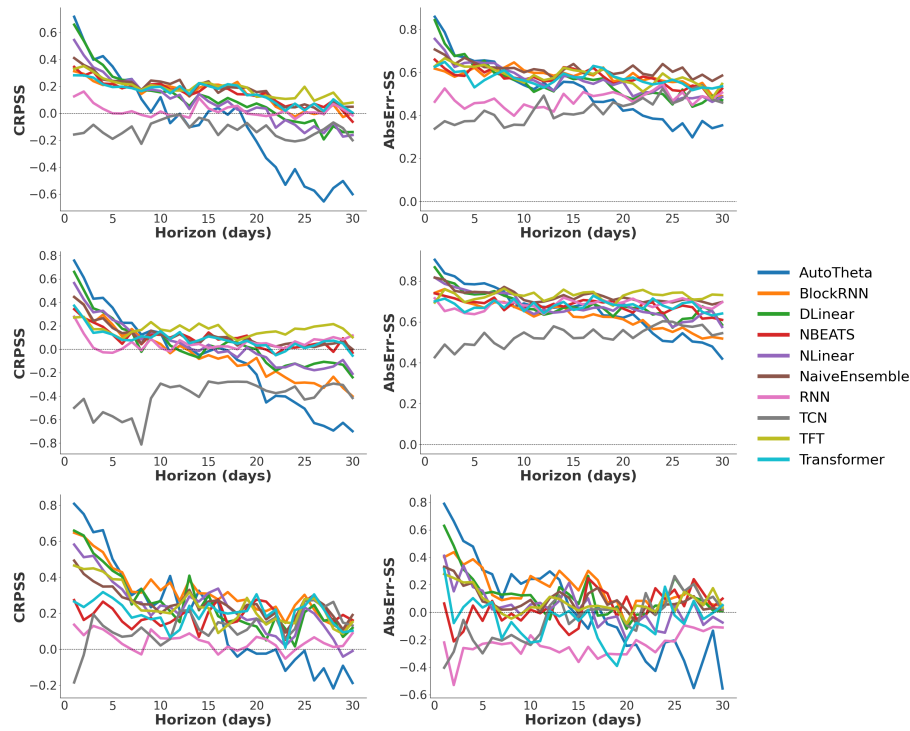


Figure 4.5: Mean skill scores within the 30-day forecast horizon for oxygen, water temperature and chl-a in descending order vertically. A guideline is plotted at the threshold for underperformance: contrary to the skill score distribution plots, a score above the guideline here denotes that a forecast is outperforming the reference model; a score below this line denotes that a forecast is underperforming the reference model. In the early phase of the forecast horizon ($t < 5$), the AutoTheta model is universally the top performing model. However, the performance of the AutoTheta model rapidly declines over the course of the 30-day horizon, concluding in AutoTheta being the worst performing model universally at the end of the horizon. The neural network models do not perform as well as AutoTheta at the beginning of the forecast horizon, but their performance declines less rapidly throughout the horizon.

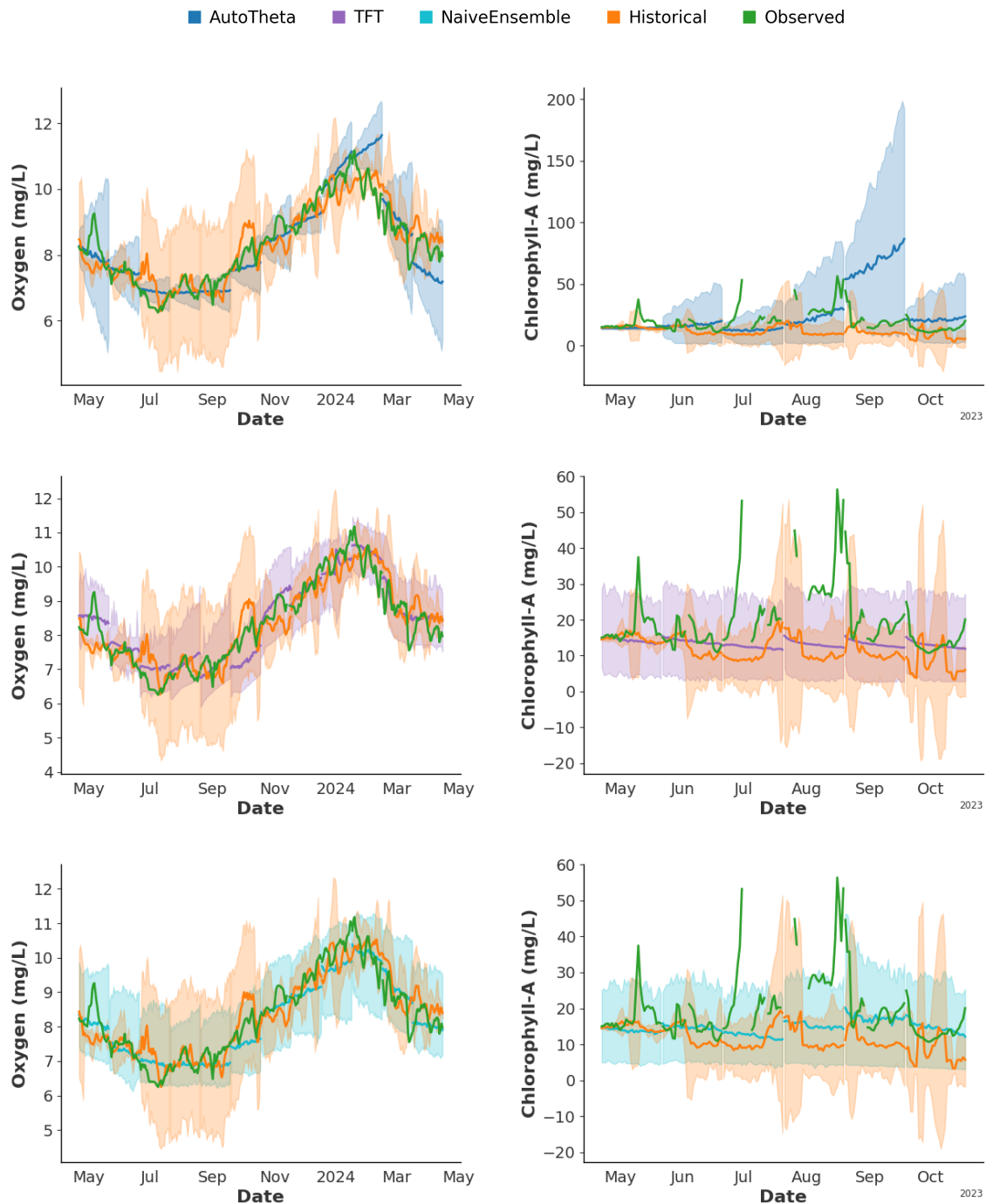


Figure 4.6: 30-day forecasts of dissolved oxygen and chla. These plots present forecasts from 12 30-day intervals spanning 2023 and 2024. The AutoTheta model produces accurate forecasts most of the year, especially for dissolved oxygen; but, there are some intervals where AutoTheta produces extremely inaccurate forecasts. After peaks in dissolved oxygen and chla, the AutoTheta model wrongly predicts that the values will continue to increase. Conversely, the neural network-based forecasts from the TFT and the Naive Ensemble model have learned from the training data that high levels of dissolved oxygen in the winter are followed by a steady decline throughout the spring. With chla, the neural networks take a conservative strategy where they predict that chla will remain in a non-bloom state and never take the risk of forecasting that a bloom will occur.

from reservoirs, enact controls to adjust nutrient levels and engage in other activities to combat events like hypoxia and toxic algae blooms that harm aquatic ecosystems. During the last 10 years, there has been a growing focus in limnology around the use of neural networks for water quality forecasting, but this focus has primarily centered on older neural network architectures with little exploration of newer methods. This chapter aims to address this research gap by performing a comprehensive comparison of neural network models that includes more recently developed models.

Instead of identifying that there is an individual model that performs exceptionally well at forecasting water quality, this chapter has affirmed the conventional wisdom that ensemble forecasts are often more accurate and robust than forecasts from individual models. We found that an ensemble model which aggregates the forecasts from all the neural network models was the best performing model overall, surpassing the accuracy of all the other models on the target variables of dissolved oxygen and water temperature, while also performing well for chl_a, attaining the second and third best CRPS and RMSE respectively. This result affirms recent work from Olsson et al. (2024) which established that multi-model ensembles offer significant advantages over individual models for water temperature forecasting.

Yet, as is often the case when judging whether one method is superior to another, we have found that subtle changes to our criteria may have produced radically different results. For instance, if we had used the same forecasts abbreviated to a 5 day horizon, then the AutoTheta model would have been the best performing model according to the scoring metrics used in this study. But, it is important to qualify that if we changed the model training configurations to consider short-time horizons ($t < 5$), this would produce forecasts that would be different than the ones we evaluated for the 30-day horizon. Thus, how neural networks models perform over different forecast horizons warrants additional exploration.

The neural network models struggled to forecast chl_a, a result that is not surprising given that forecasting algae blooms is well established as a challenging prediction problem in limnology (Chen et al. 2024). With chl_a, the neural network models displayed a tendency to take a conservative strategy, regularly predicting that the chl_a concentration would remain in a non-bloom state. It is possible that the neural network models settled on this conservative behavior because we did not provide enough information to predict blooms; the neural networks were not provided with some of the typical covariates used in process-based models like Nitrogen and Phosphorous concentrations, and photosynthetic active radiation. Yet, it is also plausible that this conservative strategy was driven by the length of the forecast horizon. Blooms are stochastic events, characterized by rapid fluctuations in chl_a. Predicting the timing of such an event in a long-time horizon ($t \sim 30$) will be more difficult than performing the same prediction on a shorter horizon as there are more opportunities in the long-time horizon to be wrong. With this reasoning, it is sensible that the neural networks would take a conservative approach.

This chapter has generated an abundance of research questions and ideas for future work. Outside of exploring how the performance of neural networks will vary with forecast horizon, other promising directions for exploration include cross-learning and hybrid models. Throughout this chapter, we trained individual models for each target time series. Cross-learning presents a different approach where one model would be trained on a collection of target time series. Recently, cross-learning has been shown to improved forecast performance as models that employ cross-learning are able to learn patterns across time series that transfer to making more accurate forecasts on an individual time series,

particularly when there is a limited amount of data points in the individual time series (Semenoglou et al. 2021). Another promising direction for future work would be to develop a model hybridized across modeling classes. Since we observed that the AutoTheta model performed well early in the forecast horizon and that neural networks performed well later in the forecast horizon, it is sensible that a hybrid ML-statistical model could achieve better performance than a purely statistical or machine learning model. For instance, a model that relies on exponential smoothing at the beginning of a forecast horizon then favors a ML model towards the end of forecast window could achieve better performance than the models explored in this chapter.

Neural networks offer some promising advantages over classical methods for limnological forecasting as neural networks can learn complex patterns from data without being restrained by a need for domain knowledge. As there are large water quality data sets that exist currently, and increasing amounts of limnological data will come online as sensor costs decline, neural networks present a way forward for the analysis of such data sets. Furthermore, the performance of neural networks has the potential to improve over time as neural networks often perform better with more data. Yet, neural networks also present a range of problems like a lack of interpretability and generalizability that could restrain them from being useful decision support tools for the safety critical problem of water resource management. Furthermore, in the era of global change, it is possible that there could be significant distribution shifts in the data that could lead to poorly performing neural network models as neural networks operate on the premise that future data will be similar to the historical data seen in training. This chapter does not attempt to provide a definitive answer as to whether neural networks will be the best method class moving forward as resolving such a question will require a much larger body of work than what is presented here. Instead, the hope is that this work will generate ideas and incite momentum towards improving our ability to forecast and manage water quality.

Chapter 5

Conclusion

This dissertation supports that neural network models can improve our ability to forecast ecological time series and helpfully inform conservation decision-making. In Chapter 2, I show that neural network-based RL agents approximate the optimal solution on a classic harvest selection problem and outperform a reasonable rule-of-thumb strategy on a non-stationary conservation management problem where the optimal solution is not known. In Chapters 3 and 4, I demonstrate that neural network-based time series forecasting models are able to outperform a selection of reference models on critical transition and water quality time series. Yet, along with their successes, neural network models have shortcomings. For instance, I found that neural network-based methods fail to outperform a simple null model when forecasting chl_a concentration in Chapter 4. To use models effectively, it is essential that we understand where they excel and fall short.

Every chapter in this dissertation possesses a clear path ahead for subsequent research. Chapter 2 presents a paradigm for how ecologists can approach decision-making problems in conservation. In this chapter, I examined some relatively simple reinforcement learning environments, but the methodology could readily be applied to environments with higher dimensionality and more complex dynamics. Recently, Equihua, Beckmann, and Seppelt (2024) have achieved this by extending RL framework from Chapter 2 to study connectivity conservation planning. Chapter 3 which centered on analyzing simulated time series could be extended to observed time series that display critical transitions. And, from Chapter 4, follow-up work could be done to improve the performance of the neural network models on the chl_a time series.

Ecology has transitioned from a discipline that was once data poor to one that is increasingly data rich. Neural networks have been used for various problems in ecological data analysis since the 1990's (Fielding 1999); but, due to recent advances in computational power and algorithmic development, neural networks are now particularly well suited to handle the large amount of ecological data that is currently available. Over the last decade, there have been hundreds of studies that have used neural networks to analyze ecological data, yet most of this work has centered on applications in automated monitoring (Borowiec et al. 2022). This dissertation contributes to the literature by providing methodological guidance on how neural networks can be used to inform conservation decision-making, and by investigating the performance of neural network models for critical transition and water quality forecasting. In these areas, I have shown that neural networks offer advantages over classical methods, but I have also been careful to highlight that neural network models have significant limitations. With further method development and careful consideration of their limitations, neural networks will play a

key role in the analysis of ecological data.

References

- Adcock, Ben, and Nick Dexter. 2020. “The Gap Between Theory and Practice in Function Approximation with Deep Neural Networks.” <https://doi.org/10.48550/ARXIV.2001.07523>.
- Aitken, Kyle, Vinay V. Ramasesh, Yuan Cao, and Niru Maheswaranathan. 2021. “Understanding How Encoder-Decoder Architectures Attend.” arXiv. <http://arxiv.org/abs/2110.15253>.
- Albert, James S., Georgia Destouni, Scott M. Duke-Sylvester, Anne E. Magurran, Thierry Oberdorff, Roberto E. Reis, Kirk O. Winemiller, and William J. Ripple. 2021. “Scientists’ Warning to Humanity on the Freshwater Biodiversity Crisis.” *Ambio* 50 (1): 85–94. <https://doi.org/10.1007/s13280-020-01318-8>.
- Arulkumaran, Kai, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. 2017. “A Brief Survey of Deep Reinforcement Learning.” *IEEE Signal Processing Magazine* 34 (6): 26–38. <https://doi.org/10.1109/MSP.2017.2743240>.
- Assimakopoulos, V., and K. Nikolopoulos. 2000. “The Theta Model: A Decomposition Approach to Forecasting.” *International Journal of Forecasting* 16 (4): 521–30. [https://doi.org/10.1016/S0169-2070\(00\)00066-2](https://doi.org/10.1016/S0169-2070(00)00066-2).
- Bai, Shaojie, J. Zico Kolter, and Vladlen Koltun. 2018. “An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling.” arXiv. <https://doi.org/10.48550/ARXIV.1803.01271>.
- Barnosky, Anthony D., Elizabeth A. Hadly, Jordi Bascompte, Eric L. Berlow, James H. Brown, Mikael Fortelius, Wayne M. Getz, et al. 2012. “Approaching a State Shift in Earth’s Biosphere.” *Nature* 486 (7401): 52–58. <https://doi.org/10.1038/nature11018>.
- Berger-Tal, Oded, Jonathan Nathan, Ehud Meron, and David Saltz. 2014. “The Exploration-Exploitation Dilemma: A Multidisciplinary Framework.” *PLOS ONE* 9 (4): e95693. <https://doi.org/10.1371/journal.pone.0095693>.
- Boettiger, Carl. 2018a. “From Noise to Knowledge: How Randomness Generates Novel Phenomena and Reveals Information.” *Ecology Letters*. <https://doi.org/10.1111/ele.13085>.
- . 2018b. “From Noise to Knowledge: How Randomness Generates Novel Phenomena and Reveals Information.” Edited by Tim Coulson. *Ecology Letters* 21 (8): 1255–67. <https://doi.org/10.1111/ele.13085>.
- Boettiger, Carl, and Alan Hastings. 2012. “Early Warning Signals and the Prosecutor’s Fallacy.” *Proceedings of the Royal Society B: Biological Sciences* 279 (1748): 4734–39. <https://doi.org/10.1098/rspb.2012.2085>.
- Bommasani, Rishi, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, et al. 2021. “On the Opportunities and Risks of Foundation Models.” *arXiv:2108.07258 [Cs]*, August. <http://arxiv.org/abs/2108.07258>.
- Borowiec, Marek L., Rebecca B. Dikow, Paul B. Frandsen, Alexander McKeeken, Gabriele Valentini, and Alexander E. White. 2022. “Deep Learning as a Tool for Ecology and

- Evolution.” *Methods in Ecology and Evolution* 13 (8): 1640–60. <https://doi.org/10.1111/2041-210X.13901>.
- Brockman, Greg, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. “OpenAI Gym.” *arXiv:1606.01540 [Cs]*, June. <http://arxiv.org/abs/1606.01540>.
- Bury, Thomas M., R. I. Sujith, Induja Pavithran, Marten Scheffer, Timothy M. Lenton, Madhur Anand, and Chris T. Bauch. 2021. “Deep Learning for Early Warning Signals of Tipping Points.” *Proceedings of the National Academy of Sciences* 118 (39): e2106140118. <https://doi.org/10.1073/pnas.2106140118>.
- Caissie, D. 2006. “The Thermal Regime of Rivers: A Review.” *Freshwater Biology* 51 (8): 1389–1406. <https://doi.org/10.1111/j.1365-2427.2006.01597.x>.
- Carpenter, S. R., J. J. Cole, M. L. Pace, R. Batt, W. A. Brock, T. Cline, J. Coloso, et al. 2011. “Early Warnings of Regime Shifts: A Whole-Ecosystem Experiment.” *Science* 332 (6033): 1079–82. <https://doi.org/10.1126/science.1203672>.
- Carpenter, Stephen R., J. J. Cole, Michael L Pace, Ryan D. Batt, William A Brock, Timothy J. Cline, J. Coloso, et al. 2011. “Early Warnings of Regime Shifts: A Whole-Ecosystem Experiment.” *Science (New York, N.Y.)* 1079 (April). <https://doi.org/10.1126/science.1203672>.
- Castelvecchi, Davide. 2016. “Can We Open the Black Box of AI?” *Nature News* 538 (7623): 20. <https://doi.org/10.1038/538020a>.
- Catherine, Quiblier, Wood Susanna, Echenique-Subiabre Isidora, Heath Mark, Villeneuve Aurélie, and Humbert Jean-François. 2013. “A Review of Current Knowledge on Toxic Benthic Freshwater Cyanobacteria – Ecology, Toxin Production and Risk Management.” *Water Research* 47 (15): 5464–79. <https://doi.org/10.1016/j.watres.2013.06.042>.
- Ceballos, Gerardo, Paul R. Ehrlich, and Rodolfo Dirzo. 2017. “Biological Annihilation via the Ongoing Sixth Mass Extinction Signaled by Vertebrate Population Losses and Declines.” *Proceedings of the National Academy of Sciences* 114 (30). <https://doi.org/10.1073/pnas.1704949114>.
- Chades, Iadine, Luz V. Pascal, Sam Nicol, Cameron S. Fletcher, and Jonathan Ferrer Mestres. 2021. “A Primer on Partially Observable Markov Decision Processes (POMDPs).” *Methods in Ecology and Evolution*, August, 2041–210X.13692. <https://doi.org/10.1111/2041-210X.13692>.
- Chapman, Melissa, William Oestreich, Timothy H. Frawley, Carl Boettiger, Sibyl Diver, Bianca Santos, Caleb Scoville, et al. 2021. “Promoting Equity in Scientific Recommendations for High Seas Governance.” Preprint. EcoEvoRxiv. <https://doi.org/10.32942/osf.io/jhbuz>.
- Chen, Cheng, Qiuwen Chen, Siyang Yao, Mengnan He, Jianyun Zhang, Gang Li, and Yuqing Lin. 2024. “Combining Physical-Based Model and Machine Learning to Forecast Chlorophyll-a Concentration in Freshwater Lakes.” *Science of The Total Environment* 907 (January): 168097. <https://doi.org/10.1016/j.scitotenv.2023.168097>.
- Chung, Junyoung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling.” *arXiv:1412.3555 [Cs]*, December. <http://arxiv.org/abs/1412.3555>.
- Clark, Colin W. 1990. *Mathematical Bioeconomics: The Optimal Management of Renewable Resources, 2nd Edition*. Wiley-Interscience.
- Clark, Colin W. 1973. “Profit Maximization and the Extinction of Animal Species.” *Journal of Political Economy* 81 (4): 950–61. <https://doi.org/10.1086/260090>.
- Clark, Colin Whitcomb. 2010. *Mathematical Bioeconomics: The Mathematics of Conservation*. 3rd ed. Pure and Applied Mathematics. Hoboken, N.J: Wiley.

- Clark, James S., Steven R. Carpenter, Mary Barber, Scott Collins, Andy Dobson, Jonathan A. Foley, David M. Lodge, et al. 2001. “Ecological Forecasts: An Emerging Imperative.” *Science* 293 (5530): 657–60. <https://doi.org/10.1126/science.293.5530.657>.
- Conroy, Michael J., and James T. Peterson. 2013. *Decision Making in Natural Resource Management: A Structured, Adaptive Approach: A Structured, Adaptive Approach*. 1st ed. Wiley. <https://doi.org/10.1002/9781118506196>.
- Costello, Christopher, Daniel Ovando, Tyler Clavelle, C. Kent Strauss, Ray Hilborn, Michael C. Melnychuk, Trevor A Branch, et al. 2016. “Global fishery prospects under contrasting management regimes.” *Proceedings of the National Academy of Sciences* 113 (18): 5125–29. <https://doi.org/10.1073/pnas.1520420113>.
- Dai, Lei, Kirill S. Korolev, and Jeff Gore. 2015. “Relation Between Stability and Resilience Determines the Performance of Early Warning Signals Under Different Environmental Drivers.” *Proceedings of the National Academy of Sciences*, 201418415. <https://doi.org/10.1073/pnas.1418415112>.
- Dai, Lei, Daan Vorselen, Kirill S Korolev, and J. Gore. 2012. “Generic Indicators for Loss of Resilience Before a Tipping Point Leading to Population Collapse.” *Science (New York, N.Y.)* 336 (6085): 1175–77. <https://doi.org/10.1126/science.1219805>.
- Dakos, Vasilis, Stephen R. Carpenter, William A. Brock, Aaron M. Ellison, Vishwesh Guttal, Anthony R. Ives, Sonia Kéfi, et al. 2012. “Methods for Detecting Early Warnings of Critical Transitions in Time Series Illustrated Using Simulated Ecological Data.” Edited by Bülent Yener. *PLoS ONE* 7 (7): e41010. <https://doi.org/10.1371/journal.pone.0041010>.
- Dar, Yehuda, Vidya Muthukumar, and Richard G. Baraniuk. 2021. “A Farewell to the Bias-Variance Tradeoff? An Overview of the Theory of Overparameterized Machine Learning.” arXiv. <http://arxiv.org/abs/2109.02355>.
- Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. “ImageNet: A Large-Scale Hierarchical Image Database.” In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–55. Miami, FL: IEEE. <https://doi.org/10.1109/CVPR.2009.5206848>.
- Dietz, Simon, James Rising, Thomas Stoerk, and Gernot Wagner. 2021. “Economic Impacts of Tipping Points in the Climate System.” *Proceedings of the National Academy of Sciences* 118 (34): e2103081118. <https://doi.org/10.1073/pnas.2103081118>.
- Dietze, Michael. 2017. *Ecological Forecasting*. Princeton University Press. <https://doi.org/10.1515/9781400885459>.
- Dietze, Michael C., Andrew Fox, Lindsay M. Beck-Johnson, Julio L. Betancourt, Mevin B. Hooten, Catherine S. Jarnevich, Timothy H. Keitt, et al. 2018. “Iterative Near-Term Ecological Forecasting: Needs, Opportunities, and Challenges.” *Proceedings of the National Academy of Sciences* 115 (7): 1424–32. <https://doi.org/10.1073/pnas.1710231115>.
- Dirzo, Rodolfo, Hillary S Young, Mauro Galetti, Gerardo Ceballos, Nick JB Isaac, and Ben Collen. 2014. “Defaunation in the Anthropocene.” *Science* 345 (6195): 401–6.
- Dobson, Andrew P., Stuart L. Pimm, Lee Hannah, Les Kaufman, Jorge A. Ahumada, Amy W. Ando, Aaron Bernstein, et al. 2020. “Ecology and Economics for Pandemic Prevention.” *Science* 369 (6502): 379–81. <https://doi.org/10.1126/science.abc3189>.
- Du, Shengdong, Tianrui Li, Yan Yang, and Shi-Jinn Horng. 2020. “Multivariate Time Series Forecasting via Attention-Based Encoder–Decoder Framework.” *Neurocomputing* 388 (May): 269–79. <https://doi.org/10.1016/j.neucom.2019.12.118>.
- Equihua, Julián, Michael Beckmann, and Ralf Seppelt. 2024. “Connectivity Conservation Planning Through Deep Reinforcement Learning.” *Methods in Ecology and Evolution*

- 15 (4): 779–90. <https://doi.org/10.1111/2041-210X.14300>.
- Farley, Scott S, Andria Dawson, Simon J Goring, and John W Williams. 2018. “Situating Ecology as a Big-Data Science: Current Advances, Challenges, and Solutions.” *BioScience* 68 (8): 563–76. <https://doi.org/10.1093/biosci/biy068>.
- Ferrer-Mestres, Jonathan, Thomas G. Dietterich, Olivier Buffet, and Iadine Chades. 2021. “K-N-MOMDPs: Towards Interpretable Solutions for Adaptive Management.” *Proceedings of the AAAI Conference on Artificial Intelligence* 35 (17): 14775–84. <https://ojs.aaai.org/index.php/AAAI/article/view/17735>.
- Fielding, Alan H., ed. 1999. *Machine Learning Methods for Ecological Applications*. Boston, MA: Springer US. <https://doi.org/10.1007/978-1-4615-5289-5>.
- Fiorucci, Jose A., Tiago R. Pellegrini, Francisco Louzada, Fotios Petropoulos, and Anne B. Koehler. 2016. “Models for Optimising the Theta Method and Their Relationship to State Space Models.” *International Journal of Forecasting* 32 (4): 1151–61. <https://doi.org/10.1016/j.ijforecast.2016.02.005>.
- Fischer, Joern, Garry D Peterson, Toby A. Gardner, Line J Gordon, Ioan Fazey, Thomas Elmqvist, Adam Felton, Carl Folke, and Stephen Dovers. 2009. “Integrating Resilience Thinking and Optimisation for Conservation.” *Trends in Ecology & Evolution* 24 (10): 549–54. <https://doi.org/10.1016/j.tree.2009.03.020>.
- Folke, Carl, Steve Carpenter, Brian Walker, Marten Scheffer, Thomas Elmqvist, Lance Gunderson, and C. S. Holling. 2004. “Regime Shifts, Resilience, and Biodiversity in Ecosystem Management.” *Annual Review of Ecology, Evolution, and Systematics* 35 (1): 557–81. <https://doi.org/10.1146/annurev.ecolsys.35.021103.105711>.
- Frankenhuis, Willem E., Karthik Panchanathan, and Andrew G. Barto. 2019. “Enriching Behavioral Ecology with Reinforcement Learning Methods.” *Behavioural Processes* 161 (April): 94–100. <https://doi.org/10.1016/j.beproc.2018.01.008>.
- Fujimoto, Scott, Herke van Hoof, and David Meger. 2018. “Addressing Function Approximation Error in Actor-Critic Methods.” *arXiv:1802.09477 [Cs, Stat]*, October. <http://arxiv.org/abs/1802.09477>.
- Getz, Wayne M., Charles R. Marshall, Colin J. Carlson, Luca Giuggioli, Sadie J. Ryan, Stephanie S. Romañach, Carl Boettiger, et al. 2018. “Making Ecological Models Adequate.” Edited by Tim Coulson. *Ecology Letters* 21 (2): 153–66. <https://doi.org/10.1111/ele.12893>.
- Gneiting, Tilmann, and Matthias Katzfuss. 2014. “Probabilistic Forecasting.” *Annual Review of Statistics and Its Application* 1 (1): 125–51. <https://doi.org/10.1146/annurev-statistics-062713-085831>.
- Gneiting, Tilmann, and Adrian E Raftery. 2007. “Strictly Proper Scoring Rules, Prediction, and Estimation.” *Journal of the American Statistical Association* 102 (477): 359–78. <https://doi.org/10.1198/016214506000001437>.
- Gneiting, Tilmann, and Adrian E. Raftery. 2005. “Weather Forecasting with Ensemble Methods.” *Science* 310 (5746): 248–49. <https://doi.org/10.1126/science.1115255>.
- Grande, Robert, Thomas Walsh, and Jonathan How. 2014. “Sample Efficient Reinforcement Learning with Gaussian Processes.” In *Proceedings of the 31st International Conference on Machine Learning*, edited by Eric P. Xing and Tony Jebara, 32:1332–40. Proceedings of Machine Learning Research 2. Beijing, China: PMLR. <http://proceedings.mlr.press/v32/grande14.html>.
- Gregory, R., L. Failing, M. Harstone, G. Long, T. McDaniels, and D. Ohlson. 2012. *Structured Decision Making: A Practical Guide to Environmental Management Choices*. 1st ed. Wiley. <https://doi.org/10.1002/9781444398557>.
- Gu, Shixiang, Timothy Lillicrap, Zoubin Ghahramani, Richard E. Turner, and Sergey Levine. 2017. “Q-Prop: Sample-Efficient Policy Gradient with An Off-Policy Critic.”

- arXiv:1611.02247 [Cs]*, February. <http://arxiv.org/abs/1611.02247>.
- Ha, Sehoon, Peng Xu, Zhenyu Tan, Sergey Levine, and Jie Tan. 2020. “Learning to Walk in the Real World with Minimal Human Effort.” *arXiv:2002.08550 [Cs]*, November. <http://arxiv.org/abs/2002.08550>.
- Haarnoja, Tuomas, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. “Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor.” *arXiv:1801.01290 [Cs, Stat]*, August. <http://arxiv.org/abs/1801.01290>.
- Hadfield-Menell, Dylan, Smitha Milli, Pieter Abbeel, Stuart Russell, and Anca Dragan. 2020. “Inverse Reward Design.” *arXiv:1711.02827 [Cs]*, October. <http://arxiv.org/abs/1711.02827>.
- Hallett, T. B., T. Coulson, J. G. Pilkington, T. H. Clutton-Brock, J. M. Pemberton, and B. T. Grenfell. 2004. “Why Large-Scale Climate Indices Seem to Predict Ecological Processes Better Than Local Weather.” *Nature* 430 (6995): 71–75. <https://doi.org/10.1038/nature02708>.
- Hanin, Boris. 2019. “Universal Function Approximation by Deep Neural Nets with Bounded Width and ReLU Activations.” *Mathematics* 7 (10): 992. <https://doi.org/10.3390/math7100992>.
- Hanson, Paul C., Aviah B. Stillman, Xiaowei Jia, Anuj Karpatne, Hilary A. Dugan, Cayelan C. Carey, Jemma Stachelek, et al. 2020. “Predicting Lake Surface Water Phosphorus Dynamics Using Process-Guided Machine Learning.” *Ecological Modelling* 430 (August): 109136. <https://doi.org/10.1016/j.ecolmodel.2020.109136>.
- Hastings, Alan. 1996. *Population Biology: Concepts and Models*. Springer.
- Hastings, Alan, and Louis J. Gross, eds. 2012. *Encyclopedia of Theoretical Ecology*. Oakland, CA: University of California Press.
- Henderson, Peter, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. 2019. “Deep Reinforcement Learning That Matters.” *arXiv:1709.06560 [Cs, Stat]*, January. <http://arxiv.org/abs/1709.06560>.
- Hernandez, Danny, and Tom B. Brown. 2020. “Measuring the Algorithmic Efficiency of Neural Networks.” *arXiv:2005.04305 [Cs, Stat]*, May. <http://arxiv.org/abs/2005.04305>.
- Herzen, Julien, Francesco Lässig, Samuele Giuliano Piazzetta, Thomas Neuer, Léo Tafti, Guillaume Raille, Tomas Van Pottelbergh, et al. 2021. “Darts: User-Friendly Modern Machine Learning for Time Series.” <https://doi.org/10.48550/ARXIV.2110.03224>.
- Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. “Long Short-Term Memory.” *Neural Computation* 9 (8): 1735–80. <https://doi.org/10.1162/neco.1997.9.8.1735>.
- Hornik, Kurt, Maxwell Stinchcombe, and Halbert White. 1989. “Multilayer Feedforward Networks Are Universal Approximators.” *Neural Networks* 2 (5): 359–66. [https://doi.org/10.1016/0893-6080\(89\)90020-8](https://doi.org/10.1016/0893-6080(89)90020-8).
- Huang, Dan. 2018. “How Much Did AlphaGo Zero Cost?” <https://www.yuzeh.com/data/agz-cost.html>.
- Hyndman, R. J., and G. Athanasopoulos. 2018. *Forecasting: Principles and Practice*. 2nd ed. Melbourne, Australia: OTexts. [OTexts.com/fpp2](https://www.otexts.com/fpp2).
- Janner, Michael, Justin Fu, Marvin Zhang, and Sergey Levine. 2019. “When to Trust Your Model: Model-Based Policy Optimization.” *arXiv:1906.08253 [Cs, Stat]*, November. <http://arxiv.org/abs/1906.08253>.
- Joseph, Liana N., Richard F. Maloney, and Hugh P. Possingham. 2009. “Optimal Allocation of Resources Among Threatened Species: A Project Prioritization Protocol.” *Conservation Biology* 23 (2): 328–38. <https://doi.org/10.1111/j.1523-1739.2008.01124.x>.
- Joseph, Maxwell B. 2020. “Neural Hierarchical Models of Ecological Populations.” *Ecology Letters* 23 (4): 734–47. <https://doi.org/10.1111/ele.13462>.

- Kampen, N. G. van. 1992. *Stochastic Processes in Physics and Chemistry*. Rev. and enl. ed. North-Holland Personal Library. Amsterdam ; New York: North-Holland.
- Kao, I-Feng, Yanlai Zhou, Li-Chiu Chang, and Fi-John Chang. 2020. “Exploring a Long Short-Term Memory Based Encoder-Decoder Framework for Multi-Step-Ahead Flood Forecasting.” *Journal of Hydrology* 583 (April): 124631. <https://doi.org/10.1016/j.jhydrol.2020.124631>.
- Karpatne, Anuj, Xiaowei Jia, and Vipin Kumar. 2024. “Knowledge-Guided Machine Learning: Current Trends and Future Prospects.” arXiv. <https://doi.org/10.48550/ARXIV.2403.15989>.
- Kazak, Yafim, Clark Barrett, Guy Katz, and Michael Schapira. 2019. “Verifying Deep-RL-Driven Systems.” In *Proceedings of the 2019 Workshop on Network Meets AI & ML - NetAI’19*, 83–89. Beijing, China: ACM Press. <https://doi.org/10.1145/3341216.3342218>.
- Knape, Jonas, and Perry de Valpine. 2012. “Are Patterns of Density Dependence in the Global Population Dynamics Database Driven by Uncertainty about Population Abundance?” *Ecology Letters* 15 (1): 17–23. <https://doi.org/10.1111/j.1461-0248.2011.01702.x>.
- Lapeyrolerie, Marcus, and Carl Boettiger. 2021. “Teaching Machines to Anticipate Catastrophes.” *Proceedings of the National Academy of Sciences* 118 (40): e2115605118. <https://doi.org/10.1073/pnas.2115605118>.
- Levins, Richard. 1966. “The Strategy of Model Building in Population Biology.” *American Scientist* 54 (4): 421–31.
- Liang, Shiyu, and R. Srikant. 2016. “Why Deep Neural Networks for Function Approximation?” arXiv. <https://doi.org/10.48550/ARXIV.1610.04161>.
- Lim, Bryan, Sercan O. Arik, Nicolas Loeff, and Tomas Pfister. 2019. “Temporal Fusion Transformers for Interpretable Multi-Horizon Time Series Forecasting.” arXiv. <https://doi.org/10.48550/ARXIV.1912.09363>.
- Lyu, Pingyang, Ning Chen, Shanjun Mao, and Mei Li. 2020. “LSTM Based Encoder-Decoder for Short-Term Predictions of Gas Concentration Using Multi-Sensor Fusion.” *Process Safety and Environmental Protection* 137 (May): 93–105. <https://doi.org/10.1016/j.psep.2020.02.021>.
- Madhyastha, Pranava, and Rishabh Jain. 2019. “On Model Stability as a Function of Random Seed.” *arXiv:1909.10447 [Cs, Stat]*, September. <http://arxiv.org/abs/1909.10447>.
- Maier, Holger R., and Graeme C. Dandy. 2000. “Neural Networks for the Prediction and Forecasting of Water Resources Variables: A Review of Modelling Issues and Applications.” *Environmental Modelling & Software* 15 (1): 101–24. [https://doi.org/10.1016/S1364-8152\(99\)00007-9](https://doi.org/10.1016/S1364-8152(99)00007-9).
- Makridakis, Spyros, and Michèle Hibon. 2000. “The M3-Competition: Results, Conclusions and Implications.” *International Journal of Forecasting* 16 (4): 451–76. [https://doi.org/10.1016/S0169-2070\(00\)00057-1](https://doi.org/10.1016/S0169-2070(00)00057-1).
- Makridakis, Spyros, Evangelos Spiliotis, and Vassilios Assimakopoulos. 2018. “Statistical and Machine Learning Forecasting Methods: Concerns and Ways Forward.” Edited by Alejandro Raul Hernandez Montoya. *PLOS ONE* 13 (3): e0194889. <https://doi.org/10.1371/journal.pone.0194889>.
- . 2020. “The M4 Competition: 100,000 Time Series and 61 Forecasting Methods.” *International Journal of Forecasting* 36 (1): 54–74. <https://doi.org/10.1016/j.ijforecast.2019.04.014>.
- Marescot, Lucile, Guillaume Chapron, Iadine Chadès, Paul L. Fackler, Christophe Duchamp, Eric Marboutin, and Olivier Gimenez. 2013. “Complex Decisions Made

- Simple: A Primer on Stochastic Dynamic Programming.” *Methods in Ecology and Evolution* 4 (9): 872–84. <https://doi.org/10.1111/2041-210X.12082>.
- Matthews, David. 2018. “Supercharge Your Data Wrangling with a Graphics Card.” *Nature* 562 (7725): 151–52. <https://doi.org/10.1038/d41586-018-06870-8>.
- May, Robert M. 1977. “Thresholds and Breakpoints in Ecosystems with a Multiplicity of Stable States.” *Nature* 269 (5628): 471–77. <https://doi.org/10.1038/269471a0>.
- May, Robert M., and Roy M. Anderson. 1979. “Population Biology of Infectious Diseases: Part II.” *Nature* 280 (5722): 455–61. <https://doi.org/10.1038/280455a0>.
- Mehta, Pankaj, Marin Bukov, Ching-Hao Wang, Alexandre G. R. Day, Clint Richardson, Charles K. Fisher, and David J. Schwab. 2019. “A High-Bias, Low-Variance Introduction to Machine Learning for Physicists.” *Physics Reports* 810 (May): 1–124. <https://doi.org/10.1016/j.physrep.2019.03.001>.
- Meir, Eli, Sandy Andelman, and Hugh P. Possingham. 2004. “Does Conservation Planning Matter in a Dynamic and Uncertain World?” *Ecology Letters* 7 (8): 615–22. <https://doi.org/10.1111/j.1461-0248.2004.00624.x>.
- Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, et al. 2015. “Human-Level Control Through Deep Reinforcement Learning.” *Nature* 518 (7540): 529–33. <https://doi.org/10.1038/nature14236>.
- Moritz, Max A., Enric Batllori, Ross A. Bradstock, A. Malcolm Gill, John Handmer, Paul F. Hessburg, Justin Leonard, et al. 2014. “Learning to Coexist with Wildfire.” *Nature* 515 (7525): 58–66. <https://doi.org/10.1038/nature13946>.
- Nicholson, A. J., and V. A. Bailey. 1935. “The Balance of Animal Populations.—Part I.” *Proceedings of the Zoological Society of London* 105 (3): 551–98. <https://doi.org/10.1111/j.1096-3642.1935.tb01680.x>.
- Nicholson, Aj. 1954a. “An Outline of the Dynamics of Animal Populations.” *Australian Journal of Zoology* 2 (1): 9. <https://doi.org/10.1071/ZO9540009>.
- . 1954b. “Compensatory Reactions of Populations to Stresses, and Their Evolutionary Significance.” *Australian Journal of Zoology* 2 (1): 1. <https://doi.org/10.1071/ZO9540001>.
- Nordhaus, W. D. 1992. “An Optimal Transition Path for Controlling Greenhouse Gases.” *Science* 258 (5086): 1315–19. <https://doi.org/10.1126/science.258.5086.1315>.
- Olsson, Freya, Tadhg N. Moore, Cayelan C. Carey, Adrienne Breef-Pilz, and R. Quinn Thomas. 2024. “A Multi-Model Ensemble of Baseline and Process-Based Models Improves the Predictive Skill of Near-Term Lake Forecasts.” *Water Resources Research* 60 (3): e2023WR035901. <https://doi.org/10.1029/2023WR035901>.
- OpenAI, Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, et al. 2019. “Learning Dexterous In-Hand Manipulation.” *arXiv:1808.00177 [Cs, Stat]*, January. <http://arxiv.org/abs/1808.00177>.
- Oreshkin, Boris N., Dmitri Carпов, Nicolas Chapados, and Yoshua Bengio. 2019. “N-BEATS: Neural Basis Expansion Analysis for Interpretable Time Series Forecasting.” *arXiv*. <https://doi.org/10.48550/ARXIV.1905.10437>.
- Oreskes, Naomi, Kristin Shrader-Frechette, and Kenneth Belitz. 1994. “Verification, Validation, and Confirmation of Numerical Models in the Earth Sciences.” *Science* 263 (5147): 641–46. <https://doi.org/10.1126/science.263.5147.641>.
- Ouellet-Proulx, Sébastien, André St-Hilaire, and Marie-Amélie Boucher. 2017. “Water Temperature Ensemble Forecasts: Implementation Using the CEQUEAU Model on Two Contrasted River Systems.” *Water* 9 (7): 457. <https://doi.org/10.3390/w9070457>.
- Ovaskainen, Otso, and Baruch Meerson. 2010. “Stochastic Models of Population Extinc-

- tion.” *Trends in Ecology & Evolution* 25 (11): 643–52. <https://doi.org/10.1016/j.tree.2010.07.009>.
- Pacala, Stephen W., Charles D. Canham, John Saponara, John A. Silander, Richard K. Kobe, and Eric Ribbens. 1996. “Forest Models Defined by Field Measurements: Estimation, Error Analysis and Dynamics.” *Ecological Monographs* 66 (1): 1–43. <https://doi.org/10.2307/2963479>.
- Pascanu, Razvan, Tomas Mikolov, and Yoshua Bengio. 2013. “On the Difficulty of Training Recurrent Neural Networks.” *arXiv:1211.5063 [Cs]*, February. <http://arxiv.org/abs/1211.5063>.
- Perolat, Julien, Joel Z. Leibo, Vinicius Zambaldi, Charles Beattie, Karl Tuyls, and Thore Graepel. 2017. “A Multi-Agent Reinforcement Learning Model of Common-Pool Resource Appropriation.” *arXiv:1707.06600 [Cs, q-Bio]*, September. <http://arxiv.org/abs/1707.06600>.
- Polasky, Stephen, Stephen R. Carpenter, Carl Folke, and Bonnie Keeler. 2011a. “Decision-Making Under Great Uncertainty: Environmental Management in an Era of Global Change.” *Trends in Ecology & Evolution*, May, 1–7. <https://doi.org/10.1016/j.tree.2011.04.007>.
- . 2011b. “Decision-Making Under Great Uncertainty: Environmental Management in an Era of Global Change.” *Trends in Ecology & Evolution* 26 (8): 398–404. <https://doi.org/10.1016/j.tree.2011.04.007>.
- Pollock, M. S., L. M. J. Clarke, and M. G. Dubé. 2007. “The Effects of Hypoxia on Fishes: From Ecological Relevance to Physiological Effects.” *Environmental Reviews* 15 (NA): 1–14. <https://doi.org/10.1139/a06-006>.
- Pong, Vitchyr, Shixiang Gu, Murtaza Dalal, and Sergey Levine. 2020. “Temporal Difference Models: Model-Free Deep RL for Model-Based Control.” *arXiv:1802.09081 [Cs]*, February. <http://arxiv.org/abs/1802.09081>.
- Popova, Mariya, Olexandr Isayev, and Alexander Tropsha. 2018. “Deep Reinforcement Learning for de Novo Drug Design.” *Science Advances* 4 (7): eaap7885. <https://doi.org/10.1126/sciadv.aap7885>.
- Punt, André E, Doug S Butterworth, Carryn L de Moor, José A A De Oliveira, and Malcolm Haddon. 2016. “Management Strategy Evaluation: Best Practices.” *Fish and Fisheries* 17 (2): 303–34. <https://doi.org/10.1111/faf.12104>.
- RAM Legacy Stock Assessment Database. 2020. “RAM Legacy Stock Assessment Database V4.491.” <https://doi.org/10.5281/zenodo.3676088>.
- Raschka, Sebastian. 2020. “Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning.” *arXiv*. <http://arxiv.org/abs/1811.12808>.
- Read, Jordan S., Xiaowei Jia, Jared Willard, Alison P. Appling, Jacob A. Zwart, Samantha K. Oliver, Anuj Karpatne, et al. 2019. “Process-Guided Deep Learning Predictions of Lake Water Temperature.” *Water Resources Research* 55 (11): 9173–90. <https://doi.org/10.1029/2019WR024922>.
- Reed, William J. 1979. “Optimal escapement levels in stochastic and deterministic harvesting models.” *Journal of Environmental Economics and Management* 6 (4): 350–63. [https://doi.org/10.1016/0095-0696\(79\)90014-7](https://doi.org/10.1016/0095-0696(79)90014-7).
- Rudin, Cynthia. 2019. “Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead.” *Nature Machine Intelligence* 1 (5): 206–15. <https://doi.org/10.1038/s42256-019-0048-x>.
- Schaefer, Milner B. 1954. “Some aspects of the dynamics of populations important to the management of the commercial marine fisheries.” *Bulletin of the Inter-American Tropical Tuna Commission* 1 (2): 27–56. <https://doi.org/10.1007/BF02464432>.
- Scheffer, Marten, Jordi Bascompte, William A. Brock, Victor Brovkin, Stephen R. Car-

- penter, Vasilis Dakos, Hermann Held, Egbert H. van Nes, Max Rietkerk, and George Sugihara. 2009. “Early-Warning Signals for Critical Transitions.” *Nature* 461 (7260): 53–59. <https://doi.org/10.1038/nature08227>.
- Scheffer, Marten, Stephen R. Carpenter, Vasilis Dakos, and Egbert van Nes. 2015. “Generic Indicators of Ecological Resilience.” *Annual Review of Ecology, Evolution, and Systematics* 46 (1): 145–67. <https://doi.org/10.1146/annurev-ecolsys-112414-054242>.
- Scheffer, Marten, Stephen R. Carpenter, Jonathan A. Foley, Carl Folke, and Brian H Walker. 2001a. “Catastrophic Shifts in Ecosystems.” *Nature* 413 (6856): 591–96. <https://doi.org/10.1038/35098000>.
- Scheffer, Marten, Stephen R. Carpenter, Timothy M. Lenton, Jordi Bascompte, William Brock, Vasilis Dakos, Johan van de Koppel, et al. 2012. “Anticipating Critical Transitions.” *Science* 338 (6105): 344–48. <https://doi.org/10.1126/science.1225244>.
- Scheffer, Marten, Steve Carpenter, Jonathan A. Foley, Carl Folke, and Brian Walker. 2001b. “Catastrophic Shifts in Ecosystems.” *Nature* 413 (6856): 591–96. <https://doi.org/10.1038/35098000>.
- Schindler, Daniel E, Jonathan B Armstrong, and Thomas E Reed. 2015. “The Portfolio Concept in Ecology and Evolution.” *Frontiers in Ecology and the Environment* 13 (5): 257–63. <https://doi.org/10.1890/140275>.
- Scoville, Caleb, Melissa Chapman, Razvan Amironesei, and Carl Boettiger. 2021. “Algorithmic Conservation in a Changing Climate.” *Current Opinion in Environmental Sustainability* 51 (August): 30–35. <https://doi.org/10.1016/j.cosust.2021.01.009>.
- Semenoglou, Artemios-Anargyros, Evangelos Spiliotis, Spyros Makridakis, and Vassilios Assimakopoulos. 2021. “Investigating the Accuracy of Cross-Learning Time Series Forecasting Methods.” *International Journal of Forecasting* 37 (3): 1072–84. <https://doi.org/10.1016/j.ijforecast.2020.11.009>.
- Sherstinsky, Alex. 2020. “Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network.” *Physica D: Nonlinear Phenomena* 404 (March): 132306. <https://doi.org/10.1016/j.physd.2019.132306>.
- Silver, David, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, et al. 2016. “Mastering the Game of Go with Deep Neural Networks and Tree Search.” *Nature* 529 (7587): 484–89. <https://doi.org/10.1038/nature16961>.
- Silver, David, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, et al. 2018. “A General Reinforcement Learning Algorithm That Masters Chess, Shogi, and Go Through Self-Play.” *Science* 362 (6419): 1140–44. <https://doi.org/10.1126/science.aar6404>.
- Silver, David, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, et al. 2017. “Mastering the Game of Go Without Human Knowledge.” *Nature* 550 (7676): 354–59. <https://doi.org/10.1038/nature24270>.
- Silvestro, Daniele, Stefano Gorla, Thomas Sterner, and Alexandre Antonelli. 2022. “Improving Biodiversity Protection Through Artificial Intelligence.” *Nature Sustainability* 5 (5): 415–24. <https://doi.org/10.1038/s41893-022-00851-6>.
- Stajkowski, Stephen, Mohammad Zeynoddin, Hani Farghaly, Bahram Gharabaghi, and Hossein Bonakdari. 2020. “A Methodology for Forecasting Dissolved Oxygen in Urban Streams.” *Water* 12 (9): 2568. <https://doi.org/10.3390/w12092568>.
- Steenbeek, Jeroen, Joe Buszowski, Villy Christensen, Ekin Akoglu, Kerim Aydin, Nick Ellis, Dalai Felinto, et al. 2016. “Ecopath with Ecosim as a Model-Building Toolbox: Source Code Capabilities, Extensions, and Variations.” *Ecological Modelling* 319 (January): 178–89. <https://doi.org/10.1016/j.ecolmodel.2015.06.031>.

- Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. 2014. “Sequence to Sequence Learning with Neural Networks.” *arXiv:1409.3215 [Cs]*, December. <http://arxiv.org/abs/1409.3215>.
- Sutton, Richard S, and Andrew G Barto. 2018. *Reinforcement Learning: An Introduction*. MIT press.
- Thomas, R Quinn, Carl Boettiger, Cayelan C Carey, Michael C Dietze, Leah R Johnson, Melissa A Kenney, Jason S McLachlan, et al. 2023. “The NEON Ecological Forecasting Challenge.” *Frontiers in Ecology and the Environment* 21 (3): 112–13. <https://doi.org/10.1002/fee.2616>.
- Thomas, R Quinn, Ryan P McClure, Tadhg N Moore, Whitney M Woelmer, Carl Boettiger, Renato J Figueiredo, Robert T Hensley, and Cayelan C Carey. 2023. “Near-term Forecasts of NEON Lakes Reveal Gradients of Environmental Predictability Across the US.” *Frontiers in Ecology and the Environment* 21 (5): 220–26. <https://doi.org/10.1002/fee.2623>.
- Tickner, David, Jeffrey J Opperman, Robin Abell, Mike Acreman, Angela H Arthington, Stuart E Bunn, Steven J Cooke, et al. 2020. “Bending the Curve of Global Freshwater Biodiversity Loss: An Emergency Recovery Plan.” *BioScience* 70 (4): 330–42. <https://doi.org/10.1093/biosci/biaa002>.
- Treloar, Neythen J., Alex J. H. Fedorec, Brian Ingalls, and Chris P. Barnes. 2020. “Deep Reinforcement Learning for the Control of Microbial Co-Cultures in Bioreactors.” Edited by Lingchong You. *PLOS Computational Biology* 16 (4): e1007783. <https://doi.org/10.1371/journal.pcbi.1007783>.
- Valletta, John Joseph, Colin Torney, Michael Kings, Alex Thornton, and Joah Madden. 2017. “Applications of Machine Learning in Animal Behaviour Studies.” *Animal Behaviour* 124 (February): 203–20. <https://doi.org/10.1016/j.anbehav.2016.12.005>.
- Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. “Attention Is All You Need.” *arXiv*. <https://doi.org/10.48550/ARXIV.1706.03762>.
- Villarroel, J. Andrei, John E. Taylor, and Christopher L. Tucci. 2013. “Innovation and Learning Performance Implications of Free Revealing and Knowledge Brokering in Competing Communities: Insights from the Netflix Prize Challenge.” *Computational and Mathematical Organization Theory* 19 (1): 42–77. <https://doi.org/10.1007/s10588-012-9137-7>.
- Vinyals, Oriol, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, et al. 2019. “Grandmaster Level in StarCraft II Using Multi-Agent Reinforcement Learning.” *Nature* 575 (7782): 350–54. <https://doi.org/10.1038/s41586-019-1724-z>.
- Walters, Carl J, and Ray Hilborn. 1978. “Ecological Optimization and Adaptive Management.” *Annual Review of Ecology and Systematics* 9 (1): 157–88. <https://doi.org/10.1146/annurev.es.09.110178.001105>.
- Wang, Xueting, Jun Cheng, and Lei Wang. 2020. “A Reinforcement Learning-Based Predator-Prey Model.” *Ecological Complexity* 42 (March): 100815. <https://doi.org/10.1016/j.ecocom.2020.100815>.
- Wheeler, Kathryn I., Michael C. Dietze, David LeBauer, Jody A. Peters, Andrew D. Richardson, Arun A. Ross, R. Quinn Thomas, et al. 2024. “Predicting Spring Phenology in Deciduous Broadleaf Forests: NEON Phenology Forecasting Community Challenge.” *Agricultural and Forest Meteorology* 345 (February): 109810. <https://doi.org/10.1016/j.agrformet.2023.109810>.
- Williams, John W., and Stephen T. Jackson. 2007. “Novel Climates, No-Analog Com-

- munities, and Ecological Surprises.” *Frontiers in Ecology and the Environment* 5 (9): 475–82. <https://doi.org/10.1890/070037>.
- Wilson, Kerrie A., Marissa F. McBride, Michael Bode, and Hugh P. Possingham. 2006. “Prioritizing Global Conservation Efforts.” *Nature* 440 (7082): 337–40. <https://doi.org/10.1038/nature04366>.
- Worm, Boris, Edward B Barbier, Nicola Beaumont, J Emmett Duffy, Carl Folke, Benjamin S Halpern, Jeremy B C Jackson, et al. 2006. “Impacts of biodiversity loss on ocean ecosystem services.” *Science (New York, N.Y.)* 314 (5800): 787–90. <https://doi.org/10.1126/science.1132294>.
- Worm, Boris, Ray Hilborn, Julia K Baum, Trevor A Branch, Jeremy S Collie, Christopher Costello, Michael J Fogarty, et al. 2009. “Rebuilding global fisheries.” *Science (New York, N.Y.)* 325 (5940): 578–85. <https://doi.org/10.1126/science.1173146>.
- Xu, Lily, Andrew Perrault, Fei Fang, Haipeng Chen, and Milind Tambe. 2021. “Robust Reinforcement Learning Under Minimax Regret for Green Security.” arXiv. <http://arxiv.org/abs/2106.08413>.
- Zhi, Wei, Alison P. Appling, Heather E. Golden, Joel Podgorski, and Li Li. 2024. “Deep Learning for Water Quality.” *Nature Water* 2 (3): 228–41. <https://doi.org/10.1038/s44221-024-00202-z>.
- Zhou, Zhenpeng, Xiaocheng Li, and Richard N. Zare. 2017. “Optimizing Chemical Reactions with Deep Reinforcement Learning.” *ACS Central Science* 3 (12): 1337–44. <https://doi.org/10.1021/acscentsci.7b00492>.
- Zwart, Jacob A., Jeremy Diaz, Scott Hamshaw, Samantha Oliver, Jesse C. Ross, Margaux Sleckman, Alison P. Appling, et al. 2023. “Evaluating Deep Learning Architecture and Data Assimilation for Improving Water Temperature Forecasts at Unmonitored Locations.” *Frontiers in Water* 5 (June): 1184992. <https://doi.org/10.3389/frwa.2023.1184992>.

