

**Technical Report**

**wtCNN: Wavelet-Transform Convolutional Neural Networks for Identifying Regime Shifts in Ecological Timeseries**

By

Ryan Taylor

8 May 2020

B.S. in Engineering Science

B.A. in Physics

On my honor as a University student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

Signed: *Ryan Taylor*

Approved: \_\_\_\_\_ Date: \_\_\_\_\_

Jonathan Walter, Department of Environmental Science

## ***Problem***

Monitoring and forecasting the health of our lakes grows more important as global temperatures rise, local ecologies become more unpredictable, and environmental water protections are relaxed (Stafford et al, 2013). The ecology of these aquatic ecosystems is heavily dependent on the composition and volume of nutrients flowing into and out of them. With an overloading of nutrients, particularly phosphorus and nitrogen compounds, the state of an ecosystem can become unstable. The structure of the food web, balance of nutrients, and population of each species can shift rapidly in what is called a regime shift, a transition from one ecological equilibrium to another. One such transition in lake ecosystems is from an oligotrophic state with low-biomass to a eutrophic state with high phytoplankton biomass (Carpenter, 2003). This transition is what occurs in cyanobacterial algal blooms that produce toxins and endanger humans, animals, and the long-term health of the ecosystem (Kidwell, 2015). The ability to predict if and when these regime shifts will occur would prove a useful tool for mitigating harmful ecological transitions like these (Ontario Ministry of the Environment and Climate Change, 2017).

While they occur on the relatively short timescale of a few months, regime shifts in lake ecosystems are accompanied by pretransition indicators that can act like early-warning systems in ecological timeseries (Carpenter et al, 2011). There are three theoretical features that indicate an increased likelihood of transition: critical slowing down, flickering, and frequency multimodality. In the first of these, the system equilibrates from a perturbation more slowly because of an underlying instability in the ecosystem. Existing techniques use an increase in temporal autocorrelation to measure this feature. Next, in flickering, an ecosystem transiently “flips” into another regime because of stochastic fluctuations or patchiness. This is marked by an

increase in both autocorrelation and signal variance. Lastly, frequency multimodality is found when characteristic oscillations of other regimes begin forming underneath the current regime (Scheffer et al, 2012). This requires spectral analysis. Autocorrelative methods to detect trends in multimodality have tended to produce false positives, weakening their predictive power (Andersen et al, 2008; Hsieh, 2007).

These early warning features and their corresponding metrics are insufficient to provide a comprehensive understanding of an ecological regime and its stability. There is currently no single metric that reliably evaluates the presence of all three pretransition characteristics. Additionally, measures like autocorrelation and variance are calculated in a fixed-length moving window across the timeseries; they cannot measure across the diverse timescales characteristic to aquatic ecology (Cazelles et al, 2008).

We need more advanced techniques for monitoring ecosystems at a point when climate change, anthropogenic over-enrichment, and hydrological modifications threaten the health and stability of our lakes and waterways (Burford et al, 2019). Harmful regime shifts like algal blooms can kill local aquatic life, endanger the health of residents and tourists, and cost millions to the economy (Wilson et al, 2018). Controlled experiments by Pace et al (2017) have shown that these harmful algal blooms can be prevented if early-warning signs are detected and acted on. This remains impossible if we cannot predict real-world regime shifts.

### ***Goal***

Our goal is to design and test a method for identifying and predicting regime shifts in aquatic ecosystems. We build our method using continuous wavelet transforms and convolutional neural networks (CNNs) to analyze timeseries data. Incorporating wavelet analysis allows for non-stationary timeseries to be analyzed while maintaining evaluation at large

timescales (Torrence & Compo, 1998). This method of analysis convolves time series with a parent function stretched over multiple timescales, and characterizes the autocorrelation, variance, and multimodality of a timeseries in a transformed dataset (Rouyer et al, 2008). The CNN acts to classify features in the wavelet-transformed dataset. What features are identified depends on how the neural network is trained. In this project, we tried two approaches: classifying by domain and classifying by the presence of a transition. We refer to our approach of combining wavelet analysis with convolutional neural networks as “wtCNN,” and develop a software package for the R programming language, having the same name, that implements our techniques.

In order to forecast possible changes in timeseries domain, we break down the timeseries into windows, comparing the wtCNN output of the most recent window to those prior. By implementing iterative wavelet transformations of non-stationary timeseries, we attempt to add predictive power to our analysis.

Our novel approach has potential upon further development. Because the indicators regime shifts are encoded in the wavelet transform data, the neural networks are only fed the most important information to make classifications and predictions. Additionally, neural networks can distinguish non-theoretical features of ecological regime shifts that may be present in an applied setting.

Here, we describe the design and testing of our software package in R. The system was trained and tested on simulated data of phytoplankton-nutrient interactions.

## Background

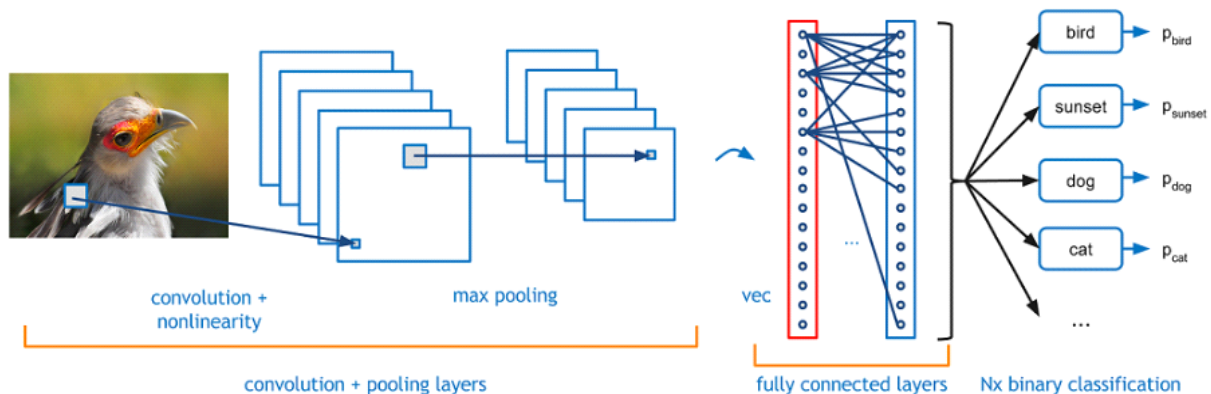
**Key terms:** Convolutional Neural Network, Wavelet Analysis, Regime, Regime Shift

**Convolutional neural networks** (CNNs) are powerful tools for image analysis and feature identification. They work by applying layers of interconnected filters and operators to an image until the output is an encoded piece of information about that image, like a classification.

A CNN is composed of multiple sub-layers of convolution and pooling. These sublayers typically include a convolutional layer, a thresholding layer, and a pooling layer. The first layer applies a windowed filter or *kernel* to the input through element-wise multiplications, a convolution. The second layer sets a threshold for passing the value into the next layer. The third layer is a pooling layer which decreases the size and resolution of its input.

These convolution and pooling layers are stacked on top of each other to increase the complexity of feature recognition. The first layer may recognize a simple line, the second layer groups of lines, and the third a collection of lines that makes the beak of a bird (see Figure 1).

Filters at each layer of convolution are trained to identify their own particular patterns.



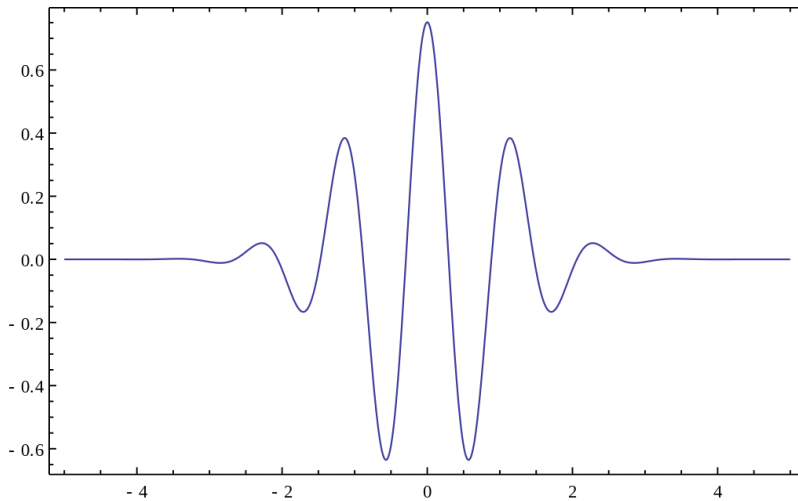
**Figure 1** – Structure of a Convolutional Neural Network – Image from Adit Deshpande (2016)

The output from an arbitrary number of convolutional layers is encoded into classifications or feature identification by fully connected layers (see Figure 1). Fully connected layers help translate the convolutional output into easier understand classifications.

Labeled data, which associate images with their classification, are required to train a convolutional network. During training, filters and layer connections are tuned to reduce the error between network output and the data label (Desphande, 2016).

Once a network is trained, new label data from the same dataset is passed through the network to test its accuracy. This testing allows the network to be validated as effective, since it has not been able to train on the testing data.

**Wavelet Analysis** is a technique for spectral analysis of a timeseries, through which frequency-based features can be extracted across time. A wavelet transformation is similar to a Fourier transformation, which can be thought of as a convolution of sine waves with a timeseries. The waves used are moved across the timeseries and stretched to different frequencies to produce a two-dimensional transformation. Wavelet analysis is very similar, but the waves it uses are localized in time (see Image 2). Although windowed Fourier transforms can provide time-localized information, the continuous wavelet transform has enhanced temporal resolution. In other words, wavelets provide a clearer picture of frequency phenomena at a more precise point in time.



**Figure 2** – Morlet wavelet – Image from Jon McLoone (2012)

Wavelet analysis can be used to characterize autocorrelation, variance, and multimodality of a timeseries, all of which are indicators of regime shifts (Rouyer et al, 2008). The technique also allows for non-stationary timeseries to be analyzed while maintaining evaluation at large timescales (Torrence & Compo, 1998).

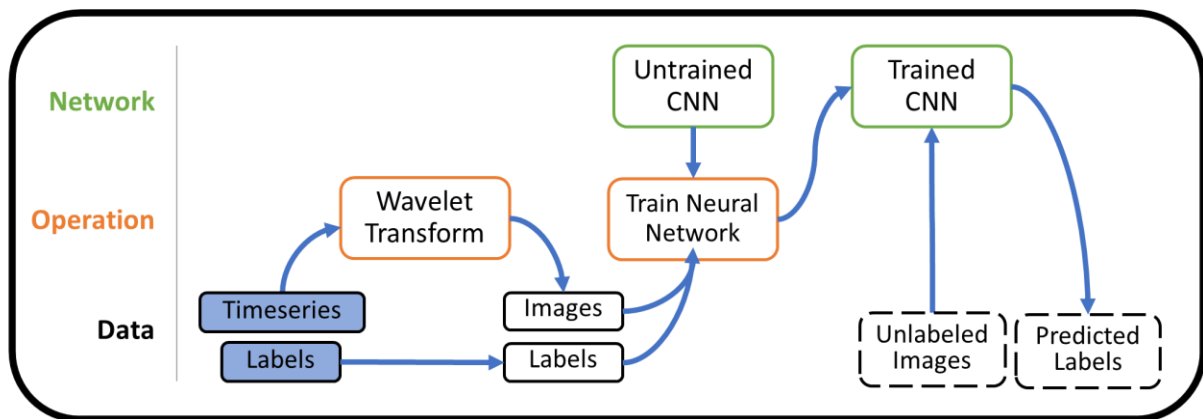
The fundamental transformation in wavelet analysis is from a one-dimensional timeseries of a variable over time to a two-dimensional image of wavelet power across time and frequency. This transformation allows for a one-variable timeseries to be analyzed with a convolutional neural network setup to process one-channel images.

### ***Design***

Our R-package is a timeseries analysis tool that employs a wavelet transform convolutional neural network (wtCNN). Timeseries are analyzed using wavelet transforms to produce images. These images are then passed through a convolutional neural network for classification. This composite approach allows for the power of CNNs to be applied to timeseries by amplifying important characteristics with wavelet transforms.

### Core Functionality

To create a trained CNN that can identify features in a timeseries, we start with a labeled timeseries and apply a wavelet transformation to produce images, still with their labels. We then use those labeled images to train a convolutional neural network. Our package is constrained to binary classifications, so the label can be either 0 or 1.



*Figure 3 – Core data flow of our package*

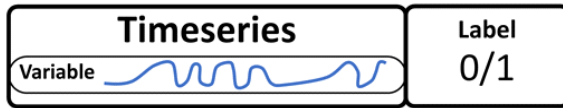
A trained CNN is the pivotal tool for monitoring ecological timeseries in the real world. Timeseries without labelled features can be wavelet transformed and put through the trained CNN to give predictions of feature presence. These lead to image classification according to how the network was originally trained.

There are two ways for the CNN to output in our R package. The first is by giving the input timeseries a label that is most likely given the training data. The second is to give probabilities of either category. This second method is useful for forecasting, which requires additional processing of the timeseries.

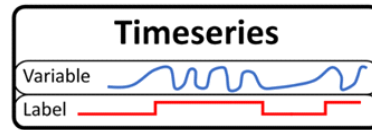


## Windowed Analysis

A: Standard Labeled Timeseries



B: Timeseries with Detailed Label



C: Windowed Analysis allows a detailed timeseries to be processed by wtCNN

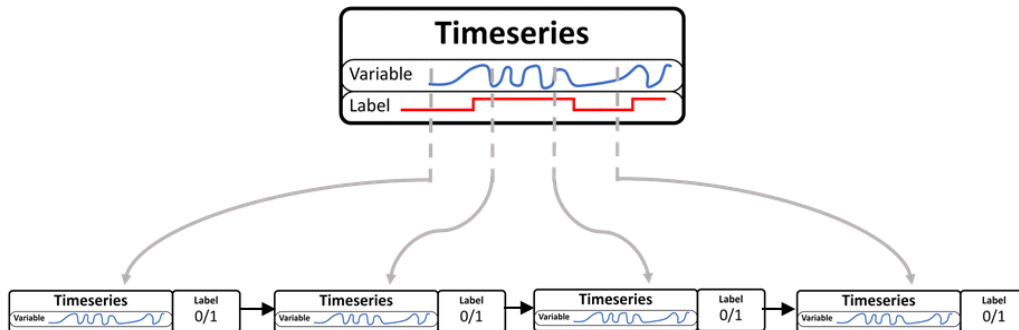


Figure 4 – Input timeseries and labeling

To process timeseries and their labels, we employ two approaches. The first is to classify the entire timeseries with a single label (see Figure 4A). This label can indicate a particular domain behavior or the presence of a regime shift, as long as there is a one-to-one relationship between the sets of labels and timeseries. In our package, labels are binary.

The second approach is what we called a **windowed analysis**, which breaks down the timeseries into smaller segments of time before processing them. These windows have a set width and stride, so they can be designed to overlap or space out by a set number of data indices.

After breaking down the timeseries, windows are sent through the same wtCNN process as the whole timeseries. In order to appropriately label the windows, each point of the timeseries must have a corresponding label before windowed analysis begins. This means that the labelled data must have a *variable* timeseries as well as a *label timeseries* of the same length (see Figure 4B). Each window is then output with a label corresponding to the majority key within its range

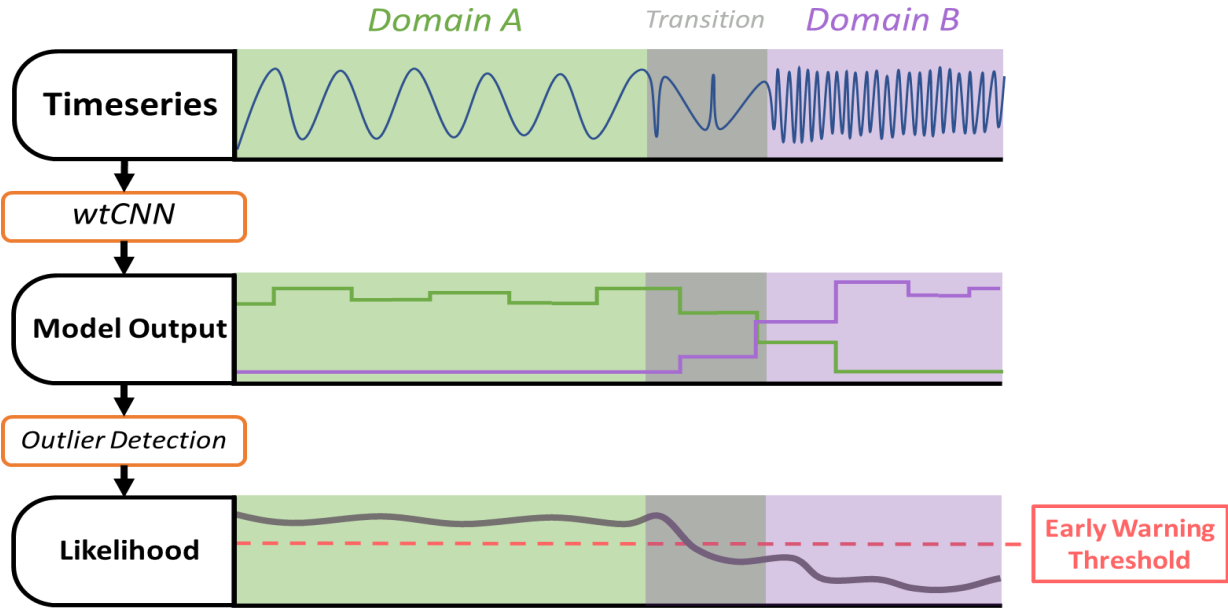
of values. With each window from a timeseries labeled with a single key, they can be processed in sequence (see Figure 4C).

Why employ windowed analysis? Windowed analysis allows a non-stationary timeseries to be analyzed as data comes in, making it ideal for real-world application. The output from the most recent window can be compared to those all previous windows to detect outliers.

*Outlier Detection and Forecasting*

By comparing model output from a current window to outputs from all previous windows, we can evaluate if the window is an outlier. If the CNN used was trained to classify domains or regime shifts, an outlier sample is a useful indicator that the system is moving out of the previous domain space.

Outlier detection works with our two-category wtCNNs by creating a two-dimensional probability distribution of all previous observations. The two dimensions are the two outputs of the wtCNN, which contain the “power” of each category/domain in an input.



*Figure 5 – an early warning system for regime shift using wtCNN and outlier detection*

Figure 5 illustrates how a timeseries is fed through a windowed wtCNN model and our outlier detection function. In this example, the wtCNN model was trained to recognize *domains*. The model output has two categories, so it is a two-value timeseries.

Because of the bimodality of features in the grey “Transition” region, the model output increases for Domain B because of the bimodality present before complete transition. This corresponds to a decrease in our likelihood function. As the transition nears completion into Domain B, the signal of that domain from our wtCNN becomes stronger and the likelihood continues to fall. This outlier detector only works because there were no other occurrences of Domain B before the transition.

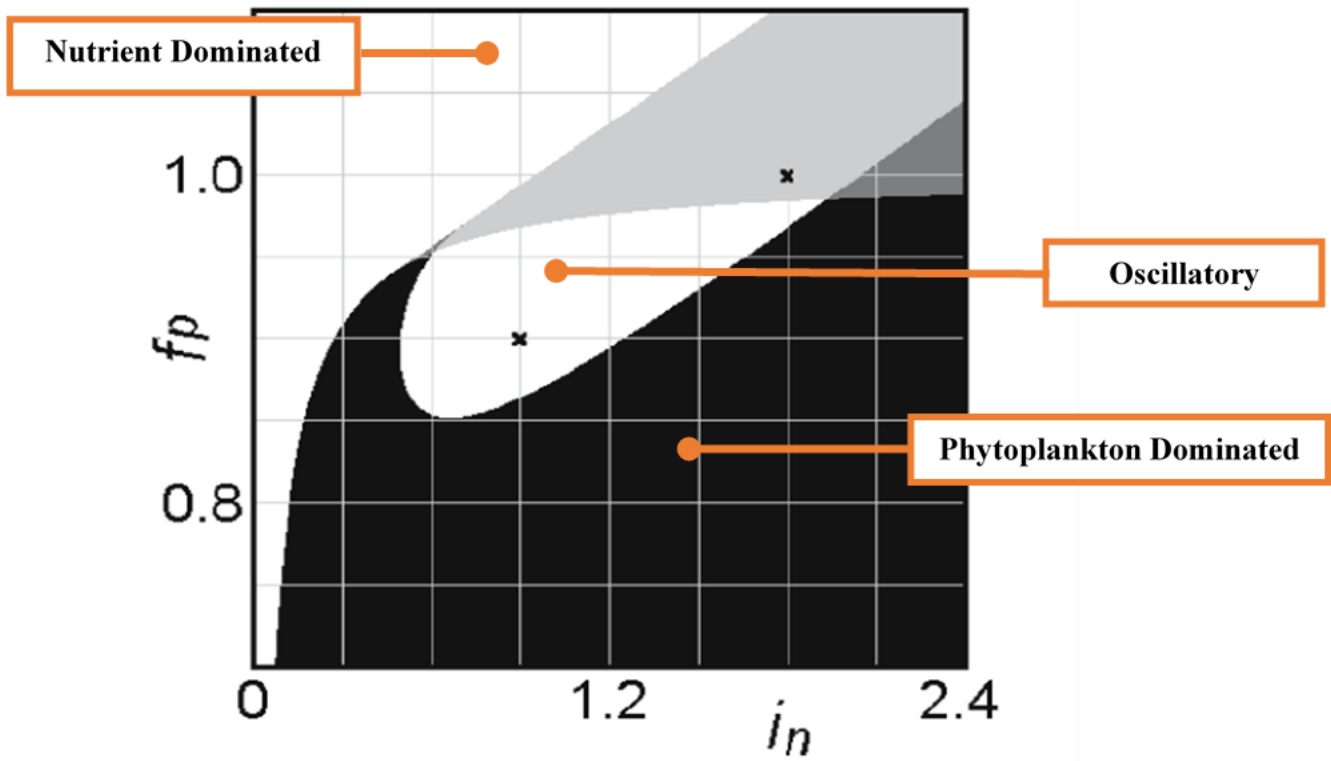
By applying a threshold to likelihood, we can make decisions about when a transition is imminent. Because of the analysis by the wtCNN, the threshold is crossed before complete transition into Domain B. In real world applications, this can provide crucial time for reversal of the transition through human intervention.

### ***Testing***

To test and develop wtCNN, we evaluated its effectiveness on simulated time series data of phytoplankton-nutrient interactions. The regime shifts present in the model are similar to those involved in harmful algal blooms. The parameters for a regime shift to occur in this model were sourced from Serizawa et al (2008).

In this model, two parameters comprised could move the system between regimes. The first  $f_p$ , is the feeding rate of zooplankton on phytoplankton. This variable was fixed for each simulation, and ranged from 0.85 to 0.95. The second variable,  $i_n$ , is the inflow of nutrients into the system. This was the value we manipulated over time to induce regime shifts. Red noise, with lag-1 autocorrelation, was added to  $i_n$  during simulation.

Each timeseries was a four-month simulation of lake ecosystem dynamics sampled by the hour in-simulation. We manipulated the nutrient inflow of each simulation differently to produce transitions at different speeds and times. In scenarios where there was no regime shift, the nutrient inflow was still manipulated, but within the range consistent with a particular domain. The three regimes of ecosystem dynamics in our simulations were nutrient-dominated, phytoplankton-dominated, and oscillatory (see Figure 6).



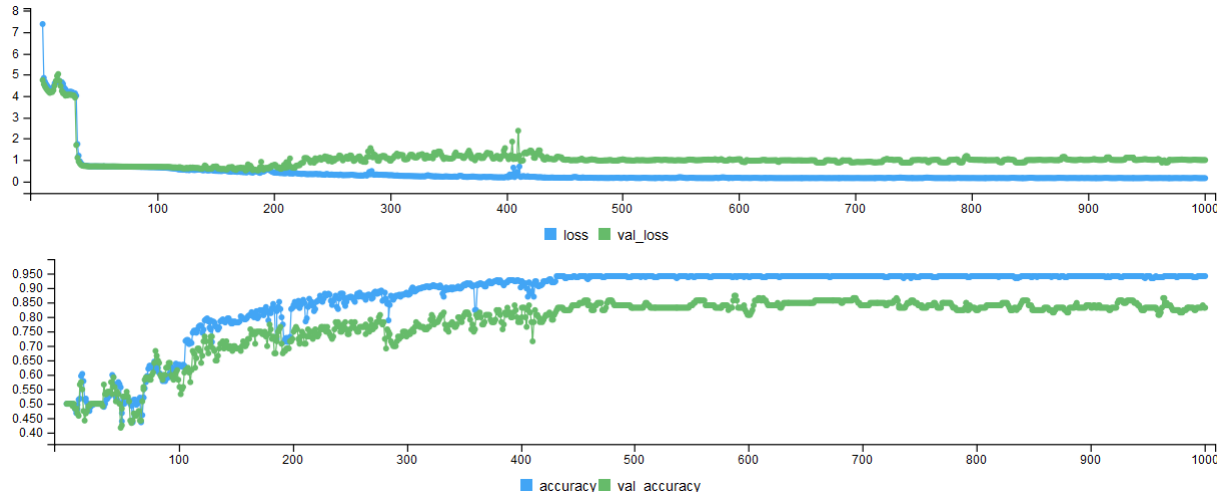
*Figure 6 – Parameter-space of lake model regimes*

		<b>Scope</b>	
		<i>Broad – Stationary Timeseries</i>	<i>Localized – Windowed Timeseries</i>
<b>Classification</b>	<i>Transition</i>	Model 1	—
	<i>Domain</i>	—	Model 2

**Table 1** – Analysis methods, blank squares indicate areas of potential future approach

We tested two methods of analysis, which differ in two respects illustrated in Figure 7. The first method, Model 1, analyzed an entire timeseries and classified it by the presence of a regime shift. For this model, at a set beginning and end time, the mean nutrient inflow would rise or fall linearly between two values. If these two values corresponded to points in different domains, the timeseries was labelled with TRUE, for the presence of a regime shift. FALSE was used to indicate there was no regime shift present.

In Model 2, we applied windows of five days with a stride of one day. Our simulated lake had an instantaneous change in nutrient inflow at a variable point in time. In each simulation, these changes corresponded to regime shifts. When the timeseries was broken into windows, each window was labelled according to which domain had the majority presence within its range. Because wtCNN is limited to binary classifications, the oscillatory and nutrient dominated domains were grouped together as FALSE. The phytoplankton-dominated domain was labelled as TRUE. This was done for two reasons. Firstly, this labelling ensures that each regime shift is a transition between a TRUE domain and a FALSE domain. Secondly, the phytoplankton-dominated domain is of greater importance to identify individually because it is associated with harmful algal blooms.



**Figure 7** – Progression of model loss and accuracy in training the CNN

Before training the CNN (as seen in Figure 7), the wavelet-transformed data set was split into a training set (blue), and a testing set (green). The testing set is meant to give the model external validity since it is data the model has not yet seen. The accuracy of the model is determined by the percentage of model classifications are correct for a labelled data set. If both the testing data and training data have high accuracies, the model is useful.

**Results**

	<i>Accuracy</i>	<i>Type I Error</i>	<i>Type II Error</i>
<b>Model 1:</b> Trained on Transitions	0.92	0.12	0.04
<b>Model 2:</b> Trained on Domains (windowed)	0.65	0.02	0.74

**Table 2** – Accuracy measures for different analysis approaches

Table 2 shows accuracy results for Model 1 and Model 2. They have been broken down into three indicators. The first, *Accuracy*, is the proportion of samples that were correctly categorized. *Type I Error* is the proportion of samples with an actual label of FALSE that were incorrectly labelled as TRUE. This would mean incorrectly claiming regime shift (Model 1) or

incorrectly claiming a phytoplankton-dominant domain (Model 2). *Type II Error* is the proportion of samples with an actual label of TRUE that were incorrectly labelled as FALSE by the model. This corresponds to missing a regime shift (Model 1) or failing to identify the presence of features of a phytoplankton-dominant domain (Model 2).

As evident by Model 1 results, wtCNN is able to identify the presence of regime shifts in simulated four-month-long timeseries with considerable accuracy. This is good evidence for the potential of the wavelet-based CNNs as applied stationary timeseries classification. This particular accuracy report was obtained from the example data set and model included in the wtCNN package. The dataset is an evenly distributed sample of 200 timeseries.

In Model 2, the model accuracy is not considerably different from random chance, 0.50. As such, the model is not much use. The results reflect a problem that occurred frequently during training. The Type I error would approach near zero while the Type II error would skyrocket. This happened because the model began classifying all training samples as FALSE, settling on a poor accuracy mid-training and hovering around it for the rest of training.

### ***Future work***

Because of Model 2's low accuracy, we were unable to test the efficacy of the timeseries forecasting technique described earlier and included in the wtCNN package. The approach requires a CNN accurately trained on a windowed timeseries dataset. Therefore, future work needs to refine our windowed analysis to improve its accuracy after training. One method for doing this could be to increase the width of the window. Model 1 worked with the entire four-month-long timeseries, and produced good results. It is possible that Model 2 was based on windows too small to capture critical features. The tradeoff of larger windows is less localized model results. It would be less clear at what point in the timeseries a new domain is present.

Another possible approach towards an accurate window-based model is by changing the classification feature from Domain to Transition (see Figure 7 again). The model would be trained on data labelled by the presence of a regime shift rather than by which domain is dominating the window. Given Model 1 also used this approach, it could be a promising new direction. Once a successful windowed model is created, we will be able to measure the effectiveness of our early warning system described earlier.

We are not sure how accurate this model, trained on simulated data, would be when tested on empirical data. It would be useful to test this and compare it to results from a model both trained and tested on empirical data. Application and testing of this wtCNN approach on real-world data would be required before application in any applied setting.



## ***References***

- Andersen T, Carstensen J, Duarte C (2008) Ecological thresholds and regime shifts: Approaches to identification. *Trends in Ecology & Evolution*. doi.org/10.1016/j.tree.2008.07.014
- Burford M, Carey C, Hamilton D, Huisman J, Paerl H, Wood S, Wulff A (2019) Perspective: Advancing the research agenda for improving understanding of cyanobacteria in a future of global change. *Harmful Algae*. doi.org/10.1016/j.tree.2019.04.004
- Carpenter S (2003) Regime shifts in lake Ecosystems: pattern and variation. *University of Wisconsin –Madison Center for Limnology*. Retrieved from limnology.wisc.edu October 15, 2019
- Carpenter S, Cole J, Pace M, Batt R, Brock W, Cline T, ... , Weidel B (2011) Early warnings of regime Shifts: a whole-ecosystem experiment. *Science* 332: 1079-1082
- Cazelles B, Chavez M, Berteaux D, Menard F, Vik J, Jenouvrier S, Stenseth N (2008) Wavelet analysis of ecological time series. *Oecologia* 156: 287–304
- Hsieh W (2007) Nonlinear principal component analysis of noisy data. *Neural Networks* Vol. 20 Issue 4 434-443
- Kidwell, D (2015) Mitigation of harmful algal blooms: The way forward. *PICES Press* Vol 23. No. 2: 22- 24
- Ontario Ministry of the Environment and Climate Change (2017) Canada-Ontario Lake Erie action plan: partnering on achieving phosphorus loading reductions to Lake Erie from Canadian sources. Retrieved October 16, 2019 from [www.canada.ca/en/environmentclimate-change/services/great-lakes-protection/action-plan-reduce-phosphorus-lakeerie.html](http://www.canada.ca/en/environmentclimate-change/services/great-lakes-protection/action-plan-reduce-phosphorus-lakeerie.html)

Pace M, Batt R, Buelo C, Carpenter S, Cole J, Kurtzweil J, Wilkinson G (2016) Reversal of a cyanobacterial bloom in response to early warnings. *Proc. Natl. Academy of Sciences* 114: 362-257

Scheffer M, Carpenter S, Lenton T, Bascompte J, Brock W, Dakos V, ... , Vandermeer J (2012) Anticipating Critical Transitions. *Science* 338: 334-348

Stafford R, Smith V, Husmeier D, Grima T, Guinn B (2013) Predicting ecological regime shift under climate change: New modelling techniques and potential of molecular-based approaches. *Current Zoology* 59(3): 403-417

Wilson R, Beetstra M, Reutter J, Hesse G, Fussel K, Johnson L, ..., Winslow C (2019) Commentary: Achieving phosphorus reduction targets for Lake Erie. *Journal of Great Lakes Research* 45: 4-11