

Machine learning with a snapshot of data: Spiking neural network ‘predicts’ reinforcement histories of pigeons’ choice behavior

Anna Plessas¹ , Josafath I. Espinosa-Ramos¹, Dave Parry¹, Sarah Cowie², and Jason Landon¹

¹Auckland University of Technology, New Zealand

²University of Auckland, New Zealand

An accumulated body of choice research has demonstrated that choice behavior can be understood within the context of its history of reinforcement by measuring response patterns. Traditionally, work on predicting choice behaviors has been based on the relationship between the history of reinforcement—the reinforcer arrangement used in training conditions—and choice behavior. We suggest an alternative method that treats the reinforcement history as unknown and focuses only on operant choices to accurately predict (more precisely, retrodict) reinforcement histories. We trained machine learning models known as artificial spiking neural networks (SNNs) on previously published pigeon datasets to detect patterns in choices with specific reinforcement histories—seven arranged concurrent variable-interval schedules in effect for nine reinforcers. Notably, SNN extracted information from a small ‘window’ of observational data to predict reinforcer arrangements. The models’ generalization ability was then tested with new choices of the same pigeons to predict the type of schedule used in training. We examined whether the amount of the data provided affected the prediction accuracy and our results demonstrated that choices made by the pigeons immediately after the delivery of reinforcers provided sufficient information for the model to determine the reinforcement history. These results support the idea that SNNs can process small sets of behavioral data for pattern detection, when the reinforcement history is unknown. This novel approach can influence our decisions to determine appropriate interventions; it can be a valuable addition to our toolbox, for both therapy design and research.

Key words: choice research, artificial intelligence, spiking neural networks, reinforcement history, machine learning prediction

An analysis of the interactions between behavior and with the environment provides us with information about reinforcement contingencies (Skinner, 1969). Reinforcement contingencies have been extensively studied in choice research and models developed to understand how behavior changes in relation to environmental factors (Davison & McCarthy, 1988; Pierce & Epling, 1983; Skinner, 1969; Staddon, 2016). Extensive research has shown that this relationship can be complex; choice is dependent on recent reinforcement history, such that responses that have in the past produced relatively more reinforcement occur relatively

more often (Baum, 1974; Davison & Jenkins, 1985; Davison & McCarthy, 1988; Killeen, 1972). These effects are observed both when environmental contingencies remain stable over time (e.g., Baum, 1974), and also when environmental conditions change rapidly (Davison & Baum, 2003; Landon et al., 2003; Mazur, 2016). Indeed, even when a reinforcer ratio remains in effect for just 10 reinforcers, choice varies systematically with the arranged reinforcer ratio (Davison & Baum, 2000). Thus, behavior is highly sensitive to recent reinforcers, and functional relations learned in the past contribute to current behavior in a way that allows prediction of future responses under similar circumstances.

The relation between behavior allocation—for example, when two keys are present—and the relative reinforcement arrangement, would suggest that either can be predicted from the other. Given the fundamental relationship between choice and reinforcement, different predictive questions can be asked

Address correspondence to: Anna Plessas, Department of Psychology, The Auckland University of Technology, 90 Akoranga Drive, Northcote, Auckland 0627, New Zealand (email: anleplka@gmail.com)

This is an open access article under the terms of the [Creative Commons Attribution-NonCommercial-NoDerivs](https://creativecommons.org/licenses/by-nc-nd/4.0/) License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

doi: 10.1002/jeab.759

about the effects of variables influencing the behavior of individual organisms. In behavior analysis, a traditional approach would be to predict the next distribution of responses when the reinforcer ratio is set as the independent variable (Baum, 2018). This informs us on how reinforcers obtained in a particular context, over successive instances, can lead to a change in behavior. An alternative approach to prediction is to speculate about the reinforcement history (i.e., the reinforcement arrangements used in training conditions and no longer in place) of a behavior by looking only at current behavior. This approach allows us to identify the reinforcement history when such information is not available and informs us about the conditions in which new behavior can be learned. The latter bears special significance for both applied and experimental contexts. Especially in applied settings, the reinforcement history is commonly unknown, and is speculated on in the development of assessments and interventions, but otherwise, there is little attention to the matter (Pipkin & Vollmer, 2009). Different schedules of reinforcement maintain different patterns and rates of responses, and reinforcers are variables that play a role in current behavior. However, these effects are still under investigation as earlier reinforcement history plays a role in present performance (Freeman & Lattal, 1992; Okouchi & Lattal, 2006). Recent research also supports the idea that the effects of reinforcement history were not only apparent but can reappear when the reinforcement contingency changes (Okouchi et al., 2014). Accurately identifying reinforcement history can improve our decision-making process when designing reinforcement arrangements to change behavior. To differentiate among distinct types of predictive questions, from now on we will use the term *retrodiction* to denote predicting the reinforcement history (i.e., to identify past concurrent variable-interval (VI) schedules with various reinforcer ratio arrangements).

In rapidly changing procedures, reinforcer ratios across two alternatives change multiple times in a session in an unpredictable manner. This procedure simulates natural environments to some degree, and results in behavior that changes quickly in response to the changing contingencies (Davison & Baum, 2000, 2002; Landon & Davison, 2001). In the

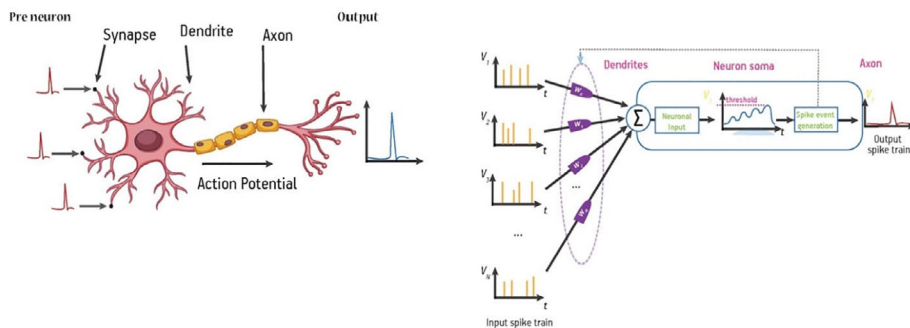
analysis of data from frequent changing procedures, large amounts of data are aggregated within and across sessions over a large period allowing for the analysis of extended patterns of behavior (Baum, 2002). In this research, we wanted to investigate whether retrodiction could be possible without the need to aggregate the data and when a limited sample of behavior was available. Therefore, we used a minimal amount of data to resemble more limited observational data from a naturalistic setting (i.e., 5-s periods postreinforcer). In this manner, we assessed the viability of approaching observed behavior and relevant processes in a way that might be attainable in both experimental and natural settings in an attempt assess whether a relationship is detectable using this small sample (5 s) of data.

Five second periods after the delivery of a reinforcer were chosen because of the close local proximity responses have to the reinforcer. Local effects of reinforcers, meaning the effects reinforcers have in specific locations in time, have proven to be strong and research has demonstrated individual reinforcers have short-term effects on momentary behavior (Davison & Baum, 2002, 2003). Local effects are evident in both stable and rapidly changing environments, and occur in the context of a more global shift toward the richer response alternative e.g., (Landon et al., 2002, 2003). Localized control by reinforcers suggests that even behavior recorded from a small interval of time should be sufficient to allow retrodiction.

Machine learning (ML) algorithms have proven to successfully complete complex tasks in only a few seconds and this advantage of ML has attracted attention across scientific domains. One subset of ML uses models inspired by neuroscience, with algorithms that simulate the properties of neurons and neural networks. Such artificial neural networks (ANN) have been used extensively to model and understand normal and abnormal brain function (Macpherson et al., 2021), as well as cognitive tasks including perception and decision-making (Zador, 2019). Although ANN are widely used in healthcare (Shatte et al., 2019), education (Korkmaz & Correia, 2019), and other fields such as speech recognition (Bhangale & Mohanaprasad, 2021) or image recognition (Cai et al., 2020), their application in behavior analysis remains limited (Turgeon & Lanovaz,

Figure 1

Left: the Structure of a Biological Neuron. Right: a Sample of a Brain-Inspired Computational Neuron (Chen et al., 2018)



2020). Thus, in the present study we asked to what extent ANN can identify reinforcement history based on current behaviors.

Although EAB has been hailed for its powerful experimental designs and identification of learning patterns, some argue that our scientific work has isolated itself (Poling, 2010; Vyse, 2013) from a world of modern technologies pervading almost all human activities. The aim of this investigation is to stimulate a discussion on how to approach data differently (i.e., using a snapshot of data) and rely on observed behavior (i.e., retrodiction; direct reverse test) and relevant processes rather than manipulating the reinforcer–behavior relation. This novel approach can influence our decisions to determine appropriate interventions or aid knowledge over a shorter time frame. Thus, this approach might lead to dealing with behavior in alternative ways that the behavioral community desires (Shahan, 2017).

Introduction to Brain-Inspired Neural Networks for This Study

ANN is a subdivision of machine learning (ML) software inspired by how the brain processes information and solves complex problems. Artificial spiking neural networks (SNNs) are termed *the third generation*¹ ANNs due to their ability to provide biologically

plausible neuronal models that capture some of the complex temporal dynamics of the data (see Gerstner & Kistler, 2002, for more on SNNs). A computational spiking neuron (also known as an artificial biological neuron model) produces spikes—discrete events that take place at points in time, coming from one neuron to another and connected through weights (Maass, 1997). This ability to work with discrete events that occur at separate times corresponds well with measures of choice data. Information transfer in artificial spiking neurons mimics the way information is transferred in the biological neuron by considering the exact time of the spike or the sequence of the spike (Fig. 1). As presented in the example, the model learns by using artificial biologically plausible algorithms (Σ) feeding the information forward across the neural network, starting from the input node to the output node. The weights represent the synaptic connections of neurons and refer to the strength of those connections, that is, the effect of the firing rates. Based on weights, the input signal may be amplified or inhibited. When the membrane potential reaches a certain threshold, the neuron will spike, generating a signal that will travel and instantly reset to a lower value. The neuron stays in a resting state for some time (the *absolute refractory period*), after which it can process new information coming from other presynaptic neurons (Gerstner & Kistler, 2002). Thus, this process attempts to resemble the natural nervous system and is considered as biologically realistic and plausible (Maass, 1997). Further, this process eliminates the need for an averaging time window and allows to process information in

¹ANNs can be classified into three generations according to their computational units (Maass, 1997): McCulloch-Pitts neurons, also called perceptrons or threshold gates, activation functions (sigmoid) and spiking neurons (“integrate-and-fire-neurons”).

continuous time. In essence, artificial SNNs' strength is that the networks learn from spatiotemporal data.

In ANN 'learning' refers to the process of extracting structure from the data that will be encoded in the parameters of an artificial network and will provide all the information needed to develop the artificial network (Zador, 2019). In other words, all ANN explicitly learn from the datasets. There are three learning methods commonly used to train an artificial neural network: supervised, unsupervised, and reinforcement learning. In this study we employed a semisupervised ML method combining supervised and unsupervised learning methods. With supervised learning the ML task is to employ the same label to the input/output set as with the training dataset. Data consists of pairs: an input item (for example, the choice made by a pigeon) and its label (e.g., the concurrent-VI schedule to the learned behavior). By training the ML algorithm, it searched for patterns on a 'labeled' dataset. After this training, any new inputs are cross-matched with the training dataset to determine the desired output. By this, the learning algorithm can deduce a pattern by identifying a relationship between the target variable and the rest of the dataset based on the information it already has. On the other hand, with unsupervised learning the training set contains no 'labels' in the data and the algorithm learns without needing to supervise the model (Rafi, 2021).

SNNs can be an appropriate tool for modeling behavioral data for several reasons. Firstly, choice and reinforcers are complex variables that constantly interact with each other in space and in time (Cowie & Davison, 2016). We hypothesized that an ANN like SNN can be beneficial since its advantage resides in handling spatiotemporal information. Secondly, SNNs lend themselves to the small-N approach that characterizes work in the field; by extracting sufficient amount of data from fewer individuals (i.e., six pigeons) rather than the reverse (i.e., small amount of data accumulated across a large number of individuals) can add value to the prediction accuracy of the model when it is applied to a single organism.

Our aim was to investigate whether a machine-learning algorithm capable of processing spatiotemporal information can identify reinforcement history when no

previous knowledge about this history is provided. Retrodicting histories of operant behavior may open exciting avenues in the field of behavior analysis. From a practical standpoint, a retrodictive outcome would be possible based solely on limited data (e.g., 5-s periods). This could parallel situations where laboratory resources are limited or when clinicians need to decide on the appropriate intervention having only limited observational data available. Also, behavior analysts could have an additional alternative tool for studying the relation between past reinforcement arrangements and current behavior when generating knowledge.

Method

Dataset for Training and Testing the Artificial Spiking Neural Networks

The dataset was extracted from Landon and Davison's (2001) study in which a frequently changing concurrent-schedule procedure arranged seven different reinforcer ratios, in *components* (1:27, 1:9, 1:3, 1:1, 3:1, 9:1, or 27:1 – see Table 1) which changed randomly within each session. The overall reinforcer rate was constant, and each component was in effect for 10 successive reinforcer deliveries, and components were separated by 10-s blackouts. The time and nature of all experimental events were recorded (for full details, see Landon & Davison 2001). We extracted the raw data for all six pigeons in all 50 training sessions from one experimental condition (Condition 1). We then created an extraction rule to use a minimal amount of data. All choice responses made in the first 5 s after every reinforcer were extracted as a frequency event and represented as discrete events in time (i.e., temporal data). All pigeons' choices were taken after the delivery of the first nine reinforcers within each component.

Extraction Rule of the Data for SNN Training

All files originally had a ".txt" extension and were generated by MED-PC[®] software. All files used in this study were programmed by the original experimenters to record all experimental events and the time at which every event occurred within the experimental conditions. To extract a snapshot of data from the

Table 1

The Relative Reinforcer Probability (Shown as Probability of Reinforcement to the Left Alternative) for Each of the Seven Concurrent VI Schedules (Referred to as Components) for Both Conditions

Component	Reinforcer Ratio (Left: Right)
1	1:27
2	1:9
3	1:3
4	1:1
5	3:1
6	9:1
7	27:1

Note. The overall probability of reinforcement per second was constant at 0.037.

whole dataset of Condition 1, the second author developed a computer code in Java. When the code was run by the first author, *samples* in “.csv” format were created, of pigeons’ left- and right-preference responses following each reinforcer delivery, during a 5-s period. Each sample consisted of a single component and was organized of aggregate data from 10 daily sessions. This resulted in a total of five samples for each bird out of all 50 sessions. Each sample contained 90 periods of 5 s as nine reinforcers were given each day (9 * 10 daily sessions). Thus, from each pigeon we took a total of 35 (7 conditions* 5 ten-daily sessions) samples, and all pigeons together generated 210 (6*35) samples. Each row in each sample represented a frequency event, which was created by calculating the ratio of left and right preferences based on the time window (5 s) and on the actual location (L or R) where the reinforcer was delivered (see Appendix A for an example).

This approach to samples gave the opportunity to allow the assessment of the explicit use of pigeons’ temporal data, and the actual location where they occurred in time (L or R) was sufficient to learn to detect patterns without any need to add further information.

Dataset for Further Generalization Tests of the Artificial Spiking Neural Networks

For generalization testing, we extracted three new datasets from the same pigeons, this time from a different experimental condition (Condition 6) of the original study (Landon & Davison, 2001). The six pigeons were exposed to Condition 6 for 35 sessions after the

completion of experimental Conditions 1-5. The same environmental arrangements as in Condition 1 were set for Condition 6. We aggregated data from the 35 sessions in three separate ways—we first created samples each containing datasets from 10 sessions (resulting in three samples per bird for each component); then we created samples each containing datasets from seven sessions; and finally, samples containing datasets from five sessions each. We did so in order to assess the model’s dependence for generalization on individual dataset size. Temporal data were taken, as in the training phase of the model, in the first 5 s after the delivery of the first nine reinforcers. Thus, from the 10-session samples we created dataset G1, where the data were aggregated from ninety 5-s periods (9 reinforcers * 10 sessions) in each sample; from the seven-session samples we created dataset G2, where the samples contained data from 63 periods (9 reinforcers * 7 sessions); and from the five-session samples we created dataset G3, where each sample included 45 periods (9 reinforcers * 5 sessions). All in all, the G1 dataset included 126 samples (6 pigeons * 7 components * 3 samples per pigeon). An equal number (126) of samples resulted from the data extraction for G2 because the data did not generate more than three samples per pigeon. Lastly, for the G3 dataset, the number of samples was 294 due to the samples containing only forty-five 5-s periods each. For a summary of all datasets extracted for training and testing the SNN model, see Table 2.

Artificial Single Spiking Neural Network Architecture Constructed for this Study

We created an artificial SNN based on the architecture initially proposed in Vazquez & Cachón, (2010) to identify the reinforcement history (i.e., frequently changing concurrent-schedule procedure arranged seven different reinforcer ratios, in components) based on pigeon choices and the firing rates these data produced. This was achievable using the Leaky Integrate-and-Fire (LIF) model, a mathematical representation of a neuron that was trained to perform the task. The LIF model is commonly used due to its simplicity and computational efficiency while attempting to mimic biology (see, e.g., Ahmed, 2020, for more on neuron models).

Table 2

Samples Extracted by the Pigeons' Temporal Data

Experimental Condition	Datasets	No of Pigeons	Points of 5-sec periods
Cond. 1	Training the model	5 Pigeons	90 periods
	Testing model	1 Pigeon	90 periods
Cond. 6	Generalization-1	6 Pigeons	90 periods
	Generalization-2	6 Pigeons	63 periods
	Generalization-3	6 Pigeons	45 periods

Note. Some samples consisted of <90 points as some of the 10 daily sessions ended at the prearranged time (45 min), and the pigeon had not consumed all reinforcers.

Figure 2 depicts a simplified schematic representation of the single SNN architecture (more technical details are provided in supplementary materials). The inputs were samples that represented discrete events of left and right choices based on the time window (5-s) and on the actual location (L or R) where the reinforcer was delivered (see extraction rule of data for SNN for Training). The output of the SNN was the firing rates associated with the seven components presented in Table 1 (multiclassification problem).

These samples were transformed into a vector of values that simulates the electrical current that is injected to the LIF neuron model. Each sample produced a spike train (sequence of ones and zeros) that was transformed into a firing rate. To achieve transformation of the spiking data, the information was encoded in the number of spikes over a specified temporal window (*rate encoding*) (see, e.g., Auge et al., 2021, for more on encoding processes). This produced a sequence of artificial spikes with a specific firing rate. Then each component was associated with a specific firing rate.

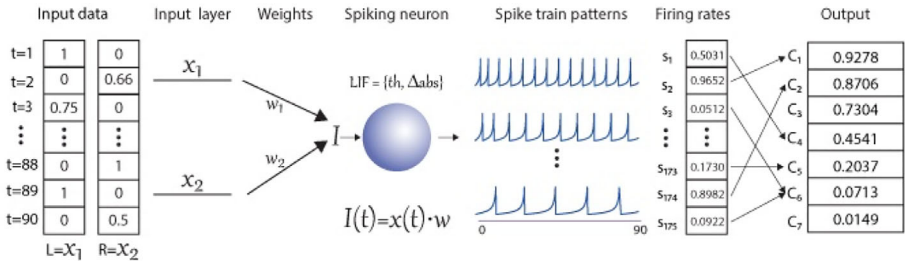
It was hypothesized that the input signal produced by samples of the same component produce similar firing rates, whereas input currents of different components produce firing rates different enough to discriminate among the various components. We used JNeuCube to develop the proposed artificial single neuron model and performed the experiments of this study. JNeuCube is a Java-based framework for building SNNs able to solve classification and prediction problems (<https://github.com/Auckland-University-of-Technology/NeuCube-java>). Specific parameters were set to train the artificial neuron to correctly perform the requested task (see supplementary materials).

Training Procedure of the SNN Model

We chose a Differential Evolution (DE) algorithm due to its effectiveness to investigate how we can understand patterns in our data (Storn & Price, 1997). It is a kind of decision-making tool, where decisions are made to optimize one or more objectives under

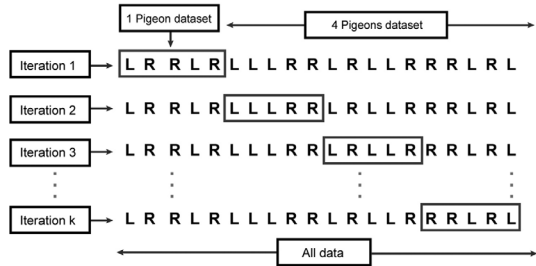
Figure 2

The SNN Architecture Constructed for Classification of Seven Components Based on Pigeons' Choices



Note. The letters in the figures represent: 90 periods of 5 s as t; left choice responses as X1; right choice responses as X2; left and right weights as w1-w2; s1....s175, as samples 1-175; C1.C7 as Components 1-7.

Figure 3
A Schematic Representation of K-Fold Cross-Validation Training When Using Right and Left Responses (Pigeons' Choices)



specific circumstances. We used a cross-validation procedure to evaluate the model's performance by using pigeon datasets for training the model and then for validating its performance on a testing dataset that the model had not encountered before. We trained the model applying the stratified k-fold cross-validation ($k = 5$) on the training dataset (five pigeons) and evaluated its performance using the testing dataset (the sixth pigeon). In the five-fold cross-validation, the data were partitioned into five sets of equal sizes that were randomly used, then the model was trained using $k-1$ folds and estimated its own postdictive ability using 1-fold (Fig. 3). The process was repeated five times, with a different fold for training ($k-1$) and testing selected each time, creating different accuracies each time. This strategy allowed for an objective, less biased and less optimistic estimation of the model's performance than other methods (James et al., 2013).

Table 3
Performance Metrics Used in this Study to Interpret Results and Formulas Used for Calculations

Measure	Description	Formula
Accuracy	The fraction of correctly predicted events in relation to all data	$\frac{\text{\#of samples predicted as having the event}}{\text{\#of total samples}}$
Recall (or sensitivity)	The proportion of correctly predicted actual events (i.e., a true positive) in reference to the total true events	$\frac{\text{\#of samples predicted as having the event}}{\text{\#of samples with the event of interest}}$
Specificity	The proportion of correctly predicted nonevents (i.e., a true negative) in reference to the total nonevents	$\frac{\text{\#of samples predicted as non events}}{\text{\#of samples without the event of interest}}$
Informedness	The probability of an informed decision (or Youden's index)	$J = \text{Recall} + \text{Specificity} - 1$
Precision	The fraction of correctly predicted actual events in reference to retrieved events	$\frac{\text{\#of samples predicted as events}}{\text{\#of samples predicted as true+false positives}}$
F1	Weighted average of precision and sensitivity	$2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$

With the completion of cross-validation, we evaluated the model's generalization ability by testing it on the data of another pigeon unknown to the model. The outcome was 40 single artificial SNNs that could accurately classify components based on a pigeon's choices. From all these we kept the fittest as the final result. We repeated the experiment 10 times and created 60 artificial SNN models by splitting the total of six pigeons' data from Condition 1 to five pigeons' data for training and one for testing each SNN model. With this method, all possible combinations of pigeons' data were used both for training and for testing for generalization.

Procedures

Model Performance Evaluation

We calculated five different measures to assess our model's performance. The most common method to evaluate model performance is prediction accuracy (Table 3). However, it has been proposed that when considering using the output for clinical decisions, additional metrics should be taken into account, as accuracy does not take into consideration other characteristics of the data (Kuhn & Johnson, 2018). Thus, going beyond accuracy metrics, we tested the model's ability to identify true positives and negatives assessed by recall (or 'sensitivity') and specificity values. An effective predictive model that can classify components based on choice behavior should be able to discriminate events from nonevents. Responses in each component should create

Table 4
The Overall Results of the Six Best Models for all Combinations

Outcomes	Pigeon 61 Overall	Pigeon 62 overall	Pigeon 63 overall	Pigeon 64 overall	Pigeon 65 overall	Pigeon 66 overall
Accuracy	0.96	0.96	0.96	0.96	0.94	0.93
Recall Specificity	0.87	0.87	0.87	0.86	0.81	0.81
Informedness	0.98	0.98	0.97	0.97	0.96	0.96
Precision	0.85	0.85	0.84	0.83	0.77	0.77
F1	0.87	0.87	0.87	0.86	0.81	0.81
	0.87	0.87	0.87	0.86	0.81	0.81

Note. The results reflect cross-validation (CV) training and generalization testing for validation. The results are listed per pigeon used for testing generalization.

unique patterns that differentiate choice behavior in one component from choice behavior in another component. In the original study (Landon & Davison, 2001), pigeons' responses in a component were significantly affected by the reinforcer ratio and the sequential effects of reinforcers. Having a model that performs equally well in recall and specificity is critical in terms of matching response ratios to reinforcer ratios in a component in conformity with the generalized matching law (GML). Additional metrics were also calculated to measure performance. See Table 3 for some critical notions used when measuring performance of artificial neurons.

Response-By-Response Analysis Based on the Time Window (5-s)

We re-analyzed the data from Landon & Davison (2001) within a 5-s window to investigate the ability of the learning algorithms to identify patterns in smaller samples by comparing the model's output with the pigeons' actual performances.

Results

SNN Performance Metrics in Modeling Learning Histories

Analysis of the Overall Performance of the Classification Model

Overall, all combinations of pigeon datasets were able to detect the reinforcement history from current operant choices with a good degree of success. The artificial SNN models correctly identified the components based on the pigeons' choice responses, ranging from 93% (Pigeon 66 – data used for testing) to 96% (Pigeons 61 to 64) correct identification.

Overall specificity performance measures were higher ($\geq 96\%$) than recall, illustrating that all artificial SNN models were better at classifying which component a choice response does not appertain to (specificity) than identifying in which component choice learning took place (recall). We combined these measures to estimate informedness, that is, error magnitude in recall and specificity, as presented in Table 4. This index had values ranging from 77%-85%, indicating only small errors in the SNN model performance. The index was highest when Pigeon 61 was used for testing and dropped slightly with Pigeon 66. The precision results revealed a similar pattern of recall and informedness metrics (Table 4). These results reveal that only a few events that should have been predicted as events were not. Lastly, F1 calculated as a weighted average of precision and recall score verified the same results. Thus, detection of patterns in choice responses with a small window is possible with SNNs. Overall, the results demonstrated high accuracy in the models' performance when making decisions regarding components and choice responses.

Analysis of the Classification Models' Performance Per Component

An additional analysis was conducted to examine the models' performance in correctly detecting unique patterns in pigeon responses in each individual component. The number of errors the algorithm made in identifying the actual component for each pigeon dataset was calculated and is shown in Figure 4. The two indicative models presented in the Figure show that most errors in identifying the component occurred when choice responses

Figure 4

A Normalized Confusion Matrix Across all Seven Components with the Horizontal Line Representing the Retrodicted Component and the Vertical Line the Actual Component

	C1	C2	C3	C4	C5	C6	C7
C1	76%	17%	7%	0%	0%	0%	0%
C2	23%	60%	17%	0%	0%	0%	0%
C3	0%	3%	94%	3%	0%	0%	0%
C4	0%	0%	3%	94%	3%	0%	0%
C5	0%	0%	0%	0%	93%	7%	0%
C6	0%	0%	0%	0%	0%	83%	17%
C7	0%	0%	0%	0%	0%	7%	93%

	C1	C2	C3	C4	C5	C6	C7
C1	80%	17%	3%	0%	0%	0%	0%
C2	3%	57%	13%	0%	0%	0%	0%
C3	0%	7%	90%	3%	0%	0%	0%
C4	0%	0%	3%	94%	3%	0%	0%
C5	0%	0%	0%	0%	90%	10%	0%
C6	0%	0%	0%	0%	0%	83%	17%
C7	0%	0%	0%	0%	0%	3%	97%

Note. The figure presents a table that is used to describe the performance of a model (*confusion matrix*). In this study the performance of the model was evaluated by assessing the number of choice responses retrodicted as learned in a component, and those that were not. For example, in line 1, 76% of choices were retrodicted as learned in Component 1, yet 17% were retrodicted as learned in Component 2 though in Component 1, and 7% as in Component 1 though learned in Component 1. From the confusion matrix we compute the rates of errors, true positives, negatives etc. The diagonal elements represent the recall values. A normalized matrix represents all components as having 1.0 samples (or 100%). The seven components are abbreviated as C1 to C7.

were trained under Component 2 (1:9 reinforcer ratio). Low recall in Component 2 was observed across pigeon datasets contrasting the classification rate for other components. Furthermore, the models' ability to identify patterns in the data was overall higher with Component 3, Component 5, and Component 7 than with Component 1, Component 2, and Component 6. Overall, these results suggest that differences in metrics presented in Table 4 were dependent on the component, as any errors in the retrodictive ability of the model were more prominent with specific components (i.e., Component 2). The phenomenon might have a specific explanation pertaining to the training environment in Component 2. The retrodictions made by our model were analyzed further to investigate whether reduced performance was due to the learning algorithm, or it reflected the actual pigeons' performance.

Analysis of the Model Performance Per Individual Pigeon Dataset

We examined the effects of individual pigeon datasets on the artificial SNN model by analyzing the machine metrics when splitting the metrics for training and testing data. The model handled individual datasets equally well (Table 5). This indicates that the generated models performed highly in classifying pigeon responses by component, showing high recall to the relevance of the data (> 82%) and

specificity (> 96%) in identifying true negatives (nonevents). The precision of the model was also high (> 82%), showing that the overall analysis was relevant to the whole data set.

Pigeon Performances Compared to SNN Model Performances

To examine the differences in the retrospective ability of the artificial SNN model, in particular with Component 2, we reanalyzed the extracted pigeon datasets (within a 5-s window following the delivery of a reinforcer) in each component from Landon & Davison's (2001) study. For each dataset, pigeons' actual choices were aggregated according to the left-key over right-key response ratios by conducting a response-by-response analysis of pigeon's choices; logarithms of these ratios were calculated and plotted as a function of each sequential response of the pigeons. When responses were occurring only on one alternative, the log(L/R) was set as 3.5 to indicate the exclusive direction of the pigeon's choice. Our analysis revealed that birds' responses in Component 1 (1:27) and Component 2 (1:9) followed a similar pattern; choices seemed more extreme than in other components (Fig. 5).

Further, when we extended our analysis to all pairs of responses, as accumulated within the sequence of responding, a similar pattern was observed (Fig. 6) and became more prominent from the third response on. Across

Table 5

The Results of the Best Performing Model for all Combinations when Splitting the Data

	Pigeon 61		Pigeon 62		Pigeon 63		Pigeon 64		Pigeon 65		Pigeon 66	
	CV	Test	CV	Test	CV	Test	CV	Test	CV	Test	CV	Test
Accuracy	0.94	0.97 0.91 0.98	0.95	0.97 0.91 0.98	0.94	0.97 0.91 0.98	0.95	0.96 0.89 0.98	0.95	0.92 0.77 0.95	0.96	0.91 0.74 0.94
Recall	0.82	0.79	0.83	0.90	0.82	0.90	0.83	0.87	0.85	0.72	0.87	0.69
Specificity	0.96	0.91	0.97	0.91	0.96	0.91	0.97	0.88	0.97	0.77	0.97	0.74
Informedness	0.79	0.90	0.80	0.91	0.79	0.91	0.80	0.88	0.82	0.77	0.84	0.74
Precision	0.82		0.83		0.82		0.83		0.82		0.87	
F1	0.83		0.83		0.82		0.83		0.85		0.87	

Note. The results reflect cross-validation (CV) training and testing for validation.

all combinations of constructed models, Component 2 was confused with Component 1 more. The analyses in Figures 5 and 6 suggest that the artificial SNN model was unable to find distinctive patterns between Components 1 and 2. Those two components had similar patterns; in this sense, the model accurately reflected pigeon responses in those particular training environments.

SNN Model Performances in Extended Generalizations

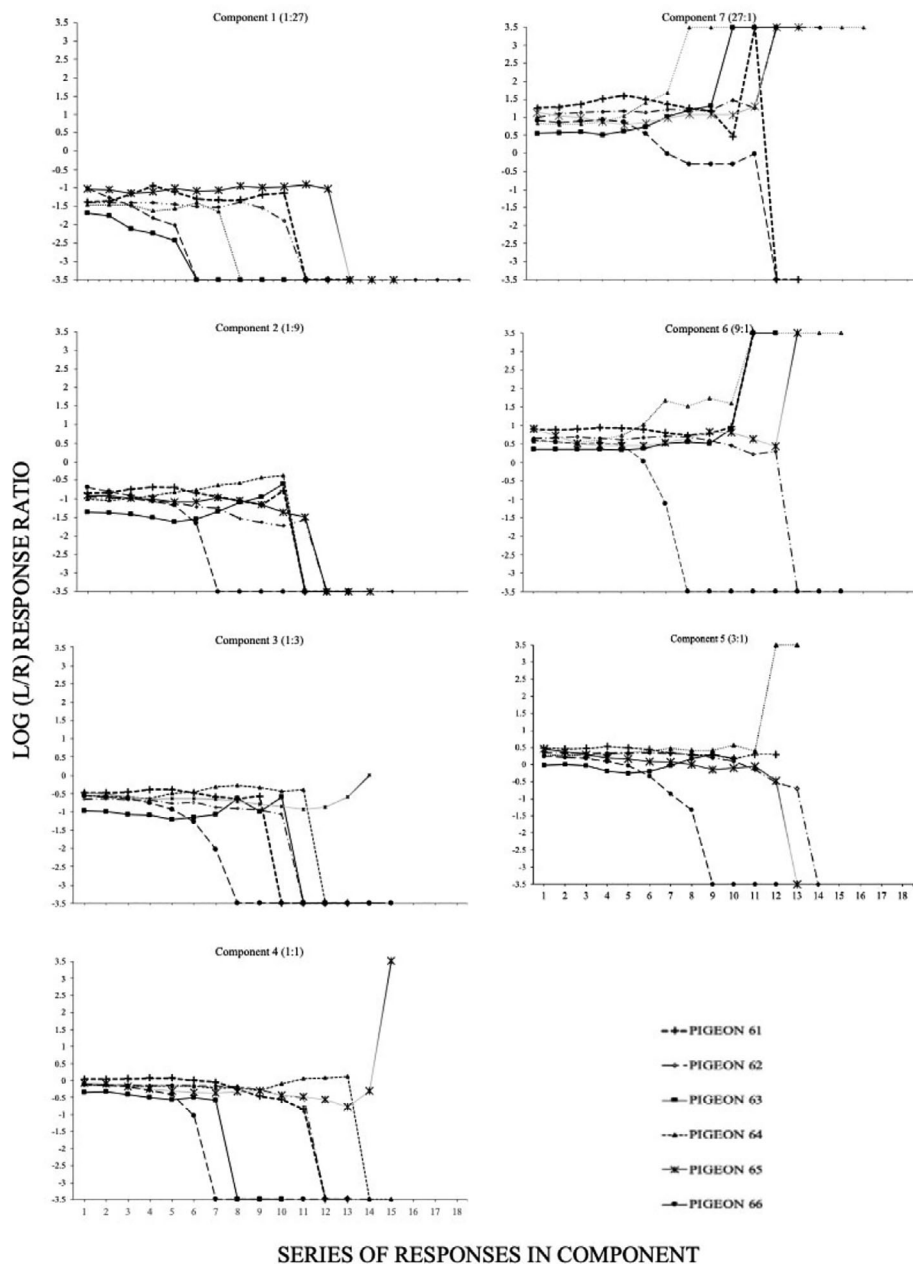
Table 6 summarizes the metrics for generalization. Accuracy remained high across all pigeons ($\leq 93\%$) and for all three generalization approaches. Informedness metrics dropped with specific pigeons' datasets, which suggests that the ability of the model to identify the component was not related to the size of the samples provided but rather to the individual pigeons' differences in patterns of responding. The only apparent exception was when tests were performed with G2, where informedness scores were slightly higher than informedness scores with G1 and G3. The same pattern is preserved in all the rest of the metrics, with a slight drop when only 50% of the data was provided (G3). Overall, the analysis for generalization suggests that the generated models can handle new datasets well and 5 days of training are sufficient when making decisions regarding components and choice responding.

Discussion

The purpose of this study was to investigate whether an artificial SNN model could be trained to identify the reinforcement history that led to current choice behavior based on small samples of choices in pigeon datasets. The results demonstrated that the single neuron architecture could identify learning histories by detecting learning patterns in choice responses. In particular, the SNN model could distinguish pigeon performance in one procedure (Condition 1) and then in another experienced much later on by the same pigeons (Condition 6). Overall, the results showed that choices can be used to retrospectively identify reinforcement history when this is unknown, demonstrating that artificial SNN models may be a useful tool for behavior analysis.

Figure 5

Log Response Ratios of Choices Emitted During the First 5 s Following Each Successive Response in Each of the Seven Components of Condition 1, of Landon & Davison (2001)



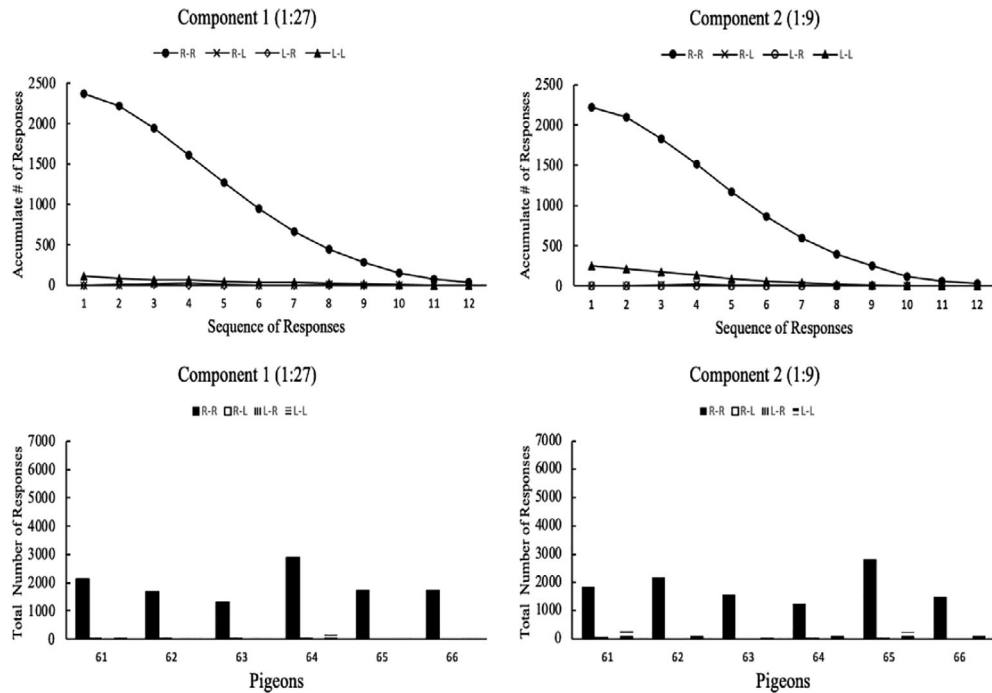
Note. Response numbers were summed across 50 sessions. On each graph, separate plots show data from each individual subset. On the x-axis, the successive responses for all pigeons are presented. When responses were occurring only on one alternative, the log(l/r) was set at 3.5 to indicate the exclusive direction of the choice.

Artificial neural networks learn solely based on datasets; therefore, they reflect patterns encoded in the parameters of the artificial

network. In this study the SNN model was developed to detect patterns of two-alternative choice responses (input data) without any

Figure 6

Total Number of Pair Responses Emitted for the First 5 s Following Each Successive Response in Each of the Seven Components of Condition 1, from Landon & Davison (2001)



Note. Response numbers were summed across 50 sessions as right–right, right–left, left–right, left–left responses. On the x-axis, the total number of responses for each bird is presented.

Table 6

The Overall Results of the Best Model for Each Generalization Test

Generalization Test			#	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
G1	10d	Accuracy		0.96	0.95	0.95	0.95	0.94	0.94
		Sensitivity		0.89	0.85	0.85	0.84	0.82	0.82
		Specificity		0.98	0.97	0.97	0.96	0.96	0.96
		Informedness		0.87	0.82	0.82	0.81	0.78	0.79
		Precision		0.89	0.85	0.85	0.84	0.82	0.82
		F1		0.89	0.85	0.85	0.84	0.82	0.82
G2	7d	Accuracy		0.97	0.95	0.95	0.95	0.95	0.95
		Sensitivity		0.89	0.86	0.87	0.84	0.86	0.86
		Specificity		0.98	0.98	0.98	0.97	0.97	0.97
		Informedness		0.87	0.83	0.83	0.81	0.83	0.83
		Precision		0.89	0.86	0.86	0.84	0.86	0.86
		F1		0.89	0.86	0.86	0.84	0.86	0.86
G3	5d	Accuracy		0.93	0.93	0.93	0.93	0.93	0.93
		Sensitivity		0.82	0.79	0.79	0.79	0.80	0.79
		Specificity		0.96	0.96	0.96	0.96	0.96	0.96
		Informedness		0.76	0.75	0.76	0.75	0.75	0.75
		Precision		0.81	0.79	0.80	0.79	0.79	0.79
		F1		0.81	0.79	0.80	0.79	0.79	0.79

Note. The results reflect generalization tests per pigeon.

knowledge of what experimental conditions or reinforcement arrangements the pigeons had been exposed to. The ability of the artificial SNN model to predict (retrodict) shows that limited observational data from naturalistic settings alone could potentially provide information on the relation between choice and reinforcer. Further, this ability to retrodict the history of novel operant behaviors (i.e., novel as in not participating in the development of the artificial SNN model) allows us to shape future behaviors without necessarily having to know how these functional relations were established in the environment; rather, by relying only on the information that the functional relation is apparent. Overall, knowing what relation exists might provide a particularly effective pathway when the sole interest is to set up optimal learning environments that are task-specific and learner-specific, and a small amount of current behavior as the only source of information. Subsequently, the prediction outcome can help us decide which reinforcement arrangements can effectively shape future choices and which may not. Further, being able to determine the reinforcer history in this way allows us to investigate other matters such as how or why the reinforcer-behavior relation exists. The outcome obtained from the SNN model reconfirms the reality of a reinforcement-behavior relation regardless of the theoretical explanation for this relation (see, e.g., Cowie & Davison, 2016; Simon et al., 2020, for current debate on the mechanism of the relation).

Our results demonstrated high overall accuracy in retrodicting reinforcement arrangements ($\geq 93\%$) across all artificial SNN models. A possible advantage of using a machine learning tool compared to moment-to-moment visual analysis is that it can provide instant additional information on true and false positive and negative values. For example, confusion matrices (Fig. 4) can visualize Type 1 and Type 2 errors, providing additional information on responses that were retrodicted as belonging to a component when they did not. These errors are not easily detected with visual analysis of single case data (Lanovaz et al., 2020). Therefore, the analysis approach explored here can provide more details on learning patterns which will enrich the behavior analyst's decision-making regarding intervention characteristics.

Retrodiction was achieved quickly from a small set of robust behavioral data (spanning 5 s following the reinforcer delivery). Also, small amounts of data render artificial models simple and friendly to use, without need for expertise in training conditions. As rapidly changing environments make it difficult to predict when the number of conditions increases, artificial SNN can be a useful tool to make predictions (retrodictions) in an efficient manner. This is of interest to experimental and applied researchers because collecting continuous data for extended periods can be tedious and costly, and by using ML we demonstrate that the first 5-s periods may not be a constraint. A further question is whether less or more than 5 s of data can alter the model's ability to detect the reinforcement history.

The literature has shown that learning occurs over time and behavior takes time to stabilize. Thus, an additional factor to consider is how large a 'window' of data we choose to model the reinforcement history. It may also be important to consider the time needed for a pigeon to learn in a reinforcement arrangement, as both short-term and long-term relations between choice and reinforcement are evident (Landon & Davison, 2001). Even so, moment-to-moment analysis with 5-s periods (Figs. 5 and 6) revealed that the pigeons behaved similarly in Components 1 (1:27) and 2 (1:9) despite different learning histories in the two reinforcement arrangements. However, this was not the case for Component 6 (9:1) and 7 (27:1). A second question is from which point in time (e.g., the beginning, the middle of the training or when behavior stabilizes) we extract our data to create inputs for the SNN to make an accurate retrodiction.

Based on research into local effects of reinforcers, we included all responses immediately after the reinforcer delivery, both from training and stabilized learning. The point in time at which training occurs in the animal's learning history may affect the ML model's ability to retrodict histories correctly. If behavior changes with experience, then models developed with data both from early (learning) and later (stable) sessions are trained to detect patterns from variable data, which adds to the complexity of 'retrodicting'. By including all types of data as inputs we showed that the artificial SNN models could handle variability over time and modeling histories based on pigeon

responses, without aggregating or averaging the responses (something that is commonly required with second-generation ANN (Alaloul & Qureshi, 2020)). Choice research has shown that previous reinforcer deliveries affect subsequent behavior, and the dynamics of this relation is determined by environmental variation (Davison & Baum, 2000). We considered that, if a series of cross-sectional data were taken (i.e., data from a set period of time during the experiment), we would have missed significant information about the degree of environmental variability. In this study the pigeons' datasets of left- and right-key responses were altered by the weights of the model and combined into one element to produce one current (feature), which we used to stimulate the artificial neuron model and produce sequences of spikes. This total transformation of the data allowed us to make reliable retrodictions without the need for curve-fitting the data (Davison & Elliffe, 2009).

Yet, reducing the sample sizes (five sessions of training versus 10) reduced only slightly the model's ability to retrodict, implying that at least the sample size may play a small role below a certain number of time-events. Future studies could explore the relation between prediction accuracy, from where in time the data are extracted, how much time from the delivery of the reinforcer, and the sample size we generate as input data. Understanding how these factors impact the model's ability to predict may provide further insight into the reinforcer–behavior relation, and the effects of reinforcers on behavior changes with experience. Overall, we demonstrated that artificial SNN allows us to investigate how additional environmental variables such as time in training can detect reinforcement histories with minimal amount of operant choice.

The focus of our study was on the output of a ML model demonstrating retrodiction of reinforcement histories from current behavior when these histories are unknown, and how this outcome can be used to achieve insights into learning. Future work can take further steps to generate new research questions and investigate how a dataset is classified by alternative ANN models to compare the information derived from other models' performance. Such a comparative study of different ANNs could investigate how inputs are handled differently by other architectures and algorithms.

It is also interesting to compare ML results with the human expert classification of the same datasets from experiments arranging high- and low-discriminability histories. Such a comparison might highlight what the specific abilities are that ML can contribute to behavioral data processing.

By being able to identify the most likely learning histories, this tool has the potential to be used in two separate ways: firstly, in the experimental analysis of behavior research to further investigate fundamental behavioral principles, and secondly, to understand what maintains current behavior and to identify effective training conditions that can produce the desired future behavior. The approach we used proves to have good utility in experimental contexts with simple choice paradigms (such as left- vs. right-key responses). Future studies can extend to more complex choice situations, as ML can be accurate with complex data. If more inputs are provided (e.g., three-key alternative responses, or measures of inherent bias), it is not clear yet whether the predictive ability of the model would remain high or even improve in situations where responses follow a similar pattern but were trained under different histories (as with Components 1 and 2). So far, with the limited information provided to our model (left- and right-key responses within a 5-s window), the SNN approach was effective in identifying reinforcement histories. Therefore, an artificial SNN can be used with datasets consisting of behavioral responses alone; in other words, it does not require the inclusion of other data from the learning history and its use is simple and accessible without having to implement common training–testing procedures.

The performance metrics of the artificial SNN model can inform the development of new hypotheses by revisiting the actual datasets. Artificial SNN can complement the existing understanding of reinforcer–behavior interactions. These algorithms allow us to pose new research questions and detect relations that shed light on the mechanisms of choice behavior of individual organisms. For example, the analysis of the moment-to-moment behavior revealed that choices in Components 1 and 2 showed a similar pattern, with response initiation being quicker when the rich key was on the right. Moreover, we saw that, when the rich key was on the left (Components 5 to 7),

responses often occurred on the right key where no reinforcement was arranged. The pigeons tended to respond to the right key more than expected, regardless of the arranged reinforcement, reflecting a right-key bias (Baum, 1974). This bias meant that choice in left-versus right-key conditions was not symmetrical, which may have hindered the model's training and its resulting ability to differentiate learning patterns, if any. This factor may be critical to making a pattern discriminable with some histories, that is, when biases are present or reinforcer ratios are similar. Nevertheless, it is worth reiterating that even with this bias present, the artificial SNN model was able to detect learning histories well using a small amount of data. This indicates that SNN offers an advantage beyond conventional approaches. Future research could investigate how biases and other sources of 'confusion' (e.g., smaller ranges of reinforcer ratios, contradictory signals) may impact on the ability of SNNs to accurately identify learning histories.

Artificial spiking neural network modeling has emerged out of an interest in modeling the behavior of the biological neuron to understand human behavior. Its ability to provide insights into patterns of the brain by analyzing the effect of environmental stimuli on the spatiotemporal brain data has led to substantial research interest in modeling brain with artificial SNN (Ghosh-Dastidar & Adeli, 2009); that is, if we model how the brain works, we could understand the behavior of a living organism (in-out approach). Here, we reversed this direction by analyzing spatiotemporal information exclusively from reinforcement arrangements under which a response occurred, thus detecting patterns that are also helpful in understanding a living organism's behavior (out-in approach). Contemporaneous computerized machine learning tools like artificial SNN can open avenues to more complete accounts of behavior. This study, to our knowledge, is the first to use direct measures of behavior (choice responses) with an artificial SNN model and illustrates how analysis of within-subject designs and small group participants can be used to answer alternative questions (i.e., by training and testing a model, what behavioral data can tell us about learning histories rather than vice versa).

The modeling here has advanced our knowledge beyond what is known from

traditional analyses: We found that snapshots of data from current learned behavior and, importantly, data that are variable and extracted from unpredictable environments, contain patterns detectable by our ML model. Experimental analysis of behavior has focused mostly on current behavior-reinforcement contingencies. SNN modeling's special contribution can be its capability to transform the temporal data for us to analyze both from current contingencies and from past experience extremely fast and by using the same algorithm. There is a need to focus on past reinforcement history (Freeman & Lattal, 1992; Okouchi et al., 2014), and our reverse engineering approach can augment existing methods of looking into this. The results are promising as they illustrate how ML modeling can have a translational ability when using data from highly controlled conditions to answer applied questions. Even without us knowing how these functional relations were established in the environment (which is also an essential experimental question to respond to), ML detects past learning and gives us a basis to anticipate how future behavior is shaped. Thus, data from experimental studies can acquire clinical utility. The results promise to open some helpful avenues for translational research as we can now make use of this additional means of investigating learning patterns.

Conclusions

This study shows how a novel machine learning (ML) tool can inform us about behavior-environment contingencies using a small 'window' of data without using the common testing-training procedures. It also demonstrates how a ML tool can be utilized as a hypothesis generator directing us to look into specific behavioral data for further investigation and analysis. Artificial SNN is a new subdivision of neural networks that we preferred over other ANNs because SNN can process spatiotemporal behavioral data, an ability that makes SNN a promising tool given the role that elapsed time plays in changes of preferences. In sum, we found that artificial SNN allowed us to identify learning histories of current learned behavior within seconds, from datasets of organisms previously unknown to the machine. Moreover, the results confirm previous research findings about the critical

impact learning history has on current behavior. ANN can arguably contribute to laying the foundations for new training methodologies using optimal training conditions for specific learning goals and for specific individual organisms. This knowledge (that a relation exists rather than why it exists) also reconfirms the reinforcement–behavior relation regardless of the theoretical explanation of this relation. This becomes possible with the ability of ML for prediction (or ‘retrodiction’) of learning patterns. It was simple and not time-consuming to train our SNN model and more research could further optimize the model’s training and efficiency with small datasets.

An interesting future perspective is how artificial neuronal modeling will perform in classifying behavior of neurologically more complex organisms such as mammals. If the model’s performance is comparable to its retrodictive ability with pigeons, that would indicate its ability to reliably detect choice patterns in changing environments. Such identification of response patterns could then provide a useful indication of brain processes underpinning learning, a subject matter of current neuroscientific research.

Acknowledgment

Open access publishing facilitated by Auckland University of Technology, as part of the Wiley - Auckland University of Technology agreement via the Council of Australian University Librarians.

References

- Ahmed, K. (2020). Brain-inspired spiking neural networks. In M. K. Habib & C. Martín-Gómez (Eds.), *Biomimetics*. IntechOpen. <https://doi.org/10.5772/INTECHOPEN.93435>
- Alaloul, W. S., & Qureshi, A. H. (2020). *Data processing using artificial neural networks*. In Dynamic data assimilation - Beating the uncertainties. IntechOpen. <https://doi.org/10.5772/INTECHOPEN.91935>
- Auge, D., Hille, J., Mueller, E., & Knoll, A. (2021). A survey of encoding techniques for signal processing in spiking neural networks. *Neural Processing Letters*, 53(6), 4693–4710. <https://doi.org/10.1007/S11063-021-10562-2/TABLES/1>
- Baum, W. M. (1974). On two types of deviation from the matching law: Bias and undermatching. *Journal of the Experimental Analysis of Behavior*, 22(1), 231–242. <https://doi.org/10.1901/jeab.1974.22-231>
- Baum, W. M. (2002). From molecular to molar: A paradigm shift in behavior analysis. *Journal of the Experimental Analysis of Behavior*, 78(1), 95–116. <https://doi.org/10.1901/jeab.2002.78-95>
- Baum, W. M. (2018). Three laws of behavior: Allocation, induction, and covariance. *Behavior Analysis: Research and Practice*, 18(3), 239. <https://doi.org/10.1037/BAR0000104>
- Bhangale, K. B., & Mohanaprasad, K. (2021). A review on speech processing using machine learning paradigm. *International Journal of Speech Technology*, 24(2), 367–388. <https://doi.org/10.1007/S10772-021-09808-0>
- Cai, L., Gao, J., & Zhao, D. (2020). A review of the application of deep learning in medical image classification and segmentation. *Annals of Translational Medicine*, 8(11), 713–713. <https://doi.org/10.21037/ATM.2020.02.44>
- Chen, X., Kang, W., Zhu, D., Zhang, X., Lei, N., Zhang, Y., Zhou, Y., & Zhao, W. (2018). A compact skyrmionic leaky-integrate-fire spiking neuron device. *Nanoscale*, 10(13), 6139–6146. <https://doi.org/10.1039/c7nr09722k>
- Cowie, S., & Davison, M. (2016). Control by reinforcers across time and space: A review of recent choice research. *Journal of the Experimental Analysis of Behavior*, 105(2), 246–269. <https://doi.org/10.1002/jeab.200>
- Davison, M., & Baum, W. M. (2000). Choice in a variable environment: Every reinforcer counts. *Journal of the Experimental Analysis of Behavior*, 74(1), 1–24. <https://doi.org/10.1901/jeab.2000.74-1>
- Davison, M., & Baum, W. M. (2002). Choice in a variable environment: Effects of blackout duration and extinction between components. *Journal of the Experimental Analysis of Behavior*, 77(1), 65–89. <https://doi.org/10.1901/JEAB.2002.77-65>
- Davison, M., & Baum, W. M. (2003). Every reinforcer counts: Reinforcer magnitude and local preference. *Journal of the Experimental Analysis of Behavior*, 80(1), 95–129. <https://doi.org/10.1901/JEAB.2003.80-95>
- Davison, M., & Elliffe, D. (2009). Variance matters: The shape of a datum. *Behavioural Processes*, 81(2), 216–222. <https://doi.org/10.1016/j.beproc.2009.01.004>
- Davison, M., & Jenkins, P. E. (1985). Stimulus discriminability, contingency discriminability, and schedule performance. *Animal Learning & Behavior*, 13(1), 77–84. <https://doi.org/10.3758/BF03213368>
- Davison, M., & McCarthy, D. (1988). *The matching law: A research review*. Lawrence Erlbaum Associates Inc. <https://doi.org/10.4324/9781315638911>
- Freeman, T. J., & Lattal, K. A. (1992). Stimulus control of behavioral history. *Journal of the Experimental Analysis of Behavior*, 57(1), 5–15. <https://doi.org/10.1901/JEAB.1992.57-5>
- Gerstner, W., & Kistler, W. M. (2002). *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge University Press. <https://doi.org/10.1017/CBO9780511815706>
- Ghosh-Dastidar, S., & Adeli, H. (2009). Spiking neural networks. *International Journal of Neural Systems*, 19(4), 295–308. <https://doi.org/10.1142/S0129065709002002>
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning: with Applications in R. Springer. <https://doi.org/10.1016/j.peva.2007.06.006>
- Killeen, P. (1972). The matching law. *Journal of the Experimental Analysis of Behavior*, 17(3), 489–495. <https://doi.org/10.1901/jeab.1972.17-489>
- Korkmaz, C., & Correia, A. P. (2019). A review of research on machine learning in educational technology. *Educational Media International*, 56(3), 250–267. <https://doi.org/10.1080/09523987.2019.1669875>

- Kuhn, M., & Johnson, K. (2018). Applied predictive modeling. Springer. <https://doi.org/10.1007/978-1-4614-6849-3>
- Landon, J., & Davison, M. (2001). Reinforcer-ratio variation and its effects on rate of adaptation. *Journal of the Experimental Analysis of Behavior*, 75(2), 207–234. <https://doi.org/10.1901/jeab.2001.75-207>
- Landon, J., Davison, M., & Elliffe, D. (2002). Concurrent schedules: Short-and long-term effects of reinforcers. *Journal of the Experimental Analysis of Behavior*, 77(3), 257–271. <https://doi.org/10.1901/JEAB.2002.77-257>
- Landon, J., Davison, M., & Elliffe, D. (2003). Choice in a variable environment: Effects of unequal reinforcer distributions. *Journal of the Experimental Analysis of Behavior*, 80(2), 187–204. <https://doi.org/10.1901/JEAB.2003.80-187>
- Lanovaz, M. J., Giannakakos, A. R., & Destras, O. (2020). Machine learning to analyze single-case data: A proof of concept. *Perspectives on Behavior Science*, 43(1), 21–38. <https://doi.org/10.1007/s40614-020-00244-0>
- Maass, W. (1997). Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10(9), 1659–1671. [https://doi.org/10.1016/S0893-6080\(97\)00011-7](https://doi.org/10.1016/S0893-6080(97)00011-7)
- Macpherson, T., Churrland, A., Sejnowski, T., DiCarlo, J., Kamitani, Y., Takahashi, H., & Hikida, T. (2021). Natural and artificial intelligence: A brief introduction to the interplay between AI and neuroscience research. *Neural Networks*, 144, 603–613. <https://doi.org/10.1016/j.neunet.2021.09.018>
- Mazur, J. E. (2016). Choice. In *Learning and behavior* (8th ed., pp. 37–51). Routledge. <https://doi.org/https://doi.org/10.4324/9781315450285>
- Okouchi, H., & Lattal, K. A. (2006). An analysis of reinforcement history effects. *Journal of the Experimental Analysis of Behavior*, 86(1), 31–42. <https://doi.org/10.1901/jeab.2006.75-05>
- Okouchi, H., Lattal, K. A., Sonoda, A., & Nakamae, T. (2014). Stimulus control and generalization of remote behavioral history. *Journal of the Experimental Analysis of Behavior*, 101(2), 275–287. <https://doi.org/10.1002/jeab.75>
- Pierce, W. D., & Epling, W. F. (1983). Choice, matching, and human behavior: A review of the literature. *The Behavior Analyst*, 6(1), 57–76. <https://doi.org/10.1007/BF03391874>
- Pipkin, C. S. P., & Vollmer, T. R. (2009). Applied implications of reinforcement history effects. *Journal of Applied Behavior Analysis*, 42(1), 83–103. <https://doi.org/10.1901/JABA.2009.42-83>
- Poling, A. (2010). Looking to the future: Will behavior analysis survive and prosper? *Behavior Analyst*, 33(1), 7–17. <https://doi.org/10.1007/BF03392200>
- Rafi, T. H. (2021). A brief review on spiking neural network - A biological inspiration. *Preprints 2021*, 2021040202. <https://doi.org/10.20944/preprints202104.0202.v1>
- Shahan, T. A. (2017). Moving beyond reinforcement and response strength. *Behavior Analyst*, 40(1), 107–121. <https://doi.org/10.1007/s40614-017-0092-y>
- Shatte, A. B. R., Hutchinson, D. M., & Teague, S. J. (2019). Machine learning in mental health: A scoping review of methods and applications. *Psychological Medicine*, 49(9), 1426–1448. <https://doi.org/10.1017/S0033291719000151>
- Simon, C., Bernardy, J. L., & Cowie, S. (2020). On the “strength” of behavior. *Perspectives on Behavior Science*, 43(4), 677–696. <https://doi.org/10.1007/S40614-020-00269-5/FIGURES/2>
- Skinner, B. F. (1969). *Contingencies of reinforcement: A theoretical analysis*. Appleton-Century-Crofts.
- Staddon, J. E. R. (2016). The optimal allocation of behavior. In *Adaptive behavior and learning* (2nd ed., pp. 205–241). Cambridge University Press. <https://doi.org/10.1017/cbo9781139998369.009>
- Storn, R., & Price, K. (1997). Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4), 341–359. <https://doi.org/10.1023/A:1008202821328>
- Turgeon, S., & Lanovaz, M. J. (2020). Tutorial: Applying machine learning in behavioral research. *Perspectives on Behavior Science*, 43(4), 697–723. <https://doi.org/10.1007/s40614-020-00270-y>
- Vazquez, R. A., & Cachón, A. (2010). Integrate and Fire neurons and their application in pattern recognition. Program and Abstract Book – 2010. 7th International Conference on Electrical Engineering, Computing Science and Automatic Control, 424–428. <https://doi.org/10.1109/ICEEE.2010.5608622>
- Vyse, S. (2013). Changing course. *The Behavior Analyst*, 1(1), 129–165. https://doi.org/10.9774/gleaf.978-1-907643-44-6_19
- Zador, A. M. (2019). A critique of pure learning and what artificial neural networks can learn from animal brains. *Nature Communications*, 10(1), 1–7. <https://doi.org/10.1038/s41467-019-11786-6>

Received: March 30, 2022

Final Acceptance: March 29, 2022

Editor-in-Chief: Rick Bevins

Associate Editor: Erin Calipari

A. Appendix

A.1. An example of the stepwise process of extracting the data from the original study

Condition 1 and 6 events in the original experiment (Landon and Davison, 2001) were arranged on an IBM-PC compatible computer running MED-PC software. The computer recorded all events and the time they occurred within each Condition. We developed a computer code to identify all original experimental codes from the existing files and created four steps to extract pigeon data (Fig. A1): (1) Searching in the data for a component indicated by x000000 that denotes the start of each component (1-7), was $x = 0-6$; (2) searching for each of the nine reinforcers as they were delivered in time by looking for the 3xxxx(L-key) or 4xxxx(R-key); (3) counting and adding the responses during the following 7.5 s (2.5 s for the hopper raised and 5 s of recording response frequency to the L or R key); and (4) generating the ratio event for every 5-s period. In this way, we extracted

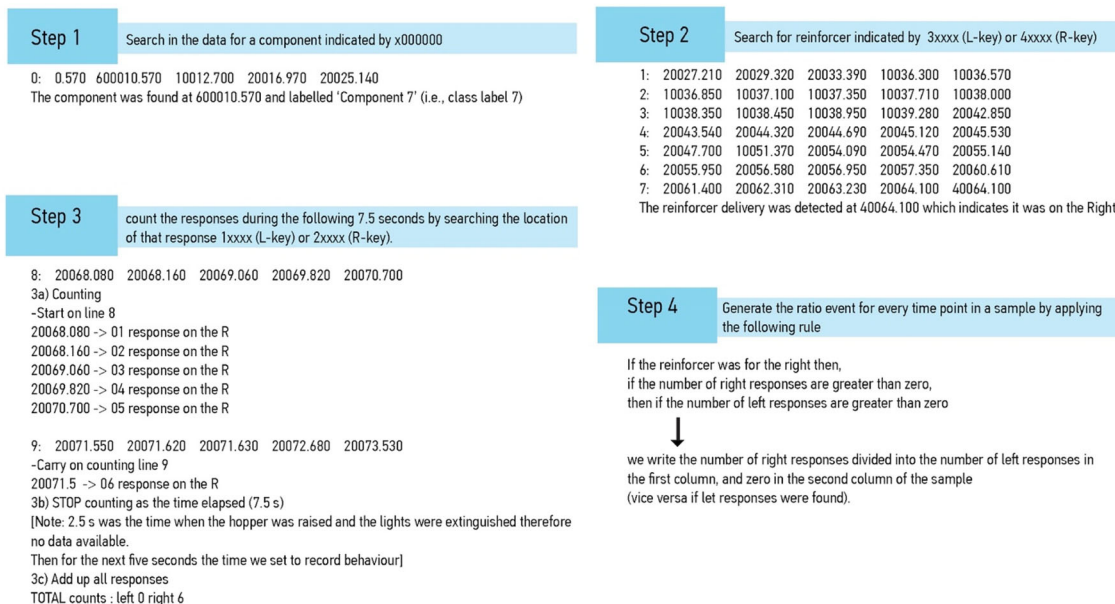
all data for input samples. The final outcome was a ratio that was a frequency event, which was created by calculating the ratio of left and right preferences based on the time window (5-s) and on the actual location (L or R) where the reinforcer was delivered.

For example, let's hypothesize that after taking steps 1-3 as presented in Figure A1, we extracted the following information: A pigeon was presented randomly with one of the seven different components (i.e., 27: 1, 9:1, 3:1, 1:1, 1:3, 1:9, and 1:27) and a reinforcer was allocated to the right, and after consuming the reinforcer, the pigeon then started pecking one time on the left-key and four times on the right-key during a period of 5 s.

The relative response rate of the pigeon would be 4/1 (the pigeon pecked four times more on the right key). In the reverse scenario, for example, if the reinforcer was allocated on the left, and the pigeon pecked one time on the right and three times on the left, then the relative response rate would be 3/1 (the pigeon pecked three times on the left

Figure A1

Examples of the Steps Taken to Extract Pigeon Data from the Original Study (Landon & Davison, 2001) As Generated By MED-PC® Software



key). By using this method, all responses extracted from the original experiment into the samples occurred after the reinforcer was delivered for 5 s and were dependent on the reinforcer location in time.

Supporting Information

Additional Supporting Information may be found in the online version of this article at the publisher's website.