Artificial Neural Networks, ADHD, and the Construction of a Virtual "Subject": Fluid Intelligence & Time Management

Markus Ville Tiitto and Robert A. Lodder*

Department of Pharmaceutical Sciences
College of Pharmacy
University of Kentucky
Lexington, KY 40536

*Author to whom correspondence should be addressed. Email: Lodder @ g.uky.edu

Abstract

Attention deficit hyperactivity disorder (ADHD) is a neurodevelopmental disorder characterized by inattention, hyperactivity, and impulsivity. Our lab is currently conducting a pilot study to assess the effects of the online game Minecraft as a therapeutic video game (TVG) to train executive function deficits in children with ADHD. The effect of the TVG intervention in combination with stimulants is being investigated to develop a drug-device combination therapy that can address both the dopaminergic dysfunction and executive function deficits present in ADHD. Although the results of this study will be used to guide the clinical development process, additional guidance for the optimization of the executive function training activities will be provided by a computational model of executive functions built with artificial neural networks (ANNs). This model uses ANNs to complete virtual tasks resembling the executive function training activities that the study subjects practice in the Minecraft world, and the schedule of virtual tasks that result in maximum improvements in ANN performance on these tasks will be investigated as a method to inform the selection of training regimens in future clinical studies. This study first proposes the use of recurrent neural networks to model the fluid intelligence executive function. This model is then combined with a previously developed model using convolutional neural networks to model working memory and prepotent impulsivity to produce virtual "subjects" who complete a computational simulation of a Time Management task that requires the use of both of these executive functions to complete. The capability of this model to produce groups of virtual "subjects" with significantly different levels of performance on the Time Management task is demonstrated.

Contents

Background

Attention Deficit Hyperactivity Disorder (ADHD) is a highly prevalent neurodevelopmental disorder in children that is characterized by symptoms of inattention, hyperactivity, and impulsivity[1]. ADHD can cause difficulties in academic and social functioning, and can lead to adverse long-term outcomes such as drug addiction and criminal behavior[2]. While the exact cause of ADHD remains unknown, drug therapies have proven successful in controlling symptoms. However, these drug therapies have not proven effective for improving long-term outcomes[2], their side effects limit use in a significant number of patients, and concerns about their safety has led to a wide preference for non-drug therapies[3].

An emerging alternative strategy for the treatment of ADHD is the training of executive functions, or a set of conscious mental processes that regulate attention and behavior[4]. Executive function deficits have been proposed as a potential cause for ADHD[5], and computerized executive function training programs have been investigated as a potential treatment option[6]. The majority of these programs have been designed to target a single core executive function, such as working memory, that is thought to exert far-reaching effects across multiple areas of behavior[4]. While these programs were developed with the expectation that amelioration of these core executive function deficits would lead to improvements in ADHD symptoms and overall functioning, evidence of these far-transfer effects has thus far been limited. Although several studies have shown improvements in parent ratings of ADHD symptoms, teacher ratings and academic performance remained largely unaffected, indicating a lack of generalization of improvements in executive function across multiple environments[6,7].

Multiple strategies have emerged to address this lack of far-transfer effects resulting from executive function training interventions. Firstly, video game environments have been developed as a setting for executive function training activities to improve patient motivation and increase engagement with the training interventions. Video gaming has been shown to increase brain dopamine levels[8], and functional deficits in children with ADHD may improve during video game play[9]. Secondly, the training strategies have expanded to target the training of multiple executive functions, as well as real-life skills requiring the use of executive functions for their application. For example, video game intervention designed to train working memory, inhibition, and cognitive flexibility improved both parent and teacher ratings of ADHD symptoms[10]. Another video game intervention was designed to train real-life skills such as time management, organization, planning, and cooperation, resulting in improvements in both parent and teacher ratings of childrens' performance in time management (ADHD symptom ratings were not assessed)[11].

An ongoing clinical pilot study is investigating the use of Minecraft as a therapeutic video game treatment for ADHD in combination with stimulants[12]. Minecraft provides an open-ended video game environment with a considerable degree of flexibility for the development of a variety of executive function training activities. Training activities to address both multiple core executive function deficits and real-life skills resulting from the use of executive functions are included in the present study. In order to address the heterogeneous nature of executive function deficits in ADHD[13], a personalized treatment intervention that can provide an optimal schedule of executive function training activities for the pattern of deficits present in each individual patient to further improve outcomes is under development. This personalized treatment intervention will also include a stimulant dosage target to reduce the trial and error required to find an optimal dose for a given child with ADHD.

A computational model utilizing artificial neural networks (ANNs) will be used to produce the personalized treatment recommendations. Artificial neural networks are a group of machine learning methods that can learn complex patterns in a dataset and use these patterns to generate predictions for new examples[14]. ANNs have generated a great deal of interest for their applications in healthcare. For example, ANNs have been shown capable of medical imaging diagnostic performance comparable to human physicians[15], and even outperformed neurologists in diagnosing intracranial hemorrhages from CT scans[16]. Neural network-based technologies have also been developed for use in institutional medical settings to identify early-stage sepsis[17] and predict the risk of deterioration in intensive care patients[18,19].

The structure and function of artificial neural networks were inspired by biological neural networks, although they remain a highly simplified approximation. Nonetheless, they can learn to make predictions in complex, changing environments and could therefore be used as a simplified model for the behavior of humans in an environment such as Minecraft. Thus, by developing a set of computational tasks performed by ANNs that resemble executive function tasks performed by human subjects in Minecraft it may be possible to gain insight into how human task performance is optimized by careful inspection of how ANN performance on their computational tasks is optimized. This hypothesis will guide the personalization of the executive function training activity schedule recommendations in the therapeutic video game intervention[20].

The objective of this work is to develop a computational task performed by ANNs that represents the use of time management skills in humans. This task was designed to require the use of the fluid intelligence and working memory executive functions, with the working memory function impeded by prepotent impulsivity. First, a fluid intelligence function represented by the performance of a factorization of the number 12 by basic recurrent neural networks (RNNs)[21] was developed. This fluid intelligence function was then combined with the previously developed working memory and prepotent impulsivity functions that used convolutional neural networks (CNNs)[22] to identify handwritten digits in the MNIST test set[23] to create the computational Time Management activity.

Methods and Results

Generating Fluid Intelligence Deficiencies (Colaboratory Notebook)

Fluid intelligence is the use of adaptive reasoning in novel situations[4]. For the purposes of this project fluid intelligence is considered as a problem solving ability consisting of the mental processes analysis and synthesis[5]. The process of analysis involves the decomposition of complex information into smaller, simpler parts, while the process of synthesis involves the combination of simple pieces of information into more complex, novel combinations. To represent the fluid intelligence process of synthesis in this experiment, a computational Factorization of 12 activity was designed. This activity is performed by RNNs that are trained to produce a sequence of integer outputs in the range 1-10 that can be multiplied together to produce the larger integer 12.

RNNs were chosen to perform the computational Factorization of 12 activity because they are capable of processing sequences consisting of multiple inputs[21]. Although the architectures of RNNs can vary considerably in complexity, the simplest form of RNN possesses a single internal layer that changes its state as the sequence is processed and three sets of weight parameters. One set of weights processes the input to the network, another set of weights modifies the state of the internal layer of the network, and the third set of weights converts the inner state of the network into an output. The inner state of the network changes as each individual element of the input sequence is processed in a series of steps; this inner state is affected by the current element being processed as well as all the previously processed elements and determines the output that the network produces at each step. As a result, the output of the network at each step depends on both the previously processed elements of the sequence and the currently processed element.

To produce a list of integers that are factors of the number 12, the RNN is provided with the start token as an initial input, and outputs a single predicted factor in the first step. This predicted factor is recorded in a list, which is then provided as an input (rather than the start token) for the RNN in the next step (Figure 1**)**. This process repeats until the RNN produces an output indicating the sequence of factors is complete.

All calculations for this experiment were performed with Python v3 in a Google Colaboratory Notebook. A basic RNN model modified from a RNN designed for language modeling and the generation of reddit comments[24] was implemented. This RNN contained a single inner layer of 100 units, used the tanh activation function (a non-linear function with outputs ranging from -1 to 1) to calculate the inner layer, and a softmax activation function to generate a list of normalized probabilities for integers in the range 1-11 as an output. The number 0 was used as a “start token”, or initial input to signal the start of a factorization prediction, while the number 11 was used as an “end token” to indicate the end of a sequence of predicted outputs.

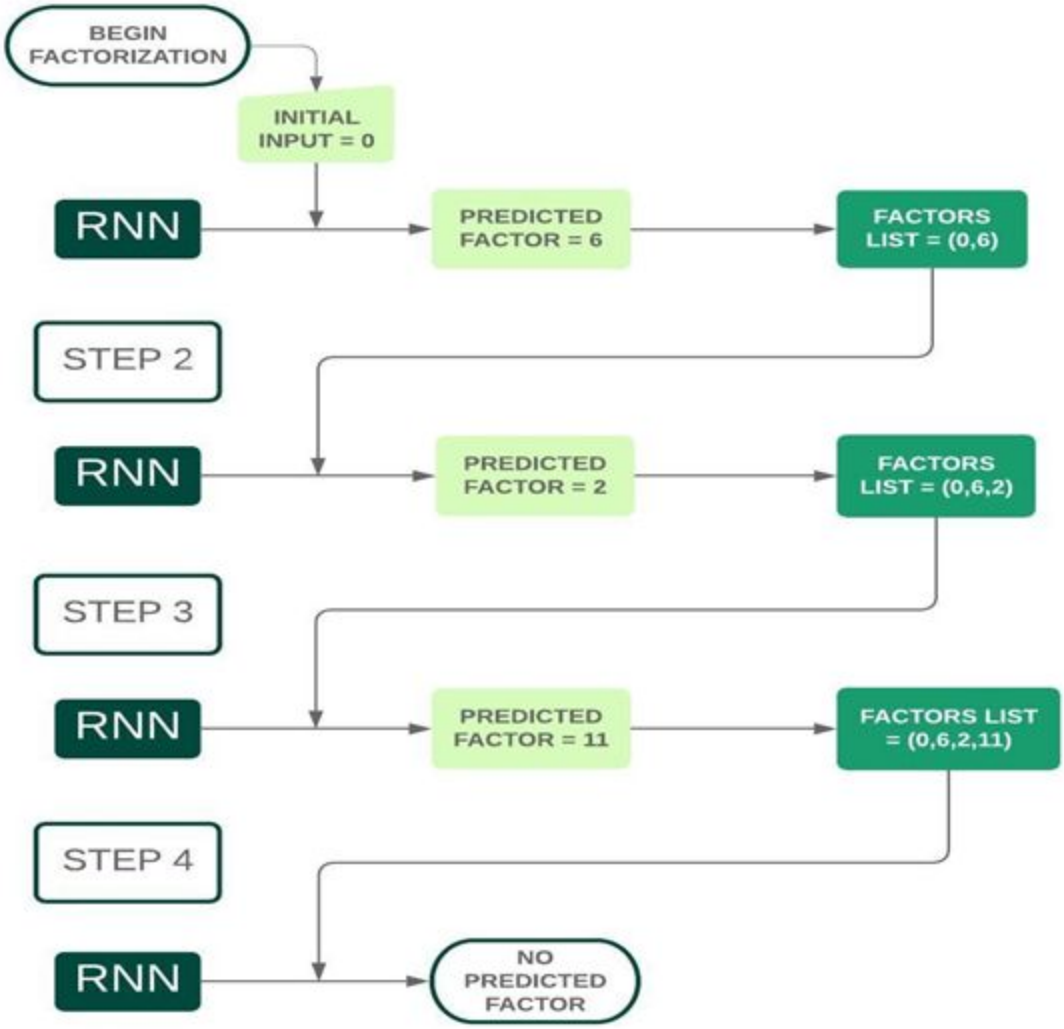

Figure 1. Factor Prediction for the Number 12 by a Recurrent Neural Network

A training set composed of 29 training examples with features (inputs) and labels (correct outputs) were generated by hand. The set of training features consisted of lists of integers between 0 and 10, and the list for each training feature began with the integer 0. Lists of integers that could be multiplied together to produce the number 12 and lists of integers that could not be multiplied together to produce the number 12 were both included. The set of training features lists that produced the number 12 included all the permutations of each set of factors, while the training features lists that did not produce the number 12 simply consisted of the start token 0 followed by the incorrect factor. A matching training label was created for each training feature. The training label for each example contained the same list of integers contained in the features for the example, but the start token 0 was removed and the end token 11 was added to the end of the list.

To produce different categories of performance for the computational Factorization of 12 activity, modifications were introduced into a supervised learning training procedure and

investigated for their effects on RNN performance. In this case, the data available for training was limited (29 training examples) so the differences in performance were created by varying the number of training cycles (or epochs) as the independent variable. All of the training examples are presented to a neural network in one training epoch, so the number of times a neural network sees the set of training examples is equivalent to the number of training epochs. Ten groups of 12 x RNNs per group were assigned unique durations for their training ranging from 1 training epoch to 5,000 training epochs. The categorical cross-entropy loss function was used to measure the deviation between RNN outputs and correct predictions and calculate gradients to adjust parameter values. Stochastic gradient descent with a learning rate of 0.005 was used for the optimization method to adjust parameter values.

After training, the performance of each RNN was evaluated by calculating its percent accuracy on 100 factorization of 12 predictions. As expected, the mean percent accuracy for the RNNs in each group increased with the number of training epochs in a logarithmic relationship **(**Figure 2**).** The minimum group mean accuracy was 0.5 % (SD 0.9%) for the RNN group trained for 1 epoch while the maximum group mean accuracy achieved was 81.1% (SD 2.9%) for the RNN group trained for 5,000 epochs. The rate of improvement in performance began to level off at a group mean accuracy of 74.2% (SD 5.4%) with 250 training epochs, although significant improvements in performance were still observed as the number of training epochs was increased further.

The training procedure for “deficient” fluid intelligence RNNs was selected to achieve a goal of approximately 50% mean factorization of 12 accuracy, while the training procedure for “healthy” fluid intelligence RNNs was selected with a consideration of performance and computational efficiency. The goal performance level for “deficient” fluid intelligence was achieved in RNNs trained for 75 epochs (group mean accuracy = 48.3% (SD 4.1%). Training for 1000 epochs was selected as the learning procedure for “healthy” fluid intelligence RNNs to achieve a mean group accuracy of 77.6 % (SD 4.2%) with a sacrifice of 3.5% accuracy in exchange for significantly improved computational efficiency compared to training for 5000 epochs.

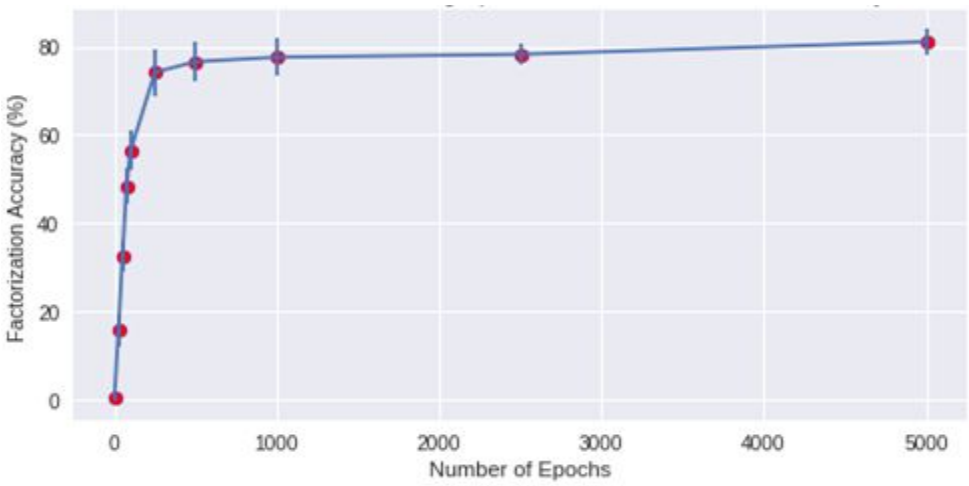

Figure 2. Effect of Number of Training Epochs on Mean Group Accuracy of Factorization of 12 in Recurrent Neural Networks (n = 12 x RNNs Per Group)

In summary, a computational representation for the synthesis function of the higher order executive function fluid intelligence in humans was developed in this experiment. This model includes basic RNNs to predict integers between 1 and 10 that can be multiplied together to produce the number 12, where a "healthy" fluid intelligence is represented by RNNs with high Factorization of 12 accuracy and a "deficient" fluid intelligence is represented by RNNs with poor Factorization of 12 accuracy. This Factorization of 12 fluid intelligence function was then integrated together with the MNIST handwritten digit recognition function representation for working memory and the prepotent impulsivity function to create the first implementation of virtual "subjects" who perform a virtual Time Management task.

Time Management Task (Colaboratory Notebook)

A pertinent example of the use of fluid intelligence is in the practice of time management. When considering how to complete a complex task most efficiently, one may first analyze the task carefully to determine the individual steps needed for its completion. There will be a finite number of ways to organize these steps, and the best sequence of steps will be the one that produces the best outcome in the shortest amount of time. Working memory and the synthesis process of fluid intelligence can be used to mentally rehearse the completion of varying combinations of these steps and determine the time required to complete them. Some combinations of steps may overlap better in time than others when they are completed sequentially, and some steps may be found to provide minimal contributions to the overall goal. The order of the overlapping steps may then be preserved in the final sequence, and the marginally important steps may be left at the end of the sequence to be completed after the more highly prioritized steps should time permit. In this way, fluid intelligence can work together with the core executive function working memory in a time management activity.

While the working memory and fluid intelligence functions are each represented by the performance of individual ANNs in performing a given activity representing an executive function, the Time Management task consists of a group of ANNs performing their given activities in tandem to complete a given task together. This combination of multiple ANNs working together to complete a task can be thought of as a virtual "subject" that utilizes multiple executive functions to complete a computational representation of an executive function training activity completed by human subjects in the therapeutic video game intervention (Figure 3).

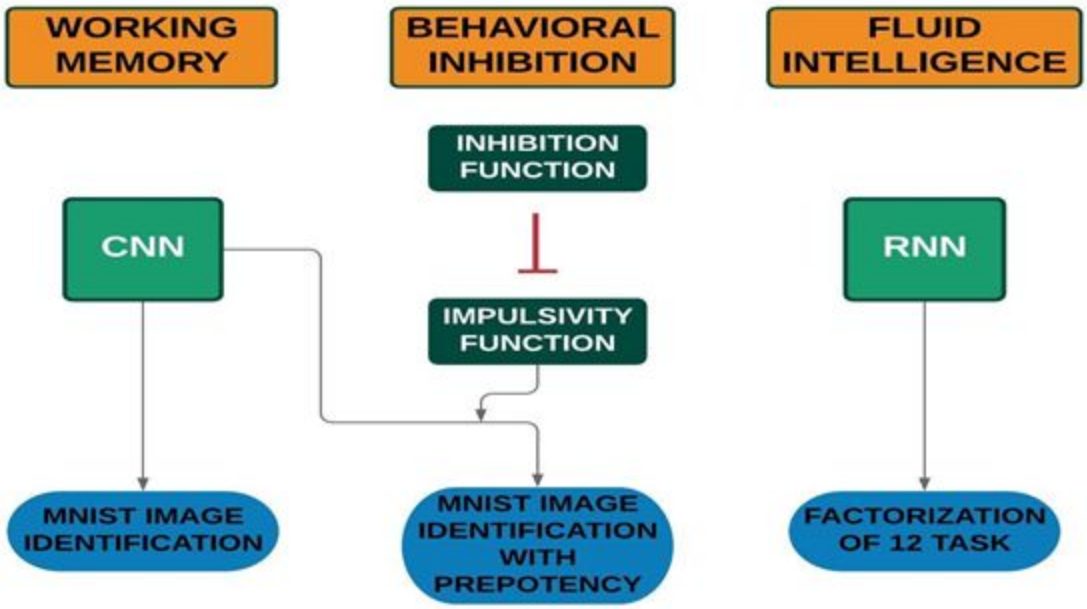

Figure 3. Building the Virtual "Subject"

Similar to the fluid intelligence task described in the previous section, the goal of the Time Management task is to create a list of integers between 1 and 10 that can be multiplied together to produce the number 12. However, the process required to achieve this goal is modified in three important ways. Firstly, the Time Management task is not complete after a single factorization of 12 attempt, but rather continues until a successful factorization of 12 is completed (Figure 4). Secondly, the Time Management task requires the use of the working memory function impeded by the prepotent impulsivity function for its completion. The working memory function with prepotent impulsivity is used after the prediction of a single factor to locate and correctly identify an MNIST image that contains the predicted factor as its handwritten digit in the MNIST test set. Finally, the performance of the Time Management task is measured by the total number of MNIST test set images searched through to complete a successful factorization of 12 rather than the percentage of successful factorizations.

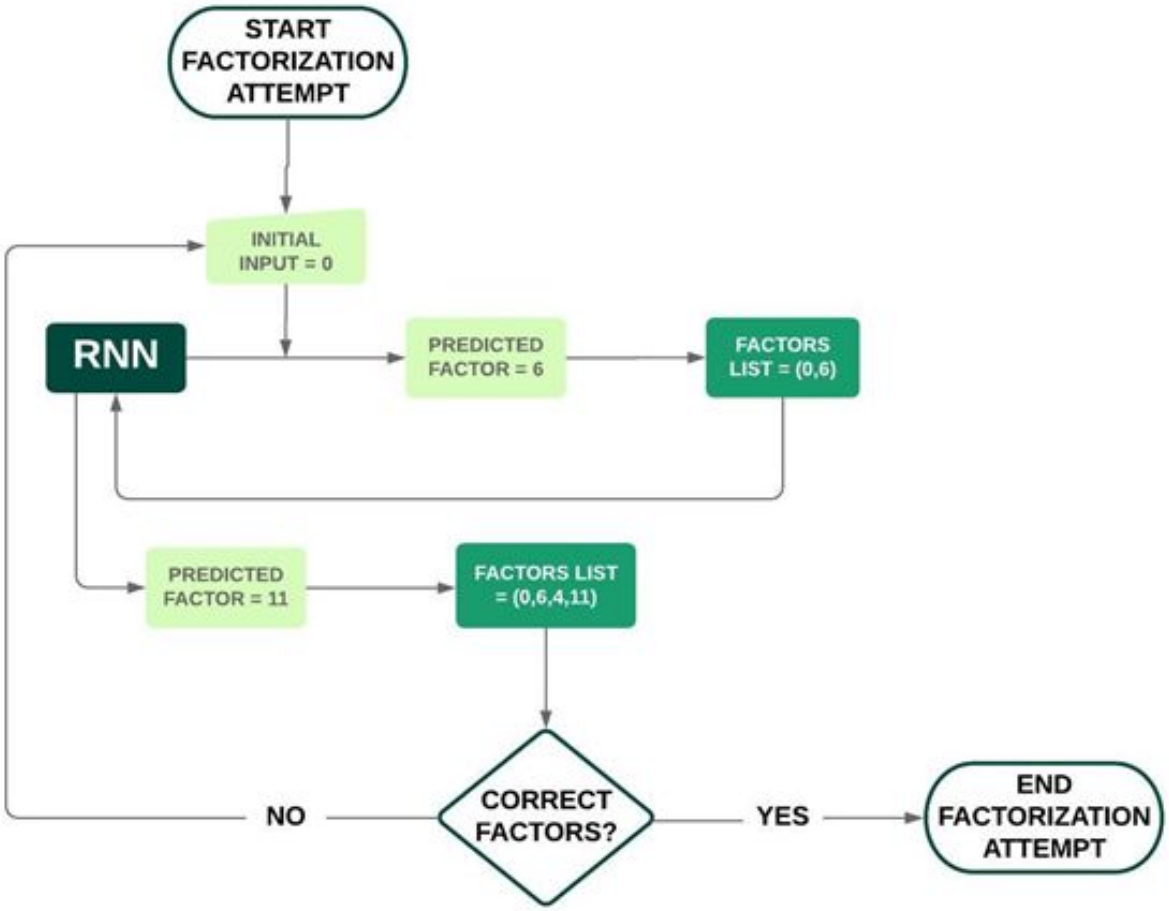

Figure 4. Recurrent Neural Networks (Fluid Intelligence) in the Time Management Task

The procedure to create virtual "subjects" who complete the virtual Time Management task is simply the combining of the previously developed neural network training procedures. Each virtual "subject" consists of a working memory function consisting of a CNN trained for handwritten digit recognition in the MNIST test set, a set of parameters for the prepotent impulsivity function that impedes the performance of the working memory function, and a fluid intelligence function consisting of an RNN trained to factorize the number 12. Two groups of virtual "subjects" with n = 6 x virtual "subjects" per group were created: a control group with "healthy" working memory and fluid intelligence and an "ADHD" group with "deficient" working memory and fluid intelligence. Prepotent impulsivity[25] was included in all "subjects" in both groups with the control group possessing a set of parameters that leads to lower impulsivity than the "subjects" in the "ADHD" group.

All computations for the Time Management task were performed with Python v3 in a Google Colaboratory Notebook, and the creation and training of CNNs were performed with the Keras Machine Learning Library[26]. Data visualizations were created with R v3.6. ANNs for the virtual "subjects" in the control group were trained with the learning procedures that produced 6x

"healthy" working memory and 6x "healthy" fluid intelligence representations, while ANNs for the virtual "subjects" in the ADHD group were trained with the learning procedures that produced 6x "deficient" working memory and 6x "deficient" fluid intelligence representations. All ANNs (CNNs[25] and RNNs) for each executive function representation possessed an identical architecture across both groups as previously described. "Healthy" working memory representations for six virtual control "subjects" were produced by training each of six CNNs with 2,500 handwritten digit images randomly selected from the MNIST training set and "deficient" working memories for six virtual "ADHD subjects" were produced by training each of six CNNs with 25 randomly selected MNIST training set images. Similarly, "healthy" fluid intelligence representations for six virtual control "subjects" were produced by training each of six basic RNNs for 1,000 epochs to factorize the number 12 and "deficient" fluid intelligence representations for six virtual "ADHD subjects" were produced by training each of six basic RNNs for 75 training epochs. Baseline accuracies of the working memory representations were measured by handwritten digit recognition by CNNs on the MNIST test set while baseline accuracies of the fluid intelligence representations were determined on 100x factorization of 12 attempts by RNNs.

After training and evaluation of baseline performance of all ANNs, the Time Management task was performed by the virtual "subjects" (Figure 5). In the first step of the task, the virtual "subject's" fluid intelligence representation uses a basic RNN to perform a single factor prediction. The identity of this factor is then provided to the virtual "subject's" working memory representation with prepotent impulsivity, and the virtual "subject" begins to identify handwritten digits in randomly selected images from the MNIST test set. The virtual "subject's" prepotent impulsivity function is operational with each identification, and may activate an automatic impulsive response where the virtual "subject" simply responds with its previous handwritten digit recognition output rather than using its working memory representation to identify the handwritten digit with a CNN. The automatic impulsive response becomes more likely as multiple MNIST images containing the same handwritten digit are encountered in succession. This search continues until the predicted factor is located AND correctly identified in the MNIST test set. Once the search is successfully completed, another factor prediction is performed and this sequence of steps continues until a complete factor list is predicted and all of its elements are correctly identified in the MNIST test set. When a correct factor list is predicted one trial of the task is complete. Otherwise, this procedure is repeated until a correct factor list is produced. Each virtual "subject" completed 25 successful factorizations of 12 trials, and a mean MNIST test set images searched count for all the trials was calculated for each "subject".

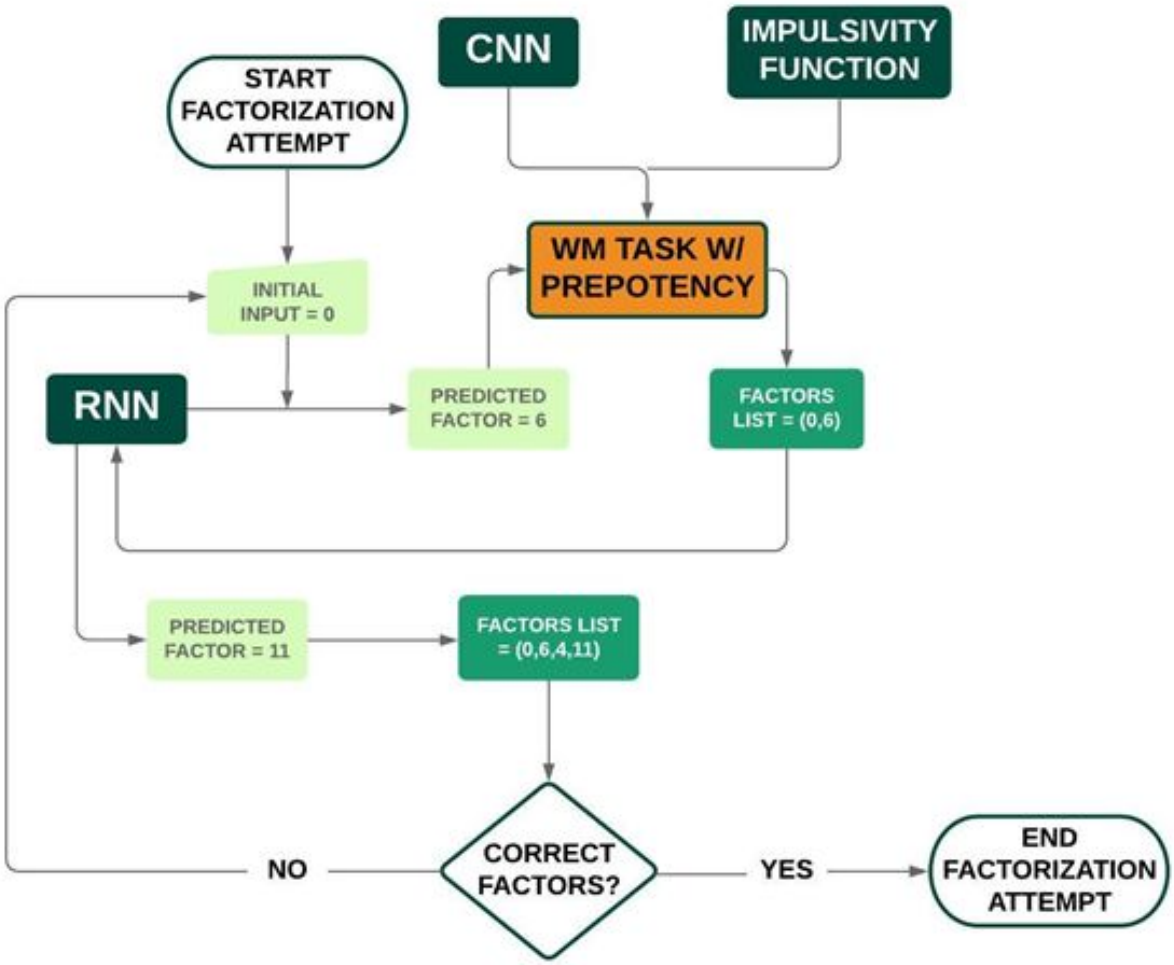

Figure 5. Time Management Task Flowchart

Due to the small sample size, the nonparametric Mann-Whitney U Test was used to test for significant differences between groups in baseline executive function performances and performance in the final Time Management task. A Type I Error Rate $\alpha = 0.05$ was chosen for significance. Since three comparisons were performed, this error rate was maintained by the use of a Bonferroni Correction to adjust the individual p-values by a factor of 3.

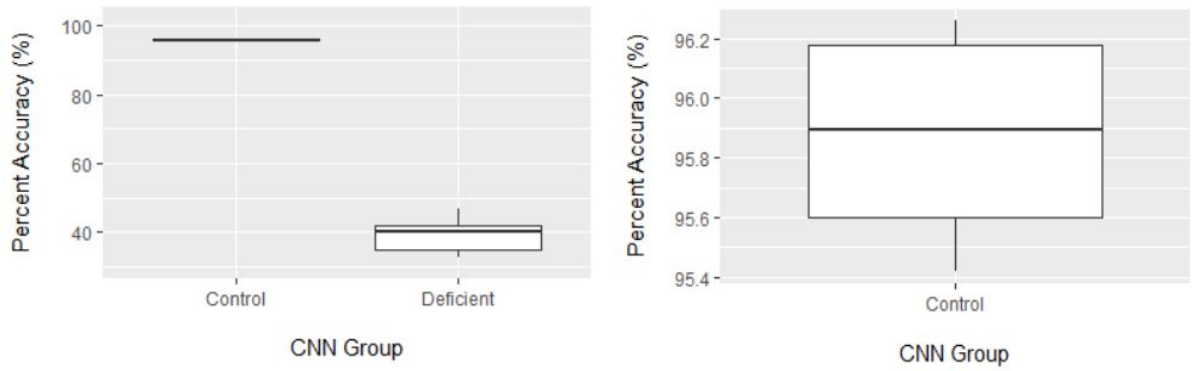

Figure 6. Baseline Difference in Mean Group Accuracy of Handwritten Digit Recognition on MNIST TestSet by Convolutional Neural Networks of Virtual "Subjects" in the Time Management Task (n = 6 x CNNs Per Group)

Results of this experiment are shown in Table 1 and Figures 6-8. As expected, a significant difference was found between both groups in the baseline performance of their executive functions. The health control "subjects" searched through a mean number of 9 to 38 images to complete their 25 trials of the Time Management task. In contrast, the performance of the ADHD "subjects" was far more variable, as they required a mean number of 44-1278 images to complete their trials. This difference between groups in the performance of the Time Management was also found to be statistically significant.

Group	Median Percent Accuracy	P-Value	Adjusted P-Value
CNNs – MNIST Image Identification (Working Memory)			
Control	95.9% (IQR 0.6%)	0.0025	0.0075
Deficient	40.2% (IQR 7.2%)		
RNNs – Factorization of 12 Task (Fluid Intelligence)			
Control	78.5% (IQR 6.0%)	0.0025	0.0075
Deficient	46.5% (IQR 5.5%)		
Virtual "Subjects" – Time Management Task			
Group	**Median of Mean Image Search Counts (Qty Images)**	**P-Value**	**Adjusted P-Value**
Control	32 (IQR 14)	0.0025	0.0075
"ADHD"	282 (IQR 332)		

Table 1. Results Summary of Neural Networks Representing Executive Functions and Virtual "Subjects" in the Time Management Task

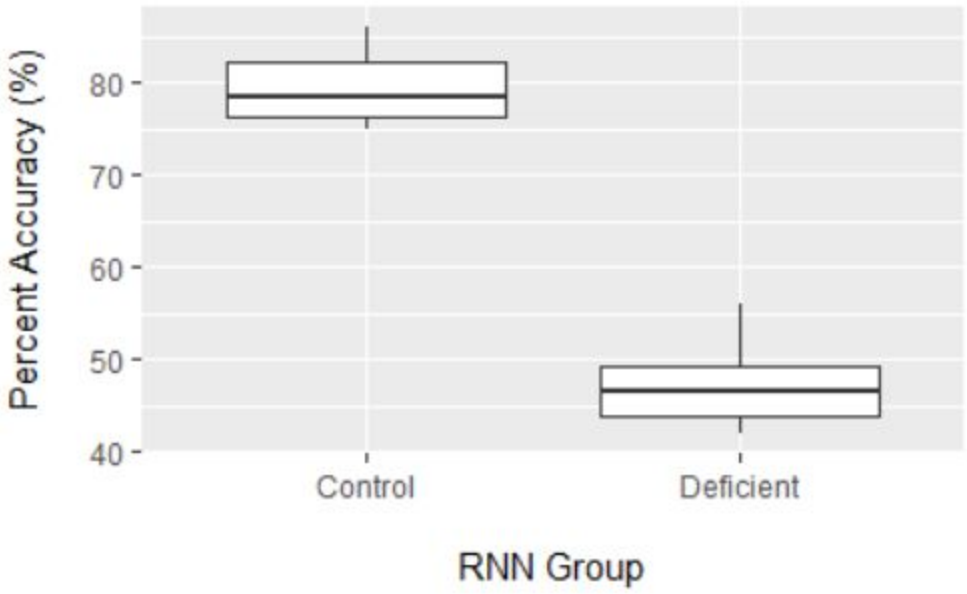

Figure 7. Baseline Difference in Mean Group Accuracy of Factorization of 12 by Recurrent Neural Networks of Virtual "Subjects" in the Time Management Task (n = 6 x RNNs Per Group)

These results indicated that this model composed of virtual "subjects" built from executive function representations of working memory and fluid intelligence with impulsivity that complete a virtual Time Management task that requires the use of these executive function representations was able to produce two groups of "subjects" with distinct levels of performance in the virtual task.

Discussion

In summary, this experiment used a basic RNN model to serve as a simplified model of fluid intelligence, and a procedure was developed to produce "healthy" and "deficient" fluid intelligence performance. This fluid intelligence representation was combined with the previously developed working memory & impulsivity representations to create a virtual "subject" who completed a simulated Time Management task that required the use of these executive function representations. Finally, a significant performance difference between "healthy" and "deficient" virtual "subjects" in this task was demonstrated.

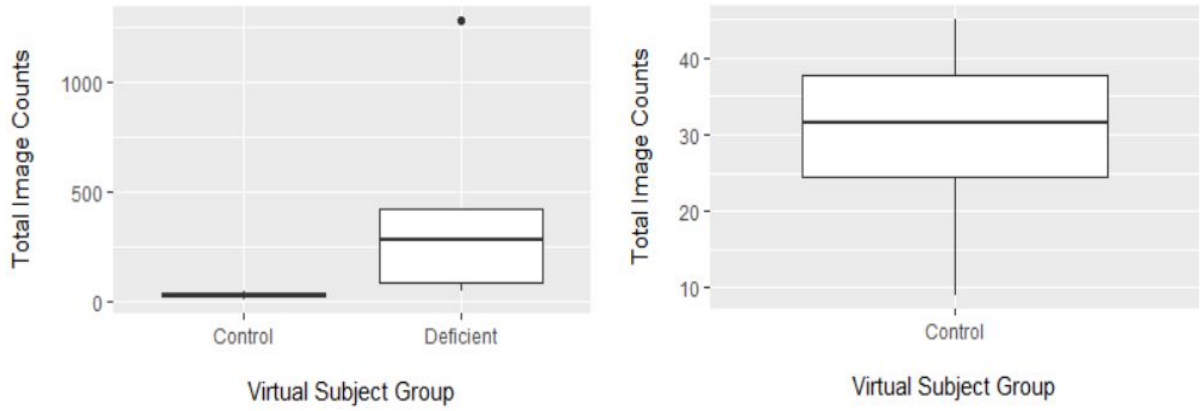

Figure 8. Performance of Virtual "Subjects" in the Time Management Task (n = 6 x Virtual "Subjects" Per Group)

The Factorization of 12 task using basic RNNs to output factor sets of the number 12 was used as a representation for the synthesis component of fluid intelligence in this experiment. This task can be improved by the inclusion of a representation for the process of analysis to create a more comprehensive model of fluid intelligence and by adopting a different variety of RNN to increase the complexity of the task. Consider the process of analysis (breaking down complex ideas) as the inverse of the process of synthesis (building up complex ideas). When the process of synthesis is then represented as the building up of the "complex" number 12 from its simpler component factors, the process of analysis may be represented as the breaking down of the number 12 into its simpler component factors. Thus, the fluid intelligence operation that currently consists of the process of synthesis will be expanded to include an RNN task to break down the number 12 as a representation for the process of analysis.

In the current version of the task, the RNN begins by producing a factor of the number 12, which is then used as an input in the next loop of the task for the RNN to output another factor. This process continues until the RNN produces an output to indicate the end of the sequence of factors and then concludes with an evaluation of the accuracy of the sequence. The reverse process can be implemented through the modification of the RNN's training set. The training set will be differentiated into two separate categories, where one category consists of factor sequences with factors ordered from smallest to largest in size and vice-versa for the other category. The training set with examples ordered by increasing size of their factors will train RNNs for the process of synthesis (where the larger number is assembled), while the training set with examples ordered by decreasing size will train RNNs for the process of analysis (where the larger number is broken down).

The fluid intelligence model can be improved by the substitution of the basic RNNs with Long-Short Term Memory networks (LSTMs)[27]. While basic RNNs indiscriminately update every parameter of their internal layer with each step of the sequence they produce, LSTMs include additional "forget" gates in this updating process that screen for and exclude the passage of extraneous information to the internal layer between steps in the sequence. This increased efficiency of updating permits the processing of larger sequences in LSTMs, and thus LSTMs

would be beneficial to permit the analysis and synthesis of larger numbers in this project. The capability to process larger numbers in this way would allow the creation of multiple difficulty levels for the task to more accurately capture the properties of the therapeutic video games tasks which have multiple difficulty levels.

A comparison of this representation of fluid intelligence to alternative modeling approaches is difficult due to a paucity of research related to the neural functions that produce this executive function. The function of a network of brain regions known collectively as the Multiple-Demand System has been proposed to play a general role across multiple complex problem solving tasks in primates[28]. Detailed functional neural network theories are lacking however. While a recent theory has proposed that fluid intelligence results from a widespread, randomly connected network structure of groups of neurons[29], to our knowledge there does not seem to be any research available that examines the computations that these types of networks conduct. However, it is recognized that this representation of fluid intelligence is grossly simplified compared to the underlying biological functions. Nonetheless, an accurate representation of the functional complexity of fluid intelligence may not be a necessary prerequisite for modeling the effects of the learning processes that improve its performance.

In addition to their potential applications in drug discovery and the practice of medicine, ANNs and other machine learning methods have also drawn interest as tools to improve clinical trial execution and design. For example, the patient selection process can be aided by search tools that are able to interpret and synthesize patient health information from a variety of sources, and patient adherence & retention can be improved by wearable technologies that report data from clinical monitoring parameters to machine learning systems that then analyze it[30]. Furthermore, machine learning models are under development to predict disease progression, such as in Alzheimer's disease[31]. These models can be used to inform both patient selection and clinical trial design to increase the likelihood that a given clinical endpoint can be achieved successfully[32]. This work also aims to enhance clinical trial design, but does so in a unique way. Rather than modelling the disease process to optimize the effects of a single treatment during a clinical study, this simulation seeks to model the effects of a variety of treatment options to inform the selection of an optimal intervention.

The Time Management task described here is the first of a series of computational representations of executive function training activities that are part of a therapeutic video game treatment utilizing the online platform Minecraft to treat children with ADHD. This intervention will utilize a novel personalized medicine approach where an individualized treatment regimen consisting of an initial stimulant dose recommendation and schedule of therapeutic video game activities will be determined from the initial ADHD assessment results of new patients. We hypothesize that the training of ANNs can serve as a computational model for the adaptations that biological neural networks undergo in response to practice of activities requiring the use of executive functions. In the model, a set of ANNs will first be trained to resemble the pattern of executive function deficits reflected in a new patient's assessment responses. A large set of virtual tasks resembling the therapeutic video game activities will be tested in the model, and the set of virtual tasks that produces the maximum improvements in the ANNs of the virtual "subject" will predict the analogous set of therapeutic video game tasks most beneficial for the human patient. A future clinical study will test this hypothesis by including an experimental group with a personalized executive function activity training schedule guided by the

computational model that will be compared with a control group completing an executive function activity training intervention that is not guided by the computational model.

References

1: American Psychiatric Association. (2013). Diagnostic and statistical manual of mental disorders (5th ed.). Arlington, VA: American Psychiatric Publishing

2: Molina, B. S., Hinshaw, S. P., Swanson, J. M., Arnold, L. E., Vitiello, B., Jensen, P. S., ... & Elliott, G. R. (2009). The MTA at 8 years: prospective follow-up of children treated for combined-type ADHD in a multisite study. *Journal of the American Academy of Child & Adolescent Psychiatry*, *48*(5), 484-500.

3: Rutledge, K. J., van den Bos, W., McClure, S. M., & Schweitzer, J. B. (2012). Training cognition in ADHD: current findings, borrowed concepts, and future directions. *Neurotherapeutics*, *9*(3), 542-558.

4: Diamond, A. (2013). Executive functions. *Annual review of psychology*, *64*, 135-168.

5: Barkley, R. A. (1997). Behavioral inhibition, sustained attention, and executive functions: constructing a unifying theory of ADHD. *Psychological bulletin*, *121*(1), 65.

6: Cortese, S., Ferrin, M., Brandeis, D., Buitelaar, J., Daley, D., Dittmann, R. W., ... & Zuddas, A. (2015). Cognitive training for attention-deficit/hyperactivity disorder: meta-analysis of clinical and neuropsychological outcomes from randomized controlled trials. *Journal of the American Academy of Child & Adolescent Psychiatry*, *54*(3), 164-174.

7: Sonuga-Barke, E., Brandeis, D., Holtmann, M., & Cortese, S. (2014). Computer-based cognitive training for ADHD: a review of current evidence. *Child and Adolescent Psychiatric Clinics*, *23*(4), 807-824.

8: Koepp, M. J., Gunn, R. N., Lawrence, A. D., Cunningham, V. J., Dagher, A., Jones, T., ... & Grasby, P. M. (1998). Evidence for striatal dopamine release during a video game. Nature, 393(6682), 266-268.

9: Lawrence, V., Houghton, S., Tannock, R., Douglas, G., Durkin, K., & Whiting, K. (2002). ADHD outside the laboratory: Boys' executive function performance on tasks in videogame play and on a visit to the zoo. Journal of Abnormal Child Psychology, 30(5), 447-462.

10: Dovis, S., Van der Oord, S., Wiers, R. W., & Prins, P. J. (2015). Improving executive functioning in children with ADHD: training multiple executive functions within the context of a computer game. A randomized double-blind placebo controlled trial. PLoS One, 10(4), e0121651.

11: Bul, K. C., Doove, L. L., Franken, I. H., Van der Oord, S., Kato, P. M., & Maras, A. (2018). A serious game for children with Attention Deficit Hyperactivity Disorder: Who benefits the most?. *PloS one*, *13*(3), e0193681.

12: Lodder R, Lodder R, Tiitto M, Smith R, Banfield A, Ensor M. A Pilot Study of a Device and Drug Therapy for ADHD. WebmedCentral PAEDIATRICS 2017;8(11):WMC005354

13: Nigg, J. T., Willcutt, E. G., Doyle, A. E., & Sonuga-Barke, EJ. (2005). Causal heterogeneity in attention-deficit/hyperactivity disorder: do we need neuropsychologically impaired subtypes?. *Biological psychiatry*, *57*(11), 1224-1230.

14: Hagan, M. T., Demuth, H. B., Beale, M. H., & De Jesús, O. (1996). *Neural network design* (Vol. 20). Boston: Pws Pub.

15: Liu, X., Faes, L., Kale, A. U., Wagner, S. K., Fu, D. J., Bruynseels, A., ... & Ledsam, J. R. (2019). A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis. *The lancet digital health*, *1*(6), e271-e297.

16: Kuo, W., Häne, C., Mukherjee, P., Malik, J., & Yuh, E. L. (2019). Expert-level detection of acute intracranial hemorrhage on head computed tomography using deep learning. *Proceedings of the National Academy of Sciences*, *116*(45), 22737-22745.
17: Fagerström, J., Bång, M., Wilhelms, D., & Chew, M. S. (2019). Lisep lstm: A machine learning algorithm for early detection of septic shock. *Scientific reports*, *9*(1), 1-8.
18: Kim, S. Y., Kim, S., Cho, J., Kim, Y. S., Sol, I. S., Sung, Y., ... & Kim, K. W. (2019). A deep learning model for real-time mortality prediction in critically ill children. *Critical Care*, *23*(1), 279.
19: Hu, S. B., Wong, D. J., Correa, A., Li, N., & Deng, J. C. (2016). Prediction of clinical deterioration in hospitalized adult patients with hematologic malignancies using a neural network model. *PloS one*, *11*(8), e0161401.
20: Tiitto M, Lodder R. Therapeutic Video Games for Attention Deficit Hyperactivity Disorder (ADHD). WebmedCentral PAEDIATRICS 2017;8(11):WMC005330.
21: Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*.
22: LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., & Jackel, L. D. (1990). Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems* (pp. 396-404).
23: LeCun et al. (1999): The MNIST Dataset Of Handwritten Digits (Images). http://yann.lecun.com/exdb/mnist/
24: Britz D (September 17, 2015). Recurrent Neural Networks Tutorial. WildML. http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/
25: Tiitto M, Lodder R. Working Memory and Impulsivity and Neural Networks. bioRxiv 2020.10.26.355990; doi: https://doi.org/10.1101/2020.10.26.355990
26: Chollet, F., & others. (2015). Keras. GitHub. Retrieved from https://github.com/fchollet/keras
27: Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.
28: Duncan, J. (2010). The multiple-demand (MD) system of the primate brain: mental programs for intelligent behaviour. *Trends in cognitive sciences*, *14*(4), 172-179.
29: Barbey, A. K. (2018). Network neuroscience theory of human intelligence. *Trends in cognitive sciences*, *22*(1), 8-20.
30: Harrer, S., Shah, P., Antony, B., & Hu, J. (2019). Artificial intelligence for clinical trial design. *Trends in pharmacological sciences*, *40*(8), 577-591.
31: Fisher, C. K., Smith, A. M., & Walsh, J. R. (2019). Machine learning for comprehensive forecasting of Alzheimer's Disease progression. *Scientific reports*, *9*(1), 1-14.
32: Romero, K., Ito, K., Rogers, J. A., Polhamus, D., Qiu, R., Stephenson, D., ... & Brown, D. (2015). The future is now: model-based clinical trial design for Alzheimer's disease. *Clinical Pharmacology & Therapeutics*, *97*(3), 210-214.