Research Article

May Kristine Jonson Carlon, Jeffrey S. Cross*

Knowledge tracing for adaptive learning in a metacognitive tutor

https://doi.org/10.1515/edu-2022-0013
received October 29, 2020; accepted March 9, 2022.

Abstract: Adaptive learning is provided in intelligent tutoring systems (ITS) to enable learners with varying abilities to meet their expected learning outcomes. Despite the personalized learning afforded by ITSes using adaptive learning, learners are still susceptible to shallow learning. Introducing metacognitive tutoring to teach learners how to be aware of their knowledge can enable deeper learning. However, metacognitive tutoring on top of cognitive tutoring can lead to unsustainable cognitive loads. Using metacognitive inputs for knowledge tracing was explored for managing cognitive loads. Hidden Markov models (HMM) and artificial neural networks were used to train models on a synthetic dataset created from predetermined learner personas. The models created with metacognitive inputs were compared with the models created without said inputs. The models using metacognitive inputs performed better than the standard models while still following learning intuitions. This indicates that combining knowledge tracing and metacognitive tutoring is a viable option for improving learning outcomes. This is an important finding since online learning, which demands metacognitive skills, is becoming popular for various topics, including those that are challenging even with immediate teacher assistance.

Keywords: knowledge tracing; hidden Markov models; artificial neural networks; intelligent tutoring systems; metacognition.

1 Introduction

There is more to learning than meets the eye. In most educational environments, learning assessment activities using quizzes and homework are used to evaluate learners' knowledge and comprehension. However, a learner's performance on these activities can be influenced by factors other than learning. These can include the assessment material's quality, learner's environmental conditions, or emotional state during the assessment. Researchers have been using latent variable models to reveal attributes hidden in observable phenomena. For instance, item response theory (IRT) is a popular latent variable modeling technique that uses learner responses to give insight into assessment item difficulty, learner ability, and learning estimate, among others (Deonovic, Yudelson, Bolsinova, Attali, & Maris, 2018).

Because of IRT's ability to differentiate learners, it has been exploited by intelligent tutoring systems (ITS) to introduce adaptive learning. However, since IRT is mainly used to assess the quality of the evaluation, it has become inherently cross-sectional (taking a snapshot of learning states) as opposed to being longitudinal (being able to track the progression of learning), which is more suitable for ITSes. Knowledge tracing algorithms that decide whether the learner needs more exercises to master a module or can already move on to succeeding modules are developed to fit ITSes better. Knowledge tracers are also latent variable models that treat learner mastery as its latent variable.

Knowledge tracing enables personalized learning as the pacing for each learner is adapted according to their abilities. However, despite ensuring the learners were exposed to sufficient exercises, there is a possibility that the learning is still not robust even with knowledge tracers (Baker, Gowda, Corbett, & Ocumpaugh, 2012). The learner may only master the skill but have difficulty applying the current learning to future learnings. Developing metacognitive skills can be crucial to overcoming this shallow learning (Aleven & Koedinger, 2002).

***Corresponding author: Jeffrey S. Cross,** Tokyo Institute of Technology, Meguro, Tokyo, Japan, E-mail: cross.j.aa@m.titech.ac.jp
May Kristine Jonson Carlon, Tokyo Institute of Technology, Meguro, Tokyo, Japan

2 Background

Metacognition, or the knowledge and regulation of one's thinking process, includes skills such as goal setting and knowledge monitoring, among others (Flavell, 1979). This can be seen through various manifestations. Examples include learners realizing that they do not understand the topic enough to explain it in their own words or learners deciding to create to-do lists to help them organize their learning activities. Multiple research studies have shown that metacognition contributes to learners' academic performance and improves their learning (Ohtani & Hisasaka, 2018). This is even more important with the emergence of online learning, which might be here to stay long after the need for social distancing measures imposed in the face of the COVID-19 pandemic (Gallagher & Palmer, 2020) are no longer needed. Metacognitive skills allow learners to calibrate their learning and are better learning predictors in online learning environments than other factors such as time spent on assignments (Zhao & Ye, 2020).

However, creating a tutoring system that effectively teaches metacognitive skills to students is challenging. Training for metacognition is practical when done in context, such as learning a cognitive domain-specific skill (e.g., mathematics, language, and others) alongside. This puts a strain on the learners' cognitive resources (Roll, Aleven, McLaren, & Koedinger, 2007). The learners must then spend effort on gaining metacognitive skills on top of learning in the cognitive domain. Fortunately, research on applying adaptive learning to metacognitive instruction already exists (Agustianto, Permanasari, Kusumawardani, & Hidayah, 2016). Research studies show that shallow learning could be addressed by metacognitive tutoring in a cognitive tutor (Baker, Gowda, Corbett, & Ocumpaugh, 2012). Nevertheless, research on adaptive learning for metacognitive instruction alongside cognitive instruction is yet to be conducted.

When metacognitive instruction is done alongside cognitive instruction, learners might concentrate more on mastering the cognitive content. Cognitive development will be more visible to the learners through markers such as higher grades, making it more important for them. Metacognitive development will be harder to see, especially when the learners cannot apply their learning outside the tutoring environment. As such, developing metacognitive skills can be easily taken for granted when cognitive resources seem to be just enough for the cognitive part. What remains to be investigated is how to combine metacognitive tutoring and cognitive adaptive learning to manage cognitive resources.

Bayesian Knowledge Tracing (BKT) is a versatile adaptive learning algorithm to which several researchers have previously introduced modifications. Some examples include estimating learner's prior knowledge based on the correctness of their first response (Pardos & Heffernan, 2010), estimating a problem's difficulty in a traditional setting (Pardos & Heffernan, 2011) and in a massive open online classroom (MOOC) setting (Pardos, Bergner, Seaton, & Pritchard, 2013), individualizing prior knowledge and learning rate estimates (Yudelson, Koedinger, & Gordon, 2013), and even using brain scans as observation inputs (Halpern et al., 2018).

Artificial neural networks are another set of algorithms that is recently gaining traction among adaptive learning researchers. These are typically composed of input and output layers connected by hidden layers. A particular interest is in using deep learning or neural networks with more than a single hidden layer.

3 Related Work

3.1 Reflection Assistant

One tool that allows for metacognitive development is the RA model created by Gama. The RA focuses on selected metacognitive skills and builds metacognitive profiles to aid metacognitive development (Gama, 2004). The RA is integrated into a cognitive tutoring environment and aims to establish the connection between their metacognitive and cognitive performance in the learners' minds. This makes the RA an ideal candidate for an investigation: it is situated within a cognitive learning context. It aims to teach metacognition, not just to enable learners to use metacognitive skills but also to improve their cognitive learning.

The RA is based on the hierarchical model of metacognition and the conceptual stages of problem-solving. The hierarchical model of metacognition is a hierarchy of skills composed of Planning, Selecting Strategies, Evaluating Learning, Knowledge Monitoring, and Controlling (Tobias & Everson, 2002). The idea of conceptual stages of problem-solving suggests that a typical problem-solving session will involve preparation to solve the problem, actual problem solving, and verification of problem-solving (Halpern D. F., 2013).

The RA was designed to include prompts where learners can provide free-form short answers in the preparation stage. These prompts invoke Planning and Selecting Strategies metacognitive skills. Another set

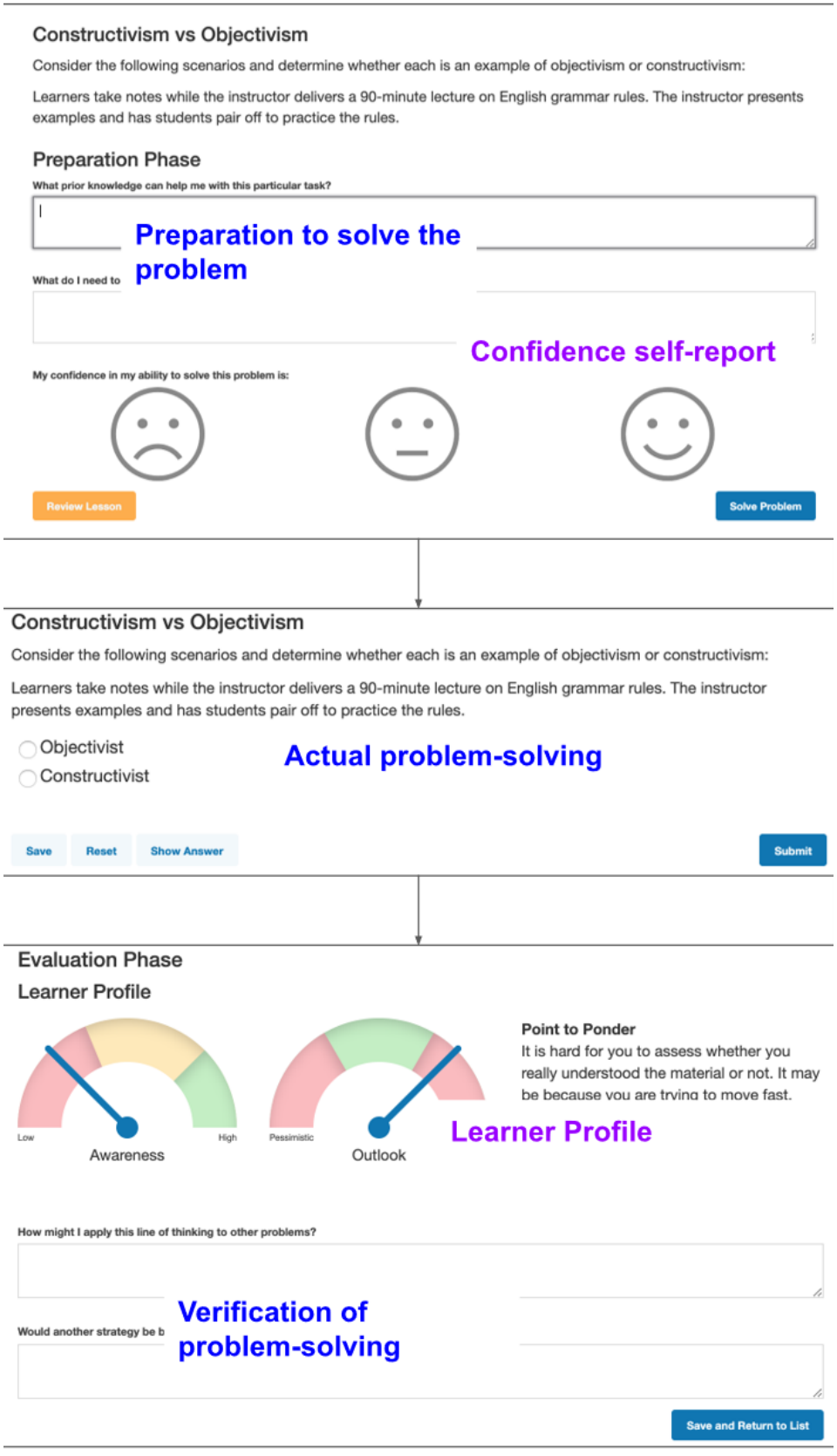

Figure 1: The interface of a web-based tutor under development following the RA model.

of prompts is presented in the verification stage, this time targeting the Evaluating Learning and Knowledge Monitoring metacognitive skills. A Learner Profile is also shown in the verification stage. Figure 1 shows the user interface of the metacognitive tutor that is a simplified version of the RA model.

The Learner Profile includes the Knowledge Monitoring Accuracy (KMA) to measure a learner's Knowledge Monitoring metacognitive skill previously developed by a different research group (Tobias & Everson, 2002). To calculate the KMA, the learners are asked during the preparation stage about their confidence - can answer correctly (C), can partially answer (P), and cannot answer (I) - in their ability to answer said exercises. The self-reported confidence is then compared with their actual performance, and the cumulative metrics Fully Correct (FC), Partially Correct (PC), Fully Incorrect (FI) are updated according to Table 1 for every problem-solving opportunity.

Table 1: *M*atrix of values for KMA.

Score	Confidence Self-Report		
	C	P	I
Correct	FC	PC	FI
Partially Correct	PC	FC	PC
Wrong	FI	PC	FC

KMA is then computed using the following formula:

$$KMA = \frac{FC - 0.5 * PC - FI}{FC + PC + FI} \quad (1)$$

To note, fully correct predictions receive full positive credit and fully incorrect predictions receive full negative credit. Partially correct predictions receive negative half credit instead of positive half credit to prevent incentivizing not taking a stance. The resulting KMA has values ranging from -1 to 1. The closer the value to 1 is, the better is the learner's performance in terms of KMA. The learners are classified from the computed KMA scores, whether they have low, average, or high KMA. In the case where only a single partially correct prediction is recorded, the KMA is -.5, indicating low awareness.

Gama further expanded the quantification of Knowledge Monitoring by adding the Knowledge Monitoring Bias (KMB) to show the learner's optimistic or pessimistic tendencies (Gama, 2004). Just like KMA, confidence self-reports are used in computing KMB. The cumulative metrics No Bias (NB), Partial Pessimistic Bias (PPB), Full Pessimistic Bias (FPB), Partial Optimistic Bias (POB), and Full Optimistic Bias (FOB) are updated according to Table 2 for every problem-solving opportunity.

Table 2: Matrix of values for KMB.

Score	Confidence Self-Report		
	C	P	I
Correct	NB	PPB	FPB
Partially Correct	POB	NB	PPB
Wrong	FOB	POB	NB

KMB is then computed using the following formula:

$$KMB = \frac{FOB + 0.5 * (POB - PPB) - FPB}{FOB + POB + NB + PPB + FPB} \quad (2)$$

Just like the KMA, the KMB has values ranging from -1 to 1. This time, the closer the value to 0 is, the better is the learner's performance in terms of KMB. Positive values are associated with optimism, while negative scores are associated with pessimism. Hence, values closer to 0 indicate less bias. When the KMA is not high, the learners are further classified as having a pessimistic, random, or optimistic outlook. The score ranges used for classification are detailed in Table 3.

Table 3: Classification according to KMA and KMB values.

Score Range	Classification	
	KMA	KMB
[-1, -0.25)	Low	Pessimistic
[-0.25, 0.25)	Average	Random
[0.25, 0.5)	Average	Optimistic
[0.5, 1]	High	Optimistic

The RA presents the KMA and KMB classifications to the learners as feedback. The RA aims to improve the learners' KMA and KMB scores through constant practice using the prompts and constant Learner Profile feedback. The RA's original research has shown that the RA improved the learners' performance, time management skills, and knowledge monitoring ability (Gama, 2004).

The modified RA illustrated in Figure 1 was tested from September 2020 to February 2021 on an undergraduate electrical engineering class delivered in

Initial Probability Distribution	
Unmastered	Mastered
1 - $P(L_0)$	$P(L_0)$

Transition Probability Matrix		
	Unmastered	Mastered
Unmastered	1 - P(T)	P(T)
Mastered	P(F) = 0	1 - P(F)

Emission Probability Matrix		
	Correct	Wrong
Unmastered	P(G)	1 - P(G)
Mastered	1 - P(S)	P(S)

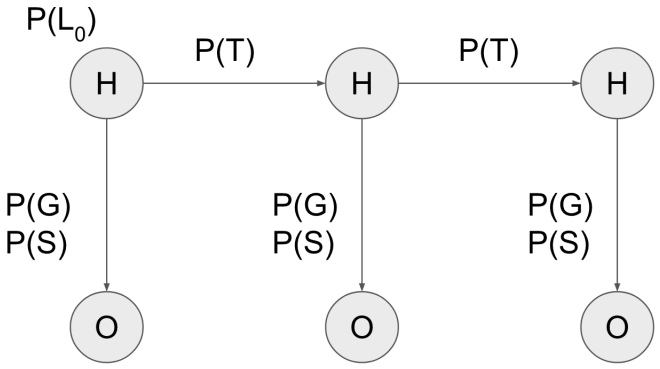

Figure 2: Graphical representation of BKT as HMM.

a blended learning format participated in by 29 learners. The learners indicated that the modified RA is usable, and the results show that it could improve learners' regulation of cognition.

The RA did account for the learners' cognitive load by introducing the metacognitive tutoring during the preparation and verification phases. Metacognitive tutoring is not conducted during the actual problem-solving stage where the demand for cognitive resources is expected to be the highest. However, its metacognitive tutoring is still on top of cognitive tutoring. Thus, while the cognitive load may not be as much as other metacognitive tutors integrated with cognitive tutors, it is still an additional burden to the learners.

3.2 Bayesian Knowledge Tracing

BKT is a knowledge tracing algorithm that is a hidden Markov Model (HMM) where learner knowledge is represented as a binary variable (whether a knowledge component is mastered or not) for each knowledge component (Corbett & Anderson, 1994). A graphical representation of BKT is shown in Figure 2.

An HMM is composed of the following components:

- A set of hidden **states**. For BKT, the states are whether the knowledge component is **mastered** or **unmastered**.
- A **transition probability matrix** that indicates the probability of transitioning from one state to another (e.g., from unmastered to mastered).
- An **initial probability distribution** that indicates the probability of starting at a hidden state (e.g., when the learner has prior knowledge).
- A sequence of **observations** drawn from a vocabulary. For BKT, the vocabulary includes whether the answer is correct = 1 or wrong = 0.
- An **emission probability matrix** that indicates the probability of an observation being generated from a hidden state (e.g., when the learner answers correctly by guessing).

As an HMM, the following assumptions are held:

- The probability of a particular hidden state depends on the previous hidden state only.
- The observations are conditionally independent of all other variables given their current hidden state.

BKT fits the following knowledge parameters, which are used in the HMM's transition probability matrix and initial probability distribution:

- $p(L_0)$ **Initial Learning**: Probability that the knowledge component is already mastered even before the first opportunity to solve a problem is presented,
- p(T) **Acquisition**: Probability that the knowledge component is mastered from solving the problem, and

– p(F) **Forget**: Probability that the knowledge component was previously mastered but is not currently mastered. This is traditionally set to 0 and is not included among the calculated parameters.

Additionally, the following performance parameters are also fitted, which are used in the HMM's emission probability matrix:

– p(G) **Guess**: Probability that the knowledge component is not yet mastered, but the learner was able to apply it correctly on the problem, and
– p(S) **Slip**: Probability that the knowledge component is already mastered, but a mistake was made when applying it to the problem.

The correctness of the learner's response at opportunity n can be predicted with the following equation, where * indicates multiplication and ¬ indicates negation:

$$p(Correct_n) = p(L_n) * p(\neg S) + p(\neg L_n) * p(G) \quad (3)$$

The probability that a knowledge component is mastered given that the problem is correctly answered is usually inferred using the following formula:

$$p(L_n|correct) = \frac{p(L_n) * p(\neg S)}{p(L_n) * p(\neg S) + p(\neg L_n) * p(G)} \quad (4)$$

When the answer is wrong, the following instead is used:

$$p(L_n|wrong) = \frac{p(L_n) * p(S)}{p(L_n) * p(S) + p(\neg L_n) * p(\neg G)} \quad (5)$$

The following equation gives the probability that the knowledge component is then mastered on the following problem:

$$p(L_n) = p(L_{n-1}|obs_{n-1}) + p(\neg L_{n-1}|obs_{n-1}) * p(T) \quad (6)$$

where obs_{n-1} is the observation (correct or wrong) at opportunity n-1.

One risk of introducing **Guess** and **Slip** parameters that offer counter-intuitive explanations for observed behavior is model degeneracy: the resulting model may not behave as it was intended to be (Baker, Corbett, & Aleven, 2008). For knowledge tracing, model degeneracy occurs when the link between learner knowledge and learner performance is lost. For example, despite the learner performing well on problems, the model still predicts the learner has not mastered the knowledge component because **Guess** is given more weight than **Acquisition**. Model degeneracy is likely to occur when either $p(S)$ or $p(G)$ is greater than 0.5.

Model degeneracy can be rare in practice. However, several factors, such as confusingly worded questions, can result in model degeneracy (Doroudi & Brunskill, 2017). A measure to avoid model degeneracy is bounding $p(S)$ and $p(G)$ to a small range of values, with some choosing to fix the values for said parameters. However, this exposes the problem of deciding the best values for $p(S)$ and $p(G)$ as their values will affect the other parameters. Also, fixing values for $p(S)$ and $p(G)$ deprives the chance to investigate the factors causing model degeneracy.

3.3 Knowledge Tracing with Deep Learning

Artificial neural networks such as the one shown in Figure 3 had been drawing interest among researchers since deep knowledge tracing (DKT) using recurrent neural networks was introduced (Piech et al., 2015). Artificial neural networks are conceptually derived from biological neurons, where neurons have inputs that can produce outputs through activation and are passed on to other neurons (McCulloch and Pitts, 1943).

An artificial neural network can be characterized by the following:

– An **input layer** with at least one node. In Figure 3, I1 and I2 correspond to the nodes of the input layer.
– An **output layer** with at least one node. In Figure 3, O1 and O2 correspond to the nodes of the output layer.
– At least one **hidden layer**. In Figure 3, there are two hidden layers, with each of the hidden layers having three nodes. A neural network learns through each layer; thus, more layers (i.e., deeper) may be better models if sufficient data is used for training. A sample heuristic being used is about tens of individual samples for each parameter to be estimated, essentially the weights.
– Nodes are connected with synapses associated with **weights**. In Figure 3, the weights were visualized with the connecting lines' thickness, where black lines are positive weights, and gray lines are negative weights. During the training of a neural network model, the combination of nodes and layers is used as a starting point. The corresponding weights (and optionally, the bias values) are calculated through a series of adjustments. Passing the inputs through the series of the weighted nodes and activation function is closest to the expected output.

– When predicting an output given an input, the connections' weighted sums are passed through an **activation function**. The activation function decides whether a node should fire or not; hence its return values are mostly either just 0 or 1.
– A **bias** value may also be needed to shift the activation function. In Figure 3, B1, B2, and B4 are the bias.

4 Proposed Mechanism

A system where metacognitive tutoring is done on top of cognitive tutoring while employing adaptive learning is proposed. Figure 4 illustrates such a system, where a learning management system (LMS) hosts the cognitive tutor and RA as the metacognitive tutor. An adaptive engine influences how the learner interacts with the LMS based on the metacognitive inputs by using knowledge tracing.

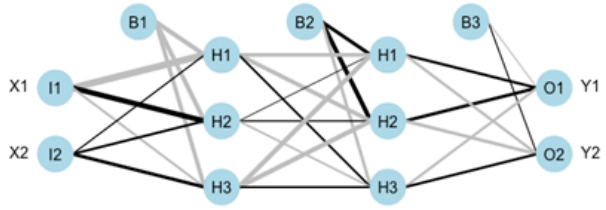

Figure 3: Sample of a deep neural network.

4.1 BKT with RA: RA-BKT

There are a myriad reasons a learner's response might be attributed to a guess or slip. Simple errors, fatigue, or the learner giving up due to frustration with how a problem is worded despite mastering the associated knowledge may all lead to slips. On the other hand, guesses can be attributed to sheer chance, assessment tool errors, or confusing a related theory with the intended theory and somehow arriving at the correct answers.

When combining RA and BKT, it may be thought that when the learner has high KMA and predicted that they could answer the problem but could not do so, then it is more likely that the incorrect answer was due to a slip. It is probably not due to the learner not being able to learn the knowledge component yet. Similarly, the case when the learner who has an optimistic KMB and a low KMA predicted that they could answer the problem and were indeed able to answer correctly must be considered. It cannot just be as quickly assume that the learner has already learned the knowledge component because it may be likely that they simply guessed the answer correctly.

It is tempting to include the KMA and KMB in the formulation of the BKT. The most straightforward way is adding the KMA and KMB values (or their resulting classifications) in the observation vocabulary. However, since KMA and KMB are cumulative values across opportunities, adding them to the observation vocabulary

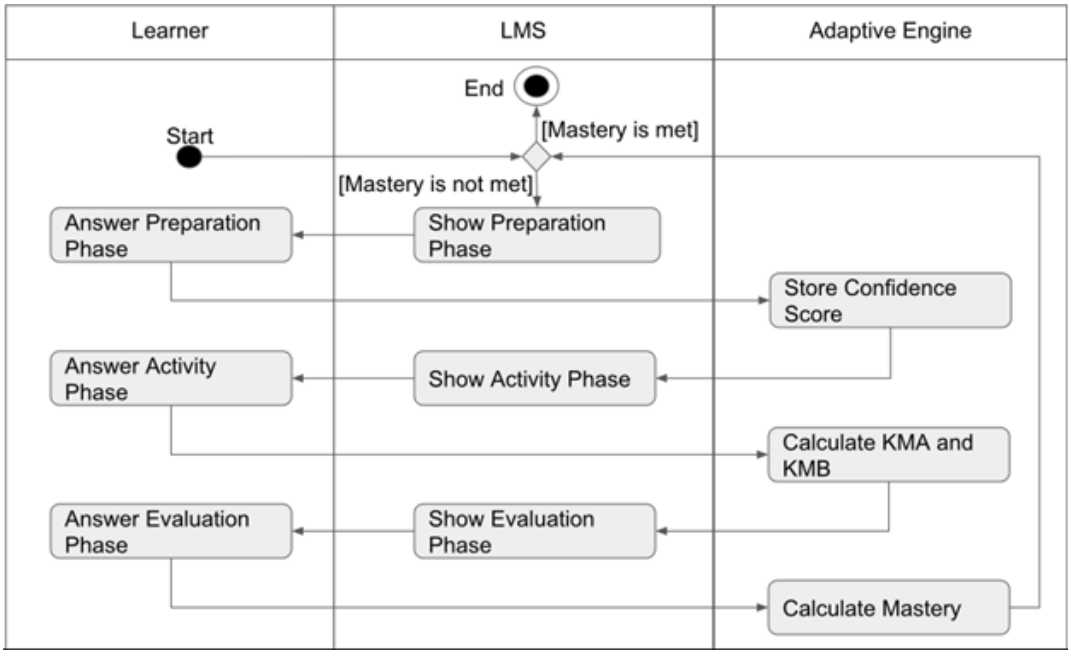

Figure 4: Envisioned activity diagram for RA model usage with an adaptive engine.

would violate the HMM assumption that observations are conditionally independent.

The combination of confidence self-reports and performance is used for the observation vocabulary to preserve the observations' conditional independence. Instead of just having {correct = 1, wrong = 0}, the observation vocabulary is made up of the combination set {confidence_performance}. Following from RA where learners can predict that they can answer correctly, partially correctly, and incorrectly, the observation vocabulary will be {c_0, p_0, i_0, c_1, p_1, i_1} where *c*, *p*, and *i* corresponds to learners predicting to answer correctly, partially correctly, and incorrectly respectively. For the performance, 0 and 1 correspond to learners answering the problem incorrectly and correctly, respectively. This BKT reconstruction is referred to in this paper as RA-BKT.

The new emission probability matrix for RA-BKT is shown in Table 4. While BKT has four parameters, RA-BKT will have 14 parameters. The knowledge parameters which make up the probability transmission matrix will be the same since the hidden states are not changed. On the other hand, the performance parameters will be replaced by the new emission probability matrix items.

Table 4: RA-BKT emission probability matrix.

State	Observation					
	***c*_0**	***p*_0**	***i*_0**	***c*_1**	***p*_1**	***i*_1**
Unmastered	$p(U_{c\_0})$	$p(U_{p\_0})$	$p(U_{i\_0})$	$p(U_{c\_1})$	$p(U_{p\_1})$	$p(U_{i\_1})$
Mastered	$p(M_{c\_0})$	$p(M_{p\_0})$	$p(M_{i\_0})$	$p(M_{c\_1})$	$p(M_{p\_1})$	$p(M_{i\_1})$

The RA allows for being able to answer the problem partially correctly, but for parallelism with the standard BKT, partially correct answers are not considered. Computations for KMA and KMB are still the same even if there are no partially correct answers. From here on, KMA and KMB will be referred to as learner awareness and outlook, respectively.

4.2 ANN with RA: RA-ANN

The formulation derived above still did not allow for taking advantage of the KMA and KMB values in predicting whether mastery is achieved or not. Artificial neural networks do not have the conditionally independent restriction on inputs as in BKT. Therefore, KMA and KMB may be used alongside the answers and confidence reports for input. Neural networks that use the number of chances given to the learner (i.e., opportunities) and another that uses all other inputs were created to parallel comparisons with BKT. These are called ANN and RA-ANN, respectively. The neural networks were constructed as classification problems: that is, the neural networks will predict whether the learner will answer the problem correctly or not. The problem construction is summarized in Table 5.

Table 5: *Input and output layers for the neural network models.*

Layer	ANN	RA-ANN
Input	Opportunity	Opportunity Predicts to answer incorrectly* Predicts to answer partially correctly* KMA KMB
Output	Answer	Answer

** [Predicts to answer correctly] is no longer used as input as it can be inferred from these inputs.*

5 Method

The conducted research is not related to either human or animals use. The following questions are to be answered:

- How do the models compare with each other in terms of training efficiency and accuracy?
- Which model might be the best in reducing cognitive load?
- How closely will the models follow learning institutions (e.g., model degeneracy; the relationship between awareness, outlook, and mastery)?

5.1 Dataset

A dataset was created for this experiment by defining learner personas. Table 6 lists possible learner behaviors based on their performance (8 behaviors) and confidence report (13 behaviors). The performance and confidence report behaviors are combined, initially resulting in 104 learner data.

In this dataset, each learner has ten opportunities to answer the problem or demonstrate learning the component. This formulation's imagined setup is a quiz with ten items for a knowledge component that the learner answers once. Alternatively, this can also mean a single problem that the learner can attempt to answer up to ten times. In most cases, the situation will be somewhere

Table 6: Assumed learner behaviors.

Performance	Confidence Report
– Always answers correctly	– Always predicts to answer correctly
– Always answers incorrectly	– Always predicts to answer partially correctly
– Occasionally answers correctly	– Always predicts to answer incorrectly
– Occasionally answers incorrectly	– Always predicts to answer correctly but occasionally predicts to answer partially correctly
– Progressively performs better	– Always predicts to answer correctly but occasionally predicts to answer incorrectly
– Regresses in performance	– Always predicts to answer partially correctly but occasionally predicts to answer correctly
– Progressively performs better then regresses in performance	– Always predicts to answer partially correctly but occasionally predicts to answer incorrectly
– Regresses in performance then progressively performs better	– Always predicts to answer incorrectly but occasionally predicts to answer correctly
	– Always predicts to answer incorrectly but occasionally predicts to answer partially correctly
	– Progressively improves in prediction
	– Regresses in prediction
	– Progressively improves then regresses in prediction
	– Regresses then improves in prediction

in between the said scenarios. BKT's assumption that learners gain more knowledge the more opportunities they are given (supposing the learner exerts honest effort to learn and is not merely gaming the system) is to be followed. Having ten opportunities is reasonable without being overbearing if the first interpretation (a ten-item quiz for every knowledge component) is applied.

The following data were also created to prevent model degeneracy:

- The performance behavior where the learner progressively improves is repeated. Data were created such that the learner initially gets the answer wrong and consistently gets the answer correctly afterward to show learner performance improvement. The improving performance condition is repeated for each opportunity count, where the learner begins to answer correctly from opportunity N, with N between 2 to 10. Combining these nine new conditions with the 13 confidence report behaviors adds 117 more data.
- The same is done for confidence report behavior, where the learner gets the prediction right from opportunity N, with N between 2 to 10. The confidence report allows for a partially correct prediction. For example, for $N = 2$, there is the case where the learner is predicting either incorrectly or partially correctly during the first opportunity. Hence, this condition is repeated ten times instead of just nine. Other than $N = 2$, the learner is set to have a partially correct prediction for opportunities above floor(N/2) but below N. After combining with the initial eight performance behaviors, 80 additional learner data were created.
- Additionally, the combination of improving performance prediction (nine times) and improving confidence prediction (ten times) is repeated. This added 90 learner data, finally resulting in 391 learner data.

In an actual learning scenario, the learners will be exposed to more than one knowledge component. However, for BKT, each knowledge component is modeled separately; each knowledge component model does not affect other knowledge components. While it may be argued that related knowledge components could be affecting each other, this case could be handled by estimating for the initial mastery. Having more than one knowledge component is thus inconsequential. Hence, only data for a single knowledge component is created.

The 391 learners were randomly assigned to five groups, with four groups combined as a model training dataset and the remaining group reserved as a validation dataset. The resulting division between training and validation sets is detailed in Table 7, where the distributions across key factors are relatively even. Thus, unbalanced data, which is typically an issue in machine learning, is not a significant concern for this research. With 391 learner data having ten opportunities each, there is a total of 3910 observation records.

Table 7: The distribution between training and validation sets.

Description		Training	Validation
Data count		3120 (79.79%)	790 (20.20%)
Answer			
-	0	1553 (49.77%)	366 (46.32%)
-	1	1567 (50.22%)	424 (53.67%)
Confidence			
-	C	1391 (44.58%)	279 (35.31%)
-	P	770 (24.67%)	275 (34.81%)
-	I	959 (30.73%)	246 (31.13%)

During test data creation, no assumptions were made about the relationship between the learner's prediction accuracy and their current mastery level. This is in line with the general assumption that metacognition is a domain-independent skill. Based upon the above information, the following columns were created for the dataset:

- **learner**: An identifier for the learner.
- **skill**: The knowledge component; currently, only one knowledge component is created with the assigned value "A." This information is not used for the current experiment.
- **opportunity**: Answer opportunity; an opportunity results in one observation. Currently, ten opportunities are created for each learner.
- **confidence**: The confidence self-report; possible values are C=predicted to be correct, P=predicted to be partially correct, and I=predicted to be incorrect.
- **answer**: The performance; possible values are 0 = wrong and 1 = correct.
- **awareness**: The KMA score.
- **outlook**: The KMB score.
- **train**: Whether the data will be used for training (0) or validation (1).

The trends between opportunities and correctness, awareness, and outlook using generalized linear modeling were visualized to ascertain how realistic the synthetic data is. The upward trend that we made in the assumption is visible in Figure 5. First, both awareness and outlook values were adjusted to limit the range of values to [0, 1], with 1 being the best value, just like correctness. The awareness was normalized to change the range of values from [-1, 1] to [0, 1]. Simultaneously, the corrected outlook was taken to be 1 - |*value*| since the original outlook values are also in the [-1, 1] range, with 0 being the desirable value.

The resulting trends were compared with the student data from ASSISTments (2015) and the Geometry Angles dataset. The ASSISTments is a widely used platform for learning research (Heffernan & Heffernan, 2014), while the Geometry Angles dataset resulted from considerable research on metacognition (Aleven & Koedinger, 2000). The attribute "Opportunity" was introduced to both datasets after sorting by Log ID attribute for ASSISTments and ID for Geometry Angles to ensure parallel comparison. The Opportunity is then incremented, starting from 1 for each Student and Problem combination. The data are further filtered such that only the Student and Problem combinations with exactly ten Opportunity counts are included. Both datasets exhibit the same upward trend, thus matching the trends of the synthetic dataset.

5.2 Modeling

For training the BKT variants, an existing R code for BKT (Nixton & Yudelson, 2013) was modified to accommodate the RA-BKT. The correspondence between the BKT and RA-BKT parameters is shown in Table 8. The Baum-Welch algorithm, a case of the Expectation-Maximization (EM) algorithm that uses the forward-backward algorithm for the Expectation step (Baum, 1972), was used for model fitting. In expectation-maximization, the parameters are first initialized randomly, and the expected values of observations are computed based on the parameter values (Dempster, Laird, & Rubin, 1977). The expected values are compared against the actual observations, and the parameters are re-calibrated based on the comparison. A threshold of 1×10^{-9} and a maximum step of 100 is set. The iteration continues until the difference between the previous and the current results is less than the threshold, or when 100 iterations are met. A seed is set for the pseudo-random number generator to ensure reproducibility.

Table 8: BKT and RA-BKT correspondence.

BKT		RA-BKT
$p(L_0)$	⇔	$p(L_0)$
$p(T)$	⇔	$p(T)$
$p(G)$	⇔	$p(U_{c\_1}) + p(U_{p\_1}) + p(U_{i\_1})$
$p(S)$	⇔	$p(M_{c\_0}) + p(M_{p\_0}) + p(M_{i\_0})$
$1 - p(G)$	⇔	$p(U_{c\_0}) + p(U_{p\_0}) + p(U_{i\_0})$
$1 - p(S)$	⇔	$p(M_{c\_1}) + p(M_{p\_1}) + p(M_{i\_1})$

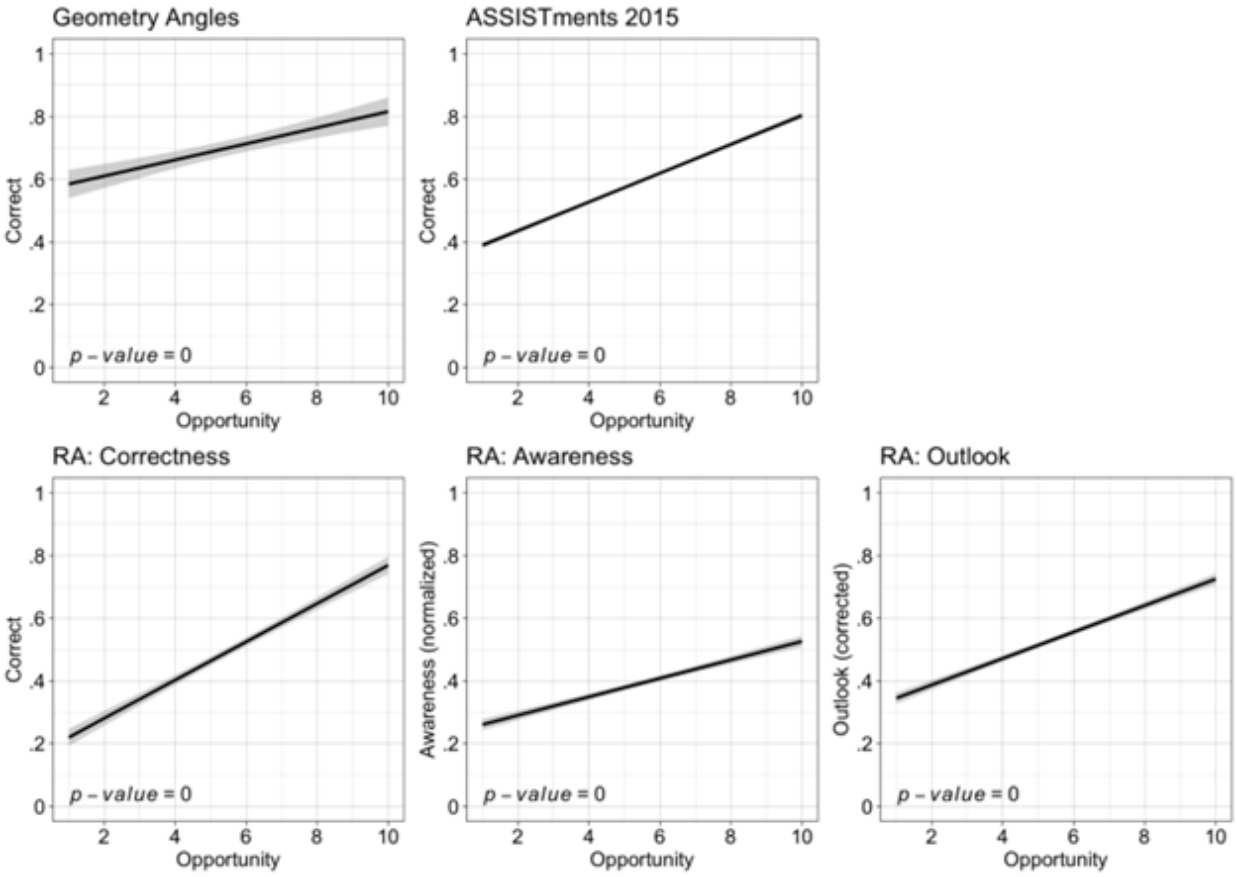

Figure 5: Trend comparison between ASSISTments and Geometry Angles and RA's correctness, awareness, and outlook.

A ten-fold cross-validation repeated five times was used for model training using the training dataset previously defined. Each iteration involved randomly distributing the dataset into ten bins, followed by ten training and testing rounds. A different bin is used for testing each round, while the remaining bins are used for training. Using cross-validation allows for more reliable results than having a single train and test set since it is less likely for the results to be just due to a convenient test/train data split. Additionally, at the beginning of each repeat, the parameters are once again re-initiated randomly. As an EM algorithm, the Baum-Welch algorithm can maximize a function with multiple peaks or optima, such as the one shown in Figure 6. The global optimum may not be reachable depending on the initial parameters. It is necessary to have different initial parameters to ensure that the global optimum is found and not just a local optimum.

For each repeat, the model is taken to be the average of the parameters in each iteration. The model corresponding to the repeat with the highest average cross-validation accuracy is selected. Arguably, accuracy is less informative than other measures such as F1 score, which accounts for precision and recall, for datasets with class imbalance (i.e., more data for a particular label than other labels). Nevertheless, accuracy has the advantage of being more intuitive than other performance measures, and since class imbalance is not prevalent in the dataset (see Table 7), measuring accuracy is sufficient for this task. The following formula gives the accuracy based on the contingency mapping in Table 9, where the first model is the actual observation, and the second model is the trained model:

$$Accuracy = \frac{A + D}{A + B + C + D} \tag{7}$$

Table 9: Contingency table mapping.

		First Model's Prediction	
		Correct	**Wrong**
Second Model's Prediction	Correct	A	B
	Wrong	C	D

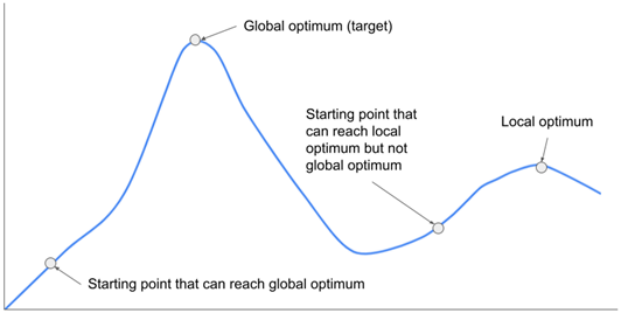

Figure 6: A function with multiple optima.

When validating, typically, the resulting model is used to predict the associated observation. However, the observation vocabulary for the RA-BKT is different from the other models ({c_0, p_0, i_0, c_1, p_1, i_1} for RA-BKT and {0,1} for the rest). To be able to make a parallel comparison between RA-BKT and the other models, the probabilities for {c_0, p_0, i_0}, and {c_1, p_1, i_1} as analogs to {0,1} are summed up when validating RA-BKT. For all other models, whether the observation corresponds to the answer is correct is predicted.

The R packages **deepnet** (Rong, 2014) and **caret** (Kuhn, 2008) were used for training the DL models. The **caret** package does the repeated cross-validation described earlier automatically. Additionally, **caret** searches for the best combination of the number of nodes and layers using the same accuracy formula mentioned earlier. The search space included up to three layers (thus, deep learning or DL) to tap the potential of higher accuracy rates. Each layer is set to possibly have 0, 3, 5, 7, or 10 nodes (0 not included for the first layer to ensure there is at least one layer). Candidate hidden dropouts (0 and 0.1) were also set. Dropout involves randomly ignoring neurons during training to prevent overfitting or the resulting model being too specific to the training data (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014). This makes the entire search space have a size of 200 (4 options for first layer × 5 options for second layer × 5 options for third layer × 2 options for hidden dropouts).

The validation dataset predictions will be used to compute the testing accuracy and the statistical difference between models using McNemar's test (McNemar, 1947) with the same contingency table as in Table 9. The resulting statistic from McNemar's test is said to follow the χ^2 distribution. For sample sizes more than 25, the statistical degree of freedom (different from the modeling degree of freedom) is typically assumed to be 1. The corresponding p-value is obtained from the readily available distribution table for χ^2 using the computed statistic from these assumptions.

$$\chi^2 = \frac{(B-C)^2}{B+C} \tag{8}$$

The null hypothesis is that none of the models predict better than the other. The p-value should be less than α, which was chosen to be 0.05 to follow conventions, to reject the null hypothesis (i.e., that the model with better accuracy scores is better).

For knowledge tracing, mastery prediction is more important than correctness prediction. The statistical difference of mastery predictions between models is also calculated, this time with the Mann-Whitney U test. This test was selected because it does not require the items to be normally distributed and allows for comparing continuous variables, unlike McNemar's test (Mann and Whitney, 1947). The same assumptions for the null hypothesis as in McNemar's test are held.

6 Results and Discussion

The training task was not resource-intensive, so a simple machine (1.4 GHz Quad-Core Intel Core i5 processor, 8 GB 2133 MHz LPDDR3 memory, Intel Iris Plus Graphics 645 1536 MB graphics) was sufficient. The R programming language (version 4.0.2) was used. Table 10 shows the resulting training metrics. A naïve model that always predicts that the learner will always answer correctly was used for baseline comparison. Based on Table 7, there are marginally more observations with correct answers for both training and validation sets. Thus, the baseline predicting answering correctly has better odds than random chance.

Table 10: Training data comparison.

Description	Base-line	BKT	RA-BKT	ANN	RA-ANN
Training time (minutes)	-	16.441	19.511	12.429	13.929
Prediction time (seconds)	-	0.163	0.051	0.023	0.007
Training Accuracy	0.502	0.786	0.845	0.648	0.864

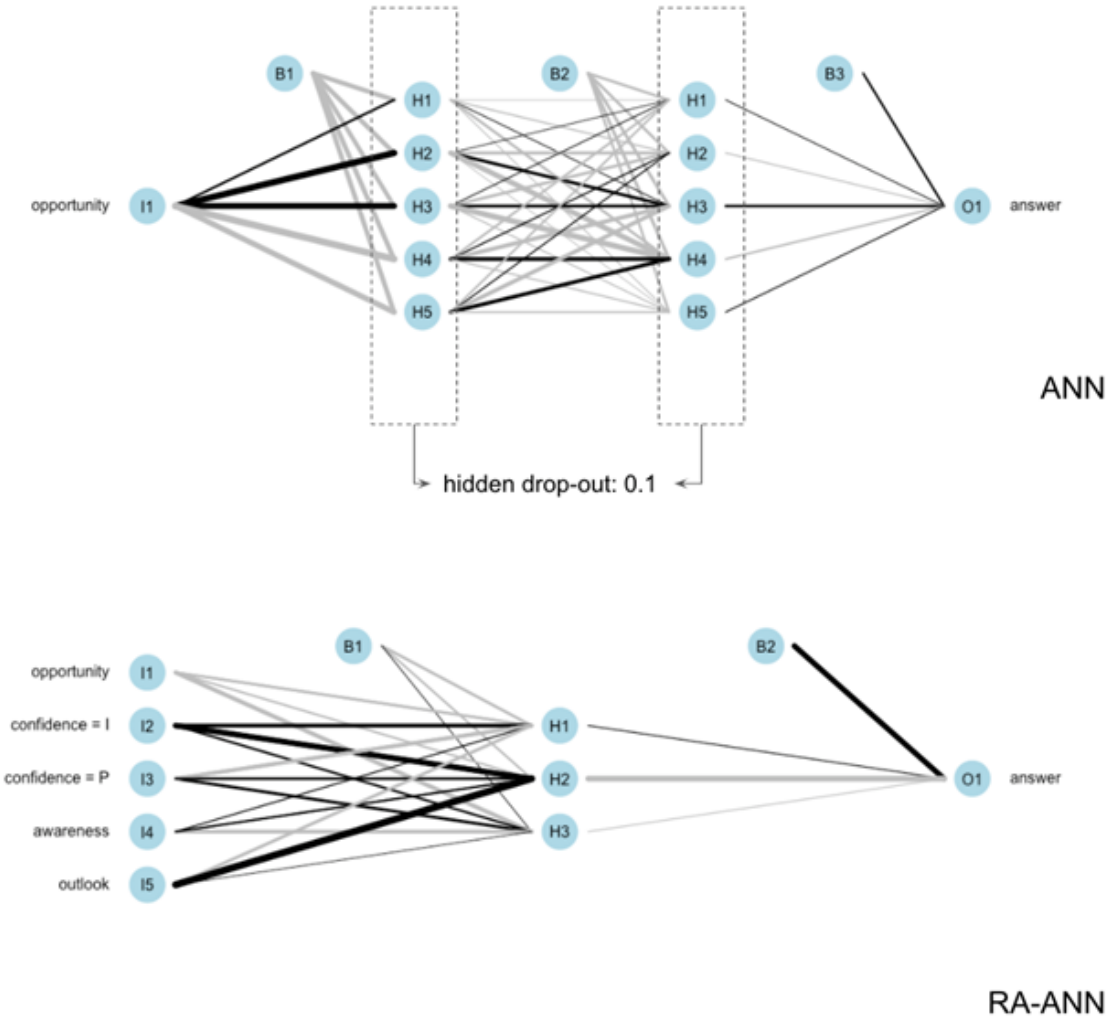

Figure 7: Resulting neural networks.

Training for RA-BKT took longer than BKT; the same is also observed for the DL models. While this result is not desirable, it is still reasonable considering the hardware specifications used for training and the increase in parameters estimated. The RA models are expected to take longer since they have more inputs than the standard models. The DL models are also expected to take more time than the HMM models since the DL's search space size is 200. In practice, the training will be done before use in tutoring software, so long training times can be acceptable if it will not demand expensive hardware. However, it is crucial to keep the prediction time low to avoid lag when the mastery prediction is being used for deciding learning paths while the learners are using software using knowledge tracing models. The difference in prediction time being negligible is thus a positive result.

Figure 7 shows the resulting neural networks for ANN and RA-ANN. The resulting network for ANN is relatively deep with two layers, while the RA-ANN consisted of only a single hidden layer. However, the ANN has lower accuracy; with insufficient data, producing good deep neural networks may be difficult. It can be seen from the RA-ANN that the neural network does not have to be deep to model the data decently for some problems. Most other deep learning applications in knowledge tracing had more complicated problem construction than presented in this paper. For instance, the learner's skill is also added as an input to potentially use the deep neural network to inform course developers which skills are related to each other (Piech et al., 2015).

Table 11 clarifies that RA-ANN is the best performer based on accuracy, followed by RA-BKT. Note that, as previously pointed, DL models will require more data for

training. Hence, if data is not enough, using RA-BKT is worth considering. Each of the models is also significantly different from the other in terms of correctness prediction except when comparing BKT and RA-BKT and ANN with RA-ANN. This means that the difference between the accuracy across models is not merely due to random chance except for BKT against RA-BKT and ANN against RA-ANN. Hence, choosing between HMM-based algorithms and DL-based algorithms matters, but whether the standard or the RA models should be used is a toss-up.

The original motive for creating RA-BKT and investigating the DL models is to reduce the learners' cognitive load when using a tutoring system that has both cognitive and metacognitive tutoring elements. With knowledge tracing, cognitive load is managed by giving the learners just enough opportunities until they master the knowledge component (e.g., provide opportunities until predicted mastery has reached a predefined value, say, 0.90). Once a predefined mastery level is reached, the learner can move on to the next knowledge component, thus spending less time and cognitive resources on the current knowledge component. For the BKT and RA-BKT, the mastery corresponds to the probability that the learner has mastered the knowledge component, which can be seen from the HMM's transition probability matrix. For the ANN and RA-ANN, this was taken to be the prediction that the answer will be correct.

When looking at mastery predictions, which matters during knowledge tracing, all models differ significantly from each other. More information is then needed if the choice is between BKT and RA-BKT. While their correctness predictions may differ only due to random chance, choosing one over the other will considerably affect adaptive learning.

Table 11: Testing accuracies (diagonal) and significant differences of correctness prediction using McNemar's test (upper triangle) and mastery prediction using Mann-Whitney U test (lower triangle) between models.

	Baseline	BKT	RA-BKT	ANN	RA-ANN
Baseline	0.463	< 0.001	< 0.001	< 0.001	< 0.001
BKT	< 0.001	0.768	0.744*	< 0.001	< 0.001
RA-BKT	< 0.001	< 0.001	0.865	< 0.001	< 0.001
ANN	< 0.001	< 0.001	< 0.001	0.682	< 0.842*
RA-ANN	< 0.001	< 0.001	< 0.001	< 0.001	0.899

* Not statistically significant

Table 12 shows the resulting knowledge parameters from the trained HMM models. Despite the efforts to create a data set intended to prevent degeneracy, BKT still resulted in a potentially degenerate model where the **Guess** and **Slip** are more than 0.5. The resulting **Initial Learning** is also high, which could be contrary to typical assumptions on the use of tutoring software (learners might use tutoring software to learn concepts not familiar to them, to begin with). Note that no assumptions were made during dataset creation related to prior knowledge, guess, and slip. Fortunately, these problems are not evident in the resulting RA-BKT model. This can be evidence of RA-BKT's better compliance with the learning intuitions set out to be investigated than the BKT.

Table 12: Resulting knowledge parameter values for HMM-based models.

Description	BKT	RA-BKT
Initial Learning	0.451	0.044
Acquisition	0.217	0.118
Guess	0.728	0.255
Slip	0.699	0.046

For this analysis, the focus is on the RA-BKT parameters which directly mapped to the BKT parameters. Those that map to the derivable parameters **Not Guess** (1 - p(G)) and **Not Slip** (1 - p(S)) were not accounted for. Nevertheless, the RA-BKT parameters that comprise these derivable parameters ($p(U_{c\_0}) + p(U_{p\_0}) + p(U_{i\_0})$ and $p(M_{c\_1}) + p(M_{p\_1}) + p(M_{i\_1})$ respectively) can provide implicit feedback about the way the knowledge component is taught when calculated using actual learner data. For example, a high value for the probability of a learner saying they can answer a question correctly despite not having mastery ($p(U_{c\_0})$) may indicate misconception. On the other hand, A high probability of a learner saying they cannot answer the question correctly despite having mastery ($p(M_{i\_1})$) could indicate that the question may have been confusing.

Figure 8 compares the resulting locally estimated scatterplot smoothing (LOESS) curves and linear regression lines (shown as dashed lines) using generalized linear modeling (GLM) for the models' predicted mastery and opportunity. The regression lines were added to visualize the directionality of the trends better. Ideally, the RA-BKT and the RA-ANN curves should sit higher than the BKT and ANN curves to reduce the opportunities (i.e., higher predicted mastery, thus less work) required. However, this is not the case for the HMM models due to BKT's high

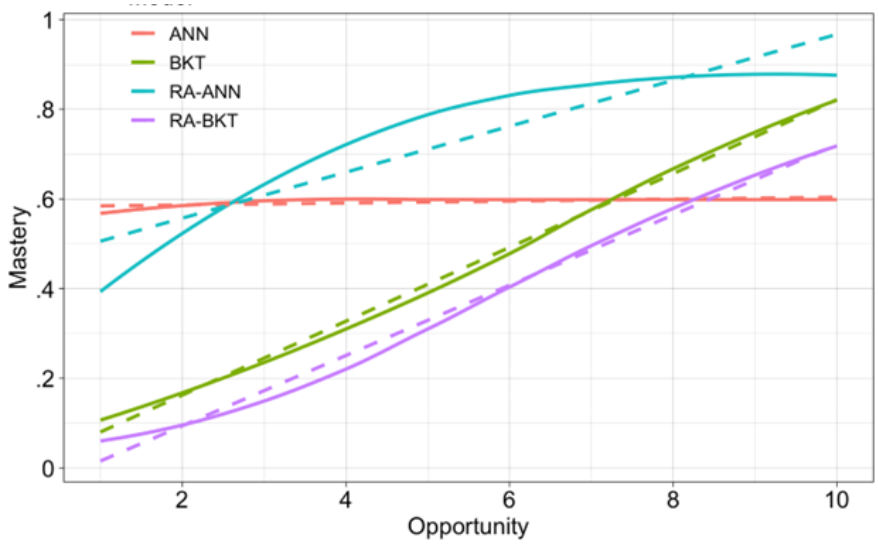

Figure 8: Predicted mastery against learning opportunity LOESS trend curves with 0.75 span (solid line) and GLM regression lines (dashed line).

Initial Learning. With higher **Initial Learning**, the BKT assumes that the learner starts with more prior knowledge than what the RA-BKT predicts.

Nevertheless, looking at the regression lines, the RA-BKT has a steeper slope. It can be deduced that if the starting point had been the same, the RA-BKT would reach the desired predicted mastery before BKT does. The BKT line slope is lower, which can be explained by the potential degeneracy discussed earlier based on the low **Acquisition** and high **Slip** and **Guess**.

A similar observation on slopes can also be seen between ANN and RA-ANN plots. Unlike in BKT and RA-BKT, where the models can be inspected for possible degeneracy, ANN and RA-ANN would require post-hoc modeling to explain the original models. This is a disadvantage of the models based on neural networks. What is known is the ANN model had lower accuracy, to begin with, making the RA-ANN model the better choice between ANN and RA-ANN. Additionally, Figure 8 shows that the RA-ANN model better reflects the learning intuition the dataset was built on than the ANN case, like BKT and RA-BKT models. Thus, the RA models better follow the intuition that the more chances given to the learners, the more they gain mastery.

To illustrate how adaptive learning will work with the created models more definitely, suppose the instructor decides that 50% mastery for a given module is sufficient for the learners to proceed to the next module. Without adaptive learning, all learners would have to attempt all ten opportunities before moving on to the next knowledge component. If the LOESS curves in Figure 8 are crudely followed for adaptive learning, a learner who answers the first opportunity correctly can already proceed based on the standard BKT and ANN. This makes the standard models suspiciously lenient. The learner would need to answer the first six opportunities to proceed to the next knowledge component if the RA-BKT followed, which is reasonably more than the opportunities that the standard models will require but still less than the complete set.

One of the motivations for developing the RA-BKT model is that the awareness and outlook scores could indicate learners' mastery (i.e., guesses and slips could be less frequent if they have desirable awareness and outlook scores). Intuitively, one might say that higher mastery could mean better awareness since the learners have the better domain knowledge to understand their knowledge levels. Figure 9 shows the LOESS curves of the predicted mastery against awareness scores. While all models exhibit an upward trend as expected, the ANN's ascend is abrupt, while the BKT's took a relatively long downward turn towards the end. Once again, the RA models outperform the standard models from this perspective.

A similar argument can be applied for the outlook measure. While being optimistic is generally seen as a positive attitude, optimism can have its costs, and pessimism has its benefits in learning (Sweeny & Shepperd, 2010). Hence, a neutral outlook is desirable. Figure 10 shows the LOESS curves of the predicted mastery of the models against the corrected outlook scores, like what was used for analyzing dataset trends. It can be seen from the figure that BKT barely follows the intuition of having an upward trend. ANN's is abrupt, just like the awareness case. The RA models continue to follow intuition.

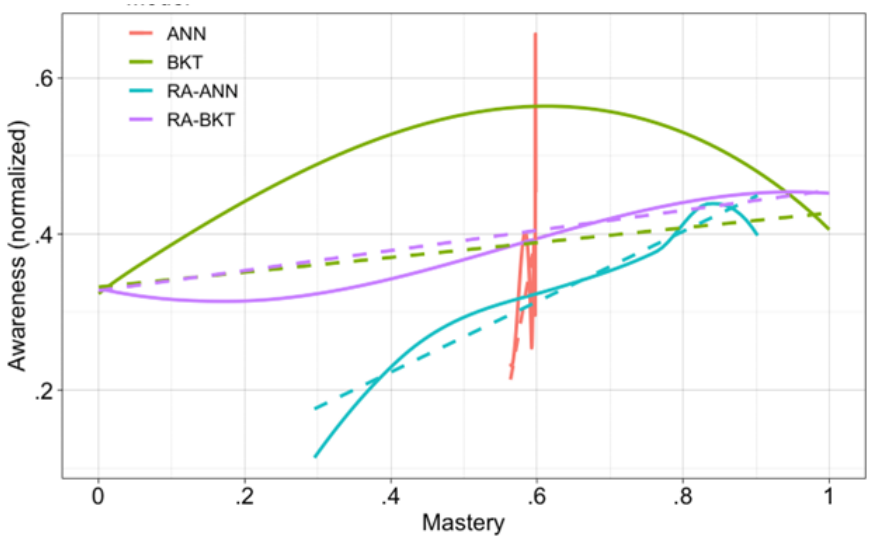

Figure 9: Awareness scores against predicted mastery LOESS trend curves with 0.75 span (solid line) and GLM regression lines (dashed line).

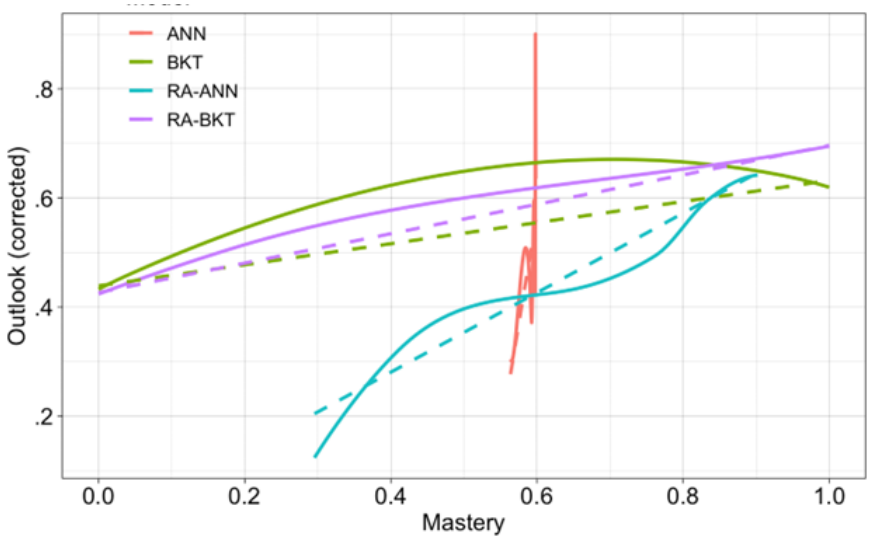

Figure 10: Corrected outlook scores against predicted mastery LOESS trend curves with 0.75 span (solid line) and GLM regression lines (dashed line).

7 Conclusion

The feasibility of using knowledge tracing to manage cognitive resources on a metacognitive tutor using the RA model was explored. The RA-BKT was constructed by expanding the observation vocabulary to include the correctness of the learners' answers at each opportunity and their confidence in self-reports. Neural networks using metacognitive inputs and not using metacognitive inputs were also constructed. All resulting models are then compared with each other. A dataset based on assumed learner behaviors was created for this purpose.

Creating a synthetic learner dataset is an original approach to conduct modeling based on a learning theory yet to be tested at scale. The approach was validated by comparing the trends from the created dataset with existing large-scale datasets. Showing similarities can be valuable when testing educational theories that have no precedent before subjecting learners to experimentation.

Even though training times for the models varied significantly, all of them had decent prediction times. RA-ANN had the best test accuracy, followed by RA-BKT. A possible reason for ANN's lack-luster accuracy is insufficient data to produce the deep neural network that it has resulted. While the RA-ANN had the best

performance, seeing its resulting neural network only had a single hidden layer can leave doubt. It could have performed better if there were sufficient data to construct a deeper neural network. All models are statistically different from each other, except when comparing correctness predictions between BKT and RA-BKT.

The resulting parameters for RA-BKT are non-degenerate. The observation follows from the predicted hidden state (i.e., whether the learner answers correctly or not follow from the prediction whether the learner has learned the knowledge component or not). The same was not observed for the BKT: the estimated guess and slip probabilities, as well as the estimated prior knowledge, are all too high (greater than 0.5). For the RA-BKT to be a better alternative to BKT in cognitive load management, it should return higher mastery predictions. This was not the case since the BKT's prior knowledge prediction is too high. However, when regression lines of opportunity against mastery are checked, RA-BKT had a steeper slope. If the BKT and RA-BKT ended up with similar prior knowledge predictions, the desired mastery level would be more quickly achieved with RA-BKT. Similar observations were also seen when comparing ANN and RA-ANN.

Another point of interest is the relationship between predicted mastery and RA measurements. The RA-BKT and RA-ANN show upward trends when comparing mastery with awareness and outlook. BKT does not consistently show an upward trend, and ANN has too abrupt slopes. These observations show that RA-BKT and RA-ANN follow cognitive and metacognitive learning intuitions better than the standard models.

In summary, RA-BKT and RA-ANN could be viable options for managing cognitive load while metacognitively tutoring given their high accuracy, efficient prediction times, and more intuitive predictions. RA-BKT can be looked at more favorably when constraints such as training time, dataset size, or hardware are present. This is critical as metacognition is crucial in succeeding in learning environments that require significant autonomy. With the emergence of online learning, concepts that can be challenging even with teacher-based support are finding their way in the online medium. This exacerbates the need to teach metacognition while ensuring deep cognitive learning.

> Developing the learners' metacognitive abilities within the context of domain instruction can benefit them as lifelong learning is becoming more and more important in increasingly knowledge-dependent economies. While metacognitive tutoring can make domain instruction more challenging, solutions such as the various knowledge tracing techniques exist and thus should not deter learners, educators, and administrators from pursuing metacognitive tutoring. However, a cost-benefit analysis must be conducted when selecting which solutions to use. Modern solutions that may lead to more accurate support may be more expensive than traditional approaches, with the improvements being marginal in typical learning scenarios.

8 Future Work

A central weak point of this experiment is the dataset used. No actual data that includes both learner performance and metacognitive measures exist as far as the researchers know. While cognitive performance might be estimated using other learning theories such as the IRT (Reise, Ainsworth, & Haviland, 2005), a similar theory is yet to be formulated for confidence ratings.

The availability of data from actual usage would raise more interesting investigation points. For example, while metacognition is generally seen as not domain-specific, there is still a belief that the domain where the metacognitive opportunity is presented does matter (Roll, Aleven, McLaren, & Koedinger, 2007). The RA models may link metacognitive and cognitive knowledge as was attempted by plotting mastery predictions against awareness and outlook scores. Having data from the metacognitive tutoring tool usage on different cognitive domains will be beneficial for this investigation.

A possible drawback of the RA models is modeling fairness. The RA models use confidence ratings that can be influenced by personal characteristics such as gender (Colbeck, Cabrera, & Terenzini, 2001) and culture (Ho, 2009). As such, the resulting mastery predictions may unjustly penalize some learners, not because of their lack of cognitive or metacognitive mastery but because of their innate attitudes. Therefore, fairness-enhancing interventions might need to be considered (Friedler et al., 2019).

Acknowledgments: The 'Geometry Angles - Fox Chapel 1998' dataset accessed via DataShop (pslcdatashop.web.cmu.edu) and the '2015 ASSISTments Skill Builder Data' dataset (https://sites.google.com/site/assistmentsdata/home/2015-assistments-skill-builder-data) were used in this work.

Conflict of Interest: Authors state no conflict of interest.

Data Availability Statement: The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Funding Information: The Japan Society for the Promotion of Science (JSPS) supported this work via the Grants-in-Aid for Scientific Research (Kakenhi) Grant Number JP20H01719.

Author Contribution: MKJC conceptualized and designed the research, developed and executed the model code, interpreted the result, and prepared the manuscript. JSC read/edited the manuscript, provided mentorship, secured funding, and reviewed all aspects of this research. The authors applied the FLAE and SDC approach for the sequence of authors.

References

Agustianto, K., Permanasari, A. E., Kusumawardani, S. S., & Hidayah, I. (2016). Design adaptive learning system using metacognitive strategy path for learning in classroom and intelligent tutoring systems. *AIP Conference Proceedings 1755* (pp. 070012-1–070012-6. doi: 10.1063/1.4958507). AIP Publishing.

Aleven, V. A., & Koedinger, K. R. (2000). Limitations of student control: Do students know when they need help? *International Conference on Intelligent Tutoring Systems* (pp. 292-303). Springer, Berlin, Heidelberg.

Aleven, V. A., & Koedinger, K. R. (2002). An effective metacognitive strategy: Learning by doing and explaining with a computer-based cognitive tutor. *Cognitive Science, 26*(2), 147-179.

Azevedo, R. (2005, December). Computer environments as metacognitive tools for enhancing learning. *Educational Psychologist, 40*(4), 193-197. doi: 10.1207/s15326985ep4004_1.

Baker, R. S., Corbett, A. T., & Aleven, V. (2008). More accurate student modeling through contextual estimation of slip and guess probabilities in Bayesian knowledge tracing. *International Conference on Intelligent Tutoring Systems* (pp. 406-415). Springer Berlin Heidelberg.

Baker, R. S., Gowda, S. M., Corbett, A. T., & Ocumpaugh, J. (2012). Towards Automatically Detecting Whether Student Learning Is Shallow. *International Conference on Intelligent Tutoring Systems* (pp. 444-453). Springer.

Baum, L. E. (1972). An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequalities, 3*(1), 1-8.

Choudhury, S., & Pattnaik, S. (2020, January). Emerging themes in e-learning: A review from the stakeholders' perspective. *Computers & Education, 144*, 103657. doi: 10.1016/j.compedu.2019.103657.

Colbeck, C. L., Cabrera, A. F., & Terenzini, P. T. (2001). Learning professional confidence: Linking teaching practices, students' self-perceptions, and gender. *The Review of Higher Education, 24*(2), 173-191. doi: 10.1353/rhe.2000.0028.

Corbett, A. T., & Anderson, J. R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction, 4*(4), 253-278.

Dempster, A. P., Laird, N. N., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological), 39*(1), 1-22. doi: 10.1111/j.2517-6161.1977.tb01600.x.

Deonovic, B., Yudelson, M., Bolsinova, M., Attali, M., & Maris, G. (2018). Learning meets assessment. *Behaviormetrika, 45*(2), 457-474.

Doroudi, S., & Brunskill, E. (2017). The misidentified identifiability problem of Bayesian knowledge tracing. *10th International Conference on Educational Data Mining* (pp. 143-149). Wuhan, China: International Educational Data Mining Society.

Flavell, J. H. (1979). Metacognition and cognitive monitoring: A new area of cognitive-developmental inquiry. *American Psychologist, 34*(10), 906-911. doi: 10.1037/0003-066X.34.10.906.

Friedler, S. A., Scheidegger, C., Venkatasubramanian, S., Choudhary, S., Hamilton, E. P., & Roth, D. (2019). A comparative study of fairness-enhancing interventions in machine learning. *Conference on Fairness, Accountability, and Transparency* (pp. 329-338. doi: 10.1145/3287560.3287589). Association for Computing Machinery.

Gallagher, S., & Palmer, J. (2020, September 29). *The Pandemic Pushed Universities Online. The Change Was Long Overdue*. Retrieved October 2020, from Harvard Business Review: https://hbr.org/2020/09/the-pandemic-pushed-universities-online-the-change-was-long-overdue

Gama, C. (2004). Metacognition in interactive learning environments: The Reflection Assistant model. *International Conference on Intelligent Tutoring Systems* (pp. 668-677). Springer.

Halpern, D. F. (2013). *Thought and knowledge: An introduction to critical thinking*. Psychology Press.

Halpern, D., Tubridy, S., Wang, H. Y., Gasser, C., Popp, P. O., Davachi, L., & Gureckis, T. M. (2018). Knowledge tracing using the brain. *International Conference on Educational Data Mining* (pp. 219-228). International Educational Data Mining Society.

Heffernan, N. T., & Heffernan, C. L. (2014). The ASSISTments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education, 24*(4), 470-497.

Himmelmann, L. (2010). *HMM: HMM - hidden Markov models*. Retrieved May 2020, from The Comprehensive R Archive Network: https://CRAN.R-project.org/package=HMM

Ho, E. S.-C. (2009). Characteristics of East Asian learners: What we learned from PISA. *Educational Research Journal, 24*(2), 327-348.

Kuhn, M. (2008). Building Predictive Models in R Using the caret Package. *Journal of Statistical Software, 25*(5), 1-26, doi:10.18637/jss.v028.i05.

Mann, H. B., & Whitney, D. R. (1947). On a test of whether two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, *18(1)*, 50-60.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 114-133. doi:10.1007/BF02478259.

McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika, 12*(2), 153-157.

Nixton, T., & Yudelson, M. (2013). *Functions for fitting Bayesian knowledge tracing (BKT) models from data*. Retrieved May 2020, from Github: https://github.com/IEDMS/REDM/tree/master/bkt

Ohtani, K., & Hisasaka, T. (2018). Beyond intelligence: a meta-analytic review of the relationship among metacognition, intelligence, and academic performance. *Metacognition and Learning, 13*(2), 179-212.

Pardos, Z. A., & Heffernan, N. T. (2010). Modeling individualization in a Bayesian networks implementation of knowledge tracing. *International Conference on User Modeling, Adaptation, and Personalization* (pp. 255-266). Springer.

Pardos, Z. A., & Heffernan, N. T. (2011). KT-IDEM: introducing item difficulty to the knowledge tracing model. *International Conference on User Modeling, Adaptation, and Personalization* (pp. 243-254). Springer.

Pardos, Z. A., Bergner, Y., Seaton, D. T., & Pritchard, D. E. (2013). Adapting Bayesian knowledge tracing to a massive open online course in edX. *Sixth International Conference on Educational Data Mining* (pp. 137-144). International Educational Data Mining Society.

Piech, C., Bassen, J., Ganguli, S., Sahami, M., Guibas, L. J., & Sohl-Dickstein, J. (2015). Deep knowledge tracing. *Advances in Neural Information Processing Systems*, (pp. 505-513).

Reise, S. P., Ainsworth, A. T., & Haviland, M. G. (2005). Item response theory: Fundamentals, applications, and promise in psychological research. *Current Directions in Psychological Science, 14*(2), 95-101. doi: 10.1111/j.0963-7214.2005.00342.x.

Roll, I., Aleven, V., McLaren, B. M., & Koedinger, K. R. (2007). Designing for metacognition—applying cognitive tutor principles to the tutoring of help seeking. *Metacognition and Learning, 2*(2-3), 125-140.

Rong, X. (2014). *deepnet: deep learning toolkit in R*. Retrieved from The Comprehensive R Archive Network: https://CRAN.R-project.org/package=deepnet

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research, 15*(1), 1929-1958.

Sweeny, K., & Shepperd, J. A. (2010). The costs of optimism and the benefits of pessimism. *Emotion, 10*(5), 750-753. doi: 10.1037/a0019016.

Tobias, S., & Everson, H. T. (2002). *Knowing what you know and what you don't: Further research on metacognitive knowledge monitoring. Research Report No. 2002-3.* College Entrance Examination Board. ERIC.

Yu, L., Schwier, J. M., Craven, R. M., Brooks, R. R., & Griffin, C. (2012). Inferring statistically significant hidden Markov models. *IEEE Transactions on Knowledge and Data Engineering, 25*(7), 1548-1558.

Yudelson, M. V., Koedinger, K. R., & Gordon, G. J. (2013). Individualized Bayesian knowledge tracing models. *International Conference on Artificial Intelligence in Education* (pp. 171-180). Springer.

Zhao, L., & Ye, C. (2020, July). Time and Performance in Online Learning: Applying the Theoretical Perspective of Metacognition. *Decision Sciences Journal of Innovative Education, 18*(3), 435-455. doi:10.1111/dsji.12216.