Dynamic Knowledge Tracing Models for Large-Scale Adaptive Learning Environments

Androniki Sapountzi[1] Sandjai Bhulai[2] Ilja Cornelisz[1] Chris van Klaveren[1]
[1]Vrije Universiteit Amsterdam, Faculty of Behavioral and Movement Sciences, Amsterdam Center for Learning Analytics
[2]Vrije Universiteit Amsterdam, Faculty of Science, Department of Mathematics
Email addresses: a.sapountzi@vu.nl, s.bhulai@vu.nl, i.cornelisz@vu.nl, c.p.b.j.van.klaveren@vu.nl

Abstract— **Large-scale data about learners' behavior are being generated at high speed on various online learning platforms. Knowledge Tracing (KT) is a family of machine learning sequence models that use these data to identify the likelihood of future learning performance. KT models hold great potential for the online education industry by enabling the development of personalized adaptive learning systems. This study provides an overview of five KT models from both a technical and an educational point of view. Each model is chosen based on the inclusion of at least one adaptive learning property. These are the recency effects of engagement with the learning resources, dynamic sequences of learning resources, inclusion of students' differences, and learning resources dependencies. Furthermore, the study outlines for each model, the data representation, evaluation, and optimization component, together with their advantages and potential pitfalls. The aforementioned dimensions and the underlying model assumptions reveal potential strengths and weaknesses of each model with regard to a specific application. Based on the need for advanced analytical methods suited for large-scale data, we briefly review big data analytics along with KT learning algorithms' scalability. Challenges and future research directions regarding learners' performance prediction are outlined. The provided overview is intended to serve as a guide for researchers and system developers, linking the models to the learner's knowledge acquisition process modeled over time.**

Keywords- adaptive learning; big data applications; deep learning models; knowledge tracing; predictive analytics; sequential machine learning.

I. INTRODUCTION

Big Data Analytics (BDA) is becoming increasingly important in the field of online education. Massive Open Online Courses (e.g., Coursera), Learning Management Systems (e.g., Moodle), social networks (e.g., LinkedIn Learning), online personalized learning platforms (e.g., Knewton), skill-based training platforms (e.g., Pluralsight), educational games (e.g., Quizlet), and mobile apps (e.g., Duolingo) are generating various types of temporal, dynamic and large-scale data about learner's behaviors during their knowledge acquisition process of a skill over time [1]–[3]. To illustrate this with an example, the 290 courses offered by MIT and Harvard in the first four years of edX produced 2.3 billion logged events from 4.5 million learners. The emerging scientific fields of educational neuroscience [4] and smart-Education [5][6] can provide new insights about how people acquire skills using these new big data sources in education.

Artificial Intelligence (AI), Learning Analytics (LA), and Educational Data Mining (EDM) are three areas under development oriented towards the inclusion and exploration of big data analytics in education [2][7]–[9]. AI, LA, EDM, and big data technologies have been progressing rapidly, including developments towards the inclusion and exploration of BDA in education. Yet, specific advanced analytic methods suited for large, diverse, streaming, dynamic or temporal data are still being under development. EDM considers a wide variety of types of data, algorithms, and methods for modeling and analysis of student data, as categorized by [2][10][11]. A critical question in this area is whether complex learning algorithms or better data in terms of higher quality [12], well pre-processed, or bigger in size [8][13]–[15] is more important for achieving improved analysis results concerning either their predictive power or explainability.

For the aforementioned reasons, the implementation of BDA in education is considered to be both a major challenge and an opportunity in education [2][3][7]–[11][13][16][17]. Table I illustrates the models used in EDM. task of Knowledge Tracing has been modeled via Neural Networks and Probabilistic Graphical supervised learning models.

Knowledge Tracing (KT) is widely applied in adaptive learning systems, and to other modal sources of big data [11] such as online standardized tests, Massive Open Online Courses (MOOCs) data, and educational apps. KT is an EDM framework for modeling the acquisition of student knowledge over time, as the student is observed to interact with a series of learning activities. The objective of the model is to either infer the knowledge state, -which stands for the depth and robustness of the specific skill- or to predict the performance on all learning resources in the sequence that assess the skill acquisition process.

KT can thus be considered as a sequence machine learning model that estimates a hidden state (i.e., the probability that a certain concept of knowledge is acquired) based on a sequence of noisy observations (i.e., the interaction-performance pairs on different learning resources on consecutive trials). The estimated probability is then considered a proxy for knowledge mastery that is leveraged in recommendation engines to dynamically adapt the learning resources or feedback returned to the learner.

There are plenty of past similar review attempts regarding the modeling of knowledge acquisition:

i. algorithms [2][10][17] and empirical evidence-based [7] EDM,
ii. issues [8], applications [9], research methods [14], and learner models [11] concerning big data in knowledge acquisition process,
iii. design principles [3], AI algorithms [16], learner models [50][51][34] for online, adaptive, intelligent learning systems,
iv. evaluation metrics for measuring learner modeling quality [26][27],

This study provides an overview of currently existing representations of KT models focused on prediction of learner performance. The educational and technical angles, inspired by the above list are then used as guides to select and analyze the modeling quality of the learner model.

The literature distinguishes between the probabilistic and deep learning representations and both are widely used to represent complex, real-world phenomena in other domains apart from learning. The former is comprised by Hidden Markov Models and Dynamic Bayesian Networks, which model the knowledge of a learner as a local, binary, stochastic hidden state. The latter is constituted by deep Recurrent Neural Networks (RNN) models with Long Short-Term Memory (LSTM) and Neural Networks augmented with an external memory (MANN). In this representation, the learner's understanding of a skillset is treated as a distributed, continuous hidden state in the LSTM and as an external memory (value) matrix in the MANN; both are updated in a non-linear, deterministic manner. This study is not focused on specific variants of one model, rather it considers all these different models for the representation of the evolution of learner knowledge. Furthermore, it is important to note that, although the challenges and advantages are usually hold for general applications in other domains, in this review we refer only to the task of knowledge acquisition.

TABLE I. MODELS FOR EDUCATIONAL DATA MINING

Predictive Analytics Methodology		
Statistical & Machine Learning		Computational Intelligence (CI)
Machine Learning Models		
	Supervised	*Unsupervised*
Continuous Output	- Decision Trees -Regression Analysis	- K-Means - PCA - SVD
Categorical Output	- Decision Trees - Logistic Regression - Naïve Bayes	- Association Rule Mining
Neural Networks, Nearest Neighbors, SVM, Probabilistic Graphical Models, Anomaly Detection and Random Forest can be applied to both outputs and learning types.		

From an educational point of view, learner models should satisfy a set of properties [3][5][36][44][45] in order to work in an adaptive, intelligent, online learning system to improve knowledge acquisition. This study focuses on the following four:

i. recent engagement of a learner with the learning activities,
ii. dynamic sequences of learning activities,
iii. inclusion of student's differences, and
iv. inclusion of the dependencies among skills that are instructed via activities.

The first feature is highly predictive for modelling knowledge acquisition over time; the second is important for the decision-making part of the model prediction; the third is concerned with the application of personalized learning; and the forth is the content-related requirement for many hierarchical knowledge domains. Embarking from the baseline Bayesian model, and based on the desired principles listed above, we outline four of the model's most recent extensions. These include the individualization of learning pace among students, the incorporation of the prerequisite relationships among multiple skills, and the continuous representation of the knowledge state. The latter enables all of the four aforementioned features to be partly estimated. There are other challenges [3][36] that are not discussed throughout the paper such as the management of optimal instructional types of learning resources, up-to-date predictions, and the lack of control over both the user experience and offline learning behavior.

From a machine learning point of view [15][34], the assumptions in the different representations [19] and the potential pitfalls and advantages of optimization [28][29][30][31] and evaluation [26][27] methods are investigated. Each model faces different challenges regarding the estimation of parameters, overfitting issues, data sample efficiency, intensity of computational operations, and overall complexity. The general idea is that, by investigating these aspects, one can gain understanding why the considered models work the way they do, under which conditions one should be preferred against another, and in which cases can a model fail to accurately model knowledge acquisition. Furthermore, based on the fact that online education systems produce large-scale data, we also consider the scalability and computational speed of the algorithms implementation phase [5][8]. Specifically, we constrain this dimension via outlying the algorithms' requirements on the following aspects [46]:

i. size of training data set,
ii. number of model parameters, and
iii. level of domain-knowledge dependence.

The review intends to serve as a guide of dynamic KT models for researchers and system developers; whose goal is to predict future learner's performance based on historical achievement trajectories and develop adaptive learning experiences. It provides a structured comparison of different models and outlines their strengths and similarities concerning the KT task. The resulting fundament may serve as inspiration for the development of more sophisticated algorithms or ways of richer data collection. The corresponding citations throughout the paper provide further guidance in implementing or extending a model for a specific data source or educational application. A shorter version of the review is available in [1].

This study proceeds as follows. Section II describes the representation component for the KT along with a brief introduction behind the probabilistic and deep learning sequential models. Section III introduces the baseline KT model and four extensions of it, after which the strengths, weaknesses, differences, and similarities are highlighted in Section IV. Section V discusses Item Response Theory (IRT), as it is the alternative family of models for predicting future performance. Section VI investigates the prospects and challenges for modeling knowledge acquisition over time, and Section VII provides with the conclusions.

II. Data Representation For knowledge tracing

Data representation refers to the choice of a mathematical structure that models the data, in this case, the hidden learner's knowledge state while interacting with learning resources. It embodies assumptions required for the generalization to new examples [19]. Identifying a sufficient dataset and representation for addressing the prediction problem is not a trivial task.

A good representation is one that can express the available kind of knowledge [15], meaning that a reasonably-sized learned representation can capture the structure of a huge number of possible input configurations. Other elements contributing to a good representation are outlined in [19]. A relevant point to note is that the choice of representation affects analytics that lie in distributed systems -a common case in BDA- in the sense of the data set decomposition into smaller components; so that analysis can be performed independently on each component.

A. Learning Task

Consider a learner interacting with an intelligent, adaptive learning platform with the purpose of acquiring a skill or a set of skills. In KT, two AI frameworks have been utilized to represent the available knowledge and disentangle the underlying explanatory factors of this process: the Bayesian (inspired by Bayesian probability theory and statistical inference) and the connectionist deep learning framework (inspired by neuroscience). Bayesian Knowledge Tracing (BKT) is the oldest -and still dominant- approach for modeling cognitive knowledge over time, while the deep learning approach to knowledge tracing, known as Deep Knowledge Tracing (DKT), is a more recently developed state-of-the-art model. Both AI approaches are used for modeling sequential data which implies that the data instances are not any more independent and hold a temporal pattern. The temporal pattern in KT is the dependency between learner's time engagement within and between learning resources.

Modeling knowledge acquisition with the objective to predict the next learner's performance, in its general form can be seen as a supervised time-series learning problem. Suppose a data set D consisting of ordered sequences of length *T,* to be trajectories of exercise-performance observation pairs X = $\{(x_{m,1}, y_{m,1}) \dots (x_{m,T}, y_{m,T})\}$ with $y_{m,t} \in \{0,1\}$ from the m^{th} student on trial $t \in \{1,..,T\}$, and with $x_{m,t}$ to be a label of a subskill that instruct one or more of N skills $\mathrm{S} = \{S^1, S^2, \dots, S^N\}$. The objective in the Bayesian approach is to estimate the probability applying a skill S^1 correctly on an upcoming exercise. This is estimated based on the sequence of observed answers that tap S^1, as determined by the concept map. Similarly, the objective of the deep learning approach -and of the logistic models described in Section V- is to predict the probability that the student will give a correct response $y_{m,t+1} = 1$ on an upcoming exercise, which could belong in any subskill. Different from Bayesian methods, the exercises in deep learning models do not have the skill notation of each exercise, thereby S is a latent structure.

Such a difference in computation is located in the logic behind generative and discriminative approaches of algorithms [20]. The former models the joint probability of both unobserved (target) y and observed (input) x random variables: $P(x, y)$ developing thus a model of each y; while the latter estimates the probability distribution of unobserved variables y conditioned on observed variables: $P(y|x)$ developing thus a model of boundary between y; in case of deep learning models, this boundary is non-linear. Neural networks are discriminative models that map out a deterministic relation of y as a function of x. Fig. 3 depicts this mapping of x sequence vectors to y sequence vectors.

B. Domain-Knowledge Dependence

Domain knowledge dependence refers to the amount of human involvement necessary to tailor the algorithm to the learning task, i.e., specify the prior knowledge built into the model before training [46]. An important distinction between the probabilistic and deep learning KT approaches is located in the existence of the concept map and the notion of a skill.

The concept map or otherwise called expert model breaks down the subject matter to chunks of knowledge. It maps an exercise and/or exercises' step to the related skill. Each skill is divided into a hierarchy of relatively fine-grained subskills, also known as Knowledge Components (KC), which need to be acquired by a learner. Skills, also referred as concepts or competencies, are abstract but intuitive notions of ideas that the exercise instructs and assesses.

Each exercise may require one or more KCs so as to be solved; the latter case is known as multi-skill learning.

To illustrate the granularity level of a KC with an example, 'the location of Kyoto' is a fine-grained KC while 'the names and locations of countries' is a coarse-grained KC [34]. The granularity of KCs is a subject of experimental research. An example of a KC could be 'declare a variable in a function definition' which should be split into 'single-variable' and 'multivariable' KCs if the data indicates such a split is warranted.

In the probabilistic KT, the sequences of X are passed through the concept map that is assumed to be accurately labelled by experts. This ensures that students master prerequisite skills, before tackling higher level skills in the hierarchy [18]. It assumes that all the learning activities are of the same difficulty level, which implies that the observation of a student struggling on some resources is occurring because there are some subskill(s) that the student has yet to acquire. In the probabilistic KT with Hidden Markov Models, a different model is initiated for each new skill.

Rather than constructing a separate model for each skill, the deep learning approach, as well as Dynamic Bayesian Networks (DBN), model all skills jointly. In contrast to deep learning models, a DBN demands a detailed concept map with the conditional relationships representing prerequisites among the KCs. The deep learning models just require exercise tags that denote the related KCs while the underlying skill and the related KCs belonging to the same skill are unknown. Examples of tags include the 'Pythagorean theorem I', 'mode', 'mean', and 'slope', which can be considered as coarse-grained KCs.

The network of DKT is presented with the whole trial sequence of exercises for all the skills practiced. The sequences are passed through featurization; that is the distributed hidden units in the hidden layers that relate the input to the output sequences. This distributed featurization is the core of the generalizing principle and is used to induce features and hence discover the similarity and prerequisite relationships among exercises.

The MANN model can also automatically discover the correlations among the KCs and cluster them based on the skill they instruct. It uses the inner product of the exercise tag with the embedding matrix that contains all KCs and passes it through the SoftMax activation function as described later in eq. (5α). Compared to DKT, which requires both a threshold to cluster the similar KC's and the network to be represented with the whole trial sequence at once, this model directly assigns exercises to concepts.

C. *Probabilistic Sequence Models: statistical learning machines*

The problem of KT was firstly posed as a 1st order, 2-state, 2-observation Hidden Markov Model (HMM) with a straightforward application of Bayesian inference. A DBN, also referred as Two-Timeslice Bayesian Network (BN), is employed to solve for a multi-skill learning task.

HMMs and DBNs are generative models called Probabilistic Graphical Models (PGM). These are statistical learning models that embody assumptions about the data generation process by modeling conditional dependencies (i.e., interactions) between random variables. Formally, a PGM is a graph $G = (X, E)$, where:

i. the random variables X are represented as nodes in a graph, and

ii. the conditional dependencies between X are described by the edges E (i.e., graph topology).

A graph is a powerful representation that can model a variety of data types by simply changing the definitions of nodes and edges. However, inference and learning over graphs is considered a difficult task. DBN is a Directed Acyclic Graph (DAG); directed graphs are useful for expressing causal relationships between random variables [20].

HMM is used to model sequences of possible latent random variables X that form a Markov process, in which the Markov property holds; that is, 'the past is independent of the future given the present' $X_{t+1} \perp (X_{t-1}, X_{t-2}, \dots, X_1)|X_t$. X have arrows pointing to the observed variables Y which are conditionally independent to each other, given the input variables X.

The interesting part of HMM is that X have unobserved states h, also referred to as *hidden* or *latent* states, which can store information for longer times in the sequence. The states h have their own internal dynamics, described by transitions between h, which are stochastic and controlled by a matrix A. At each timestep, these can take only one of N possible values. The outputs produced by an h are stochastic and hidden, in the sense that there is no direct observation about which state produced an output, much like a student's mental process. However, h produce as observables the emission probabilities Φ that govern the probability distribution of h.

The HMM model is characterized by its transition probability A, emission probability Φ and prior distribution Π. The parameters that need to be evaluated and learned are $\lambda = \{\Pi, A, \Phi\}$, where Π is the initial latent variable x_1, which is the only variable that does not depend on some other variable. Firstly, the evaluation problem is solved $P(Y|\lambda)$: the probability that the observations are generated by the parameters λ of the model, where Y is a sequence of learning activities attempts Y = $\{Y_t\}$, $t \in \{1, \dots, T\}$; and secondly the learning problem is being solved: how should λ be adjusted so as to maximize the $P(Y|\lambda)$. In the probabilistic setting of KT, X represents *the entire single skill,* h represent the knowledge states, which are 2 standing for a *mastered* and for a *not yet mastered*, as shown in Fig. 1. A indicates the learning or forgetting rate, i.e., the evolution of student's knowledge state and Φ includes the probability for a guessed or slipped answer. A detailed explanation of the HMM is provided by [23], [24].

DBNs generalize the HMM models by including a collection I of interacting input variables X linked by directional edges. The internal structure among X and E, called as graph topology, is repeated in the exact same way at each time step. The parent node set of X in G, denoted by $pa(X)$ is the set of all nodes from which an edge points to node X in G. These models hold the directed Markov property, which is $X_I \perp (non-descendants\,(X_I))|pa(X_I)$. In KT, $pa(X)$ carry the meaning of a prerequisite relationship, the nodes X indicate different KCs or skills and their realization x_i indicate the knowledge state of a student for skill i at a specific t. In KT, an observed variable is linked always to just one latent variable.

The question now becomes how the evolution of a knowledge state is modelled in a DBN. The value of a x_i at time slice t is calculated from not only the graph topology at t, described above, but also from the previous value of x_i at time $t-1$. Thereby, a DBN links knowledge state variables to each other over adjacent time steps to indicate learning or forgetting rate, as shown in Fig. 2.

In order to infer the probability distribution across the total number of hidden states h, there is a marginalization of the latent variables x over h. This is equivalent to $P(y, h| \theta) = \prod p(X|pa(X))$, denoting that the joint distribution of the observed Y and unobserved H variables is given by the product over all of the variables, i.e., nodes X of the graph conditioned on the $pa(X)$, where θ includes λ parameters together with the conditional edges of the graph topology. A detailed explanation of the computations in the DBN is provided by [20], [21].

D. *Neural Network Sequence Models: computational intelligence learning machines*

Deep RNN with LSTM internal memory units and Memory-Augmented Neural Networks (MANNs) have only recently been employed to the KT task to solve for the binary (i.e., h can take only one value), highly structured (assumptions about data generation) and memoryless (i.e., *Markov processes*) representation of the hidden knowledge state.

RNN and MANN are a family of Artificial Neural Networks (ANN) that can take variable length of inputs and the hidden state acts as a memory able to capture the temporal structure among sequences. MANN uses an external memory matrix to encode the temporal information, while LSTM uses an internal hidden state vector.

Deep learning, as it is primarily used, is a computational intelligence technique for classifying patterns (e.g. similarities found in data instances) to different targets y, based on large training data, using ANN with multiple layers [48].

ANN is a discriminative model that relates the input units x, which are amplified with weights w, to the output units y through a series of hidden layers: $y = f_1(\vec{w}_1, f_2(\vec{w}_2, \dots, f_n(\vec{w}_n, \vec{x})))$. Each hidden layer is comprised by hidden units, which are triggered to obtain a specific value by events found in x and -in case of RNN- also patterns that are found in previously hidden states. This process of triggering is implemented by a non-linear activation function f in the hidden layer.

RNNs are layered ANNs that share the same parameters w, through the activation function f. This property is illustrated in Fig. 3, with the formation of directed edges between hidden units. RNNs are powerful, as they combine the two following properties, not found in PGM's:

i. The distributed hidden state allows them to forget and store a lot of information about historical trajectories, such that they can predict efficiently.

ii. The non-linear activation functions allow them to update the hidden state in complicated ways, which can yield high-level structures found in the data (if available).

Instead of having a single hidden neural network layer (e.g., hyperbolic tangent) repeating at each step, LSTM is a type of hidden units that additionally includes *Forget, Input, and Output gates* repeating at each step. The interaction of the gates with each other is used to adjust the flow of information over time. The hidden state acts as a memory able to hold bits of information for longer periods of time and hence capable of learning complex functions from '*remembering*' even longer sequences of data (i.e., long-term dependencies).

MANN refers to the class of external-memory equipped networks rather than the inherent memory-based architectures, such as LSTM. It is a special kind of RNN and it is advantageous for

i. rapid learning from sparse data,

ii. its computational efficient storage capacity, and

iii. meta-learning tasks (i.e., it does not only learn how to solve a specific task but it also captures information on the way the task itself is structured).

Instead of a distributed hidden vector, MANNs have an external memory to model the hidden state. The external memory contains two parts, a memory matrix that stores the information and a controller that communicates with the environment and reads or writes to the memory allowing it to forget information. These operations also make use of non-linear activation functions.

In Fig. 1, Fig. 2, Fig. 3, the blue circular nodes capture the hidden students' knowledge state per skill, while the orange rectangles denote the exercise-performance observations associated with each skill. The nodes in the probabilistic models denote stochastic computations, whereas in the RNN indicate deterministic ones.

III. Dynamic Models Applied In Knowledge Tracing

The dynamic models applied in KT are described below.

A. *Standard Bayesian KT: skill-specific discrete states*

The BKT model [18] includes four binary parameters that are defined in a skill-specific way. The model emits two performance-related parameters:

i. S-slip, the probability that a student will make an error when the skill has been acquired, and

ii. G-guess, the probability that a student will guess correctly if the skill is not acquired;

The model additionally distinguishes between two learning-related transition parameters:

i.$P(\theta_{t-1}) = P(\theta_0)$, the probability of knowing the skill a priori, and

ii. $P(T)$ represents the transition probability of learning after practicing a specific skill on learning activities. The knowledge acquired is estimated using equations (1a), (1b), (1c) and (1d) as illustrated below. The acquired knowledge $P(\theta_t)$ on trial t is updated according to (1c) with $P(T) = P(\theta_{t+1} = 1 | \theta_t = 0)$. The probability of a correct or incorrect attempt is computed using equation (1a) and (1b), respectively. Equation (1d) computes the probability of a student applying the skill correctly on an upcoming practicing activity. The equations are as follows:

$$P(\theta_{t+1}|y_t = 1) = \frac{P(\theta_t) \cdot (1 - P(S))}{P(\theta_t) * \big(1 - P(S)\big) + (1 - P(\theta_t)) \cdot (P(G))} \tag{1a}$$

$$P(\theta_{t+1}|y_t = 0) = \frac{P(\theta_t) \cdot P(S)}{P(\theta_t) \cdot P(S) + (1 - P(\theta_t)) \cdot (1 - P(G))} \tag{1b}$$

$$P(\theta_{t+1}) = P(\theta_{t+1}|y_t) + \big(1 - P(\theta_{t+1}|y_t)\big) \cdot P(T) \tag{1c}$$

$$P(KC_{t+1}) = P(\theta_t) \cdot \big(1 - P(S)\big) + \big(1 - P(\theta_t)\big) \cdot P(G) \tag{1d}$$

At each t, a student m is practicing a step of a learning activity that taps a single skill S. The process of a student trying to acquire knowledge about S^1 is illustrated in Fig. 1 over one-time step. The learner state can be in one of the two states and can emit one observable. Given a series of y_t, and t for the student m and skill S^1, the learning task is the likelihood maximization of the given data $P(y| \lambda)$, where $\lambda = \{P(S), P(G), P(T), P(\theta_t)\}$. This is done through Curve Fitting or Expectation Maximization and evaluated via Mean Absolute Error.

The key idea of BKT is that it considers guessing and slipping in a probabilistic manner to infer the current state during the practicing process. Even though BKT updates the parameter estimates based on dynamic student responses, it assumes that all of the four parameters are the same for each student. It follows that, the data of all students practicing a specific skill are used to fit the BKT parameters for that skill, without conditioning on certain student's characteristics.

B. Individualized BKT: student-specific states on learning rates

Individualizing towards the learning rates $P(T)$ provides higher model accuracy and better parameters interpretability [23]. The Individualized BKT (IBKT) model [23] is developed by splitting the BKT parameters into two components *(i)* λ^k-the skill-specific, and *(ii)* λ^u-the student-specific; and combining them by summing their logit function $l(p) = log\left(\frac{p}{1-p}\right)$, and using the sigmoid function $\sigma(x) = \frac{1}{(1 + e^{-x})}$ to transform the values again to a probabilistic range. These two procedures are illustrated in (2*a*):

$$\lambda = \sigma\Big(l\big(\lambda^k\big) + l(\lambda^u)\Big) \tag{2a}$$

Finding the gradients of the parameters λ is done via forward and backward variables. Updating the gradients is possible using the chain rule, as illustrated in (2b) for the student-specific component of the parameter

$$\frac{\partial L}{\partial \lambda^u} = \frac{\partial L}{\partial \lambda} \frac{\partial \lambda}{\partial \lambda^u} \tag{2b}$$

where L simply indicates the loss function.

Fig. 1 depicts the structure for the HMM model of both BKT and IBKT. Although the underlying HMM model -and hence the process of a student practicing exercises- remains the same, the fitting process is different, i.e., λ^u is learned for each student separately.

Both the BKT and IBKT assume independent skills because thus they cannot deal with hierarchical structures. This assumption is restrictive, because it imposes that different skills cannot be related and, as a result, observing an outcome for one skill is not informative for the knowledge level of another skill. However, the expert model in educational domains is frequently hierarchical and should allow for multi-skill learning. DAG is the optimal data representation for describing the expert model in adaptive learning systems that incorporate parallel scalable architectures and BDA [3].

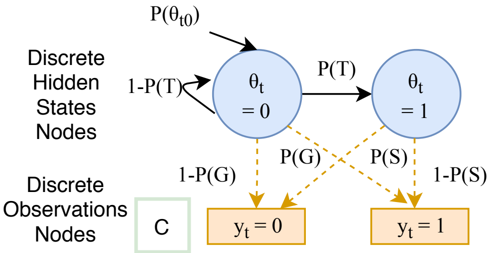

Figure 1. *Baseline and Individualized Bayesian Knowledge Tracing represented as a Hidden Markov Model over one time step*. In IBKT, the parameter {T} is learned seperately for each student.

C. Dynamic Bayesian Network: hierarchical, skill-specific, discrete states

DBN is a DAG implemented for the KT task [21] to allow for the joint representation of dependencies among skills.

On contrast to the previous models, at each timestep t, a student m receives a quiz-like assessment that contains problem steps or exercises that belong to different skills. The structure of the Bayesian network is repeating itself at each time step t with additional edges connecting the knowledge state on a skill at t to $t+1$. This is the learning or forgetting rate, previously denoted as $A = \{T\}$. Same as in BKT and IBKT, once a certain threshold for a skill mastery is reached, the user can start practicing the less mastered skills.

The enhancement of the model is based on the fact that it is possible to infer the knowledge state for a skill, say S^3, even without having observed certain outcomes for that skill y^3. To illustrate that, consider the example model depicted in Fig. 2. It depicts that, the probability of skill S^3 being mastered at t_2 depends not only on the state of S^3 at the previous time-step t_1, but also on the states of S^1 and S^2 at t_2.

The set of variables X contains all skill nodes S as well as all observation nodes Y of the model, while H denotes the domain of the unobserved variables, *i.e.*, exercises that have not yet been attempted by students and hence their corresponding binary skill variables S are latent. Suppose that a student solves a learning activity associated with S^2 at step t_2; then the hidden variables at t_2 will be $h_m = \{S^1, S^2, S^3, y^3, y^1\}$ while the observed variables will be y^2. The objective is to estimate the parameters λ that maximize the likelihood of the joint probability $p(y_m, h_m|\lambda)$. The likelihood loss is reformulated using a log-linear model to obtain a linear combination of a lower dimensional representation of features F, as shown in (3):

$$L(w) = \sum_m \ln\left(\sum_{h_m} exp(w^{\mathrm{T}}\varphi(y_m, h_m) - \ln(Z))\right) \qquad (3)$$

where φ: $\mathcal{Y}$ x $\mathcal{H} \rightarrow R^F$ denotes a mapping from the latent space $\mathcal{H}$ and the observed space $\mathcal{Y}$ to an F-dimensional feature vector. Z is a normalizing constant and w denote the weights that can be directly linked to the parameters λ.

DBNs rely on an accurate graph topology and can handle only simple topologies. Additionally, this model grapples with the limitations of the binary representation of student understanding, the lack of student differences, and the requirement for an even more detailed concept labeling and parameter constrain sets. RNNs have only recently tried to model student understanding in order to lessen or break the aforementioned assumptions.

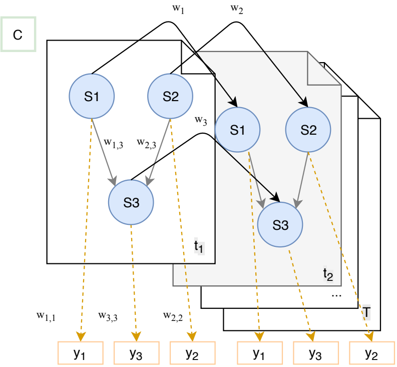

Figure 2. Bayesian Knowledge Tracing represented as a Dynamic Bayesian Network unrolled over T time steps. The hierarchical relationships between the skills (grey lines) are incorporated to the estimation of the learning rate (arrow lines) between adjacent time steps.

D. Deep Recurrent Neural Networks: continuous exercise-specific states and discovery of exercise dependencies

The complex representation in DKT is chosen based on the grounds that learning is a complex process [25] that should not rely only on simple parametric models as these models cannot capture enough of the complexity of interest, unless provided with the appropriate feature space [22].

The continuous and high dimensional representation of the latent knowledge state h_t in the hidden layer, learns the properties of sequences of the observed student interactions $x_t = \{(a_{m,0}, q_{m,0}) \ldots (a_{m,T}, q_{m,T})\}$, where a_t denotes the correctness of the response on a learning activity, which is denoted as q_t. In the deep learning context q_t denotes the corresponding activity tag, which can be roughly considered as a KC label.

DKT can discover exercise dependencies, i.e., prerequisites. Given that the knowledge state for a KC is represented as a hidden unit, the hidden-to-hidden connections encode the degree of overlapping between exercises. The researchers assign an influence metric J_{ij} on each directed pair of exercises i,j based on the correctness of the previous exercise i in the pair. They computed the correctness conditional dependencies between exercises $y(i)$, as shown in (4a):

$$J_{ij} = \frac{y(j|i)}{\sum_k y(j|k)} \qquad (4a)$$

where k is a predetermined threshold used to cluster the exercises that instruct the same skill. The possible skill labels for the clusters are manually provided.

DKT [25] exploits the utility of vanilla RNNLSTM whose fully and recurrent connections allow them to retain information of x_t for many time steps. The below equations describe the simple vanilla RNN and not the LSTM gates. Equation (4b) states that each hidden unit is activated via the hyperbolic tangent, which employs information on both the input x_t and on the previous activation h_{t-1} ,

$$h_t = tanh(W_{hx}x_t + W_{hh}h_{t-1} + b_h) \qquad (4b)$$

where b_h is the bias term and W_{hx}, W_{hh} are the weights of units corresponding to the input and hidden layers. The non-linear and deterministic output h_t will be passed to the sigmoid function σ to give the probability of getting each of the T learning activities correct $\hat{y}_t = (y_0, .., y_T)$ in the students' next interaction $t+1$, as shown in (4c):

$$\hat{y}_t = \sigma(W_{yh}h_t + b_y) \qquad (4c)$$

Finally, the loss for a single student will be the negative log-likelihood, as shown in (4d):

$$L = \sum_t l(\hat{y}^T \delta(q_{t+1}), a_{t+1}) \qquad (4d)$$

where l is the binary cross entropy $-(ylog(\hat{y})) + (1 - y)(1 - log\,(\hat{y}))$, δ denotes the one hot encoding transformation of the input $h_t = \{a_t, q_t\}$, which represents the categorical variables as binary vectors, and x_t is assigned to h_t. Compressing sensing is a suitable preprocessing step for larger data sets.

Fig. 3 depicts an example architecture of RNN, where X represents the entire sequence of exercises, which belong to multiple KCs, in the order the student receives them. After feeding X to the network, each time the student answers an exercise, the model predicts what KCs they are able to solve on their next interaction.

DKT requires large amounts of training data and it is prone to overfitting. Furthermore, it summarizes a student's knowledge state of all KCs in one hidden state vector, which makes it difficult to trace how much a student has mastered a certain skill over time. Zhang et al. (2017) [47] proposed a parameterization of MANN to address these two issues.

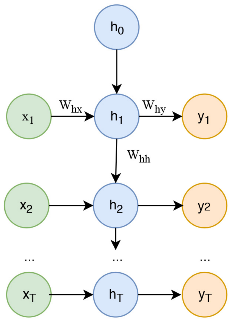

Figure 3. Deep Knowledge Tracing represented as a Recurrent Neural Network over 2 trials represented by the input $x_t = (a, q)$ and output y_t that denotes the probability of getting each of the q correctly. The lines in the hidden layer represent the learning rate between adjacent hidden knowledge states (blue nodes).

E. *Memory Augmented Neural Networks: skill-specific states & discovery of exercise clusters*

In the KT learning task, at each timestamp a MANN model takes a discrete exercise tag q_t, outputs the probability of response p$(r_t|q_t)$, and then updates the memory with the tuple (q_t, r_t). The MANN is extended to utilize a key-value, rather than a single memory matrix [47] because the exercise tags and the responses have different data types. The key component M^k, which is a static matrix, attends the latent skills underlying the exercises. The value matrix M_t^v stores, forgets, and updates the student's understanding of each skill h (skill state) via the read and write operations; and it changes over time.

Hence, the so-called Dynamic Key Value Memory Network (DKVMN) traces the knowledge of a student by reading and writing to the value matrix using the correlation weight w_t, which is commonly annotated by experts in the probabilistic KT framework. This is computed by taking the SoftMax activation of the inner product of the input exercise k_t^T and the key matrix M^k, as shown in (5α):

$$w_t(i) = softmax\,(k_t^T M^k(i)) \qquad (5α)$$

where i indicates the memory slot and k_t^T, arises after the multiplication of q_t with an embedding matrix so as to obtain a continuous embedding vector of the appropriate dimensionality size.

At each timestamp t, the learner solves an exercise tagged with q_t, the model finds that q_t requires the application of let's say skill $S1$ and reads the corresponding skill state h_{t-1}^{S1} from the read content r_t. This acts as a summary of the mastery level of the student for this exercise, as shown in (5b):

$$r_t = \sum_{i=1}^{N} w_t(i) M_t^v\,(i) \qquad (5b)$$

Then it predicts p_t, which is the probability that the student will answer q_t correctly, as shown in (5c):

$$p_t = sigmoid(W_2^T f_t + b_2) \qquad (5c)$$

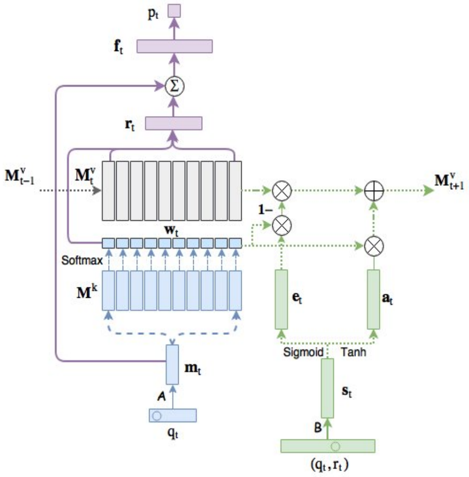

Figure 4. Deep Learning for Knowledge Tracing represented as a Dynamic Key Value Memory Network for one time step. The blue components denote the process of the correlation computation between the exercise and the underlying latent concepts, the purple components indicate the prediction process, and the green describe the update process that takes place after the students' interaction.

where f_t is a vector that contains both the student's mastery level and the exercises prior difficulty. It is calculated by a fully connected network as shown in (5d):

$$f_t = tanh(W_1^T[r_t, k_t] + b_1) \tag{5d}$$

where T and b indicate the transpose operation and the biases vectors respectively. After the student response is given, the model updates the values of h_{t-1}^{S1} . In this way, each time the student answers an exercise, the model not only predicts what exercises they are able to solve on their next interaction but also maintains a student's mastery level of each skill. The DKVMN makes use of the one-hot encoding preprocessing step and the binary cross entropy loss function. In Fig. 4 [47], the read and write processes of the model are described as purple and green components, respectively. It is implied that, the inaccurate estimation of $w_t(i)$ can lead to inaccurate predictions and updates.

IV. Comparison & Summarization of the Models

Tables III, IV, V outline important aspects of the models described above. Three dimensions are used as a guide for summarizing the models. The first is the machine learning algorithms' components [15], depicted in Table III, the second is the algorithmic scalability [46], illustrated in Table IV, and the third includes human learning related challenges faced by adaptive learning systems [36], described in Table V.

A. Machine learning components

The criterion of choosing the right algorithm is a combination of the efficiency of the available data along with the learning components of the algorithm; these are the representation, evaluation, and optimization [15]. In the below paragraphs, we briefly describe each of these components considering the KT task. The representation component has been already introduced in Section II.

1) Evaluation of the predictions

Model evaluation metrics analyze the performance of the model via the computation of training and the out-of-sample error; and are widely discussed in the context of machine learning applications including educational ones [26]–[28]. Even though, no experimental data are presented throughout the review, choosing for a metric is an open question in EDM including KT [27] for the assessment of the quality of the learner model. It depends highly on the intended use of the model and on whether absolute or relative predictions are important for this use. The metrics and their intended use are summarized in Table II based on findings from previous research [26]. This table can be used as a guide for assessing the quality of KT modeling concerning the evaluation metrics linked to it.

In KT, the metrics used for probabilistic understanding of errors include the Mean Absolute Errors (MAE) and Root Mean Square Error (RMSE). The former is considered an insufficient metric because it is biased towards the majority of classes whereas the latter is a proper score [26]. From the perspective of model comparison, the important part is only the sum of squared errors and not the square root. Note that RMSE has demonstrated a high correlation to the log-likelihood function and the *'moment of knowledge acquisition'* [26], which is highly important for mastery learning applications.

The RMSE without the squared error is sometimes referred to as the Brier Score and can give further insight to model behavior via decomposing it into three additive components. These are the following:

i. reliability, which measures the difference between predicted and observed probabilities,

ii. resolution, which captures the difference of the predictions from the base rate (proportion of positive classes), and

iii. uncertainty, which quantifies the inherent uncertainty of events.

An ideal model would, therefore, minimize the reliability term, while maximizing the resolution term. Assume that q_k are the model's predictions that they can take a set of different values c or values from c classes, n_k is the number of predictions that belong to the same category, and $f_k = \sum_{i,p_i= q_k} {}^{o_i}/_{n_k}$ is the frequency of observations. The Brier score is used by DBN whose formula is depicted in Table II.

As opposed to the probabilistic understanding of errors, values of qualitative metrics, i.e., either the prediction is correct or incorrect (0-1 loss), depend on the choice of the classification threshold. In the reviewed models, only classification accuracy was employed to evaluate the number of correctly predicted successes and failures on exercises. This measure reflects the proportion of true positives (TP) and true negatives (TN) as proportion of the total number of predictions (N). However, accuracy is not a reliable metric when the targeted classes are imbalanced. Recall is then a better metric to use, as it reflects the proportion of relevant incidents predicted correctly by the algorithm over the number of total relevant incidents. Commonly, this measure is used together with precision in F1 score, which is a more reliable metric than accuracy.

The Receiver Operating Characteristic (ROC) curve summarizes the qualitative error of the prediction model over all possible thresholds, so it summarizes performance even over those thresholds for which the algorithm would never be practically used. The predictions are considered relative to each other, and therefore the area under the ROC curve, called AUC, is better to be used as an additional metric for the evaluation of an algorithm's ability to distinguish correct from incorrect performances on exercises. It is interesting to note that, when the overall AUC is computed by averaging the per-skill AUC, namely weighing all skills equally, then its value is going to be smaller than by weighing all trials equally. This effect roots in two situations: i) the model performs poorly on a skill with only a few observations, and ii) it predicts the relative accuracy of different skills [22].

DKT and DKVMN employ the AUC on a per-trial basis instead of per-skill. Different from Bayesian methods, the deep learning models do not have the skill notation of each question and thus they cannot evaluate the results per skill.

Overfitting is a common source of error in machine learning models. That is, the model memorizes the training data and cannot generalize to out-of-sample data. The more increasing the number of model parameters, the more the danger of overfitting, since there will be less data for each subset of parameters that have to be estimated. All the aforementioned models apart from the standard BKT, compute the errors on Cross Validation (CV) so as to lessen the issue of overfitting. The folds are selected such that the mean performance of students is approximately equal to all folds, a technique referred as student-stratification.

TABLE II. EVALUATION METRICS AND APPROPRIATE USES FOR KNOWLEDGE TRACING

Metric	EDM Uses
Probabilistic	***Parameter Fitting & Model Comparison***
MAE	$^1/_n \sum_{i=1}^{n} \lvert y_i - \hat{y}_i \rvert$
RMSE	$\sqrt{^1/_n \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$
Several Numbers	***Model Comparison & Behavior***
Brier Score	$\frac{1}{N} \sum_{k} n_k \left[(q_k - {f_k}^2)(f_k - f^2) \right] + f(1-f)$
Qualitative	***Evaluation of Classification Tasks***
Accuracy	(TP+TN)/N
Recall	TP/(TP+FN)
Ranking of examples	***Interpretability of Results***
AUC	x-axis: (FP+TN), y-axis: Recall

The predictive ability of learner models is mainly a mean for improving the behavior of educational systems and for getting insight into the learning process. The automated evaluation metrics do not correlate with learning outcomes, namely, they cannot consider the fact that an adaptive learning system may not improve actual learning (e.g., decreasing learning curves). Therefore, frameworks oriented to adaptive learning systems should arise [27].

2) Optimization, Identifiability and Degeneracy

The optimization function derives the optimal values for the parameters of the objective function, which in KT is the log-likelihood function. Unlike in most other optimization problems, the function that generates the data and should be optimized is unknown and hence training error surrogates for the out-of-sample error [15]. The optimization of the log-likelihood function is performed using Curve Fitting (CF), Expectation Maximization (EM), Constrained optimization, and Gradient Descent (GD) methods.

Incremental optimization algorithms are suitable for large-scale data. GD on mini-batches is an incremental algorithm, which updates the weights using batches of data, and thus can avoid shallow local maxima. For instance, the IBKT model is built in an incremental manner by adding λ^u in batches and evaluating these additions on CV performance. It is also possible to improve the overall accuracy by incrementally updating the λ^k once a new group of students finishes a course or a course unit.

In addition to the big data handling, GD allowed IBKT to introduce student-specific parameters to BKT, without expanding the structure of the underlying HMM model and thus without increasing the computational cost of fitting. Researchers computed the gradients of the log-likelihood function given individual student and skill data samples with respect to every parameter. On every odd run, gradients are aggregated across skills to update skill component of the parameters; whereas in every even run, the gradients are aggregated across students to update respective student components. This block-coordinate descent is performed until all parameter values stabilize up to a pre-defined tolerance criterion.

In the procedure of training deep learning models, gradients tend to be unstable in the earlier layers as they either explode or vanish. Certain activations functions can cause this behavior. The exploding gradient problem refers to the large increase in the norm of the gradient, and hence it takes much time to converge to the parameters. To deal with the exploding problem, the DKVMN uses 'norm clipping' which is a function thresholding the values of the gradients before performing a gradient descent step; while DKT manually put thresholds to the values of the back-propagated gradients. The vanishing gradient, refers to the opposite behavior, when there is a small increase in the norm of the gradient, making it impossible for the model to learn from training data, i.e., find the correlation between temporally distant events. An interesting fact is that the LSTM model implemented by researchers in the DKVMN achieved better AUC than in the original paper of DKT. This could be probably because the DKVMN used 'norm clipping' and 'early stopping' instead of 'dropout' and manual thresholds to gradients.

In contrast to GD, EM takes longer to converge because even if it needs fewer evaluation steps, the computations are more difficult. In addition, due to the expectation step it does not directly maximize the likelihood of the learner's observations [23].

Constrained optimization is used in DBN [21] to ensure the interpretability of the constrained parameters and avoid the intractability issue, which can be caused by approximating the objective function. It obtains the joint distribution as a product of the exponential terms, which translates to a weighted linear combination of feature vector entries in the exponent. This implies that the search of model parameters is done to a specific direction, in order to find the feature that is responsible for a prediction outcome.

The probabilistic KT models are susceptible to the identifiability and model degeneracy issues [20][28]. An identifiable model is considered the one that converges to the true values of the parameters, given an *infinite* number of observations. Model degeneracy occurs when the same combination of model parameters fits the data equally well. Hence, the resulted model parameters can lead to paradoxical behavior [28][30]. An example of a paradoxical behavior is the probability that the student acquired the instructed knowledge after three correct answers in a row [29]. Appropriate initialization conditions of the models' parameters and constrain values of the emission parameters, i.e., guess and slip, are used as techniques to resolve these two issues [18][28]–[31]. The identifiability and degeneracy are not relevant to deep learning frameworks, since they use approximation function and cannot pinpoint the input signal that lead to specific values for the model parameters.

TABLE III. COMPARISON OF KT MODELS: SUPERVISED MACHINE LEARNING COMPONENTS

Model	**Extension**	**Representation**	**Optimization**	**Evaluation**
BKT	Baseline- Binary skill Knowledge State	HMM	Curve Fitting or Expectation Maximization	MAE
IBKT	Learning Rate Personalization	HMM	Stochastic Gradient Descent on Minibatches	RMSE
DBN	Multi-Skill, binary Knowledge State	DBN	Constrained Latent Structure	RMSE, AUC, Brier Score
DKT	Continuous Knowledge State & Discovery of concept map	RNN-LSTM	Stochastic Gradient Descent on Minibatches	AUC, Accuracy
DKVMN	Skill Knowledge State & Discovery of concept map	Memory Augmented Neural Networks with Key-Value Matrices	Stochastic Gradient Descent on Minibatches	AUC, Accuracy

DKT, IBKT and DKVMN are prone to overfitting and need large-scale training data to optimize the objective function. Compared to the DKT, DKVMN does not require such large-scale data for training and is less susceptible to overfitting. Stopping the training before the weights have converged (i.e., 'early stopping'), and dropping out units (i.e., 'dropout') are methods that address the overfitting problem. Another issue present in DKT, is the alternation between mastered and not-yet-mastered state instead of the state transiting gradually over time [31], known as the waviness of the objective function. In addition to that, the model sometimes fails to reconstruct the input, which implies that even when a student performs well on a KC, the prediction of that KC's mastery level decreases instead, and vice versa [31]. According to the current literature, potential paradoxical behaviors while fitting the DKVMN are not yet investigated.

B. *Scalability of the learning algorithms*

Online education platforms create large-scale and diverse learning behavior data. An important aspect of BDA is the algorithm's ability to scale as new data or new features come into the model. In contrast to scaling towards the number of features, scaling towards the number of learners is an easier task that can be solved via using parallel infrastructures. In this study, we just scratch the surface of algorithmic scalability. Questions of large-scale data representation typically have much more complicated extensions in algorithmic, statistical, and implementation or systems aspects that are intertwined and need to be considered jointly.

1) *Computational & Statistical Efficiency*

Computational efficiency refers to the number of computations during training [46]. These include the numbers of the following:

i. iterations of the optimization algorithm,
ii. model parameters, and
iii. resources (i.e., the number of hidden units).

In Table IV, we note only the number of model parameters; this is not necessarily the most appropriate measure of model complexity. Nonlinear functions and large datasets increase the model complexity while offering flexibility in data fitting [20].

Implementing the DKVMN and especially the DKT models demand high numbers of computational resources. Nowadays, there are many parallel and distributed computing infrastructures and there is active research on parallel algorithms that can be used to boost the efficiency of data-intensive tasks. Parallel and scalable algorithms are often utilized for BDA. DKT, DKVMN, and IBKT models took the advantage of parallel computing infrastructures.

Compared to HMM model, DBN is computationally more expensive due to its complex loopy hierarchical structure [21]. Comparing the two deep learning models, LSTM is computationally more expensive than MANN, based on the ability of the latter to not increase the number of parameters when there is increasing number of memory slots. In general, the probabilistic models are computationally more efficient than the deep learning models even in other domains apart from online skill acquisition.

Statistical efficiency refers to the number of training examples required for good generalization performance. The volume of training data examples required to establish convergence, is depicted in Table IV as learnability requirements. Learnability, which is part of statistical efficiency, appears to present a daunting challenge for deep learning. Regarding the probabilistic KT models, the inclusion of large-scale training data can prevent identifiability problems. It is also important to note that, the parameter estimates and the behavior of the probabilistic KT models should be researched in different prior cases $P(\theta_0)$

and in scalability cases in either the number of students or the increased number of interaction examples per student [30].

2) Domain-Knowledge Dependence

In the educational domain, domain-knowledge or otherwise called human involvement is not only expensive to obtain, but there are also many differences in content representations beliefs among experts. The model is more flexible if it is less domain-dependent, implying that its performance is less prone to additional error. In contrast to the probabilistic models, the deep learning models are highly flexible. Beside deep learning models, DAG models can also learn the structure of the concept map as a network. However, the relationships between content are difficult to establish and to represent in the model, even given expert labels [36].

Adding or deleting pieces of modules or content in expert models is also an important scalability dimension related to the content representation. Rule-based algorithms fail to scale while flexible graph representations are much easier to scale. DAG models are considered the optimal representation of the relationships between concepts/skills (abstract but intuitive notions of ideas that the content teaches and assesses) and KCs/content units (pieces of content). A similar scalable graph ontology based on concepts and content units is used in Knewton, a well-known, online, adaptive learning tool.

C. Adaptive Learning Properties

There are many adaptive learning properties [36] and in this study the focus is placed on individualized learning rates [23], multi-skill learning [21], recency engagement, and skill discovery [25] [47].

1) Student Differences: Prior Knowledge, Learning Rate & Recent Engagement

Modeling parameters on an individual level results in significant changes regarding instructional or mastery decisions [45]. Researchers found that the inclusion of student-specific parameters has a significant positive effect on prediction accuracy and interpretability [23][24], as well as in dealing with overfitting [23]. Researchers [24] added Dirichlet priors for the initial mastery θ_{t-1}, while IBKT [23] extended their work and found that adding variables of learning rates $P(T)$ for individual learners, provides higher model accuracy. DKT and DKVMN allow for differences in learning ability of the student by conditioning on the average accuracy of recent learner's performance across trials and skills.

Learners' data include temporal dependencies, namely, there is a correlation in time engagement within or across learning resources and student's performance on activities, as described in Section II. Furthermore, recent performance is more predictive than past performance. DKT and DKVNM inherently are more sensitive to recent trials, allow for long-term learning and can capture temporal dependencies.

The probabilistic KT tends to predict practice performance over brief intervals where forgetting the acquired knowledge is almost irrelevant; though extensions of BKT towards this direction have been proposed. These include forgetting from one day to the next and not on a much shorter time scale [22]. DBN includes the forgetting property, but it would be more insightful if the model was compared with the equivalent BKT extension that includes forgetting.

Beyond the student knowledge that is reviewed throughout the paper, there are some single-purpose KT models augmented with non-performance data such as meta-cognitive [42], affect [43], and other student differences [44][45] apart from the learning rates that we reviewed.

2) Skill Dependencies & Adaptive Instructional Policies

The effect of sequencing learning resources is an important adaptive property of learning systems. Each skill has some degree of influence on the learning of other skills, especially in hierarchical domains of knowledge, such as algebra and physics. The paths through learning resources such as exercises, KCs, or abstract skill concepts taken by learners can influence their knowledge acquisition process.

TABLE IV. COMPARISON OF KT MODELS: SCALABILITY

Model	**Learnability requirements[i]**	**Efficiency[ii]**	**Domain Knowledge Dependence[iii]**	**Limitations**
BKT	↓↓↓	4 /skill, ↑↑↑	↑	Prone to bias, independent skills assumption, local transitions between states
IBKT	↑↑	4 /skill + 1 /student *(a)* ↑↑	↑	Independent skills assumption, local transitions between states
DBN	↓	4 /skill + 2^{n-1} for n skills ↑↑	↑↑	Hard-coded skill dependencies, complex, tractable only for simple models, local transitions between states
DKT	↑↑	250K with 200 hidden units & 50 skills *(b)* ↓↓	↓	Highly complex, prone to overfitting, not interpretable
DKVMN	↑	130K with 200 states & 50 skills ↓	↓	Highly complex, not interpretable, local transitions between states

i. the higher, the more complex the model,
ii. the higher, the less complex the model,
iii. the higher, the less flexible the model,
a. only the learning rate is individualized,
b. 4(input size+1) * output size + output size^2.

DKT and DKVMN can discover the inter-skill similarities and exercise prerequisites without requiring any domain knowledge apart from the exercise tags. A set of skill labels are manually provided but the annotation part is done by the model. They can return the sequence of learning resources to a student that maximizes the expected knowledge state of that student. An interesting question is whether a sequence should contain exercises that belong to different skills or refer to one skill only. A trained DKT can be used in a Markov Decision Process for testing this scenario given a further time horizon (i.e., long-term learning). They found that presenting the exercises in an interleaved order of skills yields higher predicted knowledge after solving fewer problems, relative to presenting the exercises in a blocked order of the same skill.

Currently, the deep learning models can capture the relationships among exercises within a skill. It is worth mentioning that although deep learning models suffer from the lack of hierarchical input data structures [48], recently there is a lot of research in the direction of graph based deep learning models able to address hierarchical structures [52].

BKT and IBKT assume that each skill is independent and thus cannot be directly used to infer adaptive instructional policies; since they cannot keep the absolute sequence of exercises, given that they are not implemented in a mastery-learning way. A student's raw trial sequence is parsed into skill-specific subsequences that preserve the relative ordering of exercises within a skill, but discard the ordering relationship of exercises across skills.

TABLE V. COMPARISON OF KT MODELS: ADAPTIVE HUMAN LEARNING COMPONENTS

Model	Inclusion of forgetting rate	Inter-Skill Similarity & Instructional Policies	Learner individual differences	Multi-skill learning
BKT	✗	✗	✗	✗
IBKT	✗	✗	✓	✗
DBN	✓	✓	✗	✓
DKT	✓	✓	✓	✗
DKVMN	✓	✓	✓	✗

On the other hand, DBN allows for modeling hierarchical skill-dependencies, given a detailed expert model and can yield meaningful instructional policies. It can be used to offer an adaptive number of exercises that need to be solved for skill mastery. This is a mastery learning setting where the effort (number of practice opportunities needed to pass a skill) and score (percentage of correct observations after having the skill passed) of a learner are optimized [27][45]. Though, DBN is computationally tractable only for the simplest topologies among skills since exact inference is exponential in the number of parents a node has. Given a large-scale dataset, approximate inference can be used to exchange accuracy with computational time.

D. *Comparison of KT models' applications & performance*

All the models track at each time step the evolution of student knowledge state in real time for each skill separately [18][23], for a set of skills [21][47], or for a set of exercises [25][47]. This implies that, after a students' interaction with an exercise, the models update the knowledge state of each student in a skill [18][23] or skillset [21][25][47] way. To illustrate this with an example, let's assume there are fifty exercises where "bivariate data frequencies", "linear models of bivariate data", "plotting the line of best fit", "interpreting scatter plots", and "scatter plot construction" correspond to distinct labels of five exercises. DKT and DKVMN take the past performance of students on the sequence of 50 exercises and the current performance of a student on an exercise, and it will predict the probability of getting each of the exercise correct in their next interaction. Both models will cluster these exercises in one cluster named 'Scatter Plots' which can be roughly considered as a single skill. Different from DKT, the advantage of DKVMN is that it is more powerful in storing past performance of a learner since it can maintain the knowledge state per skill instead of only per distinct exercise label.

The three probabilistic models [18][21][23] cannot capture the relationships between the exercises. BKT and IBKT assume that, the labels provided in the example above correspond to different fine-grained skills, each one separately modelled. Each time an answer is provided for an exercise, the model will give the probability of the level of skill acquisition at time t given the probability of the level of skill acquisition on an exercise on the previous time step. Commonly, once a certain mastery level is reached, the learner can move to the next skill. Different from that, DBN will output a collection of probability distributions specifying the knowledge level for each skill or KC label. IBKT is the only model that individualizes toward learning rates of students.

KT models are commonly applied in curriculum sequencing and mastery learning frameworks, both axes are useful for smart and adaptive learning environments [5][7][10]. The former is used to either return or recommend to a learner a dynamic, optimal sequence of learning resources, whereas the latter is used to estimate the point of time that a certain skill is acquired [18] and from that point, learners are considered able to handle more advanced concepts. The DKT and DKVMN are used for curriculum sequencing while the HMMs and DBN for mastery learning. However, this study [49] suggests that KT models are better suited for discovery of concept and exercises relations, already included in DKT and DKVMN, rather than mastery learning applications. Mastery learning applications are threshold-dependent since it is difficult to automatically define an optimal threshold value.

DKT and DKVMN are complex, flexible and non-transparent models. DKT led to 25% increase in AUC when compared to the relatively simple BKT [25]. However, its success is attributed to its flexibility in capturing statistical regularities directly present in the inputs and outputs, instead of representation learning [22] which is the fundament advantage of deep learning models. When the performance of the DKT model and variations of BKT is compared [22], it is found that both models perform almost equally well. These variations allow for more flexibility in modeling statistical regularities that DKT has already the ability to explore because of the LSTM structure. DKT is presented with the whole trial sequence and thus it can discover aspects within interactions; whereas probabilistic models are given one student interaction at each time step. DKVMN performed better than the MANN baseline, DKT, BKT, and some BKT variations, as authors reported in [47].

Deep learning models can be superior towards the probabilistic ones, only if they are fed with more complex input data instead of exercise-performance interactions. Thus, they can take the full advantage of featurization and learn the representation of knowledge acquisition. These models can work only in platforms with a relatively large number of students and interactions, while not requiring significant domain expertise. All models can perform better when a bigger number of students and interactions is available to train the algorithm.

DBN led to significant improvements in prediction accuracy compared to BKT, and the logistic models of Additive Factor Model, and Performance Factor Analysis [21]. Researchers suggest that the performance differences between DBN and BKT, need to be investigated further. DBN is a highly structured and hierarchical model that can work well in hierarchical domains of the instructed concepts; and given the availability of accurate domain expertise for the detailed development of skills topology and complex constraint sets. It can infer a student's mastery on skills even if there are not any observed interactions linked to these skills, given there are interactions on other related skills. They can perform well in mastery learning applications.

IBKT also performed better compared to BKT and BKT variation of prior knowledge individualization [24]. IBKT is the only model that allows for wide variations among students.

IBKT as well as DBN are single-purpose models [32]. Combining the benefits of skill hierarchies and accounting for student differences could introduce a more holistic model. However, probabilistic models use conditional probability tables to make inferences of a learners' state, whose time and space complexity grows exponentially with the number of states and features. IBKT used logit functions to lessen this issue and incorporate user-specific features. An efficient framework allowing the integration of general features into KT via logistic models is introduced in [32].

Table VI outlines potential applications of each model in an adaptive learning platform and the effects that each model can infer.

V. Item Response Theory For Predicting Future Performance

This review focuses on KT, thereby ignoring the only available alternative, which is Item Response Theory (IRT) [34][35]-[37]. Theoretically, IRT models differ from KT in that it focuses on summative tests in which no learning occurs, or on modeling very coarse-grained skills where the overall learning is slow [33]. This implies that IRT is a static model where student's knowledge does not change over time. Technically, IRT uses logistic models, i.e., discriminative algorithms discussed in Section II and cross-sectional data where learners' interactions directly estimate the ability parameter. An important advantage of logistic models that follows up is their ability to keep a linear algorithmic complexity while integrating a variety of features into the model; but this comes with the expense of the large-scale training data requirement. Hence, especially the more sophisticated IRT variants can be directly used for multi-skill learning and to account for variability in student a-priori abilities or guesses.

TABLE VI. Applications Of KT Models

Model	IBKT	DBN	DKT	DKVMN
Appli-cation	Individualized learning pace, Personalized feedback on progress	Adaptive number of learning resources, Feedback on progress & effort minimization	Adaptive order of educational activities, Feedback on exercises progress	Adaptive order of educational activities, Feedback on concept & exercises progress
Effects	Student differences on learning rates	Multi-skill learning	Student's differences on performance Discovery of exercise relationships	Student's differences on performance Discovery of exercise relationships

The baseline IRT Rasch model, known as the One Parameter (1PL) IRT, assumes that the probability of a correct response is mathematical function of the difference between student knowledge on skill θ and an item difficulty β, as depicted in (6). The responses to items are independent and occur at constant average rate. The items are considered conditionally independent to each other and β is better estimated when there is a large amount of data to calibrate them.

$$p_i(y = 1|\theta) = \left(1 + exp\left(-(\theta - \beta_i)\right)\right)^{-1} \qquad (6)$$

The most sophisticated of 1PL IRT descendants include the Additive Factor Model (AFM), which incorporates features of learning rates and skills, and its extension the

Performance Factor Analysis (PFA). The literature has already compared the models of PFA and BKT, both in theoretical [34] and in technical [35] terms *(i.e., predictive accuracy and parameter plausibility*).

AFM is depicted in (7), where $q_{ki} = 1$ if item i uses skill k, and 0 otherwise, and γ_k and T_k denote the learning rate and the number of exercises the student has solved for skill k, respectively. This model is better estimated when there is a large number of learning responses available for calibration.

$$p_i(\theta) = (1 + \exp(-(\theta + \sum q_{ki}(\beta_k + \gamma_k \cdot T_k))))^{-1} \quad (7)$$

The PFA model developed to differentiate correct from incorrect responses. It is highly predictive but not useful for adaptive environments in the sense that it cannot optimize the subset of items presented to students according to their historical performance [27]. The PFA is depicted in (8),

$$p_i(\theta) =$$
$$(1 + \exp(-(\theta + \sum q_{ki}(\beta_k + \gamma_k \cdot S_k + \rho_k \cdot F_k))))^{-1} \quad (8)$$

where S_k and F_k denote the number of correctly and incorrectly solved items for a student at skill k, respectively. The fixed effects γ_k and ρ_k, therefore, denote the learning rates associated with correct and incorrect responses, respectively.

IRT models, which need to estimate simultaneously the entire interaction trajectory for each student with item parameters [37], or require large samples for calibration [33], are considered difficult to implement in an online environment; and together with KT are rarely evaluated with respect to real-time prediction performance [36].

It is interesting that the equation (2a) of IBKT incorporates the intuition of IRT, when summing the logistic functions to incorporate skill and student-specific parameters. AFM and PFA models are found [21] to achieve high resolution in Brier Score when compared to DBN because they are directly fitting a curve over time while the AFM achieve bad reliability, most probably because it does not differentiate correct from incorrect answers.

VI. Prospects And Challenges

The quality of a KT model is measured by its ability to predict learner performance. However, its key use is to recommend dynamic instructional policies like deciding sequences of learning resources, so as to guide learners towards achieving optimal learning outcomes in an efficient way. This raises five challenges and future directions.

Firstly, in case of instruction recommendations, deep learning KT models should be transparent and inform the learner about the underlying intuition of the recommended decisions. This requirement is also present as a “right to explanation” in the General Data Protection Regularization law in European countries. There is research oriented to explainable AI, such as the usage of a knowledge graph as reasoning evidence for the predictions of deep learning models [52] or the development of frameworks for interpretable machine learning models [53]. Both are still in early stages and commonly oriented to other domains than that of education.

Because of the importance of generalizing to new examples, which depends both on the right representation model and the sufficiency of data, it is useful to briefly approach KT from a data-centric side. In general, modeling knowledge acquisition is a complex task as human learning is grounded in the complexity of both the human brain and knowledge organization. From a social science perspective, learning is influenced by complex interactions, including affect [38], motivation [39][40], and even social identity [41]. Though, the data used as input for the described KT models are not complex; since predicting student knowledge with the mere observation of correct versus incorrect responses to learning activities provides weak evidence. As educational apps and smart learning environments increase in popularity, it may be possible to collect valuable, diverse and vast amounts of student learning data, able to capture the reality of learning; and hence create opportunities, as well as new challenges, in the utilization of the deeper insights of each learner’s knowledge acquisition trajectory.

Therefore, the second future direction concerns the inclusion of data beyond student performances. These could be conventional patterns like hint usage, exercise skipping [54], exercise difficulty perception [55], response times, involvement in discussion forums, leverage of personalized or social comparison feedback [56], or sensory patterns of facial expression, body temperature, eye movement, and body language. A shift to deep learning modeling will offer superior results only given data more behavioral and complex. It is worthy also to note that, until now, most of KT models model hint usage and exercise skipping as an incorrect answer, which is mathematically convenient but loses information about learners’ behavior.

Another example of rich data could be the inclusion of learners’ input such as their diagnosis of their prior knowledge about the instructed topic or their learning intention for topic acquisition. Such features can be included as a prior to the Bayesian or as an additional input to the network. This is important for adaptive learning systems, which face difficulties on making accurate inferences and sensible recommendations, when little data about the student is available (i.e., cold-start problem in recommendation systems) or for learners who were previously inactive for a long time [36]. Obviously, such a task is not trivial and raises other questions such as what kind of questions should be asked so as to both not overwhelm the learner and at the same time gather the maximum amount of information regarding their level of knowledge.

Both probabilistic and deep learning models are not easily scalable to include richer student and skill-specific features due to ‘the curse of dimensionality’. Large-scale datasets are necessary for alleviating this issue. Scalable frameworks and incremental, parallel algorithms are open research topics in the field of AI.

Thirdly, open issues remain the automatic setting of adaptive thresholds in mastery learning and the definition of optimality in dynamic learning paths which are conditioned on continuous learning behaviors.

The fourth future direction is related to the evaluation metrics of learner models, which should be more directed at measuring performance with respect to the learning outcomes. Frameworks and metrics specifically oriented to adaptive learning purposes should thus arise. Another related challenge is that none of these models have been evaluated on online recommendation tasks.

The fifth future direction is concerned with the expert model that describes the content relationships. The Bayesian models depend on accurate domain knowledge but defining content relationship and designing exercises is not only hard to accomplish, due to their high hierarchy and diversity, but also subject to human opinions across the globe. Deep learning models do not need domain experts but this may also be considered as an extreme in the educational domain. Hence, the utilization of knowledge graphs, crowdsourcing, or semi-supervised techniques that learn the topology of the content could possibly be considered as safer paths than the either highly structured or abstract representation of skills.

VII. Conclusions

Modeling learner's skill acquisition and predicting future performance is an integral part of online adaptive learning systems that drive personalized instruction. Knowledge Tracing has the capability to infer a student's dynamic knowledge state as the learner interacts with a sequence of learning activities. In this review, we described the probabilistic and deep learning AI approaches that are used to model the evolution of knowledge acquisition. We outline their technical and educational requirements, advantages, and limitations with respect to adaptive human learning, supervised sequential machine learning, and algorithmic scalability.

The deep learning approach models a continuous learner's state for multiple skills and can explicitly induce temporal aspects related to adaptive learning without being knowledge-domain dependent. Predictions and learning recommendations can be enhanced by including more complex data. The usage of frameworks towards AI explainability is also a beneficial step. The incorporation of regularization techniques can help in overfitting issues and inconsistent predictions.

Equivalent features in the probabilistic models are the incorporation of more flexible content representations and justified assumptions about the knowledge state dynamics. A Bayesian approach models a binary state either for one or multiple skills and is highly domain-knowledge dependent, especially in the latter case. Optimization algorithms in the Bayesian models are susceptible to local optima and multiple global optima, where proper parameter initialization and constraints have shown to alleviate these issues. Importantly, the performance of probabilistic models depends highly on the setting of a good prior probability.

Specifically, the IBKT and DBN are single-purpose models; The IBKT can be used to infer individualized learning paces while the DBN can be used for multi skill learning. The latter is tractable only for simple skill topologies. Approximation inference can improve the running time of the algorithm where justified constraint sets should be defined to ensure the interpretability and accurate estimates of parameters.

The DBN together with the DKT and the DKVMN can be used for adaptive instructional policies. The DKT is the most complex and together with the DKVMN are the only models that are non-transparent and that can discover inter-skill similarities. The IBKT, DKT, and DKVMN models require a relatively larger amount of training data than the BKT and DBN models.

An open issue regarding all models is the leverage of rich, general features and their corresponding algorithmic scalability. Furthermore, the choice of evaluation metric should be chosen based on the intended use of the model that adheres to the ultimate purpose of improving learning experiences. Lastly, to the best of our knowledge, there is a research gap on whether KT models are better for the offline discovery of exercises and skills relationships rather than the online decision-making part of mastery learning or instructional policies.

References

[1] A. Sapountzi, S. Bhulai, I. Cornelisz, and C. van Klaveren, "Dynamic Models for Knowledge Tracing and Prediction of Future Performance," IARIA, ThinkMind, In Proceedings of the 7th International Conference on Data Analytics, pp. 121-129, Nov. 2018.

[2] C. Romero and S. Ventura, "Educational Data Mining: A Review of the State of the Art," IEEE Transactions on Systems, Man, and Cybernetics Part C Applications Rev., vol. 40, no. 6, pp. 601–618, Nov. 2010.

[3] A. Essa, "A possible future for next generation adaptive learning systems," Smart Learning Environments, vol. 3, no. 1, p. 16, Dec. 2016.

[4] K. W. Fischer, U. Goswami, and J. Geake, "The Future of Educational Neuroscience," Mind, Brain, Education, vol. 4, no. 2, pp. 68–80, Jun. 2010.

[5] Z.-T. Zhu, M.-H. Yu, and P. Riezebos, "A research framework of smart education," Smart Learning Environments, vol. 3, no. 1, p. 4, Dec. 2016.

[6] S. Kontogiannis et al., "Services and high level architecture of a smart interconnected classroom," in IEEE SEEDA-CECNSM, Sep. 2018, unpublished.

[7] Z. Papamitsiou and A. Economides, "Learning Analytics and Educational Data Mining in Practice: A Systematic Literature Review of Empirical Evidence," Journal of Educational Technology & Society, International Forum of Educational Technology & Society, vol. 17, pp. 49–64, 2014.

[8] B. K. Daniel, "Big Data and data science: A critical review of issues for educational research," Review, Wiley, Br. J. Educ. Technol., Nov. 2017.

[9] K. Nadu and L. Muthu, "Application of Big Data in Education Data Mining and Learning Analytics -A Literature Review ICTACT J. Soft Computing, vol. 5, no. 4, pp. 1035–1049, Jul. 2015.

[10] C. Romero and S. Ventura, "Educational data science in massive open online courses," Wiley Interdisciplinary Review Data Mining Knowledge Discovery, vol. 7, no. 1, pp. 1-12, Jan. 2017.

[11] Z. A. Pardos, "Big data in education and the models that love them," Current Opinion in Behavioral Science, vol. 18, pp. 107–113, Dec. 2017.

[12] I. Taleb, R. Dssouli, and M. A. Serhani, "Big Data Pre-processing: A Quality Framework," in IEEE International Congress on Big Data, pp. 191–198, 2015.

[13] P. Prinsloo, E. Archer, G. Barnes, Y. Chetty, and D. Van Zyl, "Bigger data as better data in open distance learning" Review, Int. Rev. Res. Open Distributed Learning, vol. 16, no. 1, pp. 284-306, Feb. 2015.

[14] D. Gibson, "Big Data in Higher Education: Research Methods and Analytics Supporting the Learning Journey," Technology Knowledge Learning, vol. 22, no. 3, pp. 237–241, Oct. 2017.

[15] P. Domingos, "A few useful things to know about machine learning," Communication. ACM, vol. 55, no. 10, p. 78, Oct. 2012.

[16] K. Colchester, H. Hagras, D. Alghazzawi, and G. Aldabbagh, "A Survey of Artificial Intelligence Techniques Employed for Adaptive Educational Systems within E-Learning Platforms," J. Artificial Intelligence, Soft Computing, Res., vol. 7, no. 1, pp. 47–64, Jan. 2017.

[17] A. Peña-Ayala, "Educational data mining: A survey and a data mining-based analysis of recent works," Expert Systems Applications, vol. 41, no. 4, pp. 1432–1462, Mar. 2014.

[18] A. T. Corbett and J. R. Anderson, "Knowledge tracing: Modeling the acquisition of procedural knowledge," User Model User-Adapted Interactions, vol. 4, no. 4, pp. 253–278, 1995.

[19] Y. Bengio, A. Courville, and P. Vincent, "Representation Learning: A Review and New Perspectives," IEEE Transactions Pattern Analytics Machine Intelligence, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[20] C. M. Bishop, Pattern recognition and machine learning, Editors: M. Jordan J. Kleinberg B. Scholkopf, Springer, 2006.

[21] T. Kaser, S. Klingler, A. G. Schwing, and M. Gross, "Dynamic Bayesian Networks for Student Modeling," IEEE Transactions Learning Technologies, vol. 10, no. 4, pp. 450–462, Oct. 2017.

[22] M. Khajah, R. V. Lindsey, and M. C. Mozer, "How deep is knowledge tracing?," arXiv preprint arXiv:1604.02416, Mar. 2016.

[23] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon, "Individualized Bayesian Knowledge Tracing Models," in International Conference on Artificial Intelligence in Education, pp. 171–180, 2013.

[24] Z. A. Pardos and N. T. Heffernan, "Modeling Individualization in a Bayesian Networks Implementation of Knowledge Tracing," Springer, Berlin, Heidelberg, pp. 255–266, 2010

[25] C. Piech et al., "Deep Knowledge Tracing," in Advances in Neural Information Processing Systems, NIPS, pp. 505–513, 2015.

[26] R. Pelanek, "Metrics for Evaluation of Student Models.," J. Educational Data Mining, vol. 7, no. 2, pp. 1–19, 2015.

[27] J. P. González-Brenes and Y. Huang, "Your model is predictive-but is it useful? Theoretical and Empirical Considerations of a New Paradigm for Adaptive Tutoring Evaluation," in Proceedings of the 8th International Conference on Educational Data Mining, pp. 187-194, 2015.

[28] J. E. Beck and K. Chang, "Identifiability: A Fundamental Problem of Student Modeling," In Proceedings of the 11th International Conference on User Modeling pp. 137–146, 2007.

[29] R. S. J. d. Baker, A. T. Corbett, and V. Aleven, "More Accurate Student Modeling through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing," in International Conference on Intelligent Tutoring Systems, pp. 406–415, 2008.

[30] Z. A. Pardos, Z. A. Pardos, and N. T. Heffernan, "Navigating the parameter space of Bayesian Knowledge Tracing models: Visualizations of the convergence of the Expectation Maximization algorithm.", In Proceedings of the 3rd International Conference on Educational Data Mining, pp. 161–170, 2010

[31] C.-K. Yeung and D.-Y. Yeung, "Addressing Two Problems in Deep Knowledge Tracing via Prediction-Consistent Regularization," In Proceedings of the 5th ACM Conference on Learning @ Scale, vol. 5, pp. 1-10, Jun. 2018.

[32] J. Gonzalez-Brenes, Y. Huang, and P. Brusilovsky, "General Features in Knowledge Tracing to Model Multiple Subskills, Temporal Item Response Theory, and Expert Knowledge," in Proceedings of the 9th International Conference on Educational Data Mining, pp. 84–91, 2014.

[33] R. Pelánek, "Applications of the Elo rating system in adaptive educational systems," Computers & Education, vol. 98, pp. 169–179, Jul. 2016.

[34] R. Pelánek, "Bayesian knowledge tracing, logistic models, and beyond: an overview of learner modeling techniques," User Model. User-adapt. Interact., vol. 27, no. 3–5, pp. 313–350, Dec. 2017.

[35] Y. Gong, J. E. Beck, and N. T. Heffernan, "Comparing Knowledge Tracing and Performance Factor Analysis by Using Multiple Model Fitting Procedures," Springer, Berlin, Heidelberg, pp. 35–44, 2010.

[36] C. Ekanadham and Y. Karklin, "T-SKIRT: Online Estimation of Student Proficiency in an Adaptive Learning System," arXiv preprint arXiv:1702.04282, 2017

[37] K. H. Wilson, Y. Karklin, B. Han, and C. Ekanadham, "Back to the basics: Bayesian extensions of IRT outperform neural networks for proficiency estimation Acknowledgements," arXiv preprint arXiv:1604.02336, 2016.

[38] E. A. Linnenbrink, P. R. Pintrich, and P. R. Pintrich, "Role of Affect in Cognitive Processing in Academic Contexts," pp. 71–102, Jul. 2004.

[39] A. J. Elliot and C. S. Dweck, Handbook of competence and motivation. Guilford Press, 2007.

[40] B. Fogg and BJ, "A behavior model for persuasive design," in Proceedings of the 4th International Conference on Persuasive Technology - Persuasive', p. 1., Sep. 2009

[41] G. L. Cohen and J. Garcia, "Identity, Belonging, and Achievement: A Model, Interventions, Implications," Current Directions in Psychological Science, Sage Publications, Inc. Association for Psychological Science, vol. 17, pp. 365–369, 2008.

[42] I. Roll, R. S. Baker, V. Aleven, B. M. Mclaren, and K. R. Koedinger, "Modeling Students' Metacognitive Errors in Two Intelligent Tutoring Systems 1 Metacognition in Intelligent Tutoring Systems." In: Proceedings of User Modeling, pp.

379–388, 2005

[43] S. Spaulding and C. Breazeal, “Affect and Inference in Bayesian Knowledge Tracing with a Robot Tutor.” Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction, pp. 219-220, USA 2015

[44] M. Khajah, R. M. Wing, R. V Lindsey, and M. C. Mozer, “Incorporating Latent Factors into Knowledge Tracing to Predict Individual Differences in Learning,” Proceedings of the 7th International Conference on Educational Data Mining, Educational Data Mining Society Press, pp. 99–106, 2014.

[45] J. I. E. Lee, “The Impact on Individualizing Student Models on Necessary Practice Opportunities.,” Int. Educ. Data Min. Soc., In Proceedings of the 5th International Conference on Educational Data Mining, pp. 118–125, Jun. 2012

[46] Y. Bengio and Y. LeCun, “Scaling learning algorithms towards AI”, Large-Scale Kernel Machines, MIT Press, 2007

[47] J. Zhang, X. Shi, I. King, and D.-Y. Yeung, “Dynamic Key Value Memory Networks for Knowledge Tracing,” In WWW, vol. 2, pp. 765–774, 2017.

[48] Marcus, G. Deep learning: A critical appraisal. arXiv preprint arXiv:1801.00631, 2018.

[49] Pelánek, R., & Řihák, J.: Experimental analysis of mastery learning criteria. In: UMAP, ACM, pp. 156-163, 2017.

[50] Brusilovsky, P., & Millán, E.: User models for adaptive hypermedia and adaptive educational systems. In: The Adaptive Web: Methods and Strategies of Web Personalization, , Springer-Verlag, Vol. 4321, pp. 3-53, 2007.

[51] Desmarais, M., & Baker, R. S., “A review of recent advances in learner and skill modeling in intelligent learning environments”, User Modeling and User-Adapted Interaction, vol. 22, no. 1, pp. 9-38, Apr. 2012

[52] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel, “Gated graph sequence neural networks,” In International Conference on Learning Representations (ICLR), vol. 1, pp. 1-20, 2016.

[53] Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. In Conference Neural Information Processing Systems (NIPS), pp. 4768–4777, 2017.

[54] Savi, A. O., Ruijs, N. M., Maris, G. K. J., and van der Maas, H. L. J. Delaying access to a problem-skipping option increases effortful practice: Application of an a/b test in large-scale online learning. Computers & Education, vol. 119, pp. 84– 94, 2018.

[55] Cornelisz, I. and Klaveren, C.: Student engagement with computerized practicing: Ability, task value, and difficulty perceptions. Journal of Computer Assisted Learning, vol. 34, pp. 828-842, 2018.

[56] Davis, D., Jivet, I., Kizilcec, R.F., Chen, G., Hauff, and C., Houben, G.J., “Follow the successful crowd: raising MOOC completion rates through social comparison at scale”, ACM, In: Proc. of LAK’17, pp. 454–463, 2017.