Machine Learning Techniques for Knowledge Tracing: A Systematic Literature Review

Sergio Iván Ramírez Luelmo[a], Nour El Mawas[b] and Jean Heutte[c]
CIREL, Centre Interuniversitaire de Recherche en Éducation de Lille, Université de Lille, Campus Cité Scientifique, Bâtiments B5 – B6, Villeneuve d'Ascq, France

Keywords: Machine Learning, Knowledge Tracing, Learner Model, Literature Review, Technology Enhanced Learning.

Abstract: Machine Learning (ML) techniques are being intensively applied in educational settings. They are employed to predict competences and skills, grade exams, recognize behavioural academic patterns, evaluate open answers, suggest appropriate educational resources, and group or associate students with similar learning characteristics or academic interests. Knowledge Tracing (KT) allows modelling the learner's mastery of skill and to meaningfully predict student's performance, as it tracks within the Learner Model (LM) the knowledge state of students based on observed outcomes from their previous educational practices, such as answers, grades and/or behaviours. In this study, we survey commonly used ML techniques for KT figuring in 51 papers on the topic, out of an original search pool of 628 articles from 5 renowned academic sources, encompassing the latest research, based on the PRISMA method. We identify and review relevant aspects of ML for KT in LM that help paint a more accurate panorama on the topic and hence, contribute to alleviate the difficulty of choosing an appropriate ML technique for KT in LM. This work is dedicated to MOOC designers/providers, pedagogical engineers and researchers who need an overview of existing ML techniques for KT in LM.

1 INTRODUCTION

Evidence from several studies has long linked having a Learner Model (LM) can make a system more effective in helping students learn, and adaptive to learner's differences (Corbett et al., 1995).

LMs represent the system's beliefs about the learner's specific characteristics, relevant to the educational practice (Giannandrea & Sansoni, 2013), encoded using a specific set of dimensions (Nakić et al., 2015). Ultimately, a perfect LM would include all features of the user's behaviour and knowledge that effect their learning and performance (Wenger, 2014). Modelling the learner has the ultimate goal of allowing the adaptation and personalization of environments and learning activities (El Mawas et al., 2019) while considering the unique and heterogeneous needs of learners. We acknowledge the difference between Learner Profile (LP) and LM in that the former can be considered an instantiation of the latter in a given moment of time (Martins et al., 2008).

Knowledge Tracing (KT) models students' knowledge as they correctly or incorrectly answer exercises (Swamy et al., 2018), or more generally, based on observed outcomes on their previous practices (Corbett & Anderson, 1994). KT is one out of three approaches for student performance prediction (Yudelson et al., 2013). In an Adaptive Educational System (AES), predicting students' performance warrants for KT. This allows for learning programs recommendation and/or level-appropriate, educational resources personalization, and immediate feedback. KT facilitates personalized guidance for students, focusing on strengthening their skills on unknown or less familiar concepts, hence assisting teachers in the teaching process (Juntao Zhang et al., 2020).

Machine Learning (ML) is a branch (or subset) of Artificial Intelligence (AI) focused on building

[a] https://orcid.org/0000-0002-7885-0123
[b] https://orcid.org/0000-0002-0214-9840
[c] https://orcid.org/0000-0002-2646-3658

Ramírez Luelmo, S., El Mawas, N. and Heutte, J.
Machine Learning Techniques for Knowledge Tracing: A Systematic Literature Review.
DOI: 10.5220/0010515500600070
In *Proceedings of the 13th International Conference on Computer Supported Education (CSEDU 2021) - Volume 1*, pages 60-70
ISBN: 978-989-758-502-9

applications that learn from data and improve their accuracy over time without being programmed to do so (IBM, 2020). To achieve this, ML algorithms build a model based on sample data (a.k.a. input data) known as 'training data'. Once trained, this model can then be reused with other data to make predictions or decisions.

ML techniques are currently applied to KT in vast and different forms. The goal of this literature review is to survey all available works in the field of "Machine Learning for Knowledge Tracing used in a Learner Model setup" in the last five years to identify the most employed ML techniques and their relevant aspects. This is, in general terms, what common ML techniques and their relevant aspects, designed to trace a learner's mastery of knowledge, also account for the creation, storage, and update of a LM. Moreover, we aim to identify relevant ML aspects to consider insuring KT in a LM. The motivation behind this work is to present a comprehensive panorama on the topic of ML for KT in LM to our target public. To our knowledge, currently there is no research work that addresses the literature review of ML techniques for KT accounting for the LM.

Thus, we decided to focus our literature review on the terms "machine learning", "knowledge tracing" and "learner model", a.k.a. "student model" (SM). Using the PRISMA method (Moher et al., 2009), we performed this research in the IEEE, Science Direct, Scopus, Springer, and Web of Science databases comprising the 2015-2020 period. The thought behind these choices is to obtain the most recent and high-quality corpus on the topic.

This work differs from other literature reviews (Das & Behera, 2017; Olsson, 2009; Shin & Shim, 2020) on two accounts. First, we focus exclusively on ML techniques for KT accounting for the LM. That is, we do not cover pure Data Mining (DM) techniques, nor AI intended for purposes other than KT, such as Natural Language Processing (NLP), gamification, computer vision, learning styles prediction, nor any processes that make pure use of LP data (instead of LM data), nor other User Model data, such as sociodemographic, biometrical, behavioural, or geographical data[1]. Second, we do not review nor compare the mathematical inner workings of ML techniques: we feel (a) the research field and the literature corpus found cover it extensively, and (b) our target public might be unable to exploit appropriately such complex form results. Instead, we shift the focus to a pragmatic report on ML for KT in a LM application and purpose(s).

The remainder of this article is structured as follows. Section 2 of this paper oversees the theoretical framework concerning this paper, namely the definition of ML and its categorization. Section 3 details the methodology steps taken. Section 4 presents the findings of this research, Section 5 discusses the results and, finally Section 6 concludes this paper and presents its perspectives.

2 THEORETICAL BACKGROUND

In this section we present the theoretical background put in motion behind this research, namely the definition of ML and how it is categorized.

2.1 Machine Learning

ML is a branch (or subset) of AI focused on building applications that learn from data and improve their accuracy over time without being programmed to do so (IBM, 2020). Additional research (Chakrabarti et al., 2006; Schmidhuber, 2015) to this definition allows us to present Figure 1 to illustrate and discern the situation of ML against other common terms used in the field.

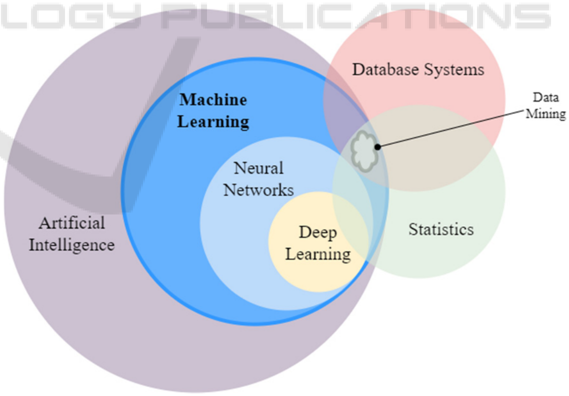

Figure 1: Situational context of ML.

2.2 ML Methods / Styles / Scenarios

Although some authors (Das & Behera, 2017; Mohri et al., 2018) admit several more ML methods (or styles or paradigms or scenarios), we retain the following categorization: Supervised ML, Semi

[1] Please note that we did include such works, if they also employed ML for KT (the core of this paper).

Supervised ML, Unsupervised ML, Reinforcement Learning, and Deep Learning (IBM, 2020). The first three differentiate each other on the labelling of the input training data while creating the model. The two latter constitute special cases altogether (Brownlee, 2019; IBM, 2020; Mohri et al., 2018).

First, in **Supervised Learning (SL)** labels are provided (metadata containing information that the model can use to determine how to classify it). However, properly labelled data is expensive[2] to prepare, and there is a risk of creating a model so tied to its training data that it cannot handle variations in new input data accurately ("overfitting") (Brownlee, 2019).

Second, **Unsupervised Learning (UL)** must use algorithms to extract meaningful features to label, sort and classify its training data (which is unlabelled) without human intervention. As such, it is usually used to identify patterns and relationships (that a human can miss) than to automate decisions and predictions. Because of this, UL requires huge amounts of training data to create a useful model (Brownlee, 2019).

Third, **Semi Supervised Learning (SSL)** is at the middle point of the two previous methods: it uses a smaller labelled dataset to extract features and guide the classification of a larger, unlabelled dataset. It is usually used when not enough labelled data is made available (or it is too expensive) to train a preferred, Supervised Model (van Engelen & Hoos, 2020).

Fourth, **Reinforcement Learning (RL)** is a behavioural machine learning model akin to SL, but the algorithm is not trained using sample data but by using trial and error. A sequence of successful outcomes will be reinforced to develop the best recommendation or policy for a given problem. RL models can also be deep learning models (IBM, 2020).

Lastly, **Deep Learning (DL)** is a subset of ML (all DL is ML, but not all ML is DL). DL algorithms define an artificial neural network[3] that is designed to learn the way the human brain learns. DL models require a large amount of data to pass through multiple layers of calculations, applying weights and biases in each successive layer to continually adjust and improve the outcomes. DL models are typically unsupervised or semi-supervised (IBM, 2020). For clarity reasons, the figure illustrating this ML categorization is available in the Appendix.

In this subsection we covered the ML definition and a categorization of ML techniques. In the following subsection we deepen into the relevant aspects in ML for KT in LM.

2.3 ML for KT in LM

An overwhelming number of ML techniques have been designed and introduced over the years (Das & Behera, 2017). They usually rely on more common ML techniques, within optimized pipelines. As such, we identify the **ML techniques** (or algorithms) upon which any new research is based.

Additionally to performing KT in LM, researchers have acknowledged that ML techniques can reliably determine the initial parameters when instantiating a LM (Eagle et al., 2016; Millán et al., 2015). This led us to consider this **purpose** when reviewing ML techniques. Different ML techniques are applied at different stages of the ML pipeline, and not all stages are responsible for KT (other applications can be NLP, computer vision, automatic grading, demographic student clustering, mood detection, etc.) We differentiate purposes related to KT and/or learner modelling, specifically if the ML technique is used for (1) either grade, skills, or knowledge prediction (and hence later, clustering, personalizing, or suggesting resources), (2) either for LM creation (or instantiation), or (3) both.

Studies highlight the importance of justifying the **rationale** when choosing a ML technique (Chicco, 2017; Wen et al., 2012; Winkler-Schwartz et al., 2019). We note such rationale, when made explicit, and contrast it to other authors' rationale for commonalities, on the same technique. This allows us to present and weigh known, favourable, and unfavourable features specific to ML techniques applied to KT accounting for the LM.

Research studies stress the ultimate importance of the input data (**dataset**) and the effects of the chosen programming language **software** employed for ML (Chicco, 2017; Domingos, 2012). Indeed, ML techniques require input data for creating a model. The feature engineering of this input data (dataset) might be determinant for a ML project to succeed or fail (Chicco, 2017). We compile and verify the availability of all public datasets presented in the reviewed articles. Furthermore, the choice of the programming language for ML plays a role in collaboration, licensing, and decision-making

[2] Mostly in terms of computational resource allocation.

[3] A quite complete and updated chart of many neural networks was made available by (van Veen & Leijnen, 2019).

processes: it helps to determine the most appropriate choices for ML implementation (purchasing licences, upgrading hardware, hiring a specialist, or considering self-training). Hence, we highlight the family of ML programming languages used by researchers on their proposals.

Thus, based on this state-of-the-art, we identify relevant aspects to consider in ML for KT in LM: the **ML technique** employed, its **purpose**, the contextual, known **rationale** for choosing it, the **programming language software** used for ML, and the **dataset(s)** employed for KT. We consider that these aspects are relevant for our target public when choosing a ML technique for KT in LM.

3 REVIEW METHODOLOGY

This review of literature follows the PRISMA (Moher et al., 2009) methodology, comprising: Rationale, Objectives & Research questions, Eligibility criteria, Information sources & Search strategy, Screening process & Study selection, and Data collection & Features.

3.1 Rationale, Objectives & Research Questions

The goal of this literature review is to present a comprehensive panorama on the topic of ML for KT in LM. This is, in general terms, what ML techniques designed to trace a learner's mastery of skill also account for the creation, storage, and update of the LM.

This article aims thus to answer the following two research questions (RQ):

- RQ1: What are the most employed ML techniques for KT in LM?
- RQ2: How do the most employed ML techniques fulfil the considered relevant aspects (identified in section 2.1.2) to insure KT in LM?

3.2 Eligibility Criteria, Information Sources & Search Strategy

In this section we describe the inclusion and exclusion criteria used to constitute the corpus of publications for our analysis. We also detail and justify our choice of in-scope publications, the search terms, and the identified databases.

In this research, we focus on recent ML techniques (and/or algorithms) that explicitly "learn" (with minimal or no human intervention) from its data input to perform KT, while accounting for the LM. Thus, we do not cover all predictive statistical methods (as they are not all ML), nor pure DM techniques, nor AI intended for purposes other than KT (such as NLP, gamification, computer vision, learning styles prediction, etc.), nor any processes that make pure use of LP data (instead of LM data), nor other User Model data, such as sociodemographic, biometrical, behavioural, or geographical data.

On one hand, our Inclusion criterion are: Works that present a ML technique for KT while accounting for the LM, in the terms presented in the previous paragraph. On the other hand, our chosen Exclusion criterion consist of: Works written not in English, under embargo, not published or in the works. We choose to keep subsequent works on the same subject from the same research team because they represent a consolidation of the techniques employed.

We performed this research at the end of October 2020 in the following scientific databases: IEEE, Science Direct, Scopus, Springer, and Web of Science, comprising 2015-2020. The thought behind these two choices is to have the most recent and quality-proven scientific works on the subject. Our general search terms were `(("learner model" OR "student model" OR "knowledge tracing") AND "machine learning")`, declined for the specificities of each scientific database (search engines parse and return verbal, noun, plural, and continuous forms of search terms). We used their 'Advanced search' function, or we queried them directly, if they allowed it. Some direct queries did not allow for year filtering, so we applied it manually on the results page. For accessibility reasons, we explicitly selected "Subscribed content" results for the scientific databases supporting it.

3.3 Screening Process & Study Selection

The paper selection process happened as follows: First, we gathered all the results in two known Citation Manager programs to benefit from the automatic metadata extraction, the report creation, and duplicate merging. We also used a spreadsheet to record, based on section 2.1.2, the following information: `doi`, `title`, `year`, `purpose`, `ml_method`, `method_rationale`, `software`, `data_source`, and `observations`. Second, we screened the abstracts of all 708 results: three categories appeared: obvious Out-of-scope results, clear Eligible results, and Pending (verification needed) results. Third, using the institutional authentication, we downloaded all the papers in the Eligible and Pending categories. Fourth, we read the full papers in the Eligible and Pending categories and re-classified them as Eligible or Out-of-scope, as needed.

We used the papers' titles and keywords metadata fields to discern if they satisfy the inclusion criteria, when the abstract was ambiguous. We highlighted the text in the abstract that made it Eligible. We registered the reason(s) for rejection NO_ML: no ML is involved but instead other prediction or classifying mechanism, NO_KT: ML is not used for KT, and NO_LM: no LM/SM accountability.

Figure 2 presents a PRISMA flow diagram of the process presented in this section. From a total of 708 results from the five academic search engines, 628 articles were collected (i.e., duplicates removed) and their abstracts read: 134 publications were thus categorized either Eligible or Pending; 494 publications were excluded. After full text read, 83 publications were again removed as they were out of scope, leading to a core of 51 papers.

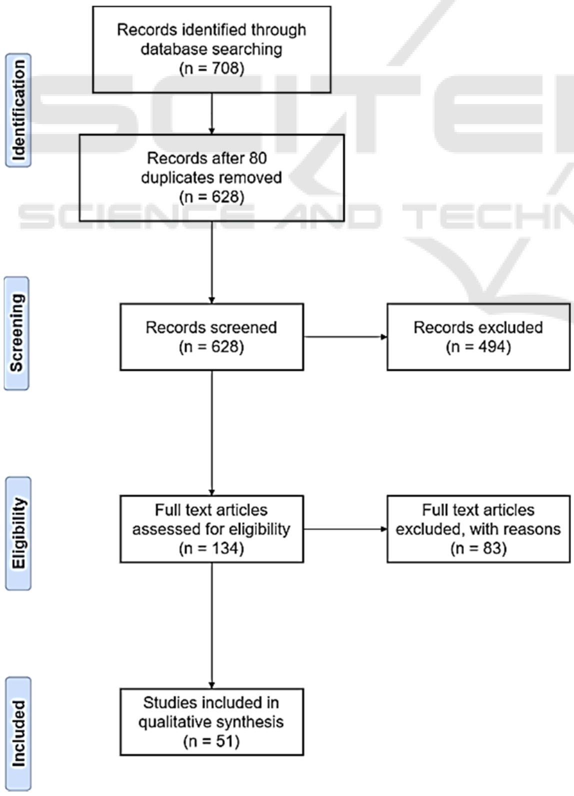

Figure 2: PRISMA Flow diagram of the publication screening process.

3.4 Data Collection & Features

In this section we review the relevant features of interest described in subsection 2.1.2 found in the reviewed literature.

During the full text read, we extracted the following information from the selected papers: (1) ML technique employed; (2) purpose of the ML technique; (3) rationale for employing that specific ML technique; (4) software employed for ML; and (5) dataset employed for KT, if any.

We note here that rarely a single, known technique ML is employed, but it is rather implemented in a pipeline, connected with another secondary ML (probabilistical, or DM) techniques. In such cases, we focused on the technique(s) employed for KT and on the reasons given for choosing it over other techniques acknowledged by the authors.

We surveyed the software used to perform the calculation of ML and we grouped them by programming language, which is a rather meaningful description, compared to combinations of libraries and platforms. We think this result shows a clear tendency on the necessary requirements to implement and perform ML for KT in LM.

We surveyed all datasets presented in the 51 reviewed papers and checked for their existence. We understand that our target public may not have data made available to perform ML for KT accounting for the LM and we feel that this resource may be invaluable when evaluating their results.

In this section we presented our literature review methodology, the considered features, and the train of thought behind them. The following section details our literature review results.

4 RESULTS

We aggregated the data collected (described in the previous section) to make it easier to digest.

First, we quickly present the seven most employed[4] ML techniques for KT in LM found in the reviewed publications. These comprise based-upon techniques for the paper proposal, techniques used as baselines, and techniques used for comparison.

Bayesian Knowledge Tracing (BKT) (Corbett & Anderson, 1994) is the most classical method used to trace students' knowledge states.

Deep Knowledge Tracing (DKT) was proposed by (Piech et al., 2015) to trace students' knowledge using Recurrent Neural Networks (RNNs), achieving

[4] With more than five applications in the last five years.

great improvement on the prediction accuracy of students' performance.

Long Short-Term (LSTM) is a special type of RNN, effective in capturing underlying temporal structures in time series data and long-term dependencies more effectively than conventional RNN (Mao et al., 2018).

Bayesian Networks (BN) are graphical models designed to explicitly represent conditional independence among random variables of interest and exploit this information to reduce the complexity of probabilistic inference (Pearl, 1988). They are a formalism for reasoning under uncertainty that has been widely adopted in AI (Conati, 2010).

Support Vector Machines (SVM) are one of the most robust prediction methods, based on statistical learning frameworks (Vapnik, 1998). The primary aim of this technique is to map nonlinear separable samples onto another higher dimensional space by using different types of kernel functions (Hämäläinen & Vinni, 2010). They distinctively afford balanced predictive performance, even in studies where sample sizes may be limited.

Dynamic Key Value Memory Network (DKVMN) is a memory augment neural network-based model, which uses the relationship between the underlying knowledge points to directly output the student's mastery of each knowledge point (Jiani Zhang et al., 2017).

Performance Factor Analysis (PFA) is one specific model from a larger class of models based on a logistic function (Pavlik et al., 2009). In PFA, the probability of learning is computed using the previous number of failures and successes.

This list answers then RQ1. "What are the most employed ML techniques for KT in LM?". Figure 3 shows a yearly heatmap of the most used techniques: the number indicates the total number of applications[5] in all 51 combined-and-reviewed papers, per year. DKT was applied eight times in 2019 (emerging of two consecutive zero years) while BKT was mostly applied in 2016 and 2017, five and six times respectively, decreasing since. LSTM peaked in 2017, with 7 applications, and has decreased since. BN remains with a steady application since 2017. For clarity reasons, the 29 ML techniques found in the 51 papers issued from this study are available in the Appendix.

ML technique	2015	2016	2017	2018	2019	2020	Cumulative
BKT	1	5	6	0	3	3	18
DKT	0	2	0	0	8	3	13
LSTM	0	0	7	1	1	3	12
BN	1	0	2	3	3	2	11
SVM	2	1	1	0	2	1	7
DKVMN	0	0	0	0	4	3	7
PFA	0	1	2	0	0	3	6

Figure 3: Yearly heatmap of the most employed ML techniques.

[5]Programming and teaching the ML model with input data.

Second, we noted the rationale (if any) given by authors when choosing a ML technique. We do not account for the rationale of the paper's unique ML proposal if its improvements are related to parameter fine-tuning, or if the justification is *à posteriori*. Instead, we account rationale for the general application of the original, unmodified technique. Also, very few publications detail the shortcomings of their choice. We grouped these rationales in the following categories:

R1-Uses Less Data and/or Metadata. These techniques handle sparse data situations better compared to others, according to the authors, e.g. DKT (Jiani Zhang & King, 2016).

R2-Extended Tracing. These techniques provide additional attributes and/or dimensional tracing with ease when compared to other techniques, according to authors, e.g. LSTM (Sha & Hong, 2017).

R3-Popularity. These techniques were chosen because of their popularity, e.g. BN (Millán et al., 2015).

R4-Persistent Data Storage. These techniques explicitly save their intermediate states to long-term memory, e.g. DKVMN (Trifa et al., 2019).

R5-Input Data Limitations. These techniques have been acknowledged to lack when the number of peers is "too high", e.g. BN (Sciarrone & Temperini, 2020).

R6-Modelling Shortcomings. Techniques in this category face difficulties when modelling either forgetting, guessing, multiple-skill questions, time-related issues, or have other modelling shortcomings, e.g. BKT (Crowston et al., 2020).

A heatmap illustrating the number of publications mentioning each of these rationales, for each of the most common ML techniques, is shown in Figure 4.

	ML technique	R1	R2	R3	R4	R5	R6
RL	BKT	0	0	1	0	0	4
DL	DKT	5	0	1	0	0	0
	LSTM	0	4	0	0	0	1
	DKVMN	0	0	0	1	0	1
SL	BN	0	1	3	0	2	0
	SVM	0	0	0	0	0	0
UL	PFA	0	1	0	0	0	1

Figure 4: Heatmap of most employed ML techniques, categorized by Method (SL, UL, SSL, RL, DL) and number of publications sharing a Rationale (R1-R6).

This heatmap includes the ML categorization presented in 2.1.1 (SL, UL, SSL, RL, DL). BKT faced mostly R6 rationales, englobing the whole of RL as well. DKT and BKT were mostly commented on R1 and R2, respectively. This leads to the DL categorization (DKT + LSTM + DKVMN) to be extensively justified in the literature, while UL (PFA) is sparsely commented, and SVM not at all, despite its non-negligeable number of applications (7). BN had the highest R3 count of all and carries all the justifications related to SL.

Third, we looked over the intended purpose of the ML implementation, besides KT. Out of the 51 publications reviewed, seven (~15%) employ ML for initializing the LM (e.g., for another course, academic year, or for determining the ML parameters in a pipeline) by accounting previous system interactions, grades, pre-tests or other data. 44 publications, the vast majority (~85%) perform some form of prediction. Finally, only one proposal (~2%) incorporates both a prediction and/or recommendation mechanism as well. A pie chart of ML techniques purpose distribution is presented in Figure 5.

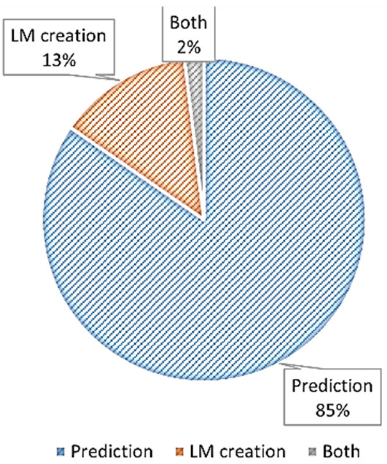

Figure 5: Pie chart distribution of ML purpose.

Fourth, we surveyed the software used to perform the ML calculations. Note that many publications (~50%) do not mention their software of choice. Python (comprising Keras, TensorFlow, PyTorch and scikit-learn) is the largest group, with 13 papers. Ad-hoc solutions follow with five papers, and finally C, Java (-based), Matlab and R, with 2 publications each. Outliers were SPSS and Stan, with 1 paper each. A pie chart illustrating the distribution of programming languages is shown in Figure 6.

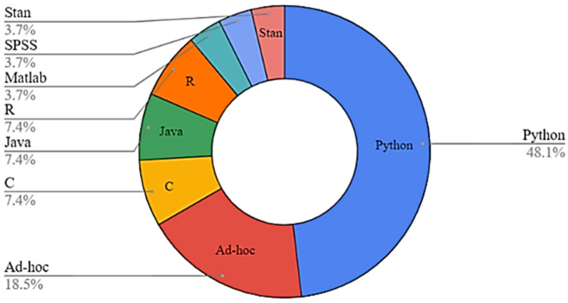

Figure 6: Pie chart distribution of ML programming language.

Fifth, we highlighted (and checked for existence) the public datasets employed, shown in Table 1. All the datasets we found in the literature were online and accessible when reviewed. We made the version

Table 1: Public datasets found.

Name	URL
ASSISTments2009	https://sites.google.com/site/assistmentsdata/home/assistment-2009-2010-data/skill-builder-data-2009-2010
ASSISTments2013	https://sites.google.com/site/assistmentsdata/home/2012-13-school-data-with-affect
ASSISTments2015	https://sites.google.com/site/assistmentsdata/home/2015-assistments-skill-builder-data
KDD Cup	https://pslcdatashop.web.cmu.edu/KDDCup/downloads.jsp
DataShop	https://pslcdatashop.web.cmu.edu/
DataShop: OLI Engineering Statics - 1.14 (Statics2011)	https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=507
The Stanford MOOCPosts Data Set	https://datastage.stanford.edu/StanfordMoocPosts/
Hour of Code	https://code.org/research
DeepKnowledgeTracing dataset	https://github.com/chrispiech/DeepKnowledgeTracing
DeepKnowledgeTracing dataset - Synthetic-5	https://github.com/chrispiech/DeepKnowledgeTracing/tree/master/data/synthetic
MOOC [Big Data and Education on the EdX platform]	https://github.com/davidjlemay/EdX-Video-Feature-Extraction

distinction (yearly or topic) of datasets from the same source (DeepKnowledgeTracing and ASSISTments, respectively) because they differ on either number of features, dimensioning, or creation method.

Thus, the elements presented here-in, namely the ML techniques, their chosen rationale, their KT in LM purpose, the most usual programming language software employed, and the subsequent required datasets, found in the 51 reviewed publications constitute the answer to "RQ2: How do the most employed ML techniques fulfil the considered relevant aspects (identified in section 2.1.2) to insure KT in LM?"

5 DISCUSSION

In this section we present our observations on the ML techniques addressed in the precedent section, issued from the 51 reviewed publications. This discussion covers the five elements mentioned in subsection 2.1.2.

ML Technique: We begin by noting that, in the reviewed papers, rarely a clear, well-defined, single ML technique is employed: very often additions or variants are employed (which make the point of the paper). Research teams seem to focus their attention on fine-tuning parameters (to improve prediction) rather than on expanding the application of ML for KT to other educational data sources or contexts. Authors recognize that additional features (or dimensions) would encumber the learning phase for limited gains, compared to parameter fine-tuning. As such, many papers propose pipelines ('chains') of ML techniques to optimize the process without increasing the calculation load. Performance improvements aside, this brings up two inconveniences: the difficulty of identifying the ML technique suitable for KT, and the difficulty to evaluate and compare any two papers employing different pipelines, as the intermediary inputs and outputs of the chain elements are quite different between papers.

ML Purpose: We distinguish two families of stated purposes in the reviewed ML techniques for KT: prediction and LM creation. Prediction is often portrayed as a probability, which can be interpreted as a mastery (or degree) of a skill (0-100), a grade (0-10), or a likelihood (0-1) of getting the answer right (in binary answers). In LM creation, ML predicts parameters for initializing the LM. We noticed that clustering, personalization, and/or resource suggestion (or other ML techniques, such as NLP) were performed once the predicting phase had taken place.

ML Choice Rationales: We condense the rationales exposed by the authors when choosing a ML technique. We omit rationales based on novelty, status-quo, or generalities, e.g., "nobody had done it before", "the existing system already uses this mechanism", "because it helps predict students' performance". The choice of BKT's was mostly driven by popularity, although it had issues on learners' individuality, multi-dimensional skill support and modelling forgetting. BN also seemed to be a common, popular choice. Its main advantage was its ability to model uncertainty, although it seems to reach its limits if the number of students is kept relatively low. On the contrary, DKT may benefit from large datasets and has proven being able to model multi-dimensional skills, although lacking in consistent predicted knowledge state across time-steps. DKVMN (based on LSTM) can model long-term memory and mastery of knowledge at the same time, as well as finding correlations between exercises and concepts, although it does not account for forgetting mechanisms. LSTM appears to additionally handle tasks other than KT satisfactory. It also models forgetting mechanisms over long-term dependencies within temporal sequences. It is then well suited for time series data with unknown time lag between long-range events. PFA does not consider answers' order (which is pedagogically relevant), nor models guessing, nor multiple-skills questions. Finally, RNNs are well suited for sequential data with temporal relationships, although long-range dependencies are difficult to learn by the model, hence the resurgence of LSTM.

Software for ML: Python (frameworks and libraries merged) is the most common programming language employed for ML, more than doubling the number of papers employing Ad-hoc languages. We think that employing platform-specific programming languages for ML assures lack of code portability (licensing issues, steep learning curve, little replicability, code isolation, etc.) and thus, little to no adoption of these research proposals. However, specialized ML software, designed by experts on the field, tends to be performance optimized for diverse hardware and software, which an ad-hoc solution cannot compete with. We were taken aback by two facts: the sparse use of specialized mathematical software (Matlab, R, SPSS) in ML, and to learn that about 50% of all reviewed publication do not specify what software was employed for their ML calculations, leaving little room for independent replication, results verification, and additional development.

Datasets: We noticed that frameworks proposal papers aim to prove the performance of their approach using publicly available datasets. An overview of the found public datasets is in Table 1, in previous section. The chosen datasets are static, mostly contain grades (or other evaluating measurements), opposed to behavioural or external sensor data, and provide the non negligeable advantages of being explained in detail and having their data already labelled, often by experts. This contrasts with the "organic" data employed in publications where ML is addressed for an existing, live system, even if it is for testing purposes. Both variants could benefit from each other's approaches, but this would require diverse, detailed, copious high-quality data that many institutions simply cannot afford to generate nor stock, let alone analyse.

One of most recurring datasets is the ASSISTment (Razzaq et al., 2005) (employed in 11 publications), of which there are different versions. A noteworthy fact is that this dataset has been acknowledged to have two main kind of data errors: (1) duplicate rows (which are removed if acknowledged by the authors) and (2) "misrepresented" skill sequences. Drawbacks of this issue have been discussed: while this does not affect the final prediction, it nevertheless might conduce the learner to being presented with less questions on one of the merged skills (the less mastered) because the global (merged) mastery of skill is achieved mainly through the mastery of the most known skill (Pelánek, 2015; Schatten et al., 2015). This raises the importance of the data cleaning process (Chicco, 2017), which processing time is not negligeable and should be accounted at early data mining stages.

6 CONCLUSION AND PERSPECTIVES

This review of literature presents a current panorama of ML techniques for KT in LM for the last five years. To our knowledge, there is no research work that addresses the literature review of such topic between 2015 and 2020. This study intents to fill in that gap by reviewing the most recent and high-quality academic publications on ML for KT that account for the LM. Its primary goal is to survey currently used ML techniques for KT in LM (methods and algorithms), their intended purpose, and their required software resources. It helps to paint a picture of the current trends in the research field, and to prepare the target public of this paper to the task of selecting a ML technique based on an argued choice.

Out of an academic database search result pool of 628 publications, 51 papers were reviewed, their employed ML technique extracted, and their employment rationale highlighted. We found a large variety of ML techniques, the most common ones are BKT (18 applications), DKT (13 applications), LSTM (12 applications), BN (11 applications), SVM (7 applications), DKVMN (7 applications), and PFA (6 applications). We found authors rationale for favouring one over another is seldomly described in publications, or very lightly. Additionally, we highlighted that combinations of ML techniques in pipelines are a common practice, with the most recent research focusing on optimizing combinations or parameter tweaking, and not in new techniques. We also noticed a steady use of public datasets, containing usually grades or other evaluating metrics, but no other pedagogical relevant data. Moreover, we insist that additional pre-treatment and cleaning is often required in these datasets before their use. Finally, our results show that ML programming language of choice is Python (libraries & frameworks combined).

This review of literature is inscribed in the context of the "Optimal experience modelling" research project, conducted by the University of Lille. This research project (Ramírez Luelmo et al., 2020) models and traces the Flow psychological state, alongside KT, via behavioural data, using the generic Bayesian Student Model (gBSM), within an Open Learner Model.

The current challenge is to incorporate the ML relevant aspects highlighted in this study, and the behavioural and psychological aspects (log traces and Flow state determination) specifically linked to the project. Namely, a ML technique supporting the gBSM, capable to initialize the LM and perform KT, supported by the most common programming language for ML, based on a sound rationale. The originality of such research lies in the use of live, behavioural, Flow-labelled data issued from the French-spoken international MOOC "Project Management"[6].

ACKNOWLEDGEMENTS

This project was supported by the French government through the Programme Investissement d'Avenir (I-SITE ULNE / ANR-16-IDEX-0004 ULNE) managed by the Agence Nationale de la Recherche.

[6] https://moocgdp.gestiondeprojet.pm/

REFERENCES

Brownlee, J. (2019, August 11). A Tour of Machine Learning Algorithms. Machine Learning Mastery. https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/

Chakrabarti, S., Ester, M., Fayyad, U., Gehrke, J., Han, J., Morishita, S., Piatetsky-Shapiro, G., & Wang, W. (2006). Data Mining Curriculum: A Proposal (version 1.0). ACM SIGKDD. https://www.kdd.org/curriculum/index.html

Chicco, D. (2017). Ten quick tips for machine learning in computational biology. BioData Mining, 10(1), 35. https://doi.org/10.1186/s13040-017-0155-3

Conati, C. (2010). Bayesian Student Modeling. In R. Nkambou, J. Bourdeau, & R. Mizoguchi (Eds.), Advances in Intelligent Tutoring Systems (pp. 281–299). Springer. https://doi.org/10.1007/978-3-642-14363-2_14

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

Corbett, A., Anderson, J., & O'Brien, A. (1995). Chapter 2 -Student modeling in the ACT programming tutor. In P. Nichols, S. Chipman, & R. Brennan (Eds.), Cognitively diagnostic assessment. Lawrence Erlbaum Associates: Hillsdale, NJ.

Crowston, K., Osterlund, C., Lee, T. K., Jackson, C., Harandi, M., Allen, S., Bahaadini, S., Coughlin, S., Katsaggelos, A. K., Larson, S. L., Rohani, N., Smith, J. R., Trouille, L., & Zevin, M. (2020). Knowledge Tracing to Model Learning in Online Citizen Science Projects. Ieee Transactions on Learning Technologies, 13(1), 123–134. https://doi.org/10.1109/TLT.2019.2936480

Das, K., & Behera, R. N. (2017). A Survey on Machine Learning: Concept, Algorithms and Applications. International Journal of Innovative Research in Computer and Communication Engineering, 5(2). https://doi.org/10.15680/IJIRCCE.2017. 0502001

Domingos, P. (2012). A few useful things to know about machine learning. Communications of the ACM, 55(10), 78–87. https://doi.org/10.1145/2347736.2347755

Eagle, M., Corbett, A., Stamper, J., McLaren, B. M., Wagner, A., MacLaren, B., & Mitchell, A. (2016). Estimating Individual Differences for Student Modeling in Intelligent Tutors from Reading and Pretest Data. In A. Micarelli, J. Stamper, & K. Panourgia (Eds.), Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 9684, pp. 133–143). Springer International Publishing. https://doi.org/10.1007/978-3-319-39583-8_13

El Mawas, N., Gilliot, J.-M., Garlatti, S., Euler, R., & Pascual, S. (2019). As One Size Doesn't Fit All, Personalized Massive Open Online Courses Are Required. In B. M. McLaren, R. Reilly, S. Zvacek, & J. Uhomoibhi (Eds.), Computer Supported Education (Vol. 1022, pp. 470–488). Springer International Publishing. https://doi.org/10.1007/978-3-030-21151-6_22

Giannandrea, L., & Sansoni, M. (2013). A literature review on intelligent tutoring systems and on student profiling. Learning & Teaching with Media & Technology, 287, 287–294.

Hämäläinen, W., & Vinni, M. (2010). Chapter 5—Classifiers for educational data mining. In Handbook of Educational Data Mining (pp. 57–74). CRC Press; Taylor & Francis.

IBM. (2020, December 18). What is Machine Learning? IBM Cloud Learn Hub. https://www.ibm.com/cloud/learn/machine-learning

Mao, Y., Lin, C., & Chi, M. (2018). Deep Learning vs. Bayesian Knowledge Tracing: Student Models for Interventions. Journal of Educational Data Mining, 10(2).

Martins, A. C., Faria, L., De Carvalho, C. V., & Carrapatoso, E. (2008). User modeling in adaptive hypermedia educational systems. Journal of Educational Technology & Society, 11(1), 194–207.

Millán, E., Jiménez, G., Belmonte, M. V., & Pérez-De-La-Cruz, J. L. (2015). Learning Bayesian networks for student modeling. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 9112, 718–721. https://doi.org/10.1007/978-3-319-19773-9_100

Moher, D., Liberati, A., Tetzlaff, J., Altman, D. G., & The PRISMA Group. (2009). Preferred Reporting Items for Systematic Reviews and Meta-Analyses: The PRISMA Statement. PLOS Medicine, 6(7), e1000097. https://doi.org/10.1371/journal.pmed.1000097

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). Foundations of Machine Learning, second edition. MIT Press.

Nakić, J., Granić, A., & Glavinić, V. (2015). Anatomy of student models in adaptive learning systems: A systematic literature review of individual differences from 2001 to 2013. Journal of Educational Computing Research, 51(4), 459–489.

Olsson, F. (2009). A literature survey of active machine learning in the context of natural language processing. SICS Technical Report, 59.

Pavlik, P. I., Cen, H., & Koedinger, K. R. (2009). Performance Factors Analysis—A New Alternative to Knowledge Tracing. Online Submission, 8.

Pearl, J. (1988). Probabilistic reasoning in intelligent systems: Networks of plausible inference (2nd ed.). Morgan Kaufmann.

Pelánek, R. (2015). Metrics for Evaluation of Student Models. https://doi.org/10.5281/ZENODO.3554665

Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L. J., & Sohl-Dickstein, J. (2015). Deep Knowledge Tracing. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, & R. Garnett (Eds.), Advances in Neural Information Processing Systems (Vol. 28, pp. 505–513). Curran Associates, Inc. https://proceedings.neurips.cc/paper/2015/file/bac9162b47c56fc8a4d2a519803d51b3-Paper.pdf

Ramírez Luelmo, S. I., El Mawas, N., & Heutte, J. (2020). Towards Open Learner Models Including the Flow State. Adjunct Publication of the 28th ACM Conference on User Modeling, Adaptation and Personalization, 305–310. https://doi.org/10.1145/3386392.3399295
Razzaq, L., Feng, M., Nuzzo-Jones, G., Heffernan, N. T., Koedinger, K., Junker, B., Ritter, S., Knight, A., Mercado, E., Turner, T. E., Upalekar, R., Walonoski, J. A., Macasek, M. A., Aniszczyk, C., Choksey, S., Livak, T., & Rasmussen, K. (2005). Blending Assessment and Instructional Assisting. Proceedings of the 2005 Conference on Artificial Intelligence in Education: Supporting Learning through Intelligent and Socially Informed Technology, 555–562.
Schatten, C., Janning, R., & Schmidt-Thieme, L. (2015). Vygotsky Based Sequencing Without Domain Information: A Matrix Factorization Approach. Communications in Computer and Information Science, 510, 35–51. https://doi.org/10.1007/978-3-319-25768-6_3
Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural Networks, 61, 85–117. https://doi.org/10.1016/j.neunet.2014.09.003
Sciarrone, F., & Temperini, M. (2020). K-OpenAnswer: A simulation environment to analyze the dynamics of massive open online courses in smart cities. Soft Computing, 24(15), 11121–11134. https://doi.org/10.1007/s00500-020-04696-z
Sha, L., & Hong, P. (2017). Neural knowledge tracing. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 10512 LNAI, 108–117. https://doi.org/10.1007/978-3-319-67615-9_10
Shin, D., & Shim, J. (2020). A Systematic Review on Data Mining for Mathematics and Science Education. International Journal of Science and Mathematics Education. https://doi.org/10.1007/s10763-020-10085-7
Swamy, V., Guo, A., Lau, S., Wu, W., Wu, M., Pardos, Z., & Culler, D. (2018). Deep knowledge tracing for free-form student code progression. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 10948 LNAI, 348–352. https://doi.org/10.1007/978-3-319-93846-2_65
Trifa, A., Hedhili, A., & Chaari, W. L. (2019). Knowledge tracing with an intelligent agent, in an e-learning platform. Education and Information Technologies, 24(1), 711–741. https://doi.org/10.1007/s10639-018-9792-5
van Engelen, J. E., & Hoos, H. H. (2020). A survey on semi-supervised learning. Machine Learning, 109(2), 373–440. https://doi.org/10.1007/s10994-019-05855-6
van Veen, F., & Leijnen, S. (2019). The Neural Network Zoo. The Asimov Institute. https://www.asimovinstitute.org/neural-network-zoo/
Vapnik, V. (1998). Statistical Learning Theory (S. Haykin, Ed.). John Wiley & Sons, Inc.
Wen, J., Li, S., Lin, Z., Hu, Y., & Huang, C. (2012). Systematic literature review of machine learning based software development effort estimation models. Information and Software Technology, 54(1), 41–59. https://doi.org/10.1016/j.infsof.2011.09.002
Wenger, E. (2014). Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge. Morgan Kaufmann.
Winkler-Schwartz, A., Bissonnette, V., Mirchi, N., Ponnudurai, N., Yilmaz, R., Ledwos, N., Siyar, S., Azarnoush, H., Karlik, B., & Del Maestro, R. F. (2019). Artificial Intelligence in Medical Education: Best Practices Using Machine Learning to Assess Surgical Expertise in Virtual Reality Simulation. Journal of Surgical Education, 76(6), 1681–1690. https://doi.org/10.1016/j.jsurg.2019.05.015
Yudelson, M. V., Koedinger, K. R., & Gordon, G. J. (2013). Individualized Bayesian Knowledge Tracing Models. In H. C. Lane, K. Yacef, J. Mostow, & P. Pavlik (Eds.), Artificial Intelligence in Education (pp. 171–180). Springer. https://doi.org/10.1007/978-3-642-39112-5_18
Zhang, Jiani, & King, I. (2016). Topological order discovery via deep knowledge tracing. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 9950 LNCS, 112–119. https://doi.org/10.1007/978-3-319-46681-1_14
Zhang, Jiani, Shi, X., King, I., & Yeung, D.-Y. (2017). Dynamic key-value memory networks for knowledge tracing. Proceedings of the 26th International Conference on World Wide Web, 765–774.
Zhang, Juntao, Li, B., Song, W., Lin, N., Yang, X., & Peng, Z. (2020). Learning Ability Community for Personalized Knowledge Tracing. Web and Big Data, 176–192. https://doi.org/10.1007/978-3-030-60290-1_14

APPENDIX

The Appendix is composed of: (a) the ML categorization figure, (b) the summary table of ML for KT in LM (for clarity reasons, the extensive column 'rationale' has been removed), and (c) the full table of the 29 ML techniques.

It can be found at the following address: https://nextcloud.univ-lille.fr/index.php/s/pDSX4c7QgDT8mdT