Informatics in Education, 2020 Vol. 19, No. 2, 283–321

DOI: 10.15388/infedu.2020.14

Teaching Machine Learning in School: A Systematic Mapping of the State of the Art

Lívia S. MARQUES, Christiane GRESSE VON WANGENHEIM, Jean C. R. HAUCK

Department of Informatics and Statistics, Federal University of Santa Catarina Florianópolis/SC, Brazil

e-mail: livia.marques@grad.ufsc.br, c.wangenheim@ufsc.br, jean.hauck@ufsc.br

Received: November 2019

Abstract. Although Machine Learning (ML) is integrated today into various aspects of our lives, few understand the technology behind it. This presents new challenges to extend computing education early to ML concepts helping students to understand its potential and limits. Thus, in order to obtain an overview of the state of the art on teaching Machine Learning concepts in elementary to high school, we carried out a systematic mapping study. We identified 30 instructional units mostly focusing on ML basics and neural networks. Considering the complexity of ML concepts, several instructional units cover only the most accessible processes, such as data management or present model learning and testing on an abstract level black-boxing some of the underlying ML processes. Results demonstrate that teaching ML in school can increase understanding and interest in this knowledge area as well as contextualize ML concepts through their societal impact.

Keywords: Machine Learning, teaching, K-12.

1. Introduction

Artificial Intelligence (AI) has become part of our everyday life deeply impacting our society. For many countries, it has also become a major strategy to promote national competitiveness (Hiner, 2017). And, as the growth of lucrative AI career opportunities far outpaces the number of interested and capable job seekers, there is a growing need for AI-literate workers (Forbes, 2019).

Although the existence of AI is well known, hardly anybody understands the technology behind it (Evangelista *et al.*, 2018). This lack of understanding also causes a misplaced fear about automation and AI, overshadowing its potential positive impact on society. Therefore, it is important to popularize a basic understanding of AI technologies (Touretzky *et al.*, 2019a). This presents new challenges to computing education, providing students starting at an early age with an understanding of AI concepts to become not just consumers of AI, but creators of intelligent solutions (Touretzky *et al.*, 2019b; Kandlhofer *et al.*, 2016). Access to basic AI literacy can also reduce the danger of social or economic exclusion of certain groups of people, especially women and minorities. Furthermore, AI literacy may encourage more students to consider STEM careers and provide a solid preparation for higher education and their future career.

While there are many programs today that focus on coding and robotics, K-12 education still needs to embrace the teaching of AI concepts. According to AI4K12 (Touretzky *et al.*, 2019c), this should cover five big ideas for a K-12 audience: perception, representation and reasoning, learning, natural interaction, and societal impact. Within this context, an important knowledge area is Machine Learning (ML) (Wollowski *et al.*, 2016; Touretzky *et al.*, 2019a). Machine Learning is the application of AI that provides systems the ability to automatically learn and improve from experience without being explicitly programmed (Royal Society, 2017). It powers a huge range of applications, from speech recognition systems to intelligent assistants, self-driving cars, healthcare, etc.

Teaching fundamental AI (including Machine Learning) concepts and techniques has traditionally been done only in higher education (Torrey, 2012; McGovern *et al.*, 2011). And, although computing education is beginning to be included in K-12 education worldwide, these computing programs rarely cover AI content on this educational stage (Hubwieser *et al.*, 2015). However, in recent years several initiatives and projects pursuing the mission of K-12 AI education have emerged. In this context, the AI for K-12 Initiative (Touretzky *et al.*, 2019c) started to develop guidelines for K-12 AI education. The guidelines are based on a set of big ideas, including teaching computers to learn from data, the challenges involved in making AI agents interact naturally with humans, and the positive and negative effects of AI on society. New AI courses, tools, and tutorials are being launched for teaching AI in schools, in the USA, China, the UK, and elsewhere.

Yet, these efforts seem to be scattered, making it difficult to obtain an overview on existing instructional units, as existing reviews on teaching computing focus mostly on computational thinking (Lye and Koh, 2014; Grover and Pea, 2013; Heintz *et al.*, 2016; Google, 2016), or related knowledge areas such as Software Engineering (da Cruz Pinheiro *et al.*, 2018). Literature providing an overview on how to teach AI/ML in K-12 is basically nonexistent, as surveys on practices and teaching of AI by focuses on higher education only (Wollowski *et al.*, 2016).

Thus, in order to analyze the question of whether and which instructional units are currently available for teaching Machine Learning in K-12, we conduct a systematic mapping study. The main contribution of this article is the mapping and synthesis of the characteristics of instructional units (IUs) for ML education from elementary to high school, regarding their content, context and the analysis of how they were developed and evaluated. Our results also show that it is possible and beneficial to introduce ML education in K-12. The overview can help instructors to select and/or curriculum developers to develop instructional units and we hope that the discussion can further foster the inclusion of ML education in K-12.

2. Background

2.1. *Artificial Intelligence Education in K-12*

Although there have been some historical AI teaching initiatives in schools from the 1970s (Papert & Solomon, 1971; Kahn, 1977) and, even specifically involving neural networks, in the 1990s (Bemley, 1999), there has been a rapid expansion of computing education in K-12 worldwide over the last few years. Standardization of what K-12 students should know about computing has been supported by the development of several curriculum guidelines, such as the CSTA K-12 Computer Science Framework (CSTA, 2017). Many instructional units, software tools, and resources have been developed to make computing accessible for young students ranging from one hour of code programming exercises (code.org) to courses allowing them to learn core computing concepts while creating meaningful artifacts that have direct impact on their lives and their communities (Tissenbaum *et al.*, 2019).

At the same time, AI has had an increasing impact on society. And, although, some countries, such as China has mandated that all high school students learn about artificial intelligence (Jing, 2018), AI education to K-12 students is still not well-defined. Existing computing curriculum guidelines such as the CSTA K-12 Computer Science Framework (CSTA, 2017) commonly only cite AI very briefly on the high school level.

In this context, the AI for K-12 Working Group (AI4K12), a joint initiative of the Association for the Advancement of Artificial Intelligence (AAAI) and the Computer Science Teachers Association (CSTA) aims at developing guidelines for teaching K-12 students about artificial intelligence. To frame these guidelines, "big ideas" in AI that every student should know are defined (Touretsky *et al.*, 2019a):

1. **Perception:** Computers perceive the world using sensors. Students should understand that machine perception of spoken language or visual imagery requires extensive domain knowledge.
2. **Representation and Reasoning:** Agents maintain models/representations of the world and use them for reasoning. Students should understand the concept of representation and understand that computers construct representations using data, and these representations can be manipulated by applying reasoning algorithms that derive new information from what is already known.
3. **Learning:** Computers can learn from data. Students should understand that machine learning is a kind of statistical inference that finds patterns in data.
4. **Natural Interaction:** Making agents interact naturally with humans is a substantial challenge for AI developers. Students should understand that while computers can understand natural language to a limited extent, at present they lack the general reasoning and conversational capabilities of even a child.
5. **Societal Impact:** AI applications can impact society in both positive and negative ways. Students should be able to identify ways that AI is contributing to their lives as well as that the ethical construction of AI systems requires attention to the issues of transparency and fairness.

Thus, while AI is “the science and engineering of making intelligent machines that have the ability to achieve goals as humans do”, Machine Learning (ML) is a subfield of AI dealing with the field of study that gives computers the ability to learn without being explicitly programmed (Mitchell, 1997). ML algorithms build a mathematical model based on sample data, known as “training data”, in order to make predictions or decisions without being explicitly programmed to perform the task. In accordance with AI4K12, Machine Learning concepts to be covered in K-12 education should include (Touretzky *et al.*, 2019c):

- What is learning?
- Approaches to machine learning (e.g., regression algorithms, instance-based algorithms, support vector machines, decision tree algorithms, Bayesian algorithms, clustering algorithms, artificial neural network algorithms).
- Types of learning algorithms by learning style.
- Fundamentals of neural networks.
- Types of neural network architecture.
- How training data influences learning.
- Limitations of machine learning.

And, although, currently there are significant efforts underway to address the need for AI curriculum guidelines (ISTE, 2018) (AI4ALL, 2018), unlike the general subject of computing, when it comes to AI, there is still little guidance available for teaching at the K-12 level.

2.2. *Machine Learning*

Machine Learning is the training of a model from data that generalizes a decision against a performance measure (Mitchell, 1997).

ML algorithms can be classified into several broad categories by their learning style (Goodfellow *et al.*, 2016). In supervised learning, the algorithm builds a mathematical model from a set of data that contains both the inputs and the desired outputs. Classification algorithms and regression algorithms are types of supervised learning. In semi-supervised learning, a combination of labeled data and unlabelled data is used in order to make better predictions for new data points than by using the labeled data alone. In unsupervised learning, the algorithm builds a mathematical model from a set of data that contains only inputs and no desired output labels. Unsupervised learning algorithms are used to find structure/patterns in the data, like grouping or clustering the data points into categories. Reinforcement learning algorithms are given feedback in the form of positive or negative reinforcement in a dynamic environment and are used, e.g., in autonomous vehicles.

Building ML applications is an iterative process that involves a sequence of steps, which typically include (Amazon, 2019):

1. **Requirements analysis.** During this stage, the main objective of the model and its target features are specified. This also includes the characterization of the inputs and expected outputs, specifying the problem.

2. **Data management.** During data collection, available datasets are identified and/or data is collected. This may also include the selection of available generic datasets (e.g., ImageNet for object detection), as well as specialized datasets for transfer learning. The type of data depends on the machine learning task (e.g., images, sound, text, etc.). They also vary greatly in terms of the number of instances ranging from a few hundred to more than a billion instances. The data is prepared by validating and cleaning the data and can also be preprocessed transforming the raw data. Data sets may be labeled in supervised learning by augmenting each piece of unlabeled data with meaningful tags manually assigned by users. The data set is typically split into a training set to train the model, validation set to select the best candidate from all models and a test set to perform an unbiased performance evaluation of the chosen model on unseen data (Ripley, 2008).
3. **Feature engineering.** Often, the raw data (input variables) and answer (target) are not represented in a way that can be used to train a machine learning model. Therefore, feature engineering is the process of using domain knowledge of the data to create features that make machine learning algorithms work. This may include feature transformation, feature generation, selecting features from large pools of features among others.
4. **Model learning.** Then a model is built or more typically chosen from well-known models that have been proven effective in comparable problems or domains (e.g., (ModelZoo, 2019)) by feeding the features/data to the learning algorithm. The quality of the model(s) is evaluated in order to understand how to iteratively improve its performance (e.g., in terms of high accuracy, lower error) by testing the model against previously unseen data (Tharwat, 2019). Hyperparameters, such as the number of training steps, learning rate, initialization values, and distribution, etc. are finetuned in order to improve performance.
5. **Model evaluation.** The quality of the model is evaluated in order to test the model providing a better approximation of how the model will perform in the real world, e.g., by analyzing the correspondence between the results of the model and human labeling.
6. **Model deployment.** During the production/deployment phase, the model is deployed into a production environment to apply it to new incoming events in real-time.

There are a number of programming languages that are popular for machine learning. Among them, Python is the most popular language followed by Java, R, and C++ (Tricon Infotech, 2019). Especially in the context of K-12 computing education, block-based programming languages are used (Weintrop, 2019). These environments improve learnability for novices by favoring recognition over recall; reducing cognitive load by chunking computational patterns into blocks; and using direct manipulation of blocks to prevent errors and enhance understanding of program structure (Bau *et al.*, 2017). Several of these block-based programming environments also provide extensions for the development of machine learning solutions, such as for App Inventor, Scratch or SNAP!.

2.3. *Development and Evaluation of Instructional Units*

An instructional unit is a set of classes (courses, workshops, etc.) designed to teach certain learning objectives to a specific target audience. It consists of a set of instructional materials for both teachers and students designed to provide learning opportunities in a specific context (Hill *et al.*, 2005).

Instructional units are typically developed in a systematic way using instructional design (Branch, 2009), in order to make the acquisition of competencies more efficient, effective, and appealing. Instructional design defines an iterative process of planning learning objectives, selecting instructional strategies, selecting or creating instructional material, and applying and evaluating instructional units. During the analysis phase, the learning needs are identified. As part of the analysis, the goals and objectives of the instructional unit are determined and the target audience is characterized. Other influencing factors, such as human and technical resources, infrastructure, cost and time, are also analyzed. During the design phase, the learning objectives of the instructional unit are specified. The content to be addressed is defined and sequenced, and the instructional methods to be used are defined. Instructional methods may include lectures, demonstrations, exercises, problem-solving activities (labs), online interactive tutorials, serious games, unplugged activities, etc. It is also defined how the students' learning will be assessed. During the development phase, the material that will be used during the instructional unit is selected and/or created in accordance with the defined instructional strategies. This step may also involve the selection and/or development of tools to support the instructional unit such as code analyzers. The implementation phase covers the preparation of the learning environment, the training of the instructors and the application of the IU in the classroom.

An essential step in the instructional design process is the evaluation of the instructional unit in order to assess its quality and whether it allows the students to achieve the defined objectives (Branch, 2009). This evaluation is typically performed through an empirical study (Wohlin *et al.*, 2012), ranging from non-experimental studies (such as case studies) to experiments (Shadish *et al.*, 2002). Several types of data collection instruments can be used, such as observation, questionnaires, interviews, or the artifacts created by the students themselves as well as test results (Branch, 2009). According to the objective of the evaluation and the nature of the data collected, different methods of qualitative or quantitative analysis can be used (Freedman *et al.*, 2007). The analyzed data are then interpreted, answering the analysis questions in order to achieve the evaluation goal.

3. Definition and Execution of the Systematic Mapping Study

To elicit the state of the art and practice on whether and how Machine Learning education is addressed from elementary to high school, we conducted a systematic mapping study following the procedure proposed by Petersen *et al.* (2008).

3.1. *Definition of the Review Protocol*

The research question is: What instructional units exist for teaching Machine Learning concepts in the context of elementary to high school (and what are their characteristics)? This research question is refined in the following analysis questions:

AQ1. Which IUs exist?
AQ2. Which Machine Learning concepts are taught in the IUs?
AQ3. What are the instructional characteristics of the IUs?
AQ4. How were the IUs developed and how was the quality of the IUs evaluated?

Inclusion/exclusion criteria. We considered any instructional unit (course, activity, tutorial) that focuses on computer teaching including ML concepts in elementary to high school published between 2009 and 2019. Instructional units that focus on teaching ML in higher education and/or instructional units for computing teaching without addressing ML concepts were excluded. We also excluded publications such as blogs, videos, or tools that do not provide an instructional unit.

Quality Criteria. We considered only articles or material for which substantial information regarding the teaching of ML concepts, indicating, for example, lesson content, instructional material, etc. were freely available.

Data source. We examined all published English-language articles or material that are available on the Web via the most important digital libraries and databases in this field (including ACM Digital Library, IEEEXplore, Scopus) with free access through the CAPES Portal[1]. To increase coverage, we also used Google, which indexes a large set of data across several different sources (Haddaway *et al.*, 2015), as in this emergent area several instructional units have not been published as scientific articles. Observing also the research focus at the MIT media lab in this area, we also searched for publications of this research group. We have also included secondary literature that has been discovered based on the primary literature found in order to obtain more detailed information.

Definition of the search string. The search string was composed of concepts related to the research question, including also synonyms, as indicated in Table 1.

Table 1
Keywords

Main concepts	Synonyms
Machine Learning	artificial intelligence, deep learning, data science
K-12	school, kids, teens, children
instructional unit	teach, learn, education, course, MOOC

[1] A web portal for access to scientific knowledge worldwide, managed by the Brazilian Ministry of Education for authorized institutions, including universities, government agencies and private companies (www.periodicos.capes.gov.br).

From these keywords, the search string was calibrated and adapted according to the specific syntax of the data source as presented in Table 2:

(*teach* OR education OR course OR MOOC OR learn**) *AND* (*"machine learning" OR "data science" OR "artificial intelligence" OR "deep learning"*) *AND* (*"k-12" OR school* OR kids OR children OR teen**)

3.2. *Search Execution*

The search has been realized in October 2019 by the first author and revised by the co-authors (Table 3). Several searches returned a large number of results even after a calibration of the search string. This is due to the fact that articles describing how to use AI techniques for education, such as learning analytics for personalized learning, correspond to the same search terms. Therefore, maintaining the search string we limited the analysis to only the most relevant ones.

In the first analysis stage, we quickly reviewed titles and abstracts to identify papers that matched the inclusion criteria, resulting in 98 potentially relevant artifacts. In the second stage, the materials were fully read to check their relevance with respect to our inclusion/exclusion criteria. Many articles were excluded due to their focus on using AI for education, or their focus on "deep learning" as a cognitive activity in the learning process. We also excluded artefacts related to other educational stages (pre-school or higher education) (Williams *et al.*, 2019a; Williams *et al.*, 2019b; Park *et al.*, 2019; Bennett, 2017; Estevez *et al.*, 2019) and the ones covering AI, but not machine learning (e.g., (CSUnplugged, 2015; AI4ALL, 2019; Ali *et al.*, 2019; Parsons and Sklar, 2004; MIT, 2019)). Furthermore, we excluded material only consisting of videos explaining ML (CS4fn, 2019) or tools ((Agassi *et al.*, 2019; Makeblock, 2019)) or demos (such as Google Teachable Machine (Google, 2017). We also excluded articles that do not provide substantial information on the instructional unit on Machine Learning (e.g. (Kandlhofer *et al.*, 2016)). Duplicates were eliminated and articles describing the same instructional unit were unified. As a result, 30 instructional units were considered relevant, as shown in Table 4.

3.3. *Data Extraction*

We systematically extracted data from the articles in order to answer the analysis questions. Data extraction was hampered in many cases by the way the material was presented. As several IUs have not been published as articles, information has been extracted based on the available instructional material, inferring characteristics such as the learning objectives. In case, no information was available, we indicate this lack as Not Informed (NI). A detailed description of the extracted data for each of the analysis questions is presented in Appendix A.

Table 2
Search strings for each source

Source		Search string
ACM	`https://dlnext.acm.org/search/advanced`	[[Abstract: teach*] OR [Abstract: education] OR [Abstract: course] OR [Abstract: mooc] OR [Abstract: learn*]] AND [[Abstract: "machine learning"] OR [Abstract: "data science"] OR [Abstract: "artificial intelligence"] OR [Abstract: "deep learning"]] AND [[Abstract: "k-12"] OR [Abstract: school*] OR [Abstract: kids] OR [Abstract: children] OR [Abstract: teen*]] AND [Publication Date: (01/01/2009 TO *)]
IEEE	`https://ieeexplore.ieee.org/search`	((("Abstract":teach*) OR ("Abstract":education) OR ("Abstract":course) OR ("Abstract":MOOC) OR ("Abstract":learn*)) AND (("Abstract":"machine learning") OR ("Abstract":"data science") OR ("Abstract":"artificial intelligence") OR ("Abstract":"deep learning")) AND (("Abstract":"k-12") OR ("Abstract":school*) OR ("Abstract":kids) OR ("Abstract":children) OR ("Abstract":teen*))) Filters Applied: 2009–2019
Scopus	`https://www2.scopus.com/search`	TITLE-ABS-KEY ((teach* OR education OR course OR mooc OR learn*) AND ("machine learning" OR "data science" OR "artificial intelligence" OR "deep learning") AND ("k-12" OR school* OR kids OR children OR teen*)) AND (LIMIT-TO (PUBYEAR , 2020) OR LIMIT-TO (PUBYEAR , 2019) OR LIMIT-TO (PUBYEAR , 2018) OR LIMIT-TO (PUBYEAR , 2017) OR LIMIT-TO (PUBYEAR , 2016) OR LIMIT-TO (PUBYEAR , 2015) OR LIMIT-TO (PUBYEAR , 2014) OR LIMIT-TO (PUBYEAR , 2013) OR LIMIT-TO (PUBYEAR , 2012) OR LIMIT-TO (PUBYEAR , 2011) OR LIMIT-TO (PUBYEAR , 2010) OR LIMIT-TO (PUBYEAR , 2009)) AND (LIMIT-TO (SUBJAREA , "COMP"))
Google	`https://www.google.com/`	"machine learning" teach "K-12" OR school
MIT media lab	`https://appinventor.mit.edu/explore/research` `https://www.media.mit.edu/groups/lifelong-kindergarten/publications/`	--

Table 3
Number of identified articles per repository per selection stage

Source	No. of search results	No. of analyzed results	No. of potentially relevant results	No. of relevant results
ACM	3,948	200	10	8
IEEE	698	200	5	3
SCOPUS	2,373	200	4	2
Google	39,900,000	500	75	16
MIT media lab	118	118	4	4

4. Data Analysis

4.1. *Which Instructional Units Exist?*

As a result of the research, a total of 30 instructional units covering the teaching Machine Learning in elementary to high school were identified (Table 4). Some instructional units focus exclusively on Machine Learning, whereas others approach ML concepts as part of a more comprehensive AI and/or programming/software engineering course.

Table 4
Instructional units for teaching Machine Learning in elementary to high school

Reference	Name of the IU	Brief description	Source
(AI Family Challenge, 2019)	AI Family Challenge	Challenge to families to learn about AI technology and solve a problem in their communities using AI tools.	https://www.curiositymachi-ne.org/about/
(ai4children, 2017)	AI 4 children	Services that allow you to teach AI to children using Scratch.	https://www.ai4children.org/
(AIinSchools, 2019)	AI in Schools	A program that aims to demystify the topic of AI.	http://aiinschools.com/
(Apps for Good, 2019a)	Apps for good: Machine Learning in a day	Taster workshop for students to gain an understanding of how machine learning impacts on their lives.	https://www.appsforgood.org/courses/ml-in-a-day
(Apps for Good, 2019b)	Apps for good: Machine Learning course	It provides an overview of diverse ML topics and aims at student teams to design and build a prototype that solves a problem they care about using ML algorithms.	https://www.appsforgood.org/courses/machine-learn-ing
(Burgsteiner *et al.*, 2016; Burgsteiner, 2016)	IRobot: Teaching the Basics of Artificial Intelligence in High Schools	AI-course covering major AI topics (problem-solving, search, planning, graphs, data structures, automata, agent systems, machine learning).	Burgsteiner, H., Kandlhofer, M., Steinbauer, G. (2016).. IRobot: Teaching the Basics of Artificial Intelligence in High Schools. Proc. of the Sixth Symposium on Educational Advances in Artificial Intelligence, Phoenix, AZ, USA. Burgsteiner, H. (2016). Design and Evaluation of an introductory artificial intelligence class in high schools. Diploma thesis, TU Graz, Austria.
(Cognimates, 2019)	Cognimates	An AI education platform for build-ing games, programming robots and training AI models.	Druga, S., Vu, S.T., Likhith, E., Qiu. T. (2019). Inclusive AI literacy for kids around the world. Proc.s of FabLearn, New York, NY, USA. Druga, S. (2018). Growing up with AI : Cognimates : from coding to teaching machines. Thesis: S.M., Massachusetts Institute of Technology, Program in Media Arts and Sciences. http://cognimates.me

Continued on next page

Table 4 – continued from previous page

Reference	Name of the IU	Brief description	Source
(CS4FN, 2011)	CS4FN – Computer Science for Fun	Diverse activities to teach computer science and AI/ML.	http://www.cs4fn.org/teachers/activities/braininabag/braininabag.pdf http://www.cs4fn.org/ai/snap/ http://www.cs4fn.org/teachers/activities/sweetcomputer/sweetcomputer.pdf
(Curiosity-machine, 2019)	Curiosity Machine – build a neural network	Design challenge to build a prototype of an unplugged artificial neural network that can classify different objects.	https://www.curiositymachine.org/challenges/126/
(Elements of AI, 2019)	Elements of AI	Online course to encourage as broad a group of people as possible to learn what AI is, what can (and can't) be done with AI, and how to start creating AI methods.	https://course.elementsofai.com/4
(Essinger and Rosen, 2019)	Machine Learning: An Introductory Unit of Study for Secondary Education	Example scenarios that give motivation to the students for learning k-means algorithm, including a recycling sorting and a biology problem.	Essinger, S. D., Rosen, G. L. (2019). Machine Learning: An Introductory Unit of Study for Secondary Education. Proc. of the 50th ACM Technical Symposium on Computer Science Education, Minneapolis, MN, USA.
(Evangelista *et al.*, 2018)	Why are we not teachingmachine learning at high school?	Workshop for the introduction to ML through a series of problem-based activities.	Evangelista, I., Blesio, G., Benatti, E. (2018). Why Are We Not Teaching Machine Learning at High School? A Proposal. Proc. of the World Engineering Education Forum - Global Engineering Deans Council, Albuquerque, NM, USA.
(Fryden curriculum, 2019)	Fryden Curriculum	Website to support anyone to learn about Machine Learning, especially neural networks.	http://www.fryden-learning.com/fryden-curriculum
(Hitron *et al.*, 2019)	Can Children Understand Machine Learning Concepts?	Proposing a gesture recognition research platform, designed to support learning from experience by uncovering ML building blocks to perform physical gestures, iterating between sampling and evaluation.	Hitron, T. *et al.* (2019). Can Children Understand Machine Learning Concepts?: The Effect of Uncovering Black Boxes. Proc. of the CHI Conference on Human Factors in Computing Systems, Glasgow, Scotland UK.
(Ho and Scadding, 2019)	Classroom Activities for Teaching Artificial Intelligence to Primary School Students	Classroom activities for teaching basic AI concepts in order to demonstrate that some seemingly complex concepts such as facial recognition and machine learning can be explained in terms of simple computer algorithms that simulate specific human-like behaviors.	Ho, J. W. K., Scadding, M. (2019). Classroom Activities for Teaching Artificial Intelligence to Primary School Students. Proc. of the Int. Conference on Computational Thinking, Hong Kong, China.

Continued on next page

Table 4 – continued from previous page

Reference	Name of the IU	Brief description	Source
(Kahn and Winters, 2018) (Kahn *et al.*, 2018)	eCraft2Learn	Programming guides that describe extensions to the Snap! programming language to enable children (and non-expert programmers) to build AI programs.	https://ecraft2learn.github.io/ai/ Kahn, K., Winters, N. (2018). AI Programming by Children. Proc. of the Constructionism Conference, Vilnius, Lithuania. Kahn, K., Megasari , R., Piantari, E., Junaeti, E. (2018). AI Programming by Children using Snap! Block Programming in a Developing Country. European Conference on Technology Enhanced Learning, Delft, Netherlands.
(MIT App Inventor, 2019)	Introduction to Machine Learning: Image Classification	Course teaching the basics of machine learning and the creation of the students'own apps that implement these concepts through image classification.	http://appinventor.mit.edu/explore/resources/ai/image-classification-look-extension
(ML4Kids, 2019)	Machine Learning for Kids	Online tutorials guiding children to create a game or interactive project that demonstrates a real-world use of artificial intelligence and machine learning.	https://machinelearningforkids.co.uk/#!/worksheets
(Mobasher *et al.*, 2019)	Data Science Summer Academy for Chicago Public School Students	Summer data science academy aimed to broaden the participation of underrepresented groups in computing by teaching a variety of data science methods and their applications, including data visualization, distance-based methods, classification, clustering, and others.	Mobasher, B. *et al.* (2019). Data Science Summer Academy for Chicago Public School Students. ACM SIGKDD Explorations Newsletter, 21(1).
(Narahara and Kobayashi., 2018)	Personalizing homemade bots with plug and play AI for STEAM education	Proposal for a new framework for hands-on educational modules to introduce ideas in AI and robotics for an autonomous toy car.	Narahara, T., Kobayashi, Y. (2018). Personalizing homemade bots with plug and play AI for STEAM education. Proc. of SIGGRAPH Asia Technical Briefs, Tokyo, Japan.
(ReadyAI, 2019)	Ready AI AI+Me	AI+ME is an online experience intended to provide young learners with the basics of AI.	ReadyAI AI+Me https://edu.readyai.org/courses/aime/
(Sakulkueakulsuk *et al.,* 2018)	Kids making AI: Integrating machine learning, gamification, and social Context in STEM Education	Approach for STEM education at the intersection of machine learning, gamification, and social context through an agricultural-based AI challenge that aims at students to learn the process of creating machine learning models in the form of a game.	Sakulkueakulsuk, B. *et al.* (2018). Kids making AI: Integrating Machine Learning, Gamification, and Social Context in STEM Education. Proc. of IEEE Int. Conference on Teaching, Assessment, and Learning for Engineering, Wollongong, Australia.
(Sperling and Lickerman, 2012)	Integrating AI and machine learning in software engineering course for high school students	Proposal for a software engineering curriculum for high-school students that includes subjects in artificial intelligence and machine learning.	Sperling, A., Lickerman, D. (2012). Integrating AI and machine learning in software engineering course for high school students. Proc. of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education, Haifa, Israel.

Continued on next page

Table 4 – continued from previous page

Reference	Name of the IU	Brief description	Source
(Srikant and Aggarwal, 2017)	Introducing Data Science to School Kids	Data science workshop to expose students to the full cycle of a typical supervised learning approach.	Srikant, S., Aggarwal, V. (2017). Introducing Data Science to School Kids. Proc. of the ACM 48th SIGCSE Technical Symposium on Computer Science Education, Seattle, WA, USA. http://www.datasciencekids.org/
(Tang, 2019; Tang *et al.*, 2019)	Empowering novices to understand and use Machine Learning with personalized image classification models, intuitive analysis tools, and MIT App Inventor.	Workshop to teach core machine learning concepts with image classification using a web interface that allows users to train and test personalized image classification models on pictures taken with computer webcams and an extension for MIT App Inventor that allows users to use the models to classify objects in their mobile applications.	Tang, D. (2019). Empowering Novices to Understand and Use Machine Learning With Personalized Image Classification Models, Intuitive Analysis Tools, and MIT App Inventor, M.Eng thesis, Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, USA. Tang, D., Utsumi, Y., Lao, N. (2019). PIC: A Personal Image Classification Webtool for High School Students. Proc. of the IJCAI EduAI Workshop, Sicily, Italy.
(TechGirlz, 2018)	Artificial Intelligence: How Computers Learn	Workshop in which students will learn about how machine learning techniques such as artificial neural networks learn from data to answer real-world questions.	https://www.techgirlz.org/topic/artificial-intelligence-computers-learn/
(Vachovsky *et al.*, 2016)	Toward More Gender Diversity in CS through an Artificial Intelligence Summer Program for High School Girls	A summer program to recruit high school girls to computer science, specifically to AI. Project topics include computer vision, robotics, NLP, and computational biology.	Vachovsky, M. E. *et al.* (2016). Toward More Gender Diversity in CS through an Artificial Intelligence Summer Program for High School Girls. Proc. of the 47th ACM Technical Symposium on Computing Science Education, Memphis, TN, USA.
(Van Brummelen, 2019) (Van Brummelen and Abelson, 2018) (Van Brummelen *et al.*, 2019)	App Inventor for Conversational AI	A workshop that aims to democratize conversational AI technology by teaching students to create Alexa Skills developing conversational App Inventor apps.	Van Brummelen, J. (2019). Tools to Create and Democratize Conversational Artificial Intelligence, M.S. thesis, Elect. Eng. Comput. Sci., Massachusetts Inst. of Technol., Cambridge MA, USA. Van Brummelen, J., Abelson, H. (2018). What's conversational AI?' with MIT App Inventor and Amazon Alexa. Proc. of Amazon Research Days, Boston, MA, USA. Van Brummelen, J., Shen, J. H., Patton, E. W. (2019). The Popstar, the Poet, and the Grinch: Relating Artificial Intelligence to the Computational Thinking Framework with Block-based Coding. Proc. of the Int. Conference on Computational Thinking, Hong Kong, China.

Continued on next page

Table 4 – continued from previous page

Reference	Name of the IU	Brief description	Source
(Zhu, 2019)	An Educational Approach to Machine Learning with Mobile Applications	Course to introduce students to what machine learning can do and allow them to build mobile ML applications with App Inventor.	Zhu, K. (2019). An Educational Approach to Machine Learning with Mobile Applications. M.Eng thesis, Elect. Eng. Comput. Sci., Massachusetts Institute of Technology, Cambridge, MA, USA.
(Zimmermann-Niefield *et al.*, 2019a; Zimmermann-Niefield *et al.*, 2019b)	Sports and machine learning: how young people can use data from their own bodies to learn about machine learning	Workshop to introduce youth to making ML models within the context of their athletic interests by building models of their own physical activity using wearable sensors.	Zimmermann-Niefield, A., Shapiro, R.B, Kane, S. (2019a). Sports and machine learning: How young people can use data from their own bodies to learn about machine learning. XRDS: Crossroads, 25(4), 44–49. Zimmermann-Niefield, A., Turner, M., Murphy, B., Kane, S.K., Shapiro, R.B. (2019b). Youth Learning Machine Learning through Building Models of Athletic Moves. Proc. of the 18th ACM Int.Conference on Interaction Design and Children, Boise, ID, USA.

This shows that so far very few IUs approach Machine Learning education in K-12. Most of them are also very recently due to the increasing importance of AI/ML as well as the increasing trend of computing education in K-12 worldwide (Fig. 1).

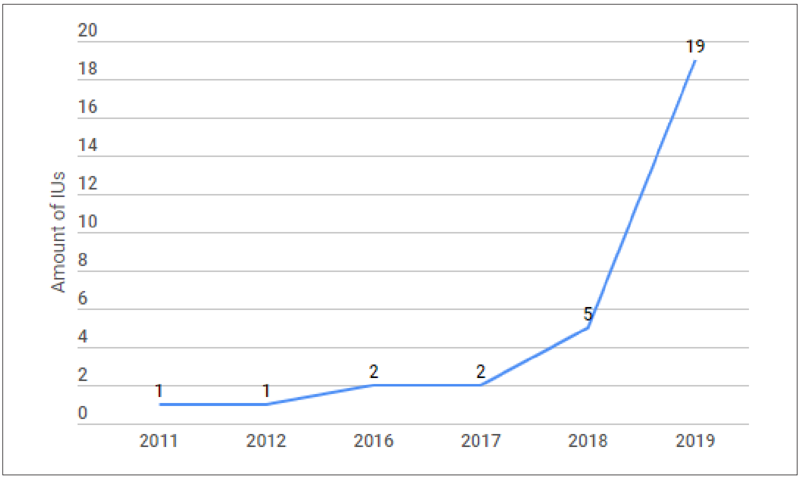

Fig. 1. Amount of IUs focusing on ML in schools published per year.

4.2. *Which Machine Learning Competencies are Taught in the IUs?*

The IUs teach competencies varying from presenting what is ML, to specific ML techniques as well as the impacts of ML. Among the topics most frequently approached by the IUs are artificial neural networks and an introduction to what is learning (Fig. 2). Several IUs also present other ML algorithms such as decision tree and/or instance-based algorithms typically using unplugged activities. A few IUs also approach the topic of social implications and ethical concerns.

The majority of the IUs focuses on supervised learning algorithms (Fig. 3), only very few approach other types of learning.

And, although several IUs approach the topic of neural networks, they typically present this content in an abstract way and/or through practical applications. We also observed that the degree of abstraction of the ML concepts varies between the IUs. Whereas some IUs only teach a general understanding of ML mechanisms and its applications, most IUs cover one or more ML algorithms typically by presenting an example, demonstration or hands-on activity in order to provide a deeper understanding.

A general strength observed in the encountered IUs is their strong focus on demonstrating the application of ML in practice, typically presenting various application examples in order to gain the attention of the students (Fig. 4). This includes mainly the demonstration of the application of ML for classification in computer vision tasks, such as facial or gesture recognition (Hitron *et al.*, 2019) for diverse domains, including recycling, biology, etc. Several IUs present various application domains (e.g., (Zhu, 2019)) including also sentiment analysis for examples of tweets, conversational AI (e.g., creating Alexa skills (Van Brummelen and Abelson, 2018)), robotics or games (e.g.,

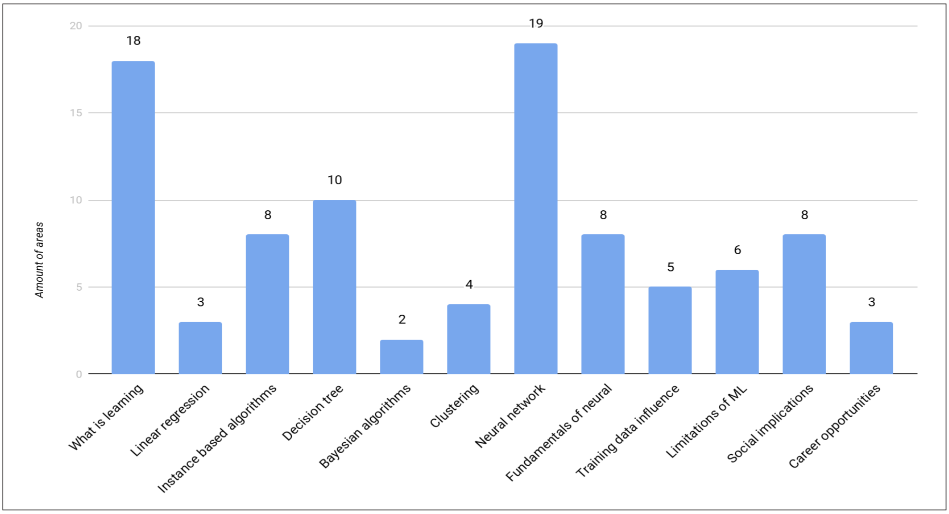

Fig. 2. Frequency of ML topics covered by the IUs.

(Zhu, 2019)). Some units also integrate ML into robotics activity, such as creating a self-learning lawn bowling robot (Ho and Scadding, 2019) or running toy cars on a physical track (Narahara and Kobayashi., 2018).

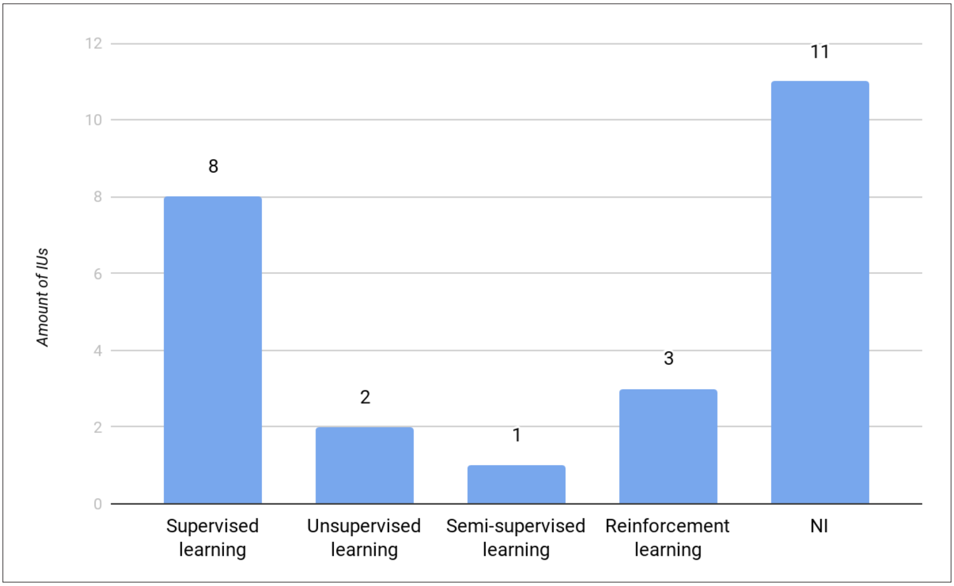

Fig. 3. Frequency of type of learning style covered by the IUs.

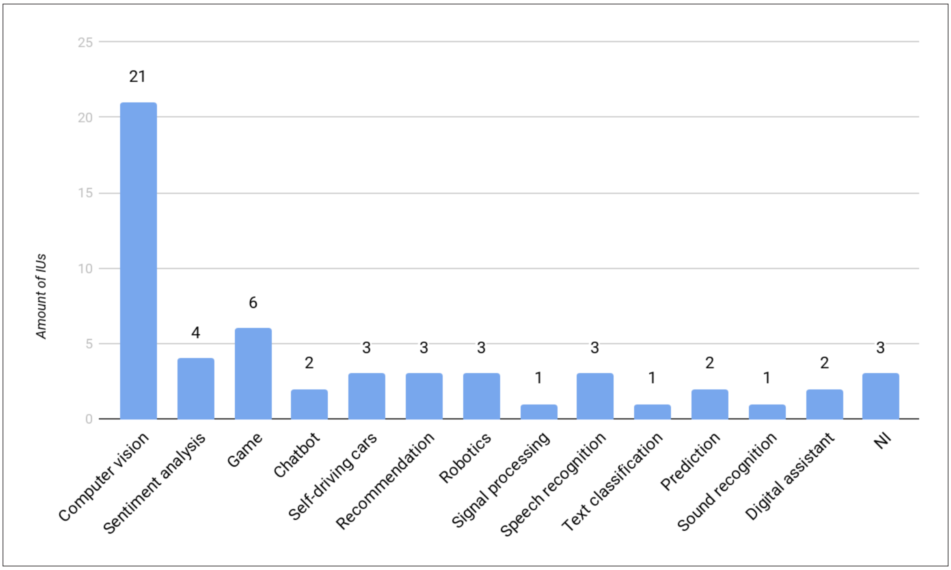

Fig. 4. Frequency of application domains covered by the IUs.

The IUs also vary largely in terms of levels of learning they are designed to achieve in accordance with Bloom's Taxonomy (Bloom *et al.*, 1956). Several instructional units focus exclusively on lower learning levels (remembering and understanding), whereas some IUs also approach the level of synthesis taking students to create their own ML model. On this level, various IUs adopt a computational action approach (Tissenbaum *et al.*, 2019) aiming at the development of an ML solution for a problem in the community (AI Family Challenge, 2019; Apps For Good, 2019b). Few IUs approach the highest level of learning evaluation by making judgments based on evidence of different ML models or techniques and/or how training data influences learning.

Observing the complexity of ML concepts, several UIs cover only the most accessible processes, such as data management (such as (Mobasher *et al.*, 2019) (Srikant and Aggarwal, 2017)). On the other hand, a considerable number of UIs also cover model learning and testing, yet, on very different levels of depth. Most of these IUs present several ML concepts only on an abstract level black-boxing some of the underlying ML processes. In these cases, the model learning process may be approached by only executing a pre-defined model learning process without any need for further interaction (e.g. (ML4Kids, 2019)). Very few IUs systematically introduce ML performance measures, such as a correctness table, confidence graph, presenting accuracy often in a more superficial way. Only a small number of IUs also include the deployment of the created ML models, for example as part of games of mobile applications.

Different ML frameworks or tools are used on this educational stage visioning the abstraction of several stages and complexity of ML models (Fig. 6). For example, ML4kids (2019) provides an abstract interface permitting young people to easily train a neural network. On the other hand, several IUs directly use general ML frameworks

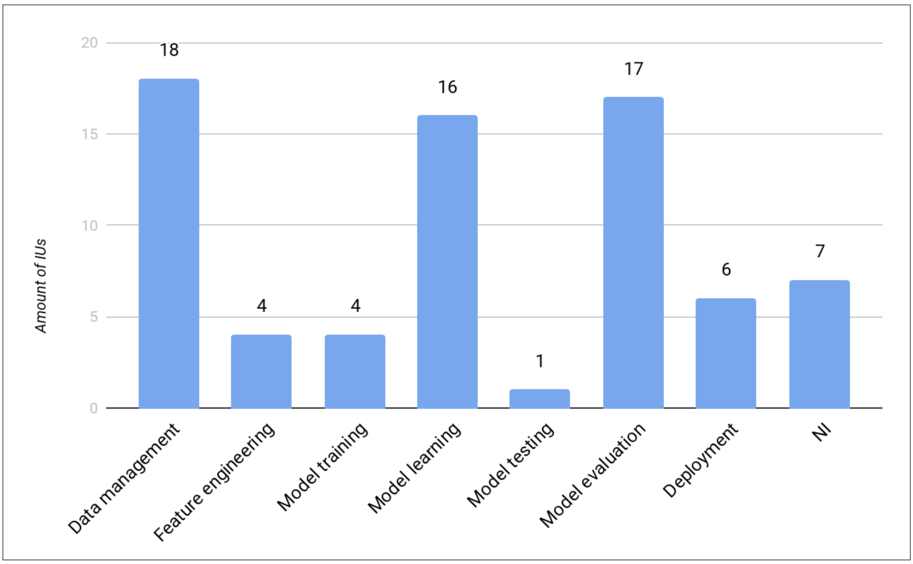

Fig. 5. Frequency of ML processes covered by the IUs.

such as TensorFlow and Jupyter Notebooks that are not specifically developed for this educational stage.

As typically used in computing education in K-12, IUs on ML also adopt predominantly block-based programming languages such as Scratch (6 IUs), Snap! (1 IU) or App Inventor (5 IUs). Six IUs also directly use Python.

Hands-on activities of the IUs mostly work with image data for classification tasks. These vary from paper images in unplugged activities to digital images ranging from Disney princesses and faces to chocolate chip cookies.

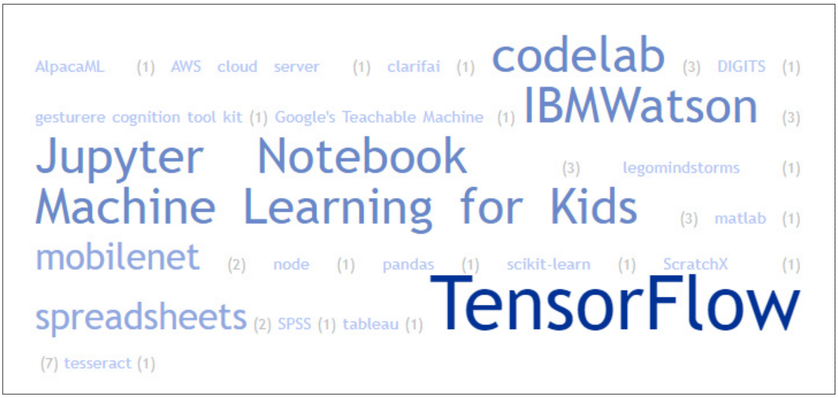

Fig. 6. Frequency of the ML frameworks/tools adopted.

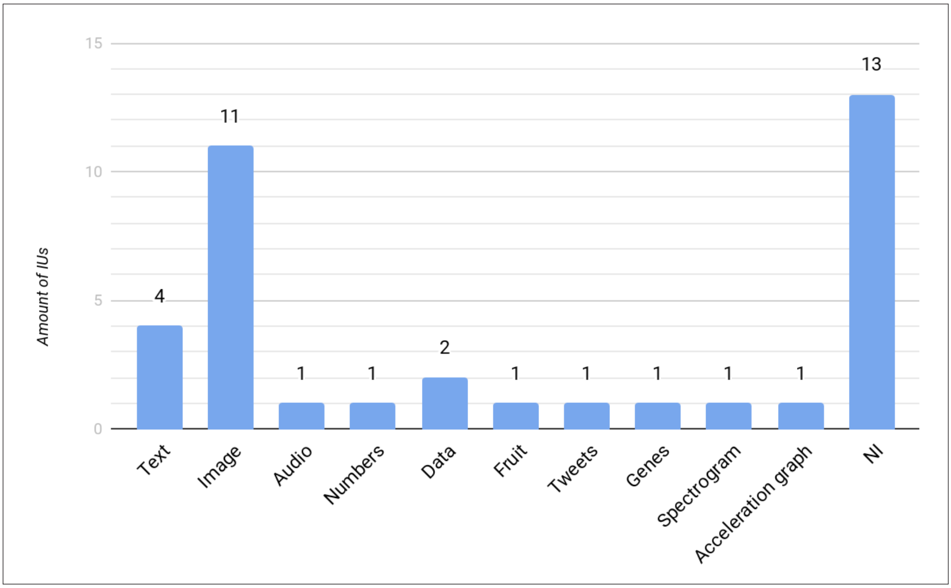

Fig. 7. Frequency of type of data used in the IUs.

Some units focusing on sports-related themes also use time series of images/acceleration graphs for the classification of gestures. For example, by applying ML to sports, students collect data from their own bodies using wearable sensors playing softball (Zimmermann-Niefield *et al.*, 2019a). Several IUs adopting a computational action approach (Tissenbaum *et al.*, 2019) in open-ended project-based activities leave the type of image used open depending on students' choice of the application domain. Other IUs also use datasets based on texts (e.g. tweets), audio clips, genes, etc. During the IU presented by Sakulkueakulsuk *et al.* (2018), students collect data on features (skin color, texture, etc.) of mango fruits.

4.3. *What are the Instructional Characteristics of the IUs?*

As the teaching of ML competencies is currently not typically included in computing education, the majority of the IUs are proposed as extracurricular activities, workshops, courses, summer camps, challenges or individual activities. Only MIT (2019) and Sperling and Lickerman (2012) propose a curricular unit as part of a computing/software engineering course. Only 3 online courses have been encountered (ReadyAI, 2019; Elements of AI, 2019; Kahn and Winters, 2018).

According to the students' current lack of knowledge regarding computing and/or ML, most IUs are aimed at beginners with no prior computing/ML competencies, with the exception of Tang *et al.* (2019) requiring prior App Inventor experience. Only ML4Kids (2019) and Curiositymachine (2019) propose also instructional units on the intermediate and advanced level.

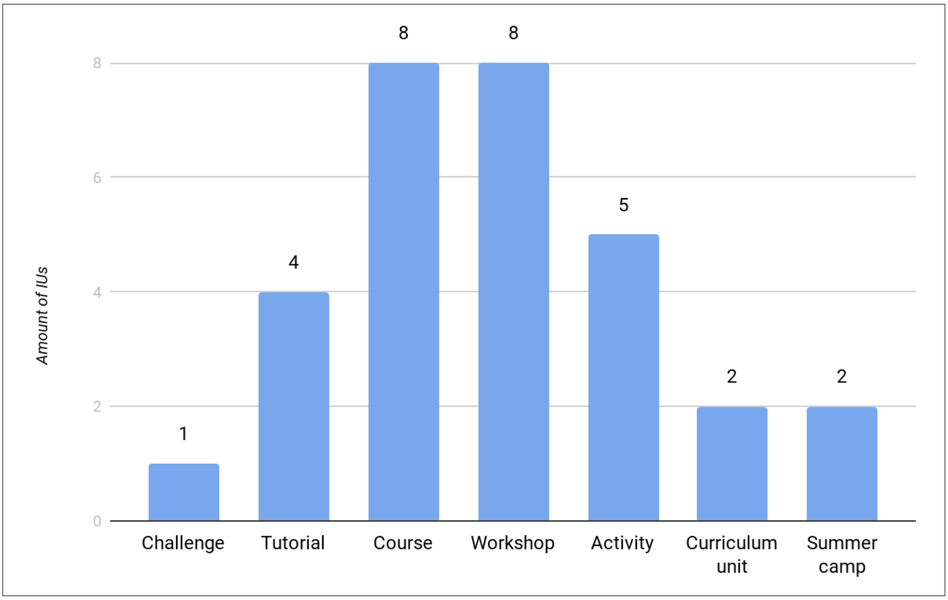

Fig. 8. Frequency of type of data used in the IUs.

Most of the IUs are focused on teaching ML in high school (Fig. 9). Also, several UIs are available for elementary and/or middle school level indicates that the insertion of ML education can be beneficial already on these earlier educational stages.

Very few IUs focus on specific groups of students such as girls (Vachovsky *et al.*, 2016), underrepresented groups in computing by targeting economically disadvantaged, African American, Hispanic, and female students (Mobasher *et al.*, 2019) or specifically at further education (Apps for Good, 2019a) (Apps for Good, 2019b). The AI Family Challenge (2019) is designed for families, teaching AI not only to the children but also to other family members.

The duration of the IUs varies largely from short and focused activities (45 minutes) to long-term courses of 100 hours, yet, with the majority being rather short units of few lessons. Several initiatives also offer instructional units of different durations, such as a one-day taster workshop (Apps for Good, 2019a) as well as a 12-sessions course (Apps for Good, 2019b).

With respect to the instructional methods, there is a strong predominance of active learning approaches aiming at the achievement of learning objectives on the application level. These range from tasks with a well-defined specification of the tasks for which an expected solution exists to tasks with ill-defined problems without a previously known solution, which aims at a higher cognitive level to take the students to create their own practical solution.

We also encountered a considerable number of IUs using unplugged activities adopting diverse materials for activities teaching mostly data management (partly supported by spreadsheet tools) or decision tree algorithms (e.g., (Curiositymachine, 2019)). Other activities also explore how biology and specifically animal brains can be the inspira-

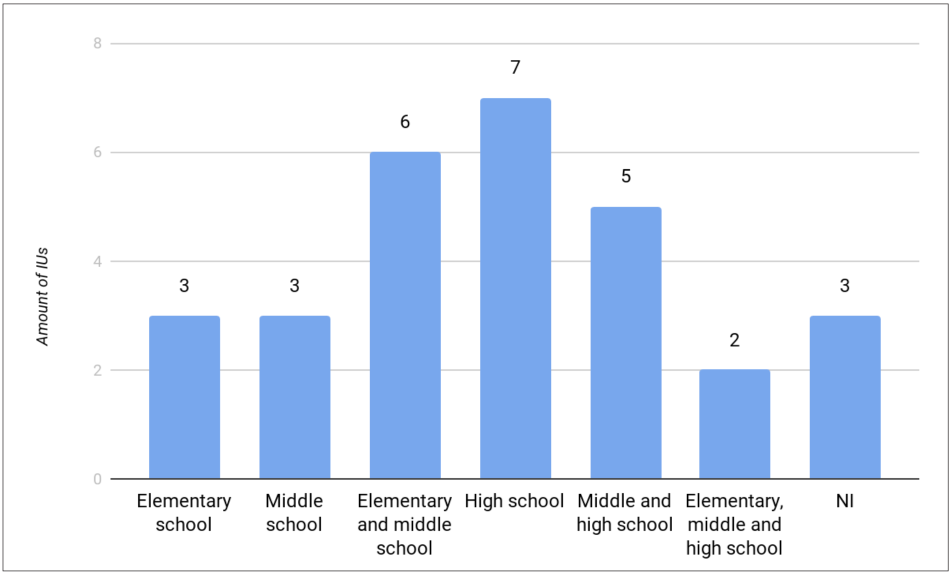

Fig. 9. Frequency of IUs per educational stage.

tion for a new way to program computers using paper cards (CS4FN, 2011). Another unplugged example is "Be the machine" (Fryden curriculum, 2019), a team role-playing game that teaches how ML works, in which each member of the team assumes a different role to manually train an ML model.

Although focusing more on active learning, several IUs also include other direct instructional methods such as lectures, videos, and demonstrations, especially in the initial part of the IU as well as the foundations of neural nets (Fig. 10). Examples include the Digit Classifier Tool, Drawing Completion Tool, Teachable Machine, and Tensorflow Playground. Interactive methods such as challenges and discussions were also used. Apps for Good (2019a) also study cases to achieve an understanding of ML. (Vachovsky *et al.*, 2016) and (Mobasher *et al.*, 2019) also included invited talks with professionals from IT companies and/or field trips in order to amplify the students' perspective on ML.

According to this variety of instructional methods, several types of instructional material are adopted (Fig. 11). Instructional videos, tutorials, etc. are specific to IUs designed as online courses. Several IUs also use worksheets to record the students' experiences. However, in general, we observed a lack of information regarding the instructional material, their availability and license, which makes it difficult for others to use them. With only one exception the materials are available in one language only (predominantly in English), which may also limit a broader adoption of IU in other countries that require instructional material in the native language at this educational stage.

The majority of the IUs does not cover the assessment of the students' learning. Only AI Family challenge (2019) and AIinSchools (2019) propose a rubric/assessment sheet for a performance-based assessment analyzing artifacts created by the students. Sakulkueakulsuk *et al.* (2018) allocate scores based on the accuracy of the ML models developed. As an alternative, AI Family challenge (2019) and Elements of AI (2019) also adopt quizzes or exercises for the students' assessment.

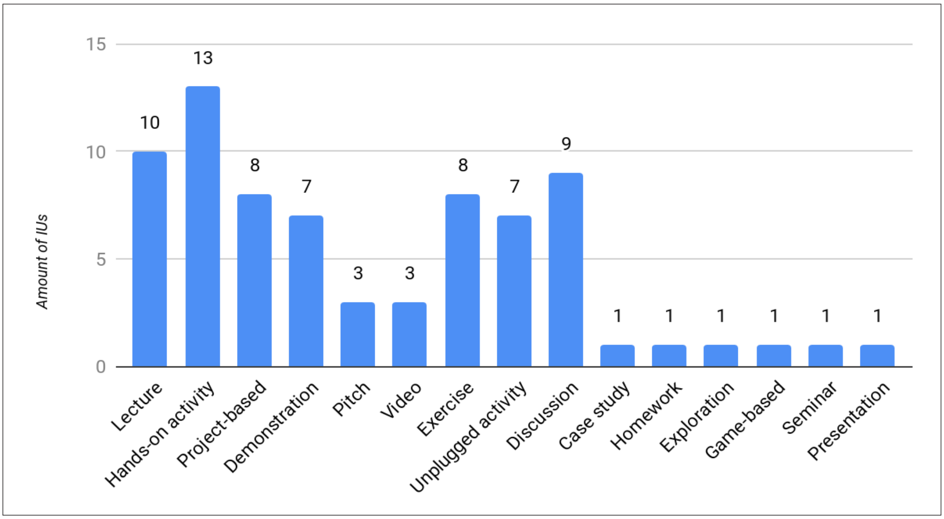

Fig. 10. Instructional methods used for ML education.

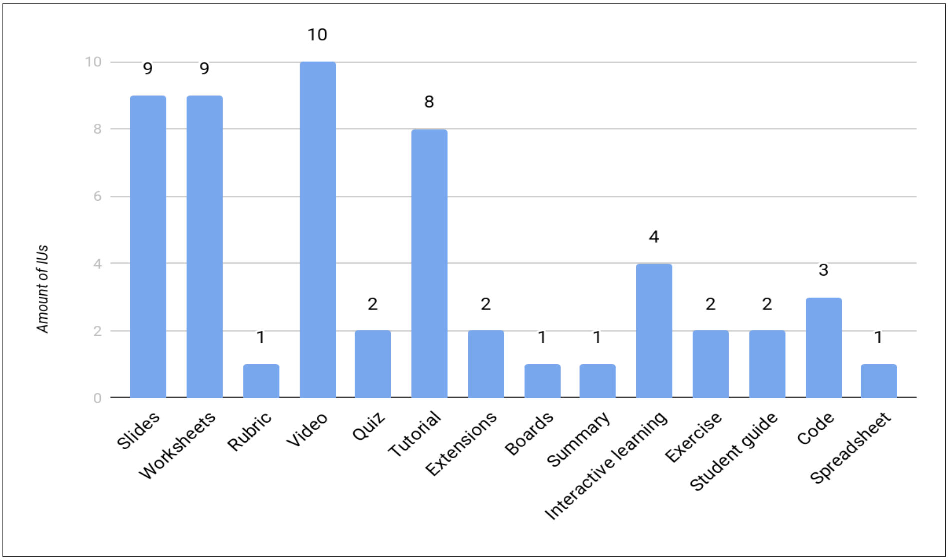

Fig. 11. Types of instructional material used.

4.4. *How Were the IUs Developed and Evaluated?*

To achieve effective learning outcomes, IUs need to be developed systematically following instructional design models. However, we observed a general lack of information in relation to the way the IUs were developed. Very few publications mention any information on this issue. For example, the IU proposed by Hitron *et al.* (2019) is based on prior work in the constructivism school of thought and in cognitive psychology. The IU designed by ReadyAI (2019) is based on the 5 big ideas as being proposed by the AI4K12 guidelines. Sakulkueakulsuk *et al.*, 2018) based the IU on the “Four P’s of Creative Learning” framework developed by MIT Media Lab and the IU designed by Zimmermann-Niefield *et al.* (2019a) is based on Interactive Machine Learning (Fiebrink, 2019). None of the encountered IUs provides more complete information on the methodology used for its development.

Most IUs were evaluated by means of a case study (Fig. 12). In these studies, the evaluation was systematically defined and, during and after the treatment (teaching ML), data was collected in relation to the objective of the evaluation. Only one study adopted a more rigorous research design. Hitron *et al.* (2019) conducted an experiment comparing the students’ understanding in three conditions: learning activity uncovering Data Labeling only, Evaluation only, or both. Two IUs indicate a more informal way of evaluation (ReadyAI, 2019) (Sperling and Lickerman, 2012), without detailed definition. In addition, no information on evaluation was being encountered for a considerable number of IUs.

Most studies evaluate more than one quality factor (Fig. 13). Learning is the most evaluated quality factor. This shows that, in fact, the main concern is the learning ef-

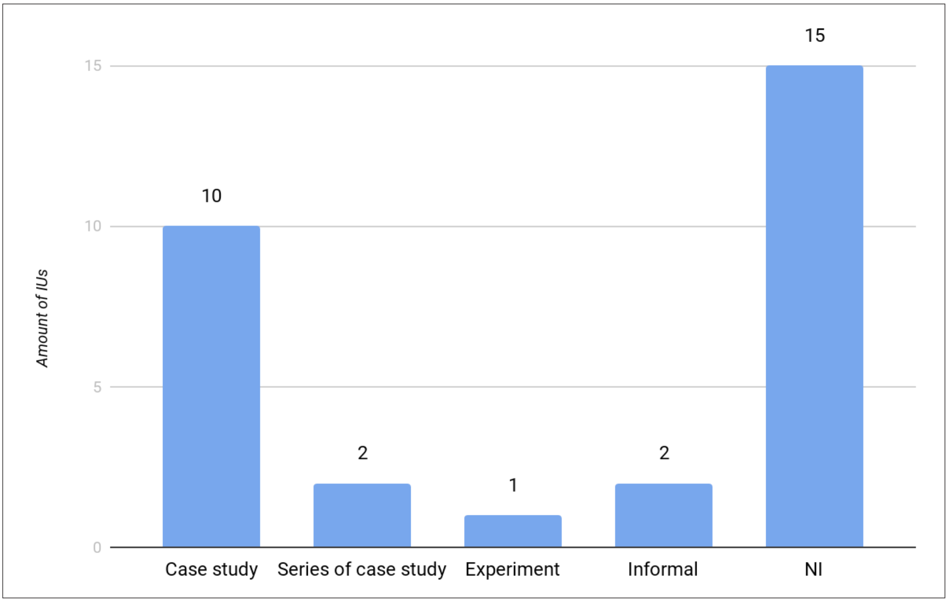

Fig. 12. Types of studies adopted for the evaluation of the IUs.

Fig. 13. Quality factors being evaluated in the studies.

fect provided by the IUs. Several studies also assess the degree of interest in a STEM/computing career motivated by the IU. Besides evaluating the impact of the IUs, several evaluations also included the measurement of feedback on the IU itself as well as the observed strengths and weaknesses.

Data regarding the evaluation is collected in several ways (Fig. 14). Most of the data is collected via questionnaires at the end of the IU. Few studies also extract data based on the performance-based assessment of artifacts created by students during the IU, tests, interviews or observations.

Taking into consideration the less rigorous research designs adopted, most studies only perform qualitative data analyses and/or descriptive quantitative analyses. Only three studies report the usage of statistical tests (Cognimates.me, 2019; Vachovsky *et al.*, 2016; Hitron *et al.*, 2019). Evaluations were performed with samples ranging from 9 to 7500+ participants, but the majority with rather small samples with less than 50 par-

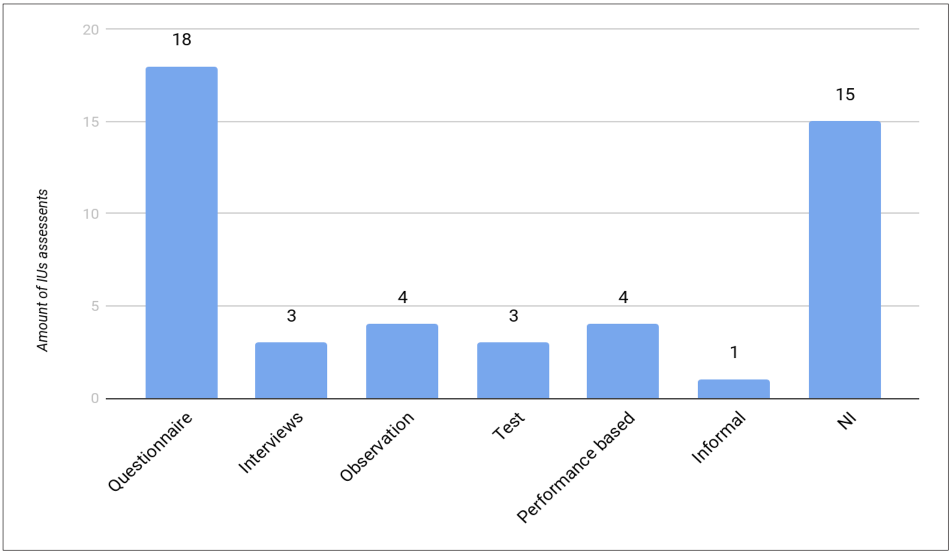

Fig. 14. Data collection methods used for the evaluation of the IUs.

ticipants. Only two studies were replicated: (Cognimates.me, 2019) in several schools worldwide and (Srikant & Aggarwal, 2017) in 4 cities in the US and India.

In general, we observed a lack of information provided on how the IUs were developed and evaluated indicating the need for a more systematic adoption of methods for the development of such instructional units.

5. Discussion

Considering the recentness of ML, we were surprised to encounter already 30 instructional units aiming at teaching ML concepts in schools. Observing, that most of these have been developed in 2019 we also expect this number to further increase in the near future.

These IUs mostly focusing on beginners at any educational stage from elementary to high school also indicates the recognition of an early exposure of students to ML concepts, not limited only to high school as typically indicated by general computing curriculum guidelines.

Being an emergent topic, most of the IUs are proposed as extracurricular units ranging from 1-hour taster workshops to semester-long courses. Providing diverse instructional materials available for free they also facilitate their application in practice. Several IUs also provide customized frameworks and tools in order to teach ML at this educational stage using e.g., block-based programming environments. However, as so far most IUs are only available in English, this may hinder their direct application in other countries. Another issue is an almost complete lack of information on the assessment of the students' learning, which is important as feedback to the learner and instructor in order to guide the learning process.

The IUs teach competencies varying from presenting what is ML, to specific ML techniques as well as the impacts of ML. However, we observed that several IUs present ML concepts only on an abstract level, black-boxing some of the underlying ML processes even as part of hands-on activities in order to reduce complexity. However, in some cases, this high level of black-boxing may limit the students to explore and construct mental models on ML (Hmelo and Guzdial, 1996) as also pointed out by Hitron *et al.* (2019). Therefore, adopting non-black-boxed processes may be imperative to acquire an effective understanding of ML. On the other hand, considering the complexity of ML, it is also important to not overwhelm novice learners (Resnick *et al.*, 2000). Therefore, it will be important to identify a balance between black-boxed processes and uncovered processes as well as a learning sequence based on the complexity of the concepts. As some of the ML concepts seem more accessible than others it seems important to analyze their difficulty using statistical methods such as the Item Response Theory (DeMars, 2010) in order to systematically guide the scaffolding process.

A general strength observed in the encountered IUs is their strong focus on demonstrating the application of ML in practice, typically presenting various application examples in order to gain the attention of the students. Furthermore, several IUs also covers the learning of how to apply ML concepts to practical problems with respect to the most diverse tasks from the context of the students, ranging from the classification of Disney princesses to the feature extraction of mango fruits for classification. However, only a few IUs go so far to guide the students to develop their own ML solution for a problem in the community adopting a computational action approach (Tissenbaum *et al.*, 2019).

In addition, it is possible to observe the existence of a concern with social aspects involved in the application of AI concepts during the practical activities. Some studies lead the student to reflect on the usage of AI in of today's society (Elements of AI, 2019; Tang, 2019). Others address moral issues and the impact of AI on humans (AIinSchools, 2019; Apps for Good, 2019a; ReadyAI, 2019; Touretzky *et al.*, 2019c). Some studies also focus on the democratization of Machine Learning/Artificial Intelligence teaching, in order to impact society not only on content but on the approach used, seeking to involve minorities (Mobasher *et al.*, 2019) (Vachovsky *et al.*, 2016). (Van Brummelen, 2019).

Another issue we observed is the lack of support for the training of instructors in order to prepare them adequately for the application of the IUs in the classroom. Besides a few IUs providing lesson plans and guides no further training is provided as part of the IUs. Taking into account that today there is a lack of K-12 teachers with computing background, most computing education is applied in a multidisciplinary way by teachers trained in other disciplines. Therefore, the motivation and training of in-service teachers become essential for a larger-scale adoption of ML education. This includes not only computing and ML knowledge but also knowledge of relevant pedagogical and technological content.

In general, we observed a lack of systematic presentation of the IUs and the way they were developed and evaluated. As many have not been published as scientific articles, no further information on their impact is available, which leaves the achievement of the learning goals questionable. However, considering the recentness of this topic, we expect more rigorous studies soon observing the large increase of IUs just this year. The systematic development of such IUs will also further supported by the development of curriculum guidelines currently underway.

Threats to validity. Some threats may affect the validity of our mapping study. We, therefore, identified potential threats and applied mitigation strategies in order to minimize their impact. Systematic mappings may suffer from the common bias that positive outcomes are more likely to be published than negative ones. However, we consider that the findings of the articles, whether positive or negative, have only a minor influence on this systematic mapping since we sought to characterize the approaches rather than analyze their impacts on learning.

Another risk is the omission of relevant studies. In order to mitigate this risk, we carefully constructed the search string to be as inclusive as possible, considering not only core concepts but also synonyms. Furthermore, considering the recentness of the topic studies, we also searched for any IU available online, not only considering scientific articles, in order to reduce the risk of excluding existing IUs. On the other hand, our observation that most IUs are available in one language only (predominantly in English), may be due to the fact that based on our search using an English search string only returned IUs available in English.

Threats to the selection of relevant IUs and data extraction were mitigated by providing a detailed definition of inclusion/exclusion and quality criteria. We defined and documented a rigid protocol for the study selection and all authors performed the selection together, discussing the selection until consensus was reached. Data extraction was hindered in some cases, as the relevant information was often not presented explicitly and, therefore, in some cases had to be inferred. However, this inference was made by the first two authors and carefully reviewed by the third author.

6. Conclusion

In this article, we present the state of the art and practice of teaching Machine Learning in elementary to high school. We have identified 30 IUs mainly focused on beginners for any of these educational stages. The results of our review indicate the importance of this topic to the rapid increase of IUs developed this year. Being an emergent topic, most of the IUs are proposed as extracurricular units ranging from 1-hour taster workshops to semester-long courses. The IUs teach competencies varying from presenting what is ML, to specific ML techniques as well as the impacts of ML with an emphasis on artificial neural networks. Observing the complexity of ML concepts, several UIs cover only the most accessible processes, such as data management or cover model learning and testing on an abstract level black-boxing some of the underlying ML processes. The IUs provide diverse instructional materials available for free as well as customized frameworks and tools in order to teach ML at this educational level, using e.g., block-based programming environments as well as Python and general ML frameworks. As a result of our study we, thus, expect to contribute to the mapping of these emergent IUs, facilitating the teaching of ML in practice. However, observing a lack of teacher training and more information on the development and evaluation of these IUs, it also becomes obvious that there is a need for further research in this area.

Acknowledgments

This work is supported by CNPq (National Council for Scientific and Technological Development), a Brazilian government entity focused on scientific and technological development.

References

Agassi, A., Erel, H., Wald, Y., Zuckerman, O. (2019). Scratch Nodes ML: A Playful System for Children to Create GestureRecognition Classifiers. *Proc. of the Conference on Human Factors in Computing Systems, Glasgow*, Scotland, UK.

AI4ALL. (2019). *Build Your Own AI Project for Social Good*. `http://ai-4-all.org/open-learning/`

AI4ALL. (2018). *Open Learning brings free and approachable AI education online with the support of Google. org*. `https://medium.com/ai4allorg/ai4all-open-learning-brings-free-and-accessible-ai-education-online-with-the-support-of-google-org-3a6360c135c9`

Ali, S. *et al.* (2019). Constructionism, Ethics, and Creativity: Developing Primary and Middle School Artificial Intelligence Education. In: *Proc. of the Int. Workshop on Education in Artificial Intelligence in K-12*, Palo Alto, CA, USA.

Amazon. (2019). *Amazon Machine Learning*, AWS Documentation. `https://docs.aws.amazon.com/machine-learning/latest/dg/building-machine-learning.html`

Bau, D., Gray, J., Kelleher, C., Sheldon, J., Turbak, F. (2017). Learnable programming: blocks and beyond. *Communications of the ACM*, 60(6), 72–80.

Bemley, J.L. (1999, July). Neural networks for precollege students. In: *IJCNN'99. International Joint Conference on Neural Networks*, 6, 4422–4427.

Bennett, B.T. (2017). Teaching artificial intelligence in a multidisciplinary computing environment. Journal of *Computing Sciences in Colleges*, 33(2), 222–228.

Bloom, B.S., Engelhart, M.D., Furst, E.J., Hill, W.H., Krathwohl, D.R. (1956). Taxonomy of educational objectives: The classification of educational goals. *Handbook I: Cognitive domain*. David McKay Company, New York.

Branch, R. (2009). *Instructional design: The ADDIE approach*. 2nd ed. Athens: Springer.

da Cruz Pinheiro, F., Gresse von Wangenheim, C., & MissfeldtFilho, R. (2018). Teaching Software engineering in K-12 education: A systematic mapping study. *Informatics in Education*, 17(2), 167–206.

CSTA. (2017). *CSTA K-12 Computer Science Standards*. `http://www.csteachers.org/standards`

CS4Fn. (2019). *Emotional Robot*. `http://www.cs4fn.org/teachers/activities/emotionalrobot/emotionalrobot.pdf`

CSUnplugged. (2015). *The Turing Test*. `https://classic.csunplugged.org/the-turing-test/`

DeMars C. (2010). *Item Response Theory*. Cary, NC: Oxford University Press.

Evangelista, I., Blesio, G., Benatti, E. (2018). Why are we not teaching machine learning at high school? A proposal. In: *Proc. of the World Engineering Education Forum – Global Engineering Deans Council*, Albuquerque, NM, USA.

Estevez, J., Garate, G., Lopez-Guede, J. M., Graña, M. (2019). *Using Scratch to Teach Undergraduate Students' Skills on Artificial Intelligence*. arXiv:1904.00296 [cs.AI].

Fiebrink, R. (2019). Machine learning education for artists, musicians, and other creative practitioners. *ACM Transactions on Computing Education*, 19(4).

Forbes.(2019).AIgoestohighschool.`https://www.forbes.com/sites/insights-intelai/2019/05/22/ai-goes-to-high-school/#68826e3f1d0c`

Freedman, D. *et al.* (2007). *Statistics*. WW Norton, New York.

Goodfellow, I., Bengio, Y., Courville, A. (2016). *Deep Learning*. Cambridge: MIT Press.

Google. (2017). *Teachable Machine*. `https://experiments.withgoogle.com/teachable-machine`

Google. (2016). *Trends in the State of Computer Science in U.S. K-12 Schools*. `http://services.google.com/fh/files/misc/trends-in-the-state-of-computer-science-report.pdf`

Grover, S., Pea, R. (2013). Computational Thinking in K–12: A Review of the State of the Field. *Educational Researcher*, 42(1), 38–43.

Haddaway, N.R. *et al.* (2015). The role of Google Scholar in evidence reviews and its applicability to grey literature searching. *PloS one*, 10(9).

Hayes, B. (2019). *Most Popular Machine Learning Frameworks and Products Used by Data Professionals*. http://businessoverbroadway.com/2019/01/27/most-popular-machine-learning-frameworks-and-products-used-by-data-professionals/

Heintz, F. *et al.* (2016). A Review of Models for Introducing ComputationalThinking, Computer Science and Computing in K–12 Education. *Proc. of IEEE Frontiers in Education Conference*, Erie, PA, USA, pp. 1–9.

Hmelo, C. E., Guzdial, M. (1996). Of black and glass boxes: Scaffolding for doing and learning. *Proc. of the Int. Conference on Learning Sciences*, Evanston, IL, USA.

Hill, H., Rowan, D., Ball, D. (2005). Effects of teachers' mathematical knowledge for teaching on student achievement. *American Educational Research Journal*, 42(2), 371–406.

Hiner, J. (2017). *AI will eliminate 1.8M jobs but create 2.3M by 2020, claims Gartner*. https://www.techrepublic.com/article/ai-will-eliminate-1-8m-jobs-but-create-2-3m-by-2020-claims-gartner

Hubwieser, P. *et al.* (2015). A Global Snapshot of Computer Science Education in K-12 Schools. *Proc. of the ITiCSE on Working Group Reports*, Vilnius, Lithuania.

ISTE. (2018). *Bold New Program Helps Teachers and Students Explore the Power of AI*. https://www.iste.org/explore/Press-Releases/Bold-New-Program-Helps-Teachers-and-Students-Explore-the-Power-of-AI

Jing, M. (2018). *China looks to school kids to win the global AI race*. https://www.scmp.com/tech/china-tech/article/2144396/china-looks-school-kids-win-global-ai-race

Kahn, K. (1977). Three Interactions between Al and Education. *Machine Intelligence*, 8.

Kandlhofer, M., Steinbauer, G., Hirschmugl-Gaisch, S., Huber, P. (2016). Artificial Intelligence and Computer Science in Education: From Kindergarten to University. *Proc. of IEEE Frontiers in Education Conference*, Erie, PA, USA.

Lye, S. Y., Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51–61.

Makeblock. *Make with Code*. (2019). https://www.mblock.cc/en-us/

MIT. (2019). *AI + Ethics Curriculum for Middle School project*. https://www.media.mit.edu/projects/ai-ethics-for-middle-school/overview/

McGovern, A., Tidwell, Z., Rushing, D. (2011). Teaching Introductory Artificial Intelligence through Java-Based Games. Proc. of the 2nd Symposium on Educational Advances in Artificial Intelligence, San Francisco, CA, USA.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill Education, New York.

ModelZoo. (2019). https://modelzoo.co

Papert, S., Solomon, C.(1971). Twenty things to do with a computer. Artificial Intilligence Memo Number 248. Massachusetts Inst. of Tech.

Park, Y. J., Ro, H., Han, T-D. (2019). Deep-ChildAR bot: educational activities and safety care augmented reality system with deep-learning for preschool. *Proc. of the ACM SIGGRAPH Conference*, Los Angeles, CA, USA.

Parsons, S., Sklar, E. (2004). Teaching AI using LEGO Mindstorms. *Proc. of the AAAI Spring Symposium*, Palo Alto, CA, USA.

Petersen, K. *et al.* (2008) Systematic mapping studies in software engineering. *Proc. of the 12th International Conference on Evaluation and Assessment in Software Engineering*, Bari, Italy, 68–77.

Resnick, M., Berg, R., Eisenberg, M. (2000). Beyond black boxes: Bringing transparency and aesthetics back to scientific investigation. *The Journal of the Learning Sciences*, 9(1), 7–30.

Ripley, B. (2008). *Pattern Recognition and Neural Networks*, University Press, Cambridge.

Royal Society. (2017). *Machine learning: the power and promise of computers that learn by example*. https://royalsociety.org/~/media/policy/projects/machine-learning/publications/machine-learning-report.pdf

Shadish, W. R. *et al.* (2002). *Experimental and quasi-experimental designs for generalized causal inference*. Houghton Mifflin Company, New York.

Tissenbaum, M., Sheldon, J., Abelson, H. (2019). From Computational Thinking to Computational Action. *Communications of the ACM*, 62(3), 34–36.

Tharwat, A. (2019). Classification assessment methods. *Applied Computing and Informatics*, in press.

Torrey, L. 2012. Teaching Problem-Solving in Algorithms and AI. *Proc. of the 3rd Symposium on Educational Advances in Artificial Intelligence*, Toronto, Ontario, Canada.

Tricon Infotech. (2019). *Top Programming Languages for Machine Learning in 2019*. `https://medium.com/@triconinfotech/top-programming-languages-for-machine-learning-in-2019-48d4640a4f37`

Touretzky, D. S., Gardner-McCune, C., Martin, F., Seehorn, D. (2019a). K-12 Guidelines for Artificial Intelligence: What Students Should Know. *Proc. of the ISTE Conference*, Philadelphia, PA, USA.

Touretzky, D. S., Gardner-McCune, C., Martin, F., Seehorn, D. (2019b). Envisioning AI for K-12: What Should Every Child Know about AI?. *Proc. of the 33rd AAAI Conference on Artificial Intelligence*, Honolulu, Hawaii, USA.

Touretzky, D. S., Gardner-McCune, C., Huang, J. (2019c). K-12 AI Playground. *Proc. of the CSTA Annual Conference*, Philadelphia, PA, USA.

Weintrop, D. (2019). Block-based Programming in Computer Science Education. *Communications of the ACM*, 62(8), 2019, 22–25.

Williams, R., Park, H. W., Oh, L., Breazeal, C. (2019a). PopBots: Designing an Artificial Intelligence Curriculum for Early Childhood Education. *Proc. of the 33rd AAAI Conference on Artificial Intelligence*, Honolulu, Hawaii, USA.

Williams, R., Park, H. W., Breazeal, C. (2019b). A is for Artificial Intelligence: The Impact of Artificial Intelligence Activities on Young Children's Perceptions of Robots. *Proc. of the CHI Conference on Human Factors in Computing Systems*, Glasgow, UK.

Wohlin, C. *et al.* (2012). *Experimentation in Software Engineering*. Springer-Verlag, Berlin.

Wollowski, M. *et al.* (2016). A Survey of Current Practice and Teaching of AI. *Proc. of the 30th AAAI Conference on Artificial Intelligence*, Phoenix, AZ, USA.

L.S. Marques is an undergraduate student of the Information System course at the Federal University of Santa Catarina (UFSC) and a scholarship student at the initiative Computing at Schools/INCoD/INE/UFSC.

C. Gresse von Wangenheim is a professor at the Department of Informatics and Statistics (INE) of the Federal University of Santa Catarina (UFSC), Florianópolis, Brazil, where she coordinates the Software Quality Group (GQS) focusing on scientific research, development and transfer of software engineering models, methods and tools and software engineering education. She also coordinates the initiative Computing at Schools that aims at bringing computing education to schools in Brazil. She received the Dipl.-Inform. and Dr. rer. nat. degrees in Computer Science from the Technical University of Kaiserslautern (Germany), and the Dr. Eng. degree in Production Engineering from the Federal University of Santa Catarina. She is also PMP – Project Management Professional and MPS.BR Assessor and Implementor.

J.C.R. Hauck holds a PhD in Knowledge Engineering and a Master's Degree in Computer Science from the Federal University of Santa Catarina (UFSC) and a degree in Computer Science from the University of Vale do Itajaí (UNIVALI). He held several specialization courses in Software Engineering at Unisul, Univali, Uniplac, Uniasselvi, Sociesc and Uniarp. He was a visiting researcher at the Regulated Software Research Center – Dundalk Institute of Technology – Ireland. He is currently a Professor in the Department of Informatics and Statistics at the Federal University of Santa Catarina.

Appendix A.

Content covered in the instructional units

Reference	Approaches to machine learning								Types of learning algorithms by learning style								
	What is learning?	Regression algorithms	Instance-based	Decision tree	Bayesian algorithms	Clustering	Neural network	Other	Supervised	Unsupervised	Semi-supervised	Reinforcement	Fundamentals of neural network	How training data influences learning	Limitations of machine learning	Ethical concerns/societal implications	Career opportunities
(AI Family Challenge, 2019)	x						x								x	x	
(ai4childre, 2017)							x										
(AIinSchools, 2019)	x						x						x		x	x	
(Apps for Good, 2019a)	x						x		x				x	x	x	x	
(Apps for Good, 2019b)	x	x		x	x	x	x		x	x			x	x	x	x	x
(Burgsteiner *et al.*, 2016; Burgsteiner, 2016)				x			x					x					
(Cognimates, 2019)							x		x								
(CS4FN, 2011)	x												x				
(Curiositymachine, 2019)	x			x			x										
(Elements of AI, 2019)	x	x	x				x						x			x	
(Essinger & Rosen, 2019)						x								x			
(Evangelista *et al.*, 2018)				x		x		Bagging/random forest						x	x	x	
Fryden curriculum	x						x						x				
(Hitron *et al.*, 2019)	x							Dynamic Time Warping (DTW) algorithm									
(Ho & Scadding, 2019)	x		x														
(Kahn & Winters, 2018) (Kan *et al.*, 208)							x						x		x	x	
(MIT, 2019)	x						x										
(ML4Kids, 2019)				x			x		x			x					
(Mobasher *et al.*, 2019)	x		x	x													x
(Narahara & Kobayashi., 2018)							x										
(ReadyAI, 2019)	x			x												x	
(Sakulkueakulsuk et.al, 2018)	x		x	x			x										
(Sperling & Lickerman, 2012)	x		x	x				LMS algorithm, Genetic algorithms									
(Srikant & Aggarwal, 2017)				x					x								
(Tang, 2019; Tang *et al.*, 2019)	x						x		x								
(Techgirlz, 2018)	x		x				x						x				
(Vachovsky *et al.*, 2016)			x		x			Background subtraction algorithm									x
(Van Brummelen, 2019) (Van Brummelen & Abelson, 2019) (Van Brummelen *et al.*, 2019)	x						x		x								
(Zhu, 2019)	x	x	x			x	x	Long short-term memory (LSTM), Ensemble learning (Random forest)	x	x	x	x					
(Zimmermann-Niefield *et al.*, 2019a) (Zimmermann-Niefield *et al.*, 2019b)								Dynamic Time Warping (DTW) algorithm						x			

Characteristics of ML content

Reference	Learning objective	Application domain	Type of ML task	ML processes	Dataset	ML frameworks/ tools	Programm-ing language(s)	Performance measure(s)	Deployment
(AI Family Challenge, 2019)	To give everyone the chance to learn, play and create with AI.	Computer vision (image classification) sentiments analysis (tweets)	Classification	Data management, Model learning,	Students collect positive/neutral/ negative words.	Cognimates/ Codelab, ML4K, tensorflow	Scratch	NI	NI
(ai4children, 2017)	To introduces machine learning.	Computer vision (image classification), game, chatbot	Classification	NI	NI	ScratchX	Scratch	NI	NI
(AIinSchools, 2019)	To learn terminology around Artificial Intelligence and reinforce computational thinking concepts. Additionally, they will learn how AI is used in specific areas and they will design, test and evaluate their own AI system based on Neural Network principles and also consider the future of AI.	Image recognition	Classification	Data management, training the system and test it	Students use images from the web against those from a known dataset to accurately identify images.	AWS cloud server, DIGITS	NI	Charts/graphics/ percentages which give information on the accuracy of the image recognition	NI
(Apps for Good, 2019a)	To understand what ML is; to understand potential issues with the application of ML; be able to design a machine learning model prototype to solve a real world problem; be able to evaluate the impact of ethical considerations in the application of ML.	Facial recognition, self-driving cars, sentiment analysis	Classification	Data management, Model learning, Model evaluation Deployment	Examples of nice and mean expressions collected by the students.	IBM Watson Assistant	Scratch	Confidence score	Deployed into a Scratch program.
(Apps for Good, 2019b)	To understand what ML is; be able to build a simple ML model.	Facial recognition, NLP (chatbot), Recommendation systems, self-driving cars	Classification, Clustering	Data management, Model learning, Model evaluation	NI	IBM Watson Assistant	Scratch, Python	Accuracy	Deployed into a Scratch program.
(Burgsteiner *et al.*, 2016) (Burgsteiner, 2016)	To introduce ML in the context of robotics.	NI	NI	NI	NI	NI	NI	NI	NI
(Cognimates, 2019)	To program and customize embodied intelligent devices.	Computer vision, sentiment analysis, games, robotics	NI	NI	NI	Codelab, Clarifai	Scratch	NI	Deployed as robots, games, etc.

Continued on next page

Table continued from previous page

Reference	Learning objective	Application domain	Type of ML task	ML processes	Dataset	ML frame-works/tools	Programm-ing language(s)	Performance measure(s)	Deployment
(CS4FN, 2011)	to give an understanding of how neurons work and how this architecture may be copied to create an artificial neural network that can lead to human-like behavior.	Game	NI	Model learning	NI	NI	NI	NI	NI
(Curiositymachine, 2019)	To understand neural network basics.	NI	Classification	Data management, feature extraction model evaluation	Any object (including small toys and stationary objects (pen, scissors, etc.)	NI	NI	NI	NI
(Elements of AI, 2019)	To encourage as broad a group of people as possible to learn what AI is, what can (and can't) be done with AI, and how to start creating AI methods.	Computer vision	NI	NI	Data on shopping history, life expectancy data	TensorFlow.js	NI	NI	NI
(Essinger & Rosen, 2019)	To understand how to instruct a computer to identify the difference between objects.	Signal processing (recycling sorting, bacteria detection)	Classification	Model learning, model evaluation	NI	Spreadsheets	NI	Accuracy	NI
(Evangelista *et al.*, 2018)	To prove that ML aims at providing a model for understanding nature based on data; Make students understand that ML is about pattern recognition, mathematics, algorithms and statistics; Provide insight on the most important concepts in ML; Show that there are similarities and differences in the way computers and people "learn".		Clustering	Data management, model training, model evaluation	Data on cookie preferences (size of cookies and number of chocolate chips)				
(Fryden curriculum, 2019)	To learn how thresholds move with training sets.	Color classification	Classification	NI	NI	NI	NI	NI	NI
(Hitron *et al.*, 2019)	To promote the understanding of ML concepts (data labeling aspects).	Tennis gestures recognition	Classification	Data management, model evaluation	NI	Gesture Recognition Toolkit (GRT)	NI	NI	NI
(Ho & Scadding, 2019)	To demystify AI by showing that it can be thought of different ways in which a computer simulates human-like behaviors.	Computer vision (facial recognition), robotics (self-learning lawn bowling robot)	Classification	Data management, feature engineering, model learning	Paper images of Disney princesses	Lego Mindstorms EV3 kit	NI	NI	Deployed with Lego Mindstorms as a selflearning lawn bowling robot.

Continued on next page

Table continued from previous page

Reference	Learning objective	Application domain	Type of ML task	ML processes	Dataset	ML frame-works/tools	Programm-ing language(s)	Performance measure(s)	Deployment
(Kahn & Winters, 2018) (Kahn *et al.*, 2018)	To build AI programs with Snap!	Computer vision, speech recognition, games	NI	Data management, model training, model evaluation	User collected data	TensorFlow.js	Snap!	Loss function, accuracy	Deployed as a game etc.
(MIT, 2019)	To learn about the basics of machine learning and create their own apps that implement these concepts through image classification.	Computer vision (object classification)	Classification	Deployment	Look Extension has been pre-trained to recognize 999 classes	Mobilenet	App Inventor, App Inventor extensions	Confidence	Deployed as App Inventor app.
(ML4Kids, 2019)	To introduces machine learning.	Computer vision (facial recognition, ob-ject classification, spe-ech recognition, handwriting recognition, sound recognition, prediction, sentiment analysis, text classification, recommendation system, digital assistant, games	Classification	Data management, model learning, model evaluation, deployment	Image, audio, text, numbers	IBM Watson	App Inventor, Scratch, Python	Confidence score, confidence threshold, confidence percentage, comparison	Deployed as an app, a game, a chatbot, etc.
(Mobasher *et al.*, 2019)	To get students excited about pursuing data science in college and careers, to teach basic data science concepts, tools & techniques, to introduce students to the wide variety of domain applications and career possibilities in data science.	Computer vision (facial recognition), recommendation system	Classification	Data management, model learning, model evaluation	Divy (bicycle ride data), Human Activity Recognition dataset, ATT face recognition dataset (for image classification) and Spotify song feature data set.	Matlab, SPSS, Tableau. pandas, scikit-learn, Jupyter Notebook	Python	NI	NI
(Narahara & Kobayashi., 2018)	To introduce ideas in AI and robotics.	Robotics (toy car with a camera controlled by Raspberry Pi)	NI	Data management, model learning, model evaluation, deployment	NI	Tensorflow	Python	NI	NI

Continued on next page

Table continued from previous page

Reference	Learning objective	Application domain	Type of ML task	ML processes	Dataset	ML frame-works/tools	Programm-ing language(s)	Performance measure(s)	Deployment
(Sakulkuea-kulsuk *et al.*, 2018)	To learn the process of making AI with the real-world context focusing on a social agriculture issue.	Prediction	Classification	Data manage-ment, feature engineering, model learning, model evaluation	Students collect data on features of real fruits (ma-ngos) (skin color, texture etc.)	NI	RapidMiner	Accuracy	NI
(Sperling & Lickerman, 2012)	To provide a background of ML.	NI	NI	NI	NI	NI	DrRacket functional programm-ing language	NI	NI
(Srikant & Aggarwal, 2017)	To understand and apply the full cycle of a typical supervised learning approach.	Computer vision (Friend predictor)	Classification	data managem-ent, feature engi-neering, model learning model evaluation	Flash-cards, each with a drawing of a face (male/ female), a name (old-fashioned/ modern) and hob-by (sports or non-sport))	Spreadsheets	NI	Accuracy	NI
(Tang, 2019) (Tang *et al.*, 2019)	To introduce ML to the students and to have them to build a model for recognizing various facial expressions.	Computer vision	Classification	Data manage-ment, model learning, model evaluation	Students collect pictures of diffe-rent objects using a webcam.	MobileNet and SqueezeNet, TensorFlow.js	App Inventor, App Inventor extension (PIC)	Correctness table and confidence graph	Deployment in App Inventor apps.
(Techgirlz, 2018)	To explain what ML is and how it is different from traditional programming; to describe on a high-level how k-nearest neighbors and neural networks operate; to use a Python machine learning library to implement and test simple machine learning techniques.	Classification, computer vision (cats)	Classification	Model learn-ing, model evaluation,	NI	Colab/Jupyter notebook	Python	Accuracy	NI
(Vachovsky *et al.*, 2016)	To increase interest in AI and contextualize technically rigorous AI concepts through societal impact, and address barriers that could discourage 10thgrade girls from pursuing computer science.	Computer vision (hospital hygiene), NLP (disaster relief), computational biology (DNA), control system (self-driving cars)	Classification	Dataset manage-ment, model learning, model evaluation,	NI	Jupyter notebook	Python	F-score metric	NI

Continued on next page

Table continued from previous page

Reference	Learning objective	Application domain	Type of ML task	ML processes	Dataset	ML frame-works/tools	Programm-ing language(s)	Performance measure(s)	Deploy-ment
(Van Brummelen, 2019) (Van Brummelen & Abelson, 2019; Van Brummelen *et al.*, 2019	To teach high school students to create conversational AI applications.	Speech recognition	(long-short term memory (LSTM) neural network)	Model learning, model evaluation, deployment	Pretrained long-short term memory (LSTM) neural network (on Dr. Seuss books, Alice, Nancy Drew, Narnia, Tom Sawyer, Wizard of Oz, Harry Potter).	Node.js environment, based on an LSTM example in the TensorFlow.js GitHub repository	App Inventor, App Inventor extension for Conversational AI	NI	NI
(Zhu, 2019)	To introduce students to what machine learning can do and allow them to build powerful applications.	Computer vision, games, OCR/NLP	Object detection, Speech processing	NI	Users provide samples of webcam data; input images, and the extension outputs the text in the image, along with the bounding boxes of each word and each word's confidence, set of labeled movie reviews, etc.	MobileNet, Tensorflow.js, Google's Teachable Machine, Tesseract.js	App Inventor, App Inventor extensions	NI	Deployed as App.Inventor app
(Zimmermann-Niefield *et al.*, 2019a) (Zimmermann-Niefield *et al.*, 2019b)	To build ML models of their own actions that provide real-time feedback so they can grow more attuned to their athletic form.	Computer Vision (Sports)	Classification	Data management, model learning, model evaluation, deployment	Users associate labels with acceleration segments to form a training set for their model captured from wearable sensors.	AlpacaML	NI	Accuracy and validity	Deployed via AlpacaML

Characteristics of the instructional units

Reference	Type of IU	Educational stage	Level of difficulty	Duration	Instructional mode	Instructional method	Instructional material	Assessment method	Idiom	License	Instructor training
(AI Family Challenge, 2019)	Challenge	Families with children 8-15 years	Novice	10 lessons each about 2 hours	Presencial	Lecture, handson activity, project-based activity, demonstration, pitch	Slides, recordings, worksheet, rubric, videos, quiz, online tutorial, etc.	Quiz, Judge rubric	English	Free	Online mentor training
(ai4children, 2017)	Tutorial	NI	NI	NI	Presencial	Video, demonstration, exercise, hands-on activity	Scratch extensions	NI	English	Free	Lesson plan
(AIinSchools, 2019)	Course	Elementary school (Year 9-13/14 ages)	Novice	6 lessons- 1 hour each	Presencial	Lecture, unplugged activity, exercise, discussion, video	Slides, worksheet	Assessment sheet	English	Free	Teacher guide
(Apps for Good, 2019a)	Workshop	Middle and high school (and further education)	Novice	5 hours	Presencial	Case study, demonstration, discussion, exercises, project-based activity, pitch	Worksheet, videos, slides	NI	English	Creative Commons	Teacher guide
(Apps for Good, 2019b)	Course (in-class or extracurricular)	Elementary and middle school (and Further Education)	Novice	12 sessions	Presencial	Demonstration, discussion, exercise, project-based activity, interaction with industry experts, pitch	Worksheet, checklist, video, slides	NI	English	Creative Commons	Teacher guide
(Burgsteiner *et al.*, 2016) (Burgsteiner, 2016)	Course	High school	Intermediate	seven weekly classes (2 hours each)	Presencial	paper-and-pencil or programming exercises, robot construction, discussions, homework, video	Exercise sheet	NI	NI	NI	NI
(Cognimates, 2019)	Tutorial	Children (aged 7-12)	Novice	varied	Presencial	Exploration, discussion, hands-on activity, project-based activity	Video, tutorial, worksheet	NI	English	NI	NI
(CS4FN, 2011)	Activity	Elementary and middle School	Novice	50min	Presencial	Unplugged activity	Boards, worksheet	NI	English	Free	NI
(Curiosity-ma-chine, 2019)	Activity	Elementary school (4th grade)	Intermediate	NI	NI	Unplugged activity	Slides, summary, video	NI	English	Free	NI
(Elements of AI, 2019)	Course	NI	Novice	6 weeks course (6 chapters each of 5-10 hours)	Online	Tutorial, exercise	Tutorial, exercise	Exercise answers	English and Finnish	Free	NI

Continued on next page

Table continued from previous page

Reference	Type of IU	Educational stage	Level of difficulty	Duration	Instructi-onal mode	Instructional method	Instructional material	Assessment method	Idiom	License	Instructor training
(Essinger & Rosen, 2019)	Activity	Middle and high	Novice	NI	Presencial	Hands-on activity, exercise, discussion	NI	NI	English	NI	NI
(Evangelista *et al.*, 2018)	Course	High school	Novice	4 sessions	Presencial	NI	NI	NI	NI	NI	NI
Fryden curriculum	Course	Middle School	Novice	NI	Presencial	Tutorial, Unplugged role playing game	Slides, game	NI	English	NI	NI
(Hitron *et al.*, 2019)	Workshop	Children aged 10-13 (Elementary and middle school)	Novice	NI	Presencial	Hands-on activity	Interactive learning system	NI	NI	NI	NI
(Ho & Scad-ding, 2019)	Activity	Elementary school (grade 6)	NI	NI	Presencial	Unplugged activity, exercise, game-based learning, discussion	Video,	NI	English	NI	NI
(Kahn & Winters, 2018) (Kahn *et al.*, 2018)	Tutorial	Children (and non-expert programm-ers)	Novice	NI	Online	Tutorial	Tutorials, exercise, project idea, video	NI	English	Creative commons	Teacher guide
(MIT, 2019)	Curriculum unit	Middle school (Gra-de 6-8) and High school (Grade 9-12)	Novice	2 lessons each 45 min	Presencial	Lecture, discussion, demonstra-tion, hands-on activity	Slides, tutorial, student guide, code, video	Multiple choice test, students self-assessment	English	Creative Commons Attribution-ShareAlike	Teacher guide, Lesson plan
(ML4Kids, 2019)	Tutorial	NI	Novice, interme-diate and advanced	45 min - 1 hour	Presencial	hands-on activity	Student guide	NI	English	Free	Teacher notes (material)
(Mobasher *et al.*, 2019)	Summer camp	High school (under-represented groups in computing by tar-geting economica-lly dis-advantaged, African Ameri-can, Hispanic, and female students)	Novice	One week	Presencial	Lecture, hands-on activity, invited talks	Slides, code	NI	English	Github	NI
(Narahara & Kobayashi., 2018)	Activity	All ages	Novice	NI	Presencial	NI	NI	NI	NI	NI	NI

Continued on next page

Table continued from previous page

Reference	Type of IU	Educational stage	Level of difficulty	Duration	Instructional mode	Instructional method	Instructional material	Assessment method	Idiom	License	Instructor training
(ReadyAI, 2019)	Course	K-12	Novice	1 hour	Online	Tutorial	Interactive tutorial, quiz	Quiz	English	Free	NI
(Sakulku-eakul-suk et.al, 2018)	Workshop	Middle school (grade 7 to 9)	Novice	3 days	Presencial	Lecture, Project-based activity, hands-on activity	NI	Scores from the challenges based on the accuracy of the models developed	NI	NI	NI
(Sperling & Lickerman, 2012)	Curricular unit (software engineering course)	High school	Novice	450 hours (100 hours on ML)	Presencial	Lectures, project-based activity	NI	NI	NI	NI	NI
(Srikant & Aggarwal, 2017)	Workshop	Children (10-15 years old)	Novice	4 hours	Presencial	hands-on activity	Flashcards, forms, spreadsheets, worksheet	NI	English	Public github	NI
(Tang, 2019) (Tang *et al.*, 2019)	Workshop	High school	Novice to ML but with prior App Inventor experience	Two lessons of 50 min each	Presencial	Hands-on activity	Tutorial	NI	English	Public	NI
(Techgirlz, 2018)	Workshop	Middle school girls	Intermediate	3 hours	Presencial	Lecture, demonstration, unplugged activity	Slides, video, resource sheet, programming notebook	NI	English	CreativeCommons Attribution-NonCommercial-ShareAlike	Lesson plan
(Vachovsky *et al.*, 2016)	Summer camp	High school girls	Novice	Two weeks	Presencial	Lecture, hands-on activity, seminar, discussion, social hour, field trip, personal growth session	Code	NI	English	NI	NI

Continued on next page

Table continued from previous page

Reference	Type of IU	Educational stage	Level of difficulty	Duration	Instructional mode	Instructional method	Instructional material	Assessment method	Idiom	License	Instructor training
(Van Brummelen, 2019) (Van Brummelen & Abelson, 2019; Van Brummelen *et al.*, 2019	Workshop	Middle and high school	NI	6 sessions each of 1 hour	Presencial	Lecture, unplugged activity, pro-ject-based activity, presentation	Worksheet, tutorial	NI	English	NI	NI
(Zhu, 2019)	Course	High school	Advanced	6 classes (each 1:30h)	Presencial	Lecture, hands-on activity	Tutorial, App Inventor extensions	NI	English	Public	NI
(Zimmermann-Niefield *et al.*, 2019a) (Zimmermann-Niefield *et al.*, 2019b)	Workshop	Middle and high school	Novice	3 hours	Presencial	Tutorial, demonstration, project-based activity	Interactive tutorial	NI	English	NI	NI