Comparison of Adaptive Instructional Strategies in Interactive Problem-Solving Environment for Programmable Logic Controller and Robot Interfacing

Dr. Sheng-Jen Hsieh, Texas A&M University

Dr. Sheng-Jen (”Tony”) Hsieh is a Professor in the Department of Engineering Technology and Industrial Distribution and a member of the Graduate Faculty at Texas A&M University, College Station, TX. His research interests include automation, robotics, cyber-manufacturing and Industry 4.0; optical/infrared imaging and instrumentation; micro/nano manufacturing; and design of technology for engineering education. He is also the Director of the Rockwell Automation Laboratory at Texas A&M University, a state-of-the-art facility for education and research in the areas of automation, robotics, and Industry 4.0 systems. He was named Honorary International Chair Professor for National Taipei University of Technology in Taipei, Taiwan, for 2015-21. Dr. Hsieh received his Ph.D. in Industrial Engineering from Texas Tech University, Lubbock, TX.

Cheng Niu, Texas A&M University

Interactive Simulation Tool to Enhance Understanding and Programming of Single Layer Perceptron Machine Learning Algorithm for High School Computer Science Education

Abstract

This paper describes the development and evaluation of an interactive simulation tool to teach high school computer science students about the Machine Learning (ML) process through understanding and implementing a Single Layer Perceptron to perform binary classification of images of numerical digits. The tool is embedded within a lesson plan that aims to enhance students' understanding of ML terms, data preparation, model training, and model evaluation. The lesson consists of four phases: initial understanding of ML concepts, using an interactive tool to understand a ML algorithm, programming the data processing and training the ML algorithm, and evaluating results and predictions. The lesson was delivered in a high school computer science classroom. Pre and post tests were administered to assess understanding of ML concepts and surveys were used to gauge student engagement and self-efficacy in programming. Results suggest that the tool was instructionally effective and had a positive impact on student engagement and self-efficacy.

Introduction and Motivation

Computer science is now a popular subject in high school with 60% of high schools offering it [1]. Many high schools now offer AP Computer Principles and AP Computer Science A courses and approximately 240,000 students take these courses every year [1]. These are college-level courses and some schools even offer more advanced computer science courses. Artificial intelligence (AI) is becoming a more in-demand subject to understand and teach [2]. We propose a project that teaches the basics of machine learning (ML) as it is becoming an increasingly important subject. ML is also becoming common in our lives so introducing the concept to children helps them understand the changing environment [3]. Students who learn ML will be better able to understand and work with AI [4]. The goal of the project is for students to understand the terms frequently used in ML and to understand the process of training an ML model, preparing the data, and evaluating the model.

It is important to understand AI because of its pervasiveness in our daily lives [5]. There has been support for AI literacy and worldwide growth in ML education for K12 students [5, 6]. Numerous educational tools and resources are available, Machine Learning for Kids, Cognimates and Google Teachable Machine [7, 8, 9]. Using these tools can help students be better prepared for future CS courses and employment [10]. However, they are black boxes with regard to showing how the underlying ML algorithms work [11]. The use of abstraction leaves the algorithms themselves unclear to the users [11].

Researchers have proposed three main areas that students should understand: 1) what a ML system is, 2) how ML models are formed and 3) what the effects are from ML applications [13]. For our work, we focus on areas 1 and 2. Focus area 3 can be used as an extension of the lesson.

There have been some attempts to increase ML education in secondary schools presented in Table 1. However, our work differs because it uses Java for programming a Perceptron from scratch. While Python dominates ML education, it often relies on libraries that obscure the underlying mechanics. Java programming is taught as part of the Advanced Placement Computer Science A course [12]. After completing the course, students can be expected to have the skills for programming a basic machine learning algorithm. The necessary skills they learn include Java classes, procedures, iteration, selection and data abstraction [12]. Leveraging Java allows students to apply these existing skills to build a basic ML algorithm directly. Also, the Perceptron was chosen because its foundation and basic arithmetic is accessible for high school students to program and comprehend.

Karalekas et al use a robotics-based approach to teach ML to students without programming experience. Their students learned the concepts of rule-based automation, supervised learning, and reinforcement through interactions with robots [14]. Kaspersen et al. engaged students to explore the effect of ML on their lives. They focus on AI literacy through an interactive web tool [15]. Zimmermann-Niefield et al. used a hybrid ML approach for young students; ML and Scratch programming were used to create ML projects with gestural inputs [16]. Long et al. explain the steps used in making ML models. They outline big ideas that K12 students should cover such as “Computers perceive the world using sensors” and “AI applications can impact society in both positive and negative ways” [17]. Hitron et al. introduced ML to children using a hands-on stick-like device [3]. They found that the children could grasp basic ML concepts with direct trial of tools and they concluded that the children should learn with feedback that is provided repeatedly [3]. Teaching AI through a simulation game has also been trialed and older students were able to quickly understand how the machine learns [18].

Table 1. Comparison of ML research for K12 Education

Paper	**ML tools**	**ML skills**
Teaching Artificial Intelligence and Machine Learning in Secondary Education: A Robotics-Based Approach [14]	3D printed robots, virtual robotics platform, cloud-based tools	Teaching a robot, training a robot with Teachable Machine
High school students exploring machine learning and its societal implications: Opportunities and challenges [13]	Ethics-first learning web tool, Computational Empowerment approach	Implication of ML for elections, understanding AI and ML, building models with data
Youth making machine learning models for gesture-controlled interactive media [16]	Mobile application with gestures, connecting with Scratch	Data from wearable sensors, building a model, evaluating a model
Envisioning AI for K-12: What Should Every Child Know about AI? [5]	Cognimates, Machine Learning for Kids, Tensorflow Playground, Google’s AIY	Sensors, models of the world, learning from data, impact on society
Developing Teaching Materials on Artificial Intelligence by Using a Simulation Game [18]	Simulation game, learning diary for students	Explain how ML works, unsupervised vs supervised learning, reinforcement learning

A ten-year research study of ML teaching in high school showed that students acquired knowledge in basic ML concepts [19]. The students also had an increase in interest for ML-related jobs. Most courses taught about supervised learning and the most used programming language was Python. There were also blocked-based languages used. Most of the students were new to programming [19].

The three objectives of this work are for students to become familiar with: 1) terms frequently used in ML, 2) the process of training an ML model, and 3) the process of preparing data and evaluating a model by programming it. Related work that we reviewed did not incorporate Java programming to enhance students' understanding. We chose to continue with Java because many CS 2 students have taken a course on it already. Our aim was to use a ML algorithm that high school students could understand and program after learning the programming skills in an introductory computer science course. We believe the students will have a deeper understanding of ML if they program an algorithm themselves. The Perceptron, a simple neural network that can be used for binary classification, was introduced by Rosenblatt in 1958 [20]. The Perceptron algorithm was selected because although it is a relatively simple algorithm, it can perform well on classification tasks. By programming all facets of the algorithm, the students will have more comprehension of the ML stages and algorithms and how ML is used for image classification. Algorithm 1 shows the steps of the Perceptron Learning Algorithm that the students will implement.

Algorithm 1: Perceptron Learning Algorithm

Input:
  D: Training dataset of (x, y) pairs, where x is an input vector (features) and y is the true binary label ($y \in \{1, -1\}$).

Output:
  w: Learned weight vector.

1. Initialize:
   1.1. Let d be the number of features in x.
   1.2. Initialize weight vector $w \in \mathbb{R}^2$ to be random between [-1.0, 1.0]
2. For each training example (x_original, y_true) in D
   2.1. Calculate Net Sum (z):
        z = w · x_original (dot product)

   2.2. Predict Label (ŷ):
        If z > 0:
          ŷ = 1
        Else:
          ŷ = -1

   2.3. Update Weights (if mistake):
        If ŷ ≠ y_true:
          w = w + (y_true - ŷ) * x_original
3. Return w

Methodology

The Perceptron algorithm is used for supervised learning in ML. It is used for classification into two groups [21]. Given points that represent two different classes, the algorithm uses weights which are adjusted to find a boundary to divide the points into the two classes [21]. By updating the weights with the algorithm in the Perceptron Learning Algorithm, the performance of the classification improves.

We create a simulation tool shown in Figure 1 that relates the decision on whether or not to go on a picnic with the Perceptron algorithm. The tool allows the user to choose the weights for sunny, windy and weekend. Depending on the weight and if it is sunny, windy or weekend, the decision is made if it is over a threshold. This tool is easy to use and understand. The students benefit by seeing the example calculations and can adjust the weights and variables to understand how a decision is made.

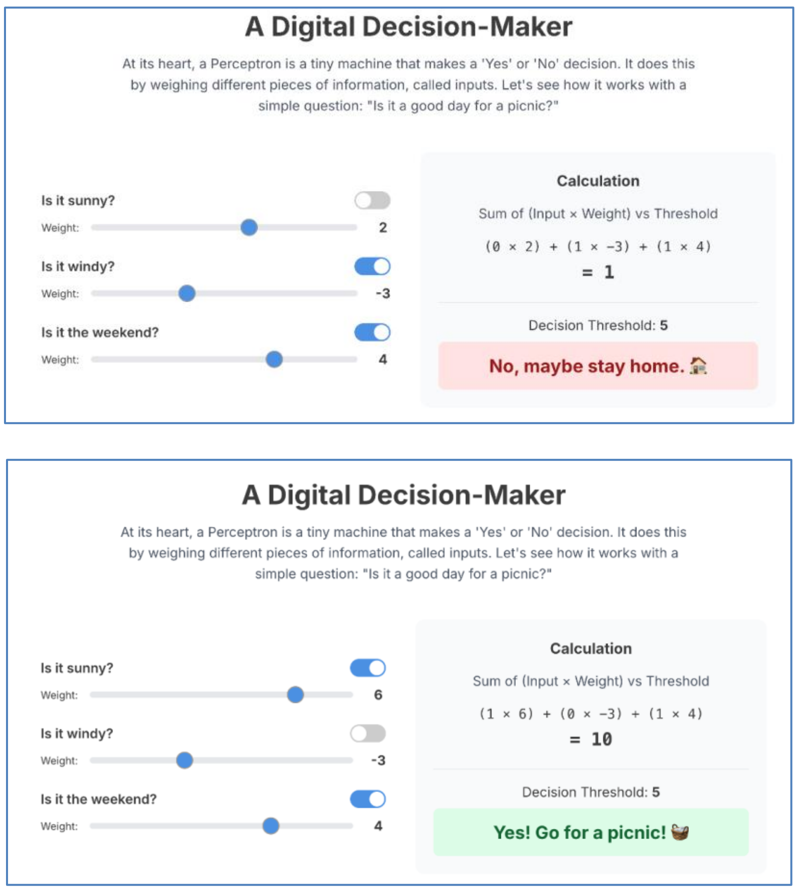

Figure 1. Screenshots from simulation tool to show how values are calculated to determine an outcome

Another simulation tool is shown in Figure 2. It is used to help the students better understand the steps of the Perceptron algorithm. The tool shows how the separating line is updated with the weight when it predicts a point to be in the wrong class. Students benefit from using this tool because they can visually see the iterations of the algorithm and they are given an explanation of the weight updates. This is important for them so that they can program the algorithm.

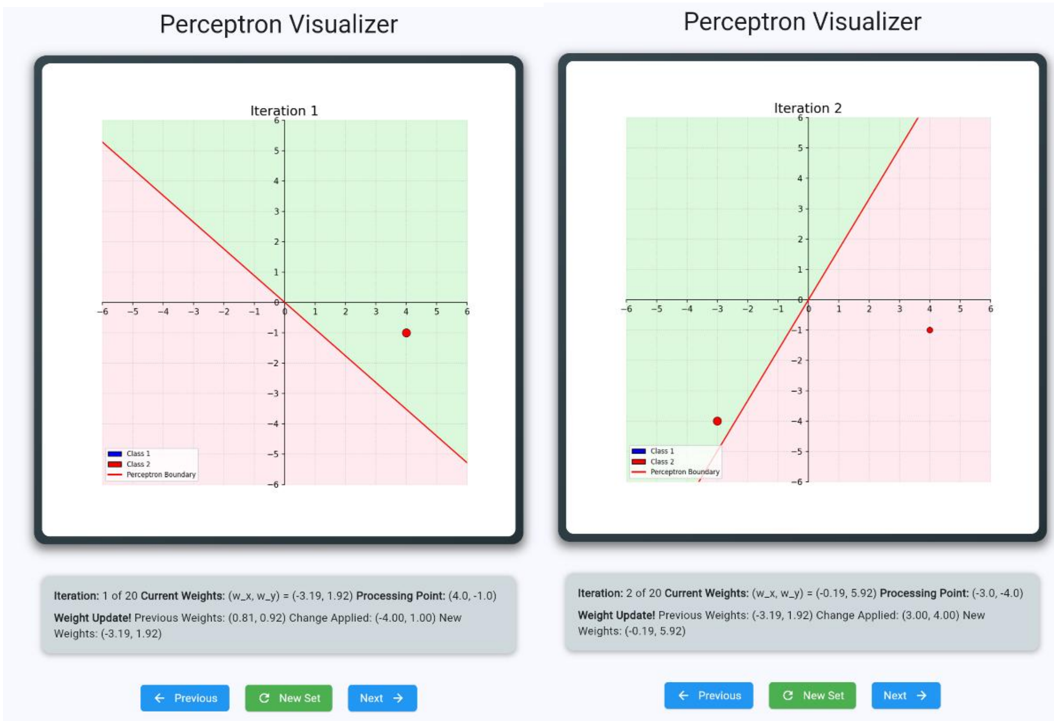

Figure 2. Simulation tool to understand the Perceptron algorithm

AI competency is needed and should be a priority according to the United Nations Educational, Scientific and Cultural Organization (UNESCO) [22]. They believe that understanding AI techniques and applications is integral and that students should be challenged to have a deep understanding by using three levels of progression: Understand, Apply and Create [22]. We addressed these levels in our work. For the Understand level, students are given the foundations of ML by learning the terms and processes involved. For the Apply level, students are introduced to the Perceptron algorithm and gain mastery of its update steps. For the Create level, students develop a program to create a Perceptron and update its weight using the algorithm learned in the Apply level.

For our work, we built web-based simulation tools that explain the Perceptron algorithm at the Apply Level. The students used these to first gain an understanding of the algorithm. The students then programmed the algorithm to perform binary classification of images of numerical digits at the Create Level. They trained their algorithm with a provided training set of images. They tested how their algorithm performed on a separate set of test images. Next, they tested their algorithm on new images that they created themselves.

Our study consists of four phases: 1) initial understanding of ML concepts through Google Teachable Machine, 2) using an interactive tool to understand a ML algorithm 3) programming the training for the ML algorithm and 4) evaluating results and predictions.

Phase 1: Introducing ML concepts at the Understand Level. In our first phase, our goal was to introduce the machine learning concepts. By the end of the phase students should have a basic understanding of the ML process. We began by explaining the steps of the machine learning process. The ML process can be broken into seven steps: gathering data, data preparation, choosing a model, training, evaluation, parameter tuning, and prediction [4]. After the explanation, students are shown Google Teachable Machine, a web-based tool for introducing ML classification [23]. We use this tool to strengthen the understanding of the terms and steps of the ML process. Students worked on their own to select two or more classes of images for training their model. Students then tested how their model performed on new images. At the end of this lesson, the students were asked to reflect on the task.

Phase 2: Evaluate Simulation tools at the Apply Level. In the second phase, the students solidify their knowledge by improving the conceptual understanding of the ML concepts. The goal of this phase is for the students to go through the lower level steps and be able to see and explain how the Perceptron updates. Students are introduced to the Perceptron model and how it works. The simulation shows the steps of how the weights are updated. It is interactive where the students are given data points and they can see how the model predicts correctly or incorrectly and how the weights are updated when there is a prediction error. After proceeding through the first section of the tool, the tool provides the students with a scaffolded experience in another section where the students use the tool to check the steps that the Perceptron should take. The students enter what they think the next step in the algorithm will be and they are given feedback of their correctness. This is a task-based learning opportunity for the students to enhance their skills in a complex situation.

Phase 3: Programming ML Process at the Create Level. The students apply their skills to program the Perceptron. Our goal is for the students to further enhance their understanding by programming the steps in the Perceptron algorithm that was introduced in Phase 2. We reviewed the pseudocode of the algorithm which matches with the steps shown to the students when they experienced the simulation tool. They are given training data from the MNIST dataset, a set of 60,000 labeled images as shown in Figure 3 [24]. With an understanding of the data format, students apply their skills to process the data for the model. The students program the Perceptron based on the algorithm pseudocode. They are able to complete an authentic ML task for binary classification.

Figure 3. Example of MNIST Images of Digits Used for Classification [24]

Phase 4: Evaluate Results at the Create Level. The students assess their model and its performance on a test set of 10,000 labeled images. With a simple Perceptron, they can achieve approximately 85% accuracy. They share feedback and review opportunities to enhance and extend their model. They can iterate on their model and gauge its success. As an extension to the lesson, students can brainstorm ways to achieve a more accurate model. Some ideas might include perform normalization on the data, add a bias term to the Perceptron, and introduce additional layers to make a multi-layer Perceptron. The model can also be extended to perform multi-class classification instead of binary classification. Future lessons could explore other machine learning methods such as decision trees or convolutional neural networks.

The lesson plan was evaluated in a high school computer science classroom with 23 students with ages ranging from 16 to 18. Students first completed a pre-survey for they rated the five statements below using a 7-point Likert scale (1=strongly disagree; 7=strongly agree):

- I enjoy programming

- I am good at programming
- I program on my own in my free time
- Programming is an important skill for my future
- I am familiar with machine learning

After the lesson, they completed a post- survey for they rated the five statements below, also using a 7-point Likert scale (1=strongly disagree; 7=strongly agree):

- I am familiar with machine learning
- The machine learning exercise was fun
- I am satisfied with how I made the machine learning model
- I put much effort in programming the machine learning model
- Programming the machine learning model helped me improve my skills

Students were also given a pre-test and a post-test to evaluate their understanding of the ML concepts. The pre-test and post-test included questions about ML terms and questions about updating the weights in the Perceptron algorithm: Sample questions include:

You are creating a system to predict a student's final exam score based on their homework grades, attendance percentage, and quiz scores. Which of these is a 'feature'?

a) Attendance percentage
b) The final exam score
c) The prediction algorithm
d) The student's name

An email service wants to build a model to automatically identify 'spam' and 'not spam'. It has a huge dataset of emails that have already been correctly categorized by users. What type of machine learning is this?

a) Unsupervised Learning
b) Reinforcement Learning
c) Data Collection
d) Supervised Learning

A perceptron has current weights (-3, 2.5). A point (1, 2) is labeled positive. What will the perceptron predict for the label? Will the weights be updated? If so, what will the new weights be?

Results

Test of pre-and post-test means. We calculate the mean scores from the pretest and posttest to measure the overall improvement in understanding. The pre- and post-test data means with n=23 are shown in Table 2.

Table 2. Pre- and Post- Test Means (n=23)

Metric	Value
Pretest Mean	16.00
Posttest Mean	22.20
t-statistic	2.3882
p-value	0.0753

On average, the post-test scores were higher than the pre-test scores. Since the p-value (0.0753) is greater than 0.05, the increase in correct answers is not statistically significant at the 0.05 level. While the scores improved, the small number of questions (sample size n=5) makes it harder to reach significance.

Test of pre- and post-test variance. We also used an F-test to analyze the variation between pre-test and post-test scores, assessing the consistency of improvement across different aspects of the concepts. The tests of variance are shown in Table 3.

Table 3. Pre- and Post-Test Variance

Metric	Value
Pretest Variance	36.5
Posttest Variance	0.2
F-statistic	182.5
p-value	0.0002

The fact that the p-value is extremely small (<0.001) suggests a significant difference in variance. The Posttest results are much more consistent than the Pretest results, suggesting that the intervention helped bring everyone to a similar increased level of understanding.

Opinion survey. The opinion survey results indicated a positive reception to the machine learning lessons and tools. Overall, participants showed high levels of engagement and agreement across all categories. Selected results include:

- Most Effective Tool: The perceptron visualization tool received the highest average rating (6.04/7), with over 95% of participants agreeing that it helped them understand the algorithm.
- High Engagement: 87% of participants reported enjoying the lessons, and 74% expressed a clear desire to learn more about machine learning.
- Educational Impact: There is strong support for AI education, with 83% agreeing that schools should teach machine learning and AI.
- Practical Understanding: 87% felt that programming the perceptron directly contributed to their understanding of the field.

Conclusions and Future Directions

We believe that ML is an important topic and that a deeper understanding is beneficial for students. It will prepare them for the increase of AI in our lives. We proposed a way to teach ML using four phases. Students were first introduced to the concepts and worked with simulation tools to strengthen their understanding. They then programmed the Perceptron algorithm in Java.

The results of our evaluation suggest that our simulation tool helped students to understand ML concepts better. Programming the algorithm was also beneficial and showed that a simple algorithm could still achieve meaningful results. One limitation of the study is the relatively small sample size, which may be why the increase in performance between the pre- and post-test was positive, but not significant at the 0.05 level. In the future, we would like to evaluate the materials again with more participants and additional questions on the pre- and post-tests. We would also like to develop more modules along with simulation tools for students to understand the progression and complexity of the ML algorithms that are used. Although the algorithms will be different, the concepts of training and evaluation would remain the same.

Acknowledgement

This material is based upon work supported by the National Science Foundation under Award No. 2206953. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] Code.org Advocacy Coalition, "State of Computer Science Education Report," *Code.org*, 2025. [Online]. Available: https://advocacy.code.org/stateofcs/. [Accessed: Jul. 02, 2025].

[2] P. Lin and J. Van Brummelen, "Engaging Teachers to Co-Design Integrated AI Curriculum for K-12 Classrooms," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, New York, NY, USA: ACM, 2021, pp. 1–12. doi: 10.1145/3411764.3445377.

[3] T. Hitron, I. Wald, H. Erel, and O. Zuckerman, "Introducing children to machine learning concepts through hands-on experience," in *Proceedings of the 17th ACM Conference on Interaction Design and Children (IDC '18)*, New York, NY, USA: ACM, Jun. 2018, pp. 563–568. doi: 10.1145/3202185.3210776.

[4] M. R. Zimmerman, *Teaching AI: Exploring New Frontiers for Learning*. Portland, OR, USA: International Society for Technology in Education, 2018.

[5] D. Touretzky, C. Gardner-McCune, F. Martin, and D. Seehorn, "Envisioning AI for K-12: What should every child know about AI?," in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI'19)*, vol. 33, Honolulu, HI, USA: AAAI Press, Jan. 2019, pp. 9795–9799. doi: 10.1609/aaai.v33i01.33019795.

[6] S. Druga, S. T. Vu, E. Likhith, and T. Qiu, "Inclusive AI literacy for kids around the world," in *Proceedings of FabLearn 2019 (FL2019)*, New York, NY, USA: ACM, Mar. 2019, pp. 104–111. doi: 10.1145/3311890.3311904.

[7] "Machine Learning for Kids," 2025. [Online]. Available: https://machinelearningforkids.co.uk. [Accessed: Jun. 27, 2025].

[8] "Home - Cognimates," *Hackidemia*. [Online]. Available: https://hackidemia.github.io/cognimates-website/home/. [Accessed: Jul. 01, 2025].

[9] "Teachable Machine," *Google*. [Online]. Available: https://teachablemachine.withgoogle.com/. [Accessed: Jun. 27, 2025].

[10] P. J. Denning and M. Tedre, *Computational Thinking*, MIT Press, 2019. [Online]. Available: https://mitpress.mit.edu/9780262536561/computational-thinking/. [Accessed: Jul. 01, 2025].

[11] R. Mariescu-Istodor and I. Jormanainen, “Machine Learning for High School Students,” in *Proceedings of the 19th Koli Calling International Conference on Computing Education Research (Koli Calling ’19)*, New York, NY, USA: ACM, Nov. 2019, pp. 1–9. doi: 10.1145/3364510.3364520.

[12] *AP Computer Science A Course and Exam Description*, College Board, New York, NY, USA, 2024.

[13] M. H. Kaspersen *et al.*, “High school students exploring machine learning and its societal implications: Opportunities and challenges,” *Int. J. Child-Comput. Interact.*, vol. 34, p. 100539, Dec. 2022, doi: 10.1016/j.ijcci.2022.100539.

[14] G. Karalekas, S. Vologiannidis, and J. Kalomiros, “Teaching Artificial Intelligence and Machine Learning in Secondary Education: A Robotics-Based Approach,” *Appl. Sci.*, vol. 15, no. 8, Apr. 2025, doi: 10.3390/app15084570.

[15] M. H. Kaspersen *et al.*, “High school students exploring machine learning and its societal implications: Opportunities and challenges,” *Int. J. Child-Comput. Interact.*, vol. 34, p. 100539, Dec. 2022, doi: 10.1016/j.ijcci.2022.100539.

[16] A. Zimmermann-Niefield, S. Polson, C. Moreno, and R. B. Shapiro, “Youth making machine learning models for gesture-controlled interactive media,” in *Proceedings of the Interaction Design and Children Conference (IDC ’20)*, New York, NY, USA: ACM, Jun. 2020, pp. 63–74. doi: 10.1145/3392063.3394438.

[17] D. Touretzky, C. Gardner-McCune, F. Martin, and D. Seehorn, “Envisioning AI for K-12: What should every child know about AI?,” in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI’19)*, vol. 33, Jan. 2019, pp. 9795–9799. doi: 10.1609/aaai.v33i01.33019795.

[18] S. Opel, M. Schlichtig, and C. Schulte, “Developing Teaching Materials on Artificial Intelligence by Using a Simulation Game (Work in Progress),” in *Proceedings of the 14th Workshop in Primary and Secondary Computing Education (WiPSCE ’19)*, New York, NY, USA: ACM, Oct. 2019, pp. 1–2. doi: 10.1145/3361721.3362109.

[19] “Findings on Teaching Machine Learning in High School: A Ten-Year Systematic Literature Review,” *ResearchGate*, 2022. [Online]. Available: www.researchgate.net. [Accessed: Jul. 03, 2025].

[20] F. Rosenblatt, “The perceptron: A probabilistic model for information storage and organization in the brain,” *Psychol. Rev.*, vol. 65, no. 6, pp. 386–408, 1958, doi: 10.1037/h0042519.

[21] “Perceptron,” *Wikipedia*, May 21, 2025. [Online]. Available: en.wikipedia.org. [Accessed: Jul. 08, 2025].

[22] “AI competency framework for students,” *UNESCO Digital Library*, 2024. [Online]. Available: https://unesdoc.unesco.org/ark:/48223/pf0000391105. [Accessed: Jul. 01, 2025].

[23] “Teachable Machine: Approachable Web-Based Tool for Exploring Machine Learning Classification,” *Google Research*. [Online]. Available: https://research.google/pubs/teachable-machine-approachable-web-based-tool-for-exploring-machine-learning-classification/. [Accessed: Jul. 01, 2025].

[24] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to

document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, doi: 10.1109/5.726791.