Embracing Computational Thinking as an Impetus for Artificial Intelligence in Integrated STEM Disciplines through Engineering and Technology Education

Paul Asunda, Miad Faezipour, Joshua Tolemy, Milo Timothy Do Engel

Abstract

The scope and versatile nature of engineering and technology education as a discipline provide a platform for the integration of computational thinking (CT) into STEM education, accomplishing the goal of bringing not only computer science principles into the K-12 education but also the fundamentals of machine learning (ML) and artificial intelligence (AI) into the curriculum. Today, it is commonplace to say that artificial intelligence and machine learning technologies impact the workplace and continue to revolutionize as well as create new demands for solving daily world challenges. This article discusses the integration of computational thinking practices of decomposition, pattern recognition, algorithmic thinking, and abstraction as key to problem-solving practices that may enhance the development of AI and ML capabilities in high school students. The intent of this article is to contribute to ongoing discussions among educators, employers, parents, and all those concerned with how best to prepare a citizenry that is digitally revolutionized. Implications are offered for the assessment of CT integrated within STEM, curriculum, pedagogy, and professional development for STEM teachers.

Keywords: STEM integration, computational thinking, problem-solving, artificial intelligence, curriculum, K-12 teaching, machine learning.

Introduction

Klaus Shwab (2016) shared that digital technology has borne the "fourth industrial revolution" altering our lives, work, and how we relate to each other. Today, we live in an increasingly computational world where humanity's dependence to solve global challenges continues to rely on technological revolution that is increasingly brought about by artificial intelligence (AI) and machine learning (ML). Artificial intelligence is the replication of human intellect and practices by machines that rationalize, solve problems, and make decisions as they learn new processes (Lind, 2019; Hintze, 2016). On the other hand, machine learning is a subdiscipline of AI that centers on the use of data and algorithms to mimic the way that humans learn, consequently empowering

Asunda, P., Faezipour, M., Tolemy, J., & Engel, M. T. D. (2023). Embracing computational thinking as an impetus for artificial intelligence in integrated STEM disciplines through engineering and technology education. *Journal of Technology Education*, *34*(2), 43–63. https://doi.org/10.21061/jte.v34i2.a.3

machines to make decisions without human intervention (IBM cloud education, 2020). Given the growing influence of digital technology, AI, and ML in the workplace, students must be prepared with computational thinking (CT) skills to genuinely connect and contribute to our digital world (Brizard & Mills, 2021). Though rarely mentioned, CT is often used subconsciously by students and teachers in finding solutions to problem-centered design challenges. A problem-centered design challenge is one that presents students with a learning opportunity to integrate relevant knowledge bases that is useful to comprehend and realize probable solutions to targeted problems (Bereiter, 1992). This may imply that educators need to help learners make meaningful connections between the learning opportunity and our everchanging digital world (Tan, et al., 2022).

The vision of CT as a powerful intellectual tool is described in the Framework for K-12 Science Education (National Research Council, NRC, 2012):

> computational theories, information and computer technologies, and algorithms have revolutionized virtually all scientific and engineering fields. These tools and strategies allow scientists and engineers to collect and analyze large data sets, search for distinctive patterns, and identify relationships and significant features in ways that were previously impossible. (p. 64)

Likewise, the Next Generation Science Standards (NGSS; NGSS Lead States, 2013), the K–12 Computer Science Framework (2016), and the Standards for Technological and Engineering Literacy (STEL; International Technology and Engineering Educators Association [ITEEA], 2020) emphasize the integration of science and engineering practices, crosscutting concepts, computational thinking, and core ideas in K-16 curricula. Based on these documents it can be rationalized that computational thinking — which applies concepts from computer science — is a vehicle by which K-12 educators may infuse into their instructional practices to develop within their students' computational capabilities and methods to solve problem-centered design challenges.

Brizard and Mills (2021) posit that these methods consist of a set of connected skills, such as pattern recognition, abstraction, critical thinking skills, communication, decomposition, etc., that may be applied to solve complex problems using, for example, automation, data analysis, or computational modeling. These skills and practices are consistently utilized in engineering and technology education classes to understand, solve and design solutions to design challenges. In line with this view, Banadaki (2020) postulated that CT integration in our teaching may be a way by which individuals may enrich problem-solving skills that when integrated with an information processing agent like a computer, may heighten the development of AI competencies in students. In our contemporary society, technological devices that support AI and

ML capabilities, i.e., computers, smartphones, smart systems, social networking, automated manufacturing, and other technologies, are woven into nearly every aspect of our daily lives (Stewart et al., 2019). ML is viewed as the backbone of AI and has become increasingly central to innovation across STEM disciplines (Banadaki, 2020).

The instruction of CT skills and practices in the K-16 curricula has been embraced by science, technology, engineering, and mathematics (STEM) educators in elementary, secondary, and higher education (Magana, & Silva Coutinho, 2017). Even though the literature (e.g., Holmlund et al., 2018; Kennedy & Odell, 2014) and educational reform documents like STEL, NGSS suggest a pedagogical change that may impact the teaching of STEM disciplines, the number of course offerings in K-16 curricula in the areas of computational thinking is minimal (Banadaki, 2020). In reaction to these concerns, various states have implemented policies to allow computer science (CS) coursework into the curriculum to fulfill high school graduation requirements and have passed legislation that requires schools to offer computer science coursework at various grade levels. This requirement has provided K-12 engineering and technology teachers with opportunities to teach, as well as to integrate CS into their instruction (Özdinç et al., 2022). This perspective may therefore place the field of engineering and technology education as a conduit to provide the context for preparing the future workforce with computational thinking concepts. However, the question then is, how can engineering and technology educators integrate computational learning opportunities into their current curriculum?

The purpose of this article is to stimulate dialogue about the place of AI and ML in the K- 12 curriculum and the role that engineering and technology education may play toward the realization of this goal. The most important question that may concern engineering and technology education as a field then may have to do with teacher preparation around AI pedagogy, and the design of learning activities that may provide students with opportunities to learn the concepts and processes of AI integrated within engineering and technology education course work. Specifically, we suggest the following two questions:

1. What does the infusion of CT practices that may lead to the development of AI and ML competencies in engineering and technology education mean for engineering and technology teacher education programs, and how do these programs then prepare engineering and technology education teachers?
2. What might a sample engineering and technology education lesson that integrates computational thinking concepts that support the development of artificial intelligence literacy hence ML look like?

In response to these questions, this essay is guided by the Opportunity to Learn (OTL) framework (Anderson, 1986). The OTL framework has four dimensions: content coverage; content exposure (time spent experimenting with course materials); content emphasis (emphasis on learning objectives around AI and ML that enhance problem-solving processes); and quality of instructional delivery (emphasis on instructional practices around AI and ML that enhance problem-solving processes). Teacher preparation tenets and instructional practices that may support the development of CT concepts in engineering and technology education classrooms are aligned with the OTL dimensions and indices. For example, practicing STEM teachers are expected to allocate instructional time to address NGSS standards that define disciplinary content and engineering design elements using a variety of pedagogical approaches. Should discourse on embracing CT as a vehicle for the inculcation of AI in integrated STEM disciplines through engineering and technology education blossom to the full, the result could be the unearthing of issues and challenges that become the basis of a framework for research in curricular initiatives that guides teacher-educators on how best STEM teacher education programs can prepare future K-12 teachers in this endeavor (Lewis, 2005).

Computational Thinking to Artificial Intelligence

Computational thinking is a prerequisite skill for understanding the technologies of the future. It is a thought process, rather than a specific body of knowledge about a device or language. CT is often associated with computers and coding, but it is important to note that it can be taught without a device (Thorson, 2018). Yadav et al. (2016) noted that CT offers an all-inclusive approach that exposes students to computing ideas and principles situated in the subject areas they are already learning. Further, Yadav et al. shared that CT involves organizing complex problems into manageable sub-problems (problem decomposition), using a sequence of steps (algorithms) to solve problems, reviewing how the solution transfers to similar problems (patterns), and finally deciding whether a computer can help us to efficiently solve those problems (automation). For example, in an engineering technology robotics class, students can create and edit functions, subroutines, and even iterative loops that depict decomposition, algorithmic thinking, pattern recognition, and automation.

CT's decomposition technique helps individuals understand the smaller aspects of a bigger picture, i.e., seeing the forest as well as the trees, and how each part works to make the forest. Through this technique, students can find solutions to problems by thinking like a computer. For example, a mathematics student may break down an equation into a step-by-step process and solve the problem in parts to get the result. Integration of CT concepts that facilitate the development of AI and ML in engineering and technology education may, for example, help connect systems to solve a given design challenge or meet a need that students may relate to in the ever-evolving digital world (Chatterjee, 2022).

Pattern recognition is dependent on a student's capability to assess and make meaning of some inherent configuration of ideas, objects, images, or any type of data, thus recognizing similar or different characteristics in the ideas, etc. Such a skill helps students to uncover the visual clues to an underlying order enabling them to ideate and or combine different patterns and unveil the sequence in a given configuration. Chatterjee (2022) shares that pattern recognition provides the basis to make a rationale when solving challenges in different situations. In other words, pattern recognition looks for similarities among and within problems (Chatterjee, 2022; Modini, 2019). Students as early as at the elementary level are taught how to recognize patterns, analyze, and find configurations to help them resolve puzzles and make decisions. Pattern recognition may play a key role, especially in ML and AI (Mendon-Plasek, 2021).

Decomposition and pattern recognition as concepts provide learners with the capabilities to analyze complex and abstract ideas. Similarly, abstraction then may be viewed as the process of sifting out unwanted parts of a given entity. It is the act of concentrating on pertinent information only, ignoring irrelevant parts (Modini, 2019), specifically, when solving a design challenge. As such, abstraction helps individuals prioritize relevant problem-solving actions in order of importance, allowing students to scrutinize a problem by keeping what is important and eliminating what is not. This way students solve problems without giving much importance to things that are insignificant toward realizing a solution. In finding solutions to a problem-centered design challenge in an engineering technology class, abstraction can be viewed through the lens of brainstorming and narrowing down feasible solutions using a design matrix. McVeigh-Murphy (2020) shared that students' comprehension of abstraction enables them to make sense of problems they encounter, consequently helping them not be overwhelmed when they encounter more complex problems, as they persist, compute, iterate, and ideate. To this end, abstraction provides students with relevance and clarity when solving problems.

Algorithmic thinking helps in developing solutions to a problem, through the device of some definite steps that are required to solve the challenge. Algorithmic thinking can therefore be viewed as a set of guidelines that are generated to solve a given problem. For example, in robotics classes, students input sequential commands enabling the robot to execute desired functions. Modini (2019), posits such are a series of steps, in sequential order, that if followed will solve a given challenge, and the related skills can be transferable or applied to solve problems that may be similar. Algorithms may then allow systems to utilize definitive data and automate themselves, sometimes creating new patterns.

It is important to recognize that these descriptions place CT at the center of instructional strategies that STEM teachers, specifically engineering and technology teachers, are already utilizing in their classrooms. For example, in engineering and technology classrooms examples of computational thinking may include analyzing and interpreting data that may involve scientific and engineering practices (e.g., developing and using models, planning and carrying out investigations, pattern recognition and pattern generalization, and analyzing and interpreting data) as outlined in the NGSS, and are synonymous to the engineering design problem-solving strategies utilized and experienced by K-12 students in their classrooms that may mirror aspects of simple AI systems (Duschl, & Bybee, 2014).

For example, suppose a high school engineering and technology student wants to answer an open-ended question that looks at an increase in population and how it impacts road expansion projects and traffic light systems. The lesson may be designed to require students to (a) create a prototype of a road expansion using a 3D printer and a laser cutter, and (b) program a microcontroller that depicts an automated traffic light system. This road expansion requires that two streets be redesigned to make a 4-way traffic light intersection, and one of the roads crosses a railway line. In order to be approved, the street and railway intersections need to be prototyped and demonstrate both safety and efficiency. To comprehend the requirements of this design challenge successfully, students must be able to decompose the design problem into different elements and figure out how best to prototype possible solutions. In this given scenario, the students might work in teams to decipher what tools and different elements they require to complete the project as well as to document the process they might employ to prototype the road intersection. Students might decide to:

- conduct research, brainstorm, and sketch how the intersection might look like,
- utilize CAD software to design and 3D print traffic light posts and railway line crossing arm,
- scale the 4-way intersection for laser machining,
- utilize tinker cad software to block code a microcontroller and generate data that will support the safe operation of a traffic light system (see Figure 1), and
- automate the model traffic lights and position the railway crossing arm as per the sketch and integrate its operation with the traffic light system.

In this example, students utilize and combine an engineering design process and CT thinking elements of decomposition, to build a traffic light algorithm with a pattern for the system to operate safely and integrate some form of automation. As they complete this assignment students will test and retest their traffic lights system and at each instance collect data about how the system is working, as they refine it.

Figure 1

Coding Traffic Light System Using Tinker CAD for Safe Operation of Traffic Lights

```
  if (!digitalRead(horizontalCarL)) pushButton = 4;

  if (pushButton==5) pushButton = 1; //if no pushPins (notice lat
  if (pushButton==6) pushButton = 2; //then these autonomously se
  if (pushButton==7) pushButton = 3; //the pushButton variable.
  if (pushButton==8) pushButton = 4; //5-8 given at the end of ea

  if (pushButton==0) pushButton = 1; //Allows light to start with
}

void verticleStraight()     //Cars going vertically straight
{
  digitalWrite (verticleRed, LOW);
  digitalWrite (verticleYellow, LOW);
  digitalWrite (verticleGreenS, HIGH);
  digitalWrite (verticleGreenL, LOW);
  digitalWrite (horizontalRed, HIGH);
  digitalWrite (horizontalYellow, LOW);
  digitalWrite (horizontalGreenS, LOW);
  digitalWrite (horizontalGreenL, LOW);

  delay(10000);     //Stays green for 10 seconds
  verticleTurnOff();     //Calls function that goes back to red l
  pushButton = 6;     //Commands next order light cycle if no pus
}

void verticleLeft()     //Cars going vertically left
{
  digitalWrite (verticleRed, LOW);
```

This argument supports the assertion that data is the foundation of scientific and engineering practices that inform computational thinking practices. Through such engineering and technology education projects, teachers need to think of ways they can integrate CT into STEM curriculum, and instructional practices to encourage students to develop AI and ML knowledge as they become more effective problem solvers and innovators.

Artificial Intelligence

Inspired by the way the human brain processes data and learns information, AI is the automation of computer systems' activities which we normally attribute to human thinking and rationality, such as problem-solving, decision-making, etc. (Lind, 2019). As a trending area that is evolving, Schroer (2022) shared that AI is deployed in various aspects of industry: smart manufacturing, automated quality control, biomedical sciences, healthcare informatics, data analytics, cybersecurity, biometrics/authentication, robotics, automation, and many more.

Relatedly, Burns et al. (2022) stated that AI provides organizations with insightful opportunities into their operations about which they may not previously have been aware. In some cases, artificial neural networks and deep learning AI technologies effect and finish work processes better than human beings, particularly when it comes to repetitive, detail-oriented tasks like analyzing data sets and numbers. They also more quickly and accurately predict outcomes than do human beings.

Neural networking is a method of learning in AI with interconnected nodes that are in a layered structure resembling the functioning of a human brain (Chen, 2022). Neural networks teach computer systems to perform a task repeatedly, each time possibly learning something new from the many layers of data it accesses and reviews for relationships between the data, tweaking the outcome to improve the results, hence the name "deep learning" (Burns & Burke, 2021; Marr, 2018). Therefore, deep neural networks allow in-depth extraction with more detailed and specific features of a system to be unearthed, and are thus seen to mostly produce highly accurate results. Nevertheless, the complexity of the algorithm increases with the number of layers in the network (Uzair & Jamil, 2020; He & Sun, 2015). These models utilize machine learning technology to review, access, and cycle through sizeable data sets quickly to learn a trend, minimize an inherent error, and aid consumers in making informed decisions. This is similar to how human beings make informed decisions from our learned experiences.

Machine Learning as the Backbone of AI

Machine learning is the science of empowering machines to make decisions without human intervention. ML, therefore, is a subdiscipline of AI that, relies on data and algorithms to impersonate the way that humans learn, steadily improving its accuracy (IBM Cloud Education, 2020). This sub-discipline (i.e., ML) together with deep learning form the backbone of AI, permitting computers to learn and decipher patterns in images, sounds, and structured data through multidimensional arrays (Lind, 2019). According to King (2019), the basic data object in ML is an array. An array is an entity that can support a fixed number of values of a single type but can also have different dimensions (i.e., multi-dimensional). For example, students test their prototypes as solutions to a given problem-solving design challenge with the intent to improve the product. In this scenario, students might be required to conduct a number of tests against some given criteria and collect and record data on each test run. Each run can then be conceptualized to capture the essence of a multi-dimensional array, where each test run may capture and record a different number depending on the testing conditions. Therefore, arrays are important data entities in machine learning. They represent images, text documents, and many other types of data.

The key to building an ML model is finding and compiling informative features that are integral toward mining in-depth information to realize commonalities in a system, hence a pattern. This implies ML models are designed to focus on a given data set, compute and learn specific information and features, and reveal a pattern, possibly enabling the system to be able to come up with a prediction (Sarker, 2021). This process can be seen as similar to when teachers are able to envisage a student's performance in a test. Teachers understand the student's strengths and can be able to predict a student's performance on a test based on the student's previous test scores (i.e., data).

Steps to Design a Machine Learning Model

The seven main steps to effectively design a general-purpose machine learning model have been comprehensively discussed by Joshi (2019) and Mayo (2022). These steps include: (a) gathering data, (b) preparing data, (c) model selection, (d) training, (e) evaluation, (f) parameter tuning, and (g) prediction. To better understand these steps let's consider one area where ML is widely used, the healthcare sector and the design of patient systems like x–ray scanners and magnetic resonance imaging (MRI) equipment, that seek to provide patients with experiences that meet their medical needs. In considering this example, these steps can then be further explained through a brain tumor MRI image classification (see Figure 2) ML model as follows:

- **Gathering data** – ML models require data. Imagine creating an ML model that is capable of distinguishing benign from malignant (cancerous) tumors in brain tumor MRI images. The first step is to collect numerous samples of brain tumor MRI images. Each instance of data should be annotated (labeled). In the case of brain tumor MRI images, a neurologist would label the images as benign or malignant.
- **Preparing data** – In order to feed data into an ML model, the data should be prepared or preprocessed. This may involve normalizing the data to have a known (limited) range, removing unwanted information, extracting features and attributes most representative of the data, etc. For example, brain tumor MRI images can be resized, filtered, and/or certain features regarding the intensity of the pixels, shape, and texture of the segmented tumor area can be extracted from the image.
- **Model selection** – The next step is to select a machine learning model. There are various machine learning techniques. Some models work better on small data sets (decision trees, support vector machines), while deep learning models work best with huge data sets. A deep neural network

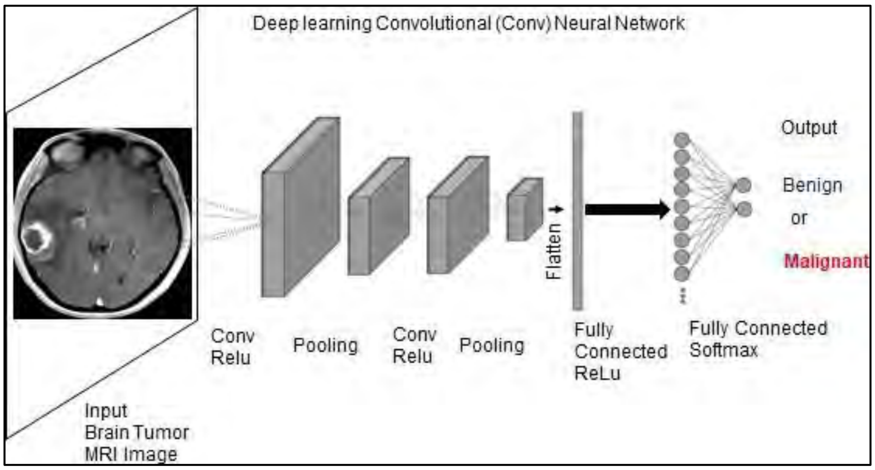
may be selected as the ML model for the brain tumor MRI image classification problem.

- **Training** – Once an ML model is selected, the model should be trained using a sample of the labeled data. Features of the labeled data are provided as input instances of the model. Through training, the model learns patterns from the input data, and the objective is to minimize the error of the function fitting the data. In the brain tumor MRI images example, the model learns patterns of images through the training phase.
- **Evaluation** – The model produces a function that should fit the input data closely enough (regression), or distinguish classes accurately enough (classification). The evaluation step involves testing the model on new data inputs in the data set as we are still in the model development and tuning phase. For example, new instances of the brain tumor MRI images can be tested, to see how accurately the model identifies the class/label (benign or malignant).
- **Parameter tuning** – After evaluation, as we are still in the model development and tuning phase, the model parameters (e.g. weights and biases) may need to be adjusted to better fit the evaluation data and for better accuracy.
- **Prediction** – The model is now ready to be used for prediction. This step involves testing the ML model on data inputs/instances that have not been provided in the training or evaluation phase. For example, the model can be tested on a new brain tumor MRI image to predict its class (see Figure 2). In machine learning, it is a common practice to use 60% of the data for training, 10% for evaluation, and 30% for testing (Joseph, 2022; Nguyen et al., 2021).

Integrated STEM Education

According to Drake and Burns (2004), integrated curriculum is about making connections, across disciplines with the goal of enhancing student learning. This position then offers a starting point for understanding how best to integrate STEM (iSTEM) disciplines. iSTEM has been viewed as an instructional approach to teaching and learning where the curriculum and content of the four individual STEM disciplines flawlessly fuse into real-world experiences contextually consistent with authentic problems and applications in STEM careers. Such amalgamation may refer to making intentional and purposeful connections between core disciplinary practices of each STEM domain being integrated, with the goal of using this combined knowledge to solve real-world problems (Mobley, 2015; STEM Taskforce Report, 2014).

Figure 2
A Deep Learning Convolutional Neural Network Model for Brain Tumor MRI Image Classification



Note. Adapted from Medical Xpress. (2009). A crystal ball for brain cancer? New method predicts which brain tumors will respond to drug. https://medicalxpress.com/news/2009-07-crystal-ball-brain-cancer-method.html and Sunsuhi, G. S., & Jose, S. A. (2022). An adaptive eroded deep convolutional neural network for brain image segmentation and classification using inception resnetV2. *Biomedical signal processing and control*, *78*, 103863. https://doi.org/10.1016/j.bspc.2022.103863

Enactment of CT within iSTEM Learning Opportunities for Students

Thibaut et al (2018) noted that teaching in context plays a significant role in the realization of iSTEM instruction. Contexts provide learning environments that contribute to strong connections of conceptual and procedural knowledge, and metacognitive opportunities that inform instructional and learning opportunities to meet student learning needs. The Opportunity to Learn framework rests on the assumption that students' learning is a function of both student factors like ability and perseverance, as well as classroom and teacher factors like time allocated for instruction on new materials, students' engagement with materials, and instructional quality. We, therefore, make an argument that for students to develop AI and ML knowledge and skills, teachers should be deliberate in designing STEM learning opportunities in context. Bokhove et al. (2019) shared that OTL can be conceptualized as the interplay of how much time educators may spend on, e.g., the integration of STEM disciplines with computer science content (instructional hours) and the proportion of the time that is spent on a specific topic (curriculum content coverage). As such, this essay provides an argument that the enactment of CT within iSTEM learning opportunities for students, should be designed to provide ample instructional time that enhances the gaining of AI and ML knowledge, inculcating in students the ability to apply their learning to solving problem-centered design challenges and situations in our increasingly computational world. In other words, purposeful and meaningful integration of learning opportunities is the basis of STEM education, and the premise of this paper, a call to infuse AI and ML as a vehicle to enhance students' 21st century skills through computational thinking.

K-12 Engineering and Technology Education and STEM Education

With the newly revised Standards for Technological and Engineering Literacy (ITEEA, 2020), the field of engineering and technology education at the K-12 level has positioned itself as a STEM-focused area of study. STEL is designed to equip educators with a better understanding of how to teach technology and engineering education. These standards present the argument that teaching engineering design at the K-12 level helps all students use an informed design process to solve technological problems. Particularly, students are afforded an opportunity to develop domain knowledge in the design process as a consequence this may promote their problem-solving skills through the application of CT concepts that may play a significant role in their development of AI and ML.

An Example of a Foundational CT Lesson that May Lead to an ML Project in Engineering and Technology Education, Designing a Coin Sorter

A coin sorter is a device that differentiates between a random collection of coins in different denominations and sorts them into separate bins based on the value and /or size of the coin. The idea of sorting coins is founded on the recognition of different arrays like measurement, weight, and appropriate sensors if necessary using an Arduino to program the sorter (Paramasivam et al., 2020; Murali, 2020). Consider the following example lesson:

Content Area: Engineering and Technology Education

Target Grade Level: 10-11 grade

Time Frame: One Activity / 50-minute Class Period

Standard(s):

- STEL-2D – Develop a plan in order to complete a task.
- STEL-3A – Apply concepts and skills from technology and engineering activities that reinforce concepts and skills across multiple content areas.

- STEL-7E – Illustrate that there are different solutions to a design and that none are perfect.

Learning Objectives:

- The student will develop the core competencies of computational thinking.
- The student will develop an understanding of how to break down complex problems into several smaller, simpler ones (decomposition).
- The student will be able to make connections between similar problems and experience (pattern recognition).
- The student will be able to identify important information while ignoring unrelated or unimportant details (abstraction).
- The student will be able to develop a step-by-step plan in order to solve a problem (algorithms).

Basic Problem:

Sort the provided bag of coins into separate piles based on types/values.

Details:

- ~30 coins total
- Coins will be a mix of values in both Canadian and American currency or any other coins of the instructor's choice
- Values will be a mix of pennies, nickels, dimes, and quarters

Step-by-Step Instructions:

1. Introduce the concepts of computational thinking:
   a. A way of thinking about complex problems and breaking them down into smaller, easier-to-solve task.
2. Explain the activity to the students.
   a. There are two different currencies.
   b. Each currency/value needs its own pile.
   c. "While I pass these out, think about how you're going to sort out these coins."
   d. "There's no one, right way to do this."

3. Distribute the bags of coins to each student, ensuring that each student will have sufficient space to complete the activity while doing so.
4. Watch the students as they go through the activity.
    a. Take note of students sorting the coins in different ways (e.g., Student A first divides the coins by currency then by value while Student B sorts the coins into piles one coin at a time).
    b. Help students by answering questions and assisting as needed, but be sure not to tell them how they should be sorting the coins outside of the specific requirements.
5. When finished, reflect on the activity with the students using leading questions:
    a. **Decomposition**: Sorting a bag of coins sounds like a simple task, but you had to sort out two different currencies, each with four different values of coins. So, you had to figure out whether a coin was American or Canadian; determine whether it was a penny, nickel, dime, or quarter; and then sort that coin into the appropriate pile you were slowly creating.
    b. **Abstraction**: "What was important to notice when sorting each coin?"
        i. American vs. Canadian currency
        ii. Value of the coin
        iii. Determining which other coins match the one you're seeing
    c. **Abstraction**: "What wasn't as important to look at when sorting each coin?"
        i. The year the coin was made
        ii. The state/province displayed on the coin
        iii. The color of the coin
        iv. The size of the coin
        v. The material the coin was made of
    d. **Algorithms**: "What step-by-step process did you go through when sorting the coins?"
        i. Call on the students you took note of who were doing things differently.

    ii. Some methods to think about:
        1. Sort by currency and value at the same time, one coin at a time.
        2. Divide the pile by currency > sort by value.
        3. Sort by relative value > divide value piles by currency.

e. **Pattern Recognition**: "What other things can you think about that seem a lot like the process you went through when sorting the coins?"
    i. Some comparisons to make:
        1. Getting ready in the morning
        2. Getting up, eating breakfast, brushing your teeth, getting dressed, etc.
        3. Doing your laundry
            a. Sorting the clothes by whites and colors, washing, drying, etc.

6. Have students re-bag their coins and deliver them to the front of the classroom.
7. Continue to the next activity or conclude class (as relevant).

Implications for Engineering and Technology Education and STEM Disciplines

This essay has laid out an argument that our schools need to provide opportunities to facilitate the development of AI and ML competencies in K-12 students. As STEM learners (high school/college students and teachers, not computer scientists) deploy AI and ML to extract patterns and make predictions, this may aid them in their development of computational thinking and critical skills, in turn improving their AI literacy. How and Hung (2019) share that such may have the potential to enhance students'abilities to efficiently solve complex problems. This position then lends us to suggest the following implications, namely (a) assessment and pedagogical implications for purposeful integration of AI and ML learning opportunities within STEM disciplines, (b) professional development for in-service teachers and STEM teacher educators, and (c) integrated STEM curriculum and theorizing. We reflect briefly upon each as the article concludes.

Assessment and Pedagogical Implications for Purposeful Integration of AI and ML Learning Opportunities within STEM Disciplines

The fourth industrial revolution and the centrality of technological revolution and its impact on solving global challenges cannot be overlooked. Nevertheless, these rapid changes have brought forth the challenge of how best to integrate STEM disciplines and prepare students through an integrated curriculum that incorporates CT as a vehicle to develop AI and ML competencies in students. There is still no clear way how best to integrate STEM disciplines through engineering design as outlined in the NGSS, hence various assessment strategies are deployed. However, this essay presents an opportunity for dialogue and research opportunities, to further explore how best to integrate CT within problem-solving instruction and accentuate the teaching, instruction, and assessment of STEM disciplines in contextual learning environments. This may call for the purposeful integration of these disciplines to facilitate the development of ML and AI skills in students through CT. All the same, more needs to be done to design integrated STEM experiences that incorporate CT opportunities explicitly to enhance learning the fundamental concepts of ML and AI at the K-12.

Implications for Professional Development for In-Service Teachers and STEM Teacher Educators

Pre and in-service teacher education and preparation programs, and for that matter engineering and technology teacher education programs, have traditionally not included coursework that explicitly delineates CT and related learning activities as key components of teaching problem-solving using contextualized design activities. Further, the challenges of offering CT-related courses at the K-12 level are even starker for rural and small school districts where administrators and school boards may be less likely to see CT as a priority. In response to these challenges, standards and reform documents like the NGSS and National Science Foundation (NSF) initiatives such as Computer Science 10K (CS10K) represent collaborations between academia, national professional associations, and organizations such as Code.Org, Microsoft, and Google that have aimed to bring CT concepts to the secondary level (Yadav et al., 2016; Google, 2015; Grover & Pea, 2013). As such, there is a clear need for increased professional development activities for in-service teachers around CT, the fundaments of AI and ML, and how the STEM design challenges they complete with their students leverage CT through problem-solving. Likewise, STEM teacher educators should seek ways to introduce CT and learning opportunities that may facilitate the development of AI and ML fundamentals into teacher preparation and education coursework. They may provide pre-service teachers opportunities to develop competencies and knowledge of CT content including AI and ML, and how best to instruct their future students.

Implications for Integrated STEM Curriculum and Theorizing

Honey et al. (2014) and Vasquez et al. (2013) assert that the basis of STEM education involves the integration of science, technology, engineering, and mathematics by breaking down the "silos" of discipline-independent teaching so that students begin to see how the concepts and skills from different disciplines can work together to help them answer intriguing questions and solve meaningful problems. Thus, the integration of CT within STEM disciplines that may facilitate the development of ML and AI competencies in students, suggests that the K-12 curriculum and the designers of such learning opportunities need to be explicit about the goals they aim to achieve, in addition to purposefully designing the integrated STEM experience to achieve these goals. They also need to better articulate their rationale about why and how a particular CT-integrated STEM experience will lead to particular outcomes around AI and ML learning in K-12 students, and how these experiences will be measured.

References

Anderson, L. W. (1986). Opportunity to learn. In T. Husén & T. Postlethwaite (Eds.), *International encyclopedia of education: Research and studies* (pp. 3682–3686). Pergamon.

Banadaki, Y. M. (2020). *Enhancing the role of machine learning in STEM disciplines through supervised undergraduate research experiences*. https://infonomics-society.org/wp-content/uploads/Enhancing-the-Role-of-Machine-Learning-in-STEM-Disciplines-through-Supervised-Undergraduate-Research-Experiences.pdf

Bokhove, C., Miyazaki, M., Komatsu, K., Chino, K., Leung, A., & Mok, I. A. (2019). The role of "Opportunity to Learn" in the geometry curriculum: A multilevel comparison of six countries. *Frontiers in Education*, *4*, 63. https://doi.org/10.3389/feduc.2019.00063

Brizard, J., & Mills, K. (2021, December 9). Opinion: Students need more computer training for our increasingly digital world. *The Hechinger Report*. https://hechingerreport.org/opinion-students-need-more-computer-training-for-our-increasingly-digital-world/

Bereiter, C. (1992). Referent-centred and problem-centred knowledge: Elements of an educational epistemology. *Interchange, 23*(4), 337–361. https://doi.org/10.1007/BF01447280

Burns, E., & Burke, J. (2021). What is a neural network? Explanation and examples. https://www.techtarget.com/searchenterpriseai/definition/neural-network

Burns, E., Laskowski, N., & Tucci, L. (2022). *What is artificial intelligence (AI)?* https://searchenterpriseai.techtarget.com/definition/AI-Artificial-Intelligence

Chatterjee, N. (2022). *What is computational thinking in 2021 – Why thinking like a computer builds skills for success*? https://codevidhya.com/what-is-computational-thinking/

Chen, J. (2022). *What is a neural network*? https://www.investopedia.com/terms/n/neuralnetwork.asp

Drake, S. M., & Burns, R. C. (2004). *Meeting standards through integrated curriculum*. Association for Supervision and Curriculum Development.

Duschl, R. A., & Bybee, R. W. (2014). Planning and carrying out investigations: An entry to learning and to teacher professional development around NGSS science and engineering practices. *International Journal of STEM Education, 1*(1), 1–9. https://doi.org/10.1186/s40594-014-0012-6

Google. (2015). *Searching for computer science: Access and barriers in U.S. K-12 education*. https://services.google.com/fh/files/misc/searching-for-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. https://doi.org/10.3102/0013189X12463051

He, K., & Sun, J. (2015). Convolutional neural networks at constrained time cost. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5353–5360). https://doi.org/10.1109/CVPR.2015.7299173

Hintze, A. (2016). *Understanding the four types of AI, from reactive robots to self-aware beings.* https://theconversation.com/understanding-the-four-types-of-ai-from- reactive-robots-to-self-aware-beings-67616

Holmlund, T. D., Lesseig, K., & Slavit, D. (2018). Making sense of "STEM education" in K-12 contexts. *International Journal of STEM Education, 5*, (32), 1–18. https://doi.org/10.1186/s40594-018-0127-2

Honey, M., Pearson, G., & Schweingruber, H. (Eds.). (2014). *STEM integration in K-12 education: Status, prospects, and an agenda for research.* The National Academies Press. https://nap.nationalacademies.org/catalog/18612/stem-integration-in-k-12-education-status-prospects-and-an

How, M. L., & Hung, W. L. D. (2019). Educing AI-thinking in science, technology, engineering, arts, and mathematics (STEAM) education. *Education Sciences*, *9*(3), 184. https://doi.org/10.3390/educsci9030184

IBM Cloud Education. (2020). *What is machine learning?* https://www.ibm.com/cloud/learn/machine-learning

International Technology and Engineering Educators Association (ITEEA). (2020). *Standards for technological and engineering literacy: The role of technology and engineering in STEM education*. https://www.iteea.org/189252.aspx

Joseph, V. R. (2022). Optimal ratio for data splitting. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, *15*(4), 531–538. https://doi.org/10.1002/sam.11583

Joshi, N. (2019). *7 types of artificial intelligence.* https://www.forbes.com/sites/cognitiveworld/2019/06/19/7-types-of-artificial-intelligence/?sh=2ac3094c233e

K-12 Computer Science Framework Steering Committee. (2016). *K-12 computer science framework. Technical report*. Association for Computing Machinery. https://k12cs.org/wp-content/uploads/2016/09/K%E2%80%9312-Computer-Science-Framework.pdf

Kennedy, J. T., & Odell, L. R. M. (2014). Engaging students in STEM education. *Science Education International* 25(3), 246–258. https://files.eric.ed.gov/fulltext/EJ1044508.pdf

King, M. (2019). *Array basics*. https://www.machinelearningmike.com/post/array-basics

Lewis, T. (2005). Coming to terms with engineering design as content. *Journal of Technology Education*, *16*(2), 37–54. https://doi.org/10.21061/jte.v16i2.a.3

Lind, N. (2019). *Foundations of AI*. https://towardsdatascience.com/foundations-of-ai-b11d6ad7ce6f

Magana, A. J., & Silva Coutinho, G. (2017). Modeling and simulation practices for a computational thinking-enabled engineering workforce. *Computer Applications in Engineering Education*, *25*(1), 62–78. https://doi.org/10.1002/cae.21779

Marr, B. (2018). What is deep learning AI? A simple guide with 8 practical examples. https://www.forbes.com/sites/bernardmarr/2018/10/01/what-is-deep-learning-ai-a-simple-guide-with-8-practical-examples/?sh=15791ebc8d4b

Mayo, M. (2022). Frameworks for approaching the machine learning process. https://www.kdnuggets.com/2018/05/general-approaches-machine-learning-process.html

McVeigh-Murphy, A. (2020). The one about abstraction in computational thinking. https://equip.learning.com/abstraction-computational-thinking/

Medical Xpress. (2009). *A crystal ball for brain cancer? New method predicts which brain tumors will respond to drug*. https://medicalxpress.com/news/2009-07-crystal-ball-brain-cancer-method.html

Mendon-Plasek, A. (2021). Mechanized significance and machine Learning: Why it became thinkable and preferable to teach machines to judge the world. In J. Roberge & M. Castelle (Eds.), *The cultural life of machine learning*. Palgrave Macmillan, Cham. https://doi.org/10.1007/978-3-030-56286-1_2

Mobley, M. C. (2015). *Development of the SETIS instrument to measure teachers' self-efficacy to teach science in an integrated STEM framework* (Unpublished doctoral dissertation). University of Tennessee, Knoxville. https://trace.tennessee.edu/cgi/viewcontent.cgi?article=4700&context=utk_graddiss

Modini, T. (2019). *Getting started with computational thinking*. https://usingtechnologybetter.com/blog/getting-started-with-computational-thinking/

Murali, K. R., (2020). *Coin sorting machine using Arduino*. https://www.skyfilabs.com/project-ideas/coin-sorting-machine-using-arduino

National Research Council. (2012). *A framework for K-12 science education: Practices, crosscutting concepts and core ideas*. The National Academies Press. https://nap.nationalacademies.org/catalog/13165/a-framework-for-k-12-science-education-practices-crosscutting-concepts

NGSS Lead States. (2013). *Next generation science standards: For states by states.* The National Academies Press. https://nap.nationalacademies.org/catalog/18290/next-generation-science-standards-for-states-by-states

Nguyen, Q. H., Ly, H. B., Ho, L. S., Al-Ansari, N., Le, H. V., Tran, V. Q., Prakesh, I, & Pham, B. T. (2021). Influence of data splitting on performance of machine learning models in prediction of shear strength of soil. *Mathematical Problems in Engineering*, *2021*, 1–15. https://doi.org/10.1155/2021/4832864

Özdinç, F., Kaya, G., Mumcu, F., & Yıldız, B. (2022). Integration of computational thinking into STEM activities: an example of an interdisciplinary unplugged programming activity. *Science Activities*, *59*(3), 151-159. https://doi.org/10.1080/00368121.2022.2071817

Paramasivam, A., Vidhushini, P., Arunprakash, N., & Srinivasan, S. P. (2021). Design and modelling of coin sorting machine. *Nveo-Natural Volatiles & Essential Oils Journal| NVEO*, *8*(4), 8202–8208. https://www.nveo.org/index.php/journal/article/view/1753/2122

Sarker, I.H. (2021). Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science, 2* (6), 420. https://doi.org/10.1007/s42979-021-00815-1

Schwab, K. (2016). *The fourth industrial revolution.* World Economic Forum.

Schroer, A. (2022). Introduction to AI. https://builtin.com/artificial-intelligence

STEM Task Force Report. (2014). *Innovate: A blueprint for science, technology, engineering, and mathematics in California public education*. Californians Dedicated to Education Foundation. https://www.cde.ca.gov/pd/ca/sc/documents/innovate.pdf

Sunsuhi, G. S., & Jose, S. A. (2022). An adaptive eroded deep convolutional neural network for brain image segmentation and classification using inception resnetV2. *Biomedical Signal Processing and Control*, *78*, 103863. https://doi.org/10.1016/j.bspc.2022.103863

Stewart, D., Loucks J. Casey M., & Wigginton C. (2019). Bringing AI to the device: Edge AI chips come into their own. https://www2.deloitte.com/content/dam/Deloitte/cn/Documents/technology-media-telecommunications/deloitte-cn-tmt-prediction-full-report-2020-en-200217.pdf

Tan, A. L., Ong, Y.S., Ng, Y. S., & Tan, J. H. H. (2022) STEM problem solving: Inquiry, concepts, and reasoning. *Science & Education*, *32*, 381–397. https://doi.org/10.1007/s11191-021- 00310-2

Thibaut, L., Knipprath, H., Dehaene, W., & Depaepe, F. (2018). How school context and personal factors relate to teachers' attitudes toward teaching integrated STEM. *International Journal of Technology and Design Education, 23*(3), 631–651. https://doi.org/10.1007/s10798-017-9416-1

Thorson, K. (2018). *Early learning strategies for developing computational thinking skills*. https://www.gettingsmart.com/2018/03/18/early-learning-strategies-for-developing-computational-thinking-skills/

Uzair, M., & Jamil, N. (2020, November). Effects of hidden layers on the efficiency of neural networks. *2020 IEEE 23rd International Multitopic Conference (INMIC)*,1–6. IEEE. https://doi.org/10.1109/INMIC50486.2020.9318195

Vasquez, J. A., Sneider C. I., & Comer, M. W. (2013). *STEM lesson essentials, grades 3-8: Integrating science, technology, engineering, and mathematics*. Heinemann.

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: Pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. *Tech Trends, 60*(6), 565–568. https://doi.org/10.1007/s11528-016-0087-7

About the Authors

Paul Asunda (pasunda@purdue.edu) is an associate professor of engineering and technology teacher education in the Department of Technology Leadership and Innovation (TLI) with a joint appointment in College of Education, Department of Curriculum and Instruction (C&I), and an affiliate faculty for the Center for Advancing the Teaching and Learning of STEM (CATALYST) at Purdue University in West Lafayette, IN, USA.

Miad Faezipour (mfaezipo@purdue.edu) is an associate professor of electrical and computer engineering technology at the School of Engineering Technology at Purdue University in West Lafayette, IN, USA.

Joshua Tolemy (jtolemy@purdue.edu) is a graduate student and Prototyping Supervisor for Mechanical Engineering Technology at Purdue University in West Lafayette, IN, USA.

Milo Timothy Do Engel (mtengel@purdue.edu) is a student at Purdue University in West Lafayette, IN, USA.

This position paper was initially received on November 21, 2022. It underwent editorial rather than peer review and was accepted for publication on March 20, 2023.