Cybersecurity Education in the Age of AI: Integrating AI Learning into Cybersecurity High School Curricula

Shuchi Grover
Looking Glass Ventures
Austin, TX, USA
shuchig@cs.stanford.edu

Brian Broll
Vanderbilt University
Nashville, TN, USA
brian.broll@vanderbilt.edu

Derek Babb
University of Nebraska, Omaha
Omaha, NE, USA
dvbabb@unomaha.edu

ABSTRACT

Artificial Intelligence (AI) and cybersecurity are becoming increasingly intertwined, with AI and Machine Learning (AI/ML) being leveraged for cybersecurity, and cybersecurity helping address issues caused by AI. The goal in our exploratory curricular initiative is to dovetail the need to teach these two critical, emerging topics in highschool, and create a suite of novel activities, 'AI & Cybersecurity for Teens' (ACT) that introduces AI/ML in the context of cybersecurity and prepares high school teachers to integrate them in their cybersecurity curricula. Additionally, ACT activities are designed such that teachers (and students) build a deeper understanding of how ML works and how the machine actually "learns". Such understanding will aid more meaningful interrogation of critical issues such as AI ethics and bias. ACT introduces core ML topics contextualized in cybersecurity topics through a range of programming activities and pre-programmed games in NetsBlox, an easy-to-use block-based programming environment. We conducted 2 pilot workshops with 12 high school cybersecurity teachers focused on ACT activities. Teachers' feedback was positive and encouraging but also highlighted potential challenges in implementing ACT in the classroom. This paper reports on our approach and activities design, and teachers' experiences and feedback on integrating AI into high school cybersecurity curricula.

CCS CONCEPTS

- **Social and professional topics → K-12 education**.

KEYWORDS

cybersecurity education, artificial intelligence education, K-12 computer science education, teacher preparation, block-based programming

ACM Reference Format:
Shuchi Grover, Brian Broll, and Derek Babb. 2023. Cybersecurity Education in the Age of AI: Integrating AI Learning into Cybersecurity High School Curricula. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2023), March 15–18, 2023, Toronto, ON, Canada.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3545945.3569750

1 NEED & MOTIVATION

As the world gallops at an unprecedented pace toward increased automation, both cybersecurity and AI are becoming increasingly key aspects of computing. The rate at which cyberthreats are developing is increasing daily. The average security operations center (SOC) receives over 11,000 alerts a day, and 28% of all alerts are never addressed [3]. Like every other arena where big data are fueling the use of automation through artificial intelligence and especially the prolific AI subfield of 'machine learning' (AI/ML), cybersecurity professionals see AI/ML as the solution to growing problems of "too much data to analyze" and "poor/insufficient automation of threat detection and response processes." As AI/ML applications pervade every corner of our day-to-day lives, innovation today cannot be separated from AI. In the context of cybersecurity, AI can learn normal patterns of data and systems to help detect and protect against malware, ransom, trojan horses and other threats; and on the flip side, increasingly sophisticated abilities to manipulate images, audio/video using "deep fake" AI techniques present new challenges and cyberthreats. Although, cybersecurity education is crucial for all learners, there can be no cybersecurity conversation today without AI/ML in the mix. Building an understanding of these two fastest-moving subfields of computing in an integrated way makes cybersecurity and AI education authentic to today's reality. Providing engaging learning experiences in secondary school develops career aspirations and interest, thus making such endeavors productive and beneficial for tomorrow's citizens.

Current K-12 cybersecurity education is largely accomplished through 'digital literacy' activities in classrooms, out-of-school initiatives such as GenCyber [13] camps, or highschool elective introductory courses (not uniformly offered everywhere). These experiences typically cover Cybersecurity First Principles, the Confidentiality-Integrity-Availability (CIA) triad, concepts such as cryptography, network security, viruses and malware, and promote cyber awareness as well as cybersafety. AI/ML do not feature in these curricula. Separately and concurrently, **AI efforts in K-12** e.g. AI4K12 [24] and AI4All are gaining momentum thanks to the availability of AI services that can now run sophisticated AI/ML algorithms in a web browser. However, most of these efforts introduce AI concepts such as machine learning, classifiers and predictors, and neural networks through extant pre-trained models, explorations are not specific to any domain or context, and involve little real programming that lifts the hood on how machine learning actually happens. This leaves learners with an amorphous understanding of AI and does little to address the opacity of the "magic of AI" in learners.

Why AI in Cybersecurity Education? Given the issues outlined above, our project tackles the need for cybersecurity and

SIGCSE 2023, March 15–18, 2023, Toronto, ON, Canada

ACM ISBN 978-1-4503-9431-4/23/03…$15.00
https://doi.org/10.1145/3545945.3569750

AI/ML education for high school students in integrated and mutually beneficial ways that are authentic to today's needs. Cybersecurity issues provide an excellent context for an introduction to AI/ML, and AI/ML issues in cybersecurity empower learners with the knowledge and wherewithal to address new challenges and opportunities in cybersecurity. Lastly, AI is also considered an exciting, game-changing technology of our time that impacts all aspects of our lives. By meaningfully intertwining the learning of cybersecurity and AI/ML for learners in their early teens through engaging real-world examples, programming exercises, and playful explorations, we also tap into the potential of building interest in more young teen learners, and especially females and students from historically marginalized groups, at a time considered critical for STEM identity and interest development [23].

Even though there has been research on AI in the context of cybersecurity (such as the symposia at IEEE ICC and Globecom), such an integrated "cybersecurity+AI" education curricular effort is new and uncharted territory in the context of K-12 education. Hence, a core goal of our exploratory project was to design and seek feedback from cybersecurity teachers as well as train them on these materials so that they could then integrate them into their curricula. This experience report describes the design of a suite of curricular activities—**AI & Cybersecurity for Teens (ACT)**—that leverage the intersection of Cybersecurity & AI, highlight suitable cybersecurity contexts and real-word situations (as "hooks") for integrating AI into cybersecurity classrooms, and teach AI/ML models in ways that lift the hood on how the machine actually "learns". We believe that such deeper understanding of AI is essential to critically interrogate issues of ethics and bias in ML models, as well as develop the kind of adversarial thinking [26] that is a core skill of both cybersecurity as well as AI. We also describe teachers' feedback and experiences from two ACT PD workshops. We believe this report will be valuable in sharing these activities and materials (freely available at cyberai4k12.org) with the broader highschool CS community and be useful for both highschool Cybersecurity and AI teachers. We also hope it will provide an impetus for upgrading K-12 cybersecurity curricula.

2 RELEVANT PRIOR WORK

Cybersecurity. Even though cybersecurity education in K-12 is not widespread, there has been a growing attention to this subject in schools. In lower grades, it takes the form of cyber-awareness and cyber-safety; at the high school level, there are efforts to make cybersecurity learning more authentic with the goal of raising interest in cybersecurity careers. Designing cybersecurity curricular that are creative, socially relevant and accessible to K-12 students is a challenge [19]. A meta-analysis of 12 published papers between 2010-2021 revealed that topics covered in K-12 cybersecurity education typically include security of data, software, systems, human-behavior, organizations, and society. [4]. The meta-analysis also revealed that teachers find it challenging to teach cybersecurity as they lack age-appropriate tools and resources to teach cybersecurity related to network issues. The authors concluded that "to make cybersecurity education a success at the K-12 level, strategies and approaches need to be used to prepare teachers for teaching cybersecurity through evidence-based curriculum, teaching materials, tools, technologies and other resources. Teachers must be provided with ongoing professional development."

AI/ML. The current "AI in schools" movement has received growing attention since the 2019 release of the "Big Ideas" for K-12 AI education [24]. Many AI for K12 teaching materials reside as curated lists of "resources" maintained by AI4K12 and CSTA. Most of these involve the use of extant pre-trained AI models that students can "play" with, such as *Quick, Draw!*, *ThisXDoesNotExist*, and Google's *Tensorflow Playground*. There are also efforts underway to build secondary school curricula that enable students to interact with AI [7], including some courses focusing on societal impact of AI [18] and AI literacy with a focus on machine learning [15].

3 DESIGN OF ACT SUITE OF ACTIVITIES

This section outlines ACT goals, pedagogy & design philosophy, NetsBlox as a block-based programming tool to introduce networking and AI/ML, a mapping of ACT activities with cybersecurity and AI/ML topics, and a description of 3 sample activities.

3.1 Goals of ACT activities

The design of relevant AI activities to be embedded within cybersecurity learning are guided by a set of "big ideas" or learning goals for our innovative designs. Through ACT activities, students will develop a sense for: (1) How AI/ML plays a role in real-world cybersecurity issues, (2) Key AI/ML techniques, (3) ***"How"*** the machine learns, (4) Data and its features, (5) Optimization (as learning), (6) Issues related to generalization in AI/ML models, (7) How bias can impact aspects (and phases) of machine ML, (8) Issues of ethics, and (9) Adversarial Thinking i.e. whenever we discuss cyber detection using AI, *also think of how it can be fooled.* Additionally, since teachers typically find it difficult to integrate programming into cybersecurity curricula [4], our goal was also to leverage the affordances of NetsBlox [2], an extension of the block-based programming environment Snap! that includes features for easy network programming, for the AI (as well as some non-AI-related cybersecurity activities such as cryptography; see Table 1).

3.2 Pedagogy & Design Philosophy

We draw inspiration from past work in turtle geometry by pioneers such as Abelson and diSessa that made sophisticated ideas in mathematics and physics accessible to younger learners through leveraging multiple representations and programming [1]. Our innovation lies in resolving the challenge of making somewhat complex AI/ML topics (that involve advanced mathematics) accessible to students that have not yet learnt (or may not be interested in learning) those mathematical ideas.

Through making real-world connections [10, 11] along with the interplay of mathematics apparent in learner-friendly ways, we contend that students will build deeper and better intuitions of ML techniques. Current cybersecurity curricula that typically use problem- and project- based learning, game-based learning, and case studies using real-world cybersecurity scenarios, but typically do not involve programming [4]. ACT uses all these pedagogies in addition to programming. In order to help learners at all levels of interest and ability succeed in engaging with non-trivial ML algorithms, we employ "levels of abstraction" [5, 25] as a scaffolding tool. We introduce the basic algorithm in pseudocode, then provide "subgoal"-inspired [16] design blocks for implementing the algorithm and Parson's problems [6, 17] for code completion that have been shown to scaffold programming in introductory courses [8]. At the lowest level, we provide the entire code that implements the algorithm. The level(s) provide teachers with options to help learners engage at varying levels, while still ensuring that they all leave with an intuition of how the ML technique works.

3.3 Enabling Technology: Block-Based Programming in NetsBlox

NetsBlox is a block-based programming environment based on Snap*!* [21] which was designed to make networking & distributed computing concepts accessible to novices. It has also been used to introduce other advanced CS topics, such as the Internet of Things and robotics, to young learners [12, 14, 22]. The main technical extensions of Snap*!* are the networking primitives for *message passing* and *Remote Procedure Calls* (RPCs). Message passing allows users to send messages to remote computers via the internet. This enables students to make engaging multi-user applications.

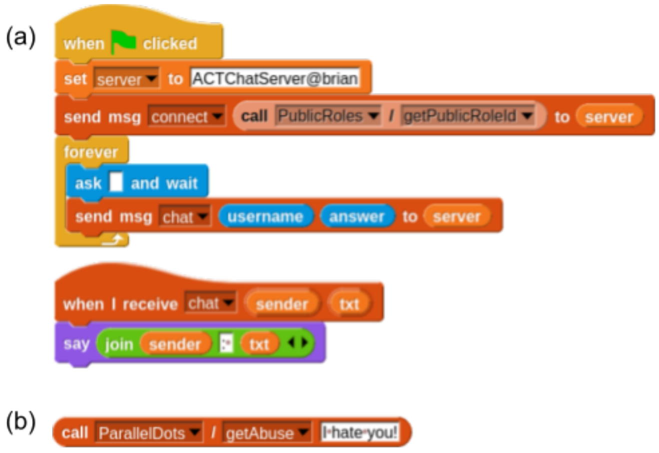
RPCs allow users to invoke functionality on the NetsBlox server. Conceptually, they are similar to custom blocks except the code is (usually) implemented in another language and runs on a remote server. RPCs with similar functionality are grouped together into NetsBlox *Services*, such as Google Maps, climate data, and the Parallel Dots API for sentiment analysis (used in the ACT cyberbullying project). Documentation for RPCs is integrated into NetsBlox and contains helpful information with examples.

As an introduction to programming in NetsBlox, ACT starts with a chat application project. The chat application is a simple message passing application with a single server and many (soon to be connected) clients. The instructor and students create their own projects and the students are first tasked to send a message to the teacher's project. After sending a message to the teacher, they extend their projects so they can view the messages of their peers from their own project. This involves adding a registration step with the server so the server can then relay chat messages to the list of registered client projects.

An example chat client is shown in Figure 1. This client connects to a hard-coded server address, "ACTChatServer@brian", and then repeatedly sends user input to the server. When it receives a chat message, it simply displays the message on the screen. Not only is this project simple (requires fewer than 20 blocks for the client) but it also provides a foundation from which other cybersecurity topics can be explored including Denial of Service attacks and identity spoofing. This project also can be easily connected to cryptography and/or cybersafety by encrypting messages before sending them or impersonating another user by changing the "sender" field.

3.4 ACT Activities Suite

Guided by the goals and big ideas described in Section 3.1 we started our work by identifying real world, topical cybersecurity issues such as bots or Denial of Service (DoS) attacks, or phishing (along with related articles or videos to serve as "hooks" in a learning setting). We then created a mapping between candidate activities and identifying linkages to cybersecurity ideas as well as AI/ML concepts contextualized by each of these issues (Table 1). This mapping was refined and put into a coherent sequence as activities were iteratively designed. We also added a few introductory programming experiences in NetsBlox with projects related to cryptography and creating/decoding ciphers. These did not have any AI connection; the goal was to (a) provide a gentle introduction to the features of Snap! and NetsBlox block-based programming and (b) introduce students to the chat app through encryption in chat messages so that we could build on that foundation and extend the chat app for the DoS and cyberbullying programming activities (that connect to Rule-based AI and Decision Tree AI models respectively). Next we describe 3 of the AI activities in some detail.



```
when clicked
set server to ACTChatServer@brian
send msg connect call PublicRoles / getPublicRoleId to server
forever
    ask and wait
    send msg chat username answer to server

when I receive chat sender txt
say join sender : txt

call ParallelDots / getAbuse I hate you!
```

Figure 1: a) Completed client code for a chat app; b) An example RPC used to check if text is abusive.

3.4.1 *Decision Tree Building.*

The first machine learning algorithm that we implement is an algorithm for building a decision tree from data. This activity starts by trying to first explore a synthetic Twitter account dataset using CODAP [9] and then implement a simple set of rules (composed largely of just `if` statements) to predict if an account is a human or bot. During the process, students iteratively explore how different features are related to the labels i.e., "bot" or "not bot" of the data and then choose a way to split the data that seems to best separate the bots from the human accounts.

After creating a rule-based classifier, we reflect on how these sets of `if` statements could be viewed as a decision tree. In a metacognitive step, we reflect on our process of manually creating rules to classify the data and consider how this could be automated. After formalizing it with pseudocode, students then complete a decision tree building algorithm presented as a Parson's problem—first at a higher level of abstraction where they use subgoals instead of primitive blocks, and then implement the subgoals in code. We use the visualization of the classifier (Fig. 2) to also engage students in adversarial thinking and discuss how the model could be fooled.

3.4.2 *Find the Minimum Game.*

"Find the Minimum Game" is an interactive game (Fig 3) to introduce the idea of learning through minimizing error (optimization) and the gradient descent ML algorithm[20] often used in neural networks. Students embody an optimizer and try to find the minimum of an unknown/invisible function/graph. At every click on the blank screen, an arrow is shown at the mouse x-coordinate and y-value of the function. The arrow shows the slope of the function and directs the player toward the minimum.

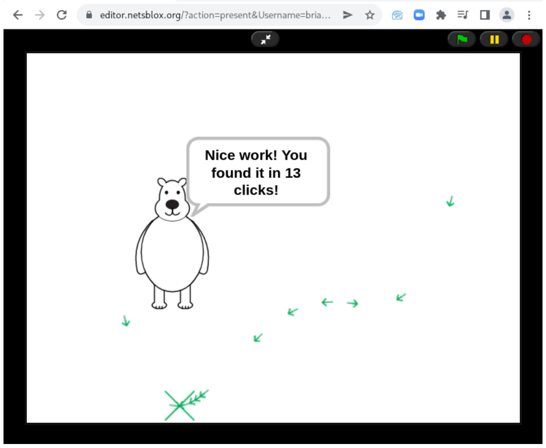

editor.netsblox.org/?action=present&Username=bria...
Nice work! You found it in 13 clicks!

Figure 3: Find the Minimum Game to explore optimization.

3.4.3 *Registration Bot Detection with Gradient Descent.*

This activity uses detection of a "registration bot" (based on mouse movement data) as a cybersecurity context for using gradient descent. Given mouse movement data of humans and bots while registering for a mock website (generated from an earlier activity), students create a ML model to classify a registration as performed by a human or a bot. First, we explore the dataset and choose a way to represent the data such as its length and average speed. Next, we come up with a simple way to classify the data points where we just learn an "importance number" (or weight) for each of the features and then combine them to get the prediction. Finally, we connect this with the earlier "Find the Minimum" activity and formalize the training algorithm through pseudocode, subgoals, then a Parson's problem.

Unlike projects in many existing AI/ML curricula, this activity introduces students to the entire lifecycle of designing an ML model. As a result, students are able to not only build intuition through hands-on experience with ML but also explore cybersecurity aspects more deeply. For example, students may try to understand the model and improve their registration bot to fool the model. Alternatively, students can perform a data poisoning attack and try to generate human data that behaves more like a bot to make the model misclassify their current bot. These types of extensions both exercise students' adversarial thinking and help them build a deeper intuition about ML concepts themselves.

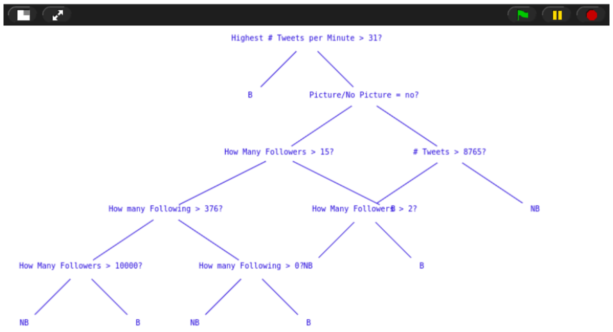

Highest # Tweets per Minute > 31?
B
Picture/No Picture = no?
How Many Followers > 15?
# Tweets > 8765?
How many Following > 376?
How Many FollowerB > 2?
NB
How Many Followers > 10000?
How many Following > 0?NB
B
NB
B
NB
B

Figure 2: Decision tree learned from a Twitter bot dataset.

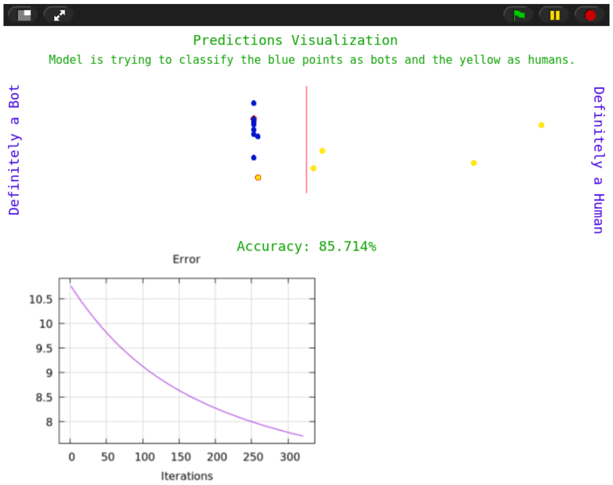


Figure 4: Visualization of model performance while training registration bot classifier.

4 PILOTING ACT WITH TEACHERS

Pilot Workshop & Feedback. We first conducted a 15 hour pilot online teacher workshop over a period of 4 sessions in Fall 2021 with 7 highschool teachers from across the US (6 Cybersecurity, 1 AI), with the stated goal of gathering feedback. 6 teachers completed the pre-post survey. Post-survey responses suggested that teachers found the pilot activities intertwining AI and cybersecurity to be suitable, innovative, and helpful for their own learning. Teachers suggested simpler activities to introduce NetsBlox which prompted the addition to Project 0—the early cryptography activities (Table 1). Other activities were also refined to add more levels of scaffolding.

The mean ranking of ACT on the innovativeness of the ACT curriculum (1=Not at all innovative; 5=Very innovative) was 4.5. Mean teacher rating on the appropriateness of the intertwining of AI & Cybersecurity in the activities (1=Not at all connected/Does not make sense; 5=Very well connected/Makes sense) was 4.3. Mean teacher comfort level with AI topics changed from 2.7 to 3.7 (1=Not at all familiar (it's totally new to me); 5= I'm an expert (I teach it to middle/high school or older students)). In rating specific ACT activities with choices: *Not a great activity (least favorite)*; *Good idea but needs a lot of improvement*; *Almost there- good activity/experience*;*Great Activity (among my faves)*; *Did not attend session*, responses to most activities was mostly "Almost there" or "Great Activity" Two or more teachers indicated that message passing (chat), DDoS, CyberBullying, Decision Tree/TwitterBot, Gradient Descent, were a "Great Activity (among my faves)". The most popular was the cyberbullying/sentiment analysis activity with 4 teachers marking that as "Great"; next was the TwitterBot/Phishing/Decision Tree activity with 3 teachers marking that as "Great Activity".

Scenario "Hook" for Discussion	**NetsBlox Programming Project**	**Cybersecurity Topics**	**AI/ML Topics**
Most common passwords (avoid these!)	Project 0: Intro to programming; Cracking the Caeser cypher	Cryptography, passwords, ciphers, hacking	
Best encrypted Chat App	Project 1: Build a text-messaging app - Intro to distributed computing in NetsBlox	Distributed computing; chat applications; network security; message passing	
Amazon AWS DDoS attack & disruption (Scenario examples)	Project 2: DDoS example & rule-based systems (Use-Modify-Discuss pedagogy that builds on initial chat app project)	DDoS, and DDoS prevention techniques	Rule-based AI systems
Cyberbullying detection - Time article on cyberbullying app - Tinder will now use AI to ask if you really want to send an offensive msg	Project 3: Sentimental writer (also introduces call block) or Moderated Chat Client (Parsons problem)	Message passing; cyberbullying (and how it is part of cybersecurity)	Use of Parallel Dots NLP API to classify chat text as abusive or not (or neutral).
Twitter bots are malicious actors that amplify negative sentiment & misinformation (e.g. about candidates in US elections).	Project 4: Twitter Bot classification (1) Discussion of features and extend Project 2 (2) Explore actual Twitter Bot dataset using CoDAP	Online bot (or phishing emails) classification	Intro Decision Trees. Introduce learning the decision tree (and machine learning)
Phishing scams. Another classification exercise used to introduce error/bias/optimization	Project 4a: Phishing data - building intuitions about data/features/classification	Phishing	Decision Trees; intuition about data

How does the machine "learn"?
How does classification and prediction happen when features cannot be enumerated/explicitly defined?
	Project 5: Play the Gradient Descent game; Engage in abstractions of GD algorithm; examine GD code	Classification in cybersecurity	Neural Networks;
			Optimization (as learning);
			Gradient Descent

Autonomous vehicles fooled by road signs that are changed subtly	Project 6: Adversarial Example game; lift hood on how to generate a data point that is classified as being part of a dataset; and how a generator and discriminator are coded	Deep fakes; cyber integrity issues	Adversarial examples
Realistic content generation via AI: Deepfakes, GPT-3 language generation, and https://thisxdoesnotexist.com/.	Project 7: Circle GA(N) exercise.	Deep fakes; cyber integrity issues	GANs

Table 1: ACT NetsBlox programming activities described with context hooks, and links to cybersecurity & AI topics

One teacher from our cohort incorporated a few ACT activities into their Fall'22 elective course, and 2 others are planning to do so in the Spring'23 term. Preliminary analyses of data just received from the Fall 2022 implementation (N=9) suggest that students found the activities to be enjoyable and are eager to learn more. They found the Chat app and its derivative activities (DDoS and Cyberbullying) to be most enjoyable. Our presentation at the SIGCSE 2023 conference will share more details on student experience.

4.1 Summer 2022 Workshop

We recruited 7 teachers for a week-long online summer workshop, *with the additional explicit understanding that the teachers would implement the activities with students in 2022-23 academic year and share their experiences from classroom implementation.* This condition made recruitment difficult (as teachers are still dealing with Covid-related challenges), and of the 7 teachers, 5 participated and completed pre-post surveys. We created a Slack channel for ongoing conversations and sharing resources and recorded Zoom sessions.

Teacher Feedback and Comments. Overall the response to the second workshop was also overwhelmingly positive. In response to the question: *How important is it to understand AI and machine learning as it is related to cybersecurity?* 4 teachers marked ***extremely important*** and 1 teacher, ***very important***. Teacher ratings and ranking to post-survey questions are presented in Table 2.

5 LESSONS LEARNED & NEXT STEPS

Overall, teachers in both workshops see the value of integrating AI as it applies to issues of cybersecurity (mitigating or causing) and appreciate having innovative and (mostly) easy-to-use curricular activities that integrate realistic cybersecurity+AI activities.

How much do you agree with the following statement 1=Strongly disagree; 2=Somewhat disagree; 3=Not sure; 4=Somewhat agree; 5=Strongly Agree	**Mean (out of 5)**
The lessons/activities in the ACT curriculum do a good job of intertwining AI and Cybersecurity	5
The designed activities in the ACT curriculum are innovative	4.8
AI is an emerging discipline and it plays an increasingly important role in cybersecurity	4.8
Inclusion of relevant AI topics in a cybersecurity curriculum can be managed	5
Teaching/coaching AI in cybersecurity will make my work as a teacher harder	2.6
I am concerned about not having enough time to prepare to teach AI in cybersecurity	3.4
I am concerned about not having enough resources to teach AI in cybersecurity	2.8
I can explain how AI fundamental/techniques relate to each of the Cybersecurity Principles/CIA Triad and its importance	4.2
I can describe the relationship of AI/ML techniques and issues of cybersecurity to my students in a way that is relevant to them	4.4
I know enough about how AI/ML techniques are implemented to teach them to students	3.6
I have a good understanding of illustrations, examples, analogies, etc., to use to help my students understand what I want to teach them about how AI & cybersecurity connect	3.6
I have a sufficient number of lessons/activities to use to teach AI / ML techniques and their relationship to cybersecurity to my students	4
Given what I know about ACT AI in cybersecurity activities, I have a very good idea about where this will fit in my cybersecurity curriculum	4.6
The ACT lessons/activities will fit nicely in my cybersecurity curriculum so I will not have to make (m)any changes to them to make them work	3.4
I can see how to connect AI/ML topics in cybersecurity content to meet the academic standards I need to meet	4.4

How likely is it that you will use the cybersecurity curriculum/individual activities	**Median Likelihood**
(Overall) How likely is it that you will teach the content learned in the ACT workshop in your cybersecurity class in the coming academic year?	97
Ciphers and Cryptography	100
Message Passing and Moderated Chat app	100
Denial of Service (and mitigating DoS attack)	88
Cyberbullying & Sentiment Analysis (Chat app extension)	99
Data exploration in CODAP	80
Decision Tree to classify Twitter Bots and phishing emails	80
Creating a registration Bot	98
Classifying a registration bot using gradient descent	70
Generating an adversarial example (registration bot)	70
Discussing deep fakes as topic	100
Coding Generative Adversarial Networks in NetsBlox	75

Table 2: How likely are teachers to use the various ACT activities (probability rating between 1-100)

Even though cybersecurity is taught as an elective and is not an AP course, teachers are aware, and sometimes wary, of the effort it will take to integrate AI activities into their curricula (since teachers often use set offerings from TeachCyber.org, CodeHS, or Cyber.org).

There is a need for high-quality support materials in addition to the lesson activities themselves in order to address teachers' concerns about teaching these topics new to them.

How deep should activities go into AI/ML code and implementation? Our goal was to provide multiple levels of engagement and options for how deep teachers could go with students. Some teachers are not too keen on integrating activities such as gradient descent and GANs and believe that these activities were "getting too much in the weeds of AI"; they could not see their relevance for an introductory cybersecurity course. Others felt that they could engage students at different levels based on interest & ability.

We surmise that the disparity in teacher reactions is due to the ground realities in their contexts and student populations, as well as their level of familiarity with programming and/or NetsBlox, which may get in the way of integrating programming activities. That said, *all teachers in our PD appreciated NetsBlox capabilities to introduce networking apps such as the chat app (which was the teachers' most-liked activity), especially since it could be extended to cover cyberbullying and encryption (which were also very well-liked).*

Given time, teachers' (and students') comfort with programming is a surmountable barrier since all the programming activities use concepts at an introductory high school level. The only exception is working with multi-dimensional lists for data (such as Twitter bot data or data returned from the ParallelDots service for sentiment analysis). Teachers urged that we create dedicated activities to familiarize students with list processing in Snap!/NetsBlox.

We believe all teachers could at least start making students aware of how AI is playing a role in detection of cybercrimes, sharing with students examples such as automated Spam email classification in gmail or classification of hate or other speech on social media, and introducing them to adversarial thinking. These can be accomplished through discussions and non-programming activities.

Conclusion & Next Steps. Despite the fact that we are in uncharted territory (and perhaps the first to be working on the integration of AI in cybersecurity) and some sobering feedback related to expected adaptations required in the classroom, teacher feedback on the need and innovativeness of the ACT suite of activities suggests that we have made a good start in designing an interesting set of curricular materials that teachers find promising. Our next steps involve gathering data from teachers' classroom implementation in the Fall and Spring terms of AY 2022-2023, and refining activities accordingly. We also plan to create a ACT curriculum package for free and open use by teachers, while concurrently developing and running additional teacher PD workshops for interested teachers. Lastly, we hope to engage highschool AI teachers who may find value in using the cybersecurity examples to introduce students to programming activities teaching ML concepts such as decision trees and optimization to classify malicious bots and emails, as well as generating adversarial examples.

6 ACKNOWLEDGEMENTS

This material is based upon work supported by the NSF (#2113803).

REFERENCES

[1] H. Abelson and A. DiSessa. *Turtle geometry: The computer as a medium for exploring mathematics.* MIT press, 1986.

[2] B. Broll, A. Lédeczi, P. Volgyesi, J. Sallai, M. Maroti, A. Carrillo, S. L. Weeden-Wright, C. Vanags, J. D. Swartz, and M. Lu. A visual programming environment for learning distributed programming. In *Proceedings of the 2017 ACM SIGCSE technical symposium on computer science education*, pages 81–86, 2017.

[3] M. Brozek. Forrester study: The 2020 state of security operations, 2020.

[4] W. Chen, Y. He, X. Tian, and W. He. Exploring cybersecurity education at the k-12 level. In *SITE Interactive Conference*, pages 108–114. Association for the Advancement of Computing in Education (AACE), 2021.

[5] A. Csizmadia, B. Standl, and J. Waite. Integrating the constructionist learning theory with computational thinking classroom activities. *Informatics in Education*, 18(1):41–67, 2019.

[6] P. Denny, A. Luxton-Reilly, and B. Simon. Evaluating a new exam question: Parsons problems. In *Proceedings of the fourth international workshop on computing education research*, pages 113–124, 2008.

[7] S. Druga, N. Otero, and A. J. Ko. The landscape of teaching resources for ai education. In *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 1*, ITiCSE '22, page 96–102, New York, NY, USA, 2022. Association for Computing Machinery.

[8] B. J. Ericson, L. E. Margulieux, and J. Rick. Solving parsons problems versus fixing and writing code. In *Proceedings of the 17th Koli Calling International Conference on Computing Education Research*, pages 20–29, 2017.

[9] W. Finzer. Common online data analysis platform (codap). *Emeryville, CA: The Concord Consortium.[Online: concord. org/codap]*, 2016.

[10] J. Gainsburg. Real-world connections in secondary mathematics teaching. *Journal of Mathematics Teacher Education*, 11(3):199–219, 2008.

[11] S. Grover, N. Jackiw, and P. Lundh. Concepts before coding: Non-programming interactives to advance learning of introductory programming concepts in middle school. *Computer Science Education*, 29(2-3):106–135, 2019.

[12] D. Jean, B. Broll, G. Stein, and Á. Lédeczi. Your phone as a sensor: Making iot accessible for novice programmers. In *2021 IEEE Frontiers in Education Conference (FIE)*, pages 1–5. IEEE, 2021.

[13] T. Ladabouche and S. LaFountain. Gencyber: Inspiring the next generation of cyber stars. *IEEE Security & Privacy*, 14(5):84–86, 2016.

[14] Á. Lédeczi, H. Zare, and G. Stein. Netsblox and wireless robots make cybersecurity fun. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, pages 1290–1290, 2019.

[15] I. Lee, S. Ali, H. Zhang, D. DiPaola, and C. Breazeal. Developing middle school students' ai literacy. In *Proceedings of the 52nd ACM technical symposium on computer science education*, pages 191–197, 2021.

[16] B. B. Morrison, L. E. Margulieux, and M. Guzdial. Subgoals, context, and worked examples in learning computing problem solving. In *Proceedings of the Eleventh Annual International Conference on International Computing Education Research*, ICER '15, page 21–29, New York, NY, USA, 2015. Association for Computing Machinery.

[17] D. Parsons and P. Haden. Parson's programming puzzles: a fun and effective learning tool for first programming courses. In *Proceedings of the 8th Australasian Conference on Computing Education-Volume 52*, pages 157–163, 2006.

[18] B. H. Payne. An ethics of artificial intelligence curriculum for middle school students. *MIT Media Lab Personal Robots Group. Retrieved Oct*, 10:2019, 2019.

[19] P. Rowland, A. Podhradsky, and S. Plucker. Cybher: A method for empowering, motivating, educating and anchoring girls to a cybersecurity career path. In *Proceedings of the 51st Hawaii International Conference on System Sciences*, 2018.

[20] S. Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[21] Snap!: a visual, drag-and-drop programming language. http://snap.berkeley.edu/snapsource/snap.html.

[22] G. Stein and A. Lédeczi. Enabling collaborative distance robotics education for novice programmers. In *2021 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 1–5. IEEE, 2021.

[23] R. H. Tai, C. Qi Liu, A. V. Maltese, and X. Fan. Planning early for careers in science. *Science*, 312(5777):1143–1144, 2006.

[24] 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 AAAI conference on artificial intelligence*, volume 33, pages 9795–9799, 2019.

[25] J. Waite, P. Curzon, W. Marsh, and S. Sentance. K-5 teachers' uses of levels of abstraction focusing on design. In *Proceedings of the 12th Workshop on Primary and Secondary Computing Education*, pages 115–116, 2017.

[26] N. Young and S. Krishnamurthi. Early post-secondary student performance of adversarial thinking. In *Proceedings of the 17th ACM Conference on International Computing Education Research*, pages 213–224, 2021.