DeapSECURE Computational Training for Cybersecurity: Progress Toward Widespread Community Adoption

Wirawan Purwanto
Old Dominion University
wpurwant@odu.edu

Bahador Dodge
Old Dominion University
bdodg001@odu.edu

Karina Arcaute
Old Dominion University
karcaute@odu.edu

Masha Sosonkina
Old Dominion University
msosonki@odu.edu

Hongyi Wu
The University of Arizona
mhwu@arizona.edu

ABSTRACT

The Data-Enabled Advanced Computational Training Program for Cybersecurity Research and Education (DeapSECURE) is a non-degree training consisting of six modules covering a broad range of cyberinfrastructure techniques, including high performance computing, big data, machine learning and advanced cryptography, aimed at reducing the gap between current cybersecurity curricula and requirements needed for advanced research and industrial projects. Since 2020, these lesson modules have been updated and retooled to suit fully-online delivery. Hands-on activities were reformatted to accommodate self-paced learning. In this paper, we summarize the four years of the project comparing in-person and online only instruction methods as well as outlining lessons learned. The module content and hands-on materials are being released as open-source educational resources. We also indicate our future direction to scale up and increase adoption of the DeapSECURE training program to benefit cybersecurity research everywhere.

KEYWORDS

Parallel computing, big data, machine learning, cybersecurity, non-degree training, hands-on, online training

1 INTRODUCTION

The world that we live in today relies heavily on connected computers and mobile devices. Furthermore, many physical instruments are now connected to form the “internet-of-things”. As such, the significance of cybersecurity cannot be underestimated. Cybersecurity in practice consists of many different tools, techniques and policies to protect and defend computing systems from potential attacks, as well as detect and mitigate attempted attacks. The research and development of novel cybersecurity tools and techniques have become more dependent on advanced cyberinfrastructure (CI) due to increasing complexity of the cyber systems being defended, as well as the growing intensity and sophistication of cyberattacks. As an area of study, cybersecurity is a multi-disciplinary field which draws from areas such as computer science and engineering, information technology, mathematics, business, law, social science, psychology, and more. At present, however, standard curricula used in many colleges and universities lack inclusion of advanced CI techniques to strengthen cybersecurity analysis, research, and development. This lack exists only in cybersecurity as a stand-alone discipline, but also in many of its “upstream” disciplines mentioned earlier. As a result, skill and knowledge gaps exist among students who are being trained to work in research areas related to cybersecurity.

With funding from the National Science Foundation (NSF), the School of Cybersecurity at Old Dominion University (ODU) developed DeapSECURE (short for *Data-Enabled Advanced Computational Training Platform for Cybersecurity Research and Education*) as an innovative, non-degree CI training program tailored for cybersecurity students and researchers. The DeapSECURE training program was created to address major curricular gaps in cybersecurity education in the areas of advanced computing. This non-degree training program consists of six modules that cover a broad range of CI topics: high performance computing (HPC), big data analytics (BD), neural networks (NN), machine learning (ML), parallel programming (PAR) and cryptography for privacy-preserving computation (CRYPT) [16]. These techniques are used extensively in state-of-the-art cybersecurity research and practice.

The goals, approach and philosophy of the DeapSECURE training program has been described in detail our earlier paper [18]. DeapSECURE emphasizes hands-on experience to fortify and connect theoretical materials with real-world applications. The primary goal of DeapSECURE lessons is to “crack open” the tough nuts of CI methods and concepts through practical use cases and codes. Application examples in the modules are carefully selected to engage learners in the field of cybersecurity and aim to train current and future researchers, engineers and practitioners with advanced techniques and skills necessary to carry out cybersecurity research and industrial projects. In a way similar to that adopted by the Carpentries [3], we leverage real cybersecurity problems (i.e. scenarios) and datasets as a way to introduce and practically learn the CI techniques through workshops. The CI technique unfolds as the lesson progresses through a series of computer codes employed to work out the solution. Quite frequently, important concepts are directly demonstrated to learners by these codes, followed by the explanation on the spot. All the lesson materials are available openly on

https://doi.org/10.22369/issn.2153-4136/15/1/1

the DeapSECURE website [16]. Workshops based on DeapSECURE lesson materials are not meant to replace comprehensive educational means such as semester-long courses; neither are the lessons intended to serve as a complete overview or an in-depth treatise on the CI topics. Rather, they are meant to give an initial practical exposure to CI and to provide learners with the first "stepping stones" to their further learning of CI for their own purposes (e.g. research). Exercises and activities in the lessons encourage learners to try, explore, and experiment with the CI tools. This training program has been in continuous development since 2018. By 2022, the lesson modules have been improved and road-tested through at least four workshop iterations.

In this paper, we present the complete conversion of all the DeapSECURE lessons from the in-person format to fully virtual delivery: the changes implemented to adapt the lessons for virtual workshops, the experience of conducting the workshops online, and the learners' feedback and reaction to the online format. We will also describe our effort of training students to become workshop teaching assistants (WTAs) and content developers, which we consider to be an important next-step to sustain the training program beyond NSF funding. The rest of the paper is organized as follows: Section 2 describes the first two years, when training was conducted in-person only. Section 3 outlines training adaptation to the online delivery and introduces our approach to training WTAs to assist the development process. In Section 4, we note on the statistics of the learners and their perception of the transition of DeapSECURE from in-person to online training. Finally, we discuss the availability of open-source training modules (Section 5) and a future roadmap in Section 6.

2 FIRST- AND SECOND-YEAR DEVELOPMENT

First year (Y1, 2018–2019 academic year)—The lesson modules of DeapSECURE were developed from scratch in the first year of the program. The unique component of DeapSECURE—combining exposure of state-of-the-art research and hands-on training on CI techniques—was developed through intensive engagement with cybersecurity researchers at ODU [18]. The hands-on component was designed with the use of HPC (i.e., parallel computers) in mind, since HPC will allow students to eventually scale up their computation when working with many challenging, real-world cybersecurity research problems. The training materials were developed collaboratively by the project's principal investigators (PIs) and WTAs using Gitlab for codes, lessons and data repositories, as well as Google Drive for document sharing and workflow coordination among team members. Three Ph.D. students assisted in the development of the lessons and the hands-on parts of the workshops. Assessments had been an integral part of the training program since its inception, utilizing pre- and post-workshop surveys, as well as focus group interviews. Findings from assessments helped drive continuous improvement of the program.

The six modules were offered twice in Y1, first as a series of workshops during the 2018-2019 academic year, and second as a week-long summer institute in June 2019. (These were in-person events, but the entire sessions were recorded with the support of ODU Distance Learning for learners' review and/or future purpose of creating a repository of video learning resources.) Each workshop lasted for three hours, which included a 30-minute cybersecurity research presentation by ODU faculty members. The bulk of the workshop consisted of hands-on introduction of CI methods using the participatory live coding method as adopted by the Carpentries [12], where the instructor narrated the method and typed on his/her own computer screen, and learners were to follow the same steps on their own computers, following the instructor's projected screen. The hands-on activities of the workshops were at that time carried on ODU's Turing HPC cluster, primarily on the UNIX terminal interface. The first-year program, the contents of the workshop materials, the demographic of the learners, and the initial assessment results were described in detail in Ref. [18].

The training program was widely advertised to ODU student body, particularly to cyber-related fields (cybersecurity, electrical and computer engineering, computer science, and modeling & simulation study programs). There were close to 50 sign-ups received; they were all accepted to the program. During the academic year, student attendance varied greatly through the semester (between 11 and over 30) based on their course workloads. Participation in summer institution was more consistent (17–21), presumably due to the absence of other commitments and contiguous workshop days. The workshops were generally well received, and students were exposed to state-of-the-art cybersecurity research topics and modern CI methods, both of which were not in the students' general awareness prior to this training. There were notable challenges in the hands-on sessions, however, due to the diversity of the participants' backgrounds as well as their computer programming experiences [17]. In particular, the command-line interface posed difficulty for many learners, who had not been familiar with such a mode of interaction with computers.

Second year (Y2, 2019–2020)—Key changes were introduced to the lessons and the workshop delivery [17], taking the lessons learned from the first year's workshop experience. Firstly, the modules were grouped into two distinct groups: (1) compute-intensive modules (HPC, CRYPT, PAR); (2) data-intensive modules (BD, ML, NN). The data-intensive modules were completely rewritten to use Pandas [13] as the data analytics toolkit (in Y1, the BD module used PySpark, which is a more difficult framework to use), together with scikit-learn [14] and Keras [4]. In an effort to streamline the lessons, a single cybersecurity use case was used for the three lesson modules, leveraging the SherLock smartphone security dataset [11]. This resulted in a more focused attention to three cybersecurity themes as the backdrop to introduce the CI techniques in the lessons: (1) spam email analysis; (2) computation with homomorphically encrypted data; (3) mobile device security. Additional hands-on sessions named "hackshops" were introduced (one session for every workshop) to provide opportunities for further hands-on learning, guided by the WTAs. Table 1 shows the lesson modules of DeapSECURE after all the changes had been completed in Y2. All the DeapSECURE lessons and their resources are available openly at DeapSECURE's project website [16].

While the training was still widely announced to any interested students at ODU, acceptance to the workshop was limited to those who had experience in writing simple computer programs (100 lines or less). This resulted in a smaller cohort at the beginning. The workshop format, structure, and length remained the same as the previous year. The workshop dates were compressed towards

Table 1: The DeapSECURE Lesson Modules (since Fall 2019)

Module	Lesson Description	Hands-on Activities	Toolkits
HPC	Introduction to HPC and how to access, use and program HPC systems	Analyzing countries of origin from a large collection of spam emails; using parallel processing on HPC to speed up data processing	UNIX shell commands, SLURM
CRYPT	Advanced cryptography for privacy-preserving computation	AES ciphertext cracking; “King Oofy” privacy-preserving census; Paillier encryption of bitmap image data	AES-Python [19], Python-paillier [5]
PAR	Parallel programming with MPI	Parallelization of image Paillier encryption	mpi4py [6], Python-paillier
BD	Big data (BD) analytics	Processing, cleaning, analyzing, and visualizing large SherLock dataset	Pandas, Matplotlib, Seaborn
ML	Machine learning (ML) modeling	Classification of smartphone apps based on system utilization data using classic ML methods	scikit-learn [14]
NN	Neural networks (NN) for deep learning modeling	Building neural networks to classify smartphone apps	TensorFlow [2] and Keras [4]

the beginning of the semesters (amounting to 3 workshops per semester) in an effort to improve retention. We were able to secure a large classroom with tables for collaborative work in small groups, which greatly improved hands-on learning. For the data-intensive module, we devised a hackish way to run Jupyter (a web-based interactive Python environment) on Turing HPC compute nodes and forward the output to learner's computers. While this was a great improvement over using vanilla Python / IPython interface, the set up procedure was very challenging for most learners, resulting in lost time. The assessment results were discussed in Ref. [17], comparing attendance and a subset of knowledge acquisition from both the first and second years. The hands-on part of the workshop was particularly well received by many learners. While there were indications of somewhat better outcome in the second year (e.g. attendance, learner's satisfaction rate), we still noticed challenges particularly in the area of knowledge acquisition from the workshops.

3 THIRD-YEAR DEVELOPMENT: FROM IN-PERSON TO ONLINE WORKSHOPS

The COVID-19 pandemic hit shortly after the second-year workshop series was completed. This forced the DeapSECURE team to change the structure and format of the workshops and make them ready for online delivery. The team conducted a pilot online workshop in the summer of 2020, using Zoom videoconferencing platform for synchronous instruction, Jupyter for hands-on activities, and Slack (a group-based messaging platform) for communications among team members and learners during and after each workshop. By this time, an Open OnDemand instance [9] has been set up for the newer Wahab cluster, which enabled convenient access to Jupyter environment. Based on the experience and lessons learned from this pilot workshop, we proceeded to convert all the DeapSECURE lesson modules to the online delivery format in the third year.

3.1 Lesson Format Redesign

While it is still possible to emulate a Carpentries-style hands-on instruction using Zoom, there are several challenges with this format: (1) It is difficult for instructors to get the sense where the learners are, and whether they are able to follow or have difficulties, since most learners tend turn their cameras off and be quiet in Zoom; (2) From the past years, the full hands-on learning of a DeapSECURE module could not be completed within the 3-hour time frame of the workshop, leading to incomplete knowledge delivery. Remote learning tends to be a self-directed process, where learners needs to have more autonomy in driving their own learning process; therefore a suitable online training format should account for this, while compensating the known challenges.

In the online format, Jupyter was the platform of choice for nearly all the modules except the first one (HPC), where command-line interaction on a UNIX shell was a major and essential part of the lesson. In Y3, a major effort was spent in producing Jupyter notebooks for the online workshops, between 2–3 notebooks per lesson module. The Jupyter notebooks were an abridged version of the web-based, Carpentry-style lessons produced by this project [16]. But unlike the web-based lessons, which contain mostly completed codes, these Jupyter notebooks contain partially completed codes which are to be completed by the learners as they are going through the notebooks. In this regard, we deviated from the teaching model of the Carpentries, which typically “unfolds” the computer codes from complete scratch. (Carpentries-style lessons are like textbooks, but they are generally intended for the instructors while preparing for their teaching, although a motivated learner can definitely use these lessons to learn hands-on computing skills independently.) This is an important design consideration that we took in order to make the notebooks usable for self-paced learning. One major challenge with online hands-on workshops is that learners can easily get lost when they fall behind the instructor. In an in-person workshops, instructors can easily identify learners that face difficulties from their gestures and facial expressions—something that is very hard to sense in a virtual workshop because most learners turn off their cameras. With their own notebooks, learners would have a way to catch up the missed part. The technical steps for creating Jupyter notebooks for online workshops was described in Ref. [7].

In addition to converting the lessons to the Jupyter format, much work was dedicated to improving and tuning the parts of the lessons to fill the gaps, devise better approaches to teach the concepts and/or skills. For every lesson, we sifted through the episodes and parts and identify the most salient parts of the concepts, codes, and exercises that will be included in the notebooks. Details that are important but too long to be included in the notebooks are referenced using links to the web-based lessons. This process was done to focus learners' attention only on those critical parts of the CI knowledge and skills:

(1) The HPC module was reworked to introduce basic parallel processing of independent tasks using only shell scripts (the previous version jumped directly to using GNU parallel, which did not give learners an opportunity to observe how the domain decomposition was performed).
(2) The CRYPT module guides learners to encrypt and decrypt data using homomorphic encryption (Paillier) as well as the standard AES encryption; compares and contrasts their strengths, limitations, as well as computational costs.
(3) In the PAR module, emphasis was placed on basic MPI "verbs" such as send, receive, broadcast and barrier; then followed by the step-by-step MPI parallelization of a simple "map-reduce"-style computation.
(4) The BD module focuses on the basic data processing building blocks (e.g. select, filter, sort, groupby, aggregate operations), followed by data wrangling and exploratory data analysis.
(5) In the ML and NN module, a greater priority was devoted to the key steps in a standard machine learning workflow, neural-network model construction, as well as basic model hyperparameter tuning. Full implementation of ML and NN on HPC became optional activities for learners that are keenly interested in the method.

All of these are the indispensable, rudimentary principles of the CI methods, which are the key opener for learning and utilizing these techniques.

3.2 Online Workshop Delivery

In the **third year (Y3, 2020–2021)**, three workshops (HPC, CRYPT, PAR) were conducted throughout the academic year, whereas the three data-intensive workshops (BD, ML, NN) in the summer of 2021. The extensive work of conversion to the online format caused delay in the scheduling of the workshops. We did not offer hackshops in the third year due to limitations in time and resources. Since learners have their own copies of notebooks, we expect that they should be able to continue learning after leaving the workshops.

The online workshops were carefully planned out, including the strict time allocation for every part therein. We still used a three-hour format (not including breaks) per workshop. The three-hour instruction was broken up to three one-hour sessions with short breaks in-between, each of which was a mix of a lecture and a hands-on work on Jupyter (or UNIX shell). The 30-minute cybersecurity research guest lectures were omitted in the online workshops conducted in the third year. Instead, faculty and advanced-stage Ph.D. students gave somewhat longer lectures with an overview of the CI methods, which included a brief overview of their own state-of-the-art cybersecurity research applications.

The Zoom breakout room feature was used to conduct the hands-on sessions in smaller groups (around 4-6 learners each). Each breakout room had a WTA that guided the learners through the notebooks. The original intent of using breakout rooms for hands-on learning was to encourage learners to open up and discuss the hands-on materials; but this generally did not occur. Initially, the learners went through the notebook on their own, which resulted in very slow pace and nearly. In latter workshops, based on a learner's input, the WTAs would actually share their Jupyter screens, talking over the materials while actively working through the code cells in their own notebooks (somewhat similar, but not identical to the Carpentries, because our Jupyter notebooks contain partially completed codes). Three breakout rooms were initially defined, designated "beginner/novice", "intermediate", and "advanced". Participants were assigned to each room based on their self-assessed computing skill levels that was self-assessed by the learners when signing up for the training. Later on, this procedure was changed to allow learners to choose any breakout room that they thought was appropriate for their skill levels. This freedom turned out to be boone for some participants: they felt they were able more comfortable at learning by choosing the appropriate level.

During Y3 workshops, we employed Kahoot online quiz platform [1] to provide additional opportunities for learners to be socially involved. With Kahoot, we did ask questions that were more specific, such as specific function names or call syntax, how a certain computation or action was programmed in Python, in addition to general questions.

3.3 Training Workshop Teaching Assistants

DeapSECURE was developed and piloted at ODU with the aim of eventually serving the community of cybersecurity research everywhere in the U.S. and beyond. There needs to be an effort to produce trainers and lesson developers to pave the way for continual development of DeapSECURE lessons and for scaling the training beyond ODU. We laid the groundwork toward this by establishing a framework to onboard and train WTAs in an ongoing basis. During the project duration, we have witnessed high turnover of the WTAs, although the 1–2 core WTAs remained with the project for at least two years. To ensure continuity and fast onboarding of new WTAs, we not only utilized collaborative lesson development practices and tools but also developed initial phases of the so-called "train-the-trainer" program, in which WTAs themselves acted as trainees first, by working through the lesson modules already existing (e.g. through the same Jupyter notebooks given to the workshop learners). They are then onboarded to the collaborative development methodology (Git/Gitlab, Jupyter, Jekyll). Afterwards, they can be brought into the ongoing collaborative work of developing, improving, and/or polishing the lesson structure and contents. We have also developed project wiki to document as much team knowledge in a single place, allowing latter WTAs to pick up the existing knowledge independently. We started this WTA training in Y2, where we trained and onboarded four Ph.D. students into the role of lesson developers [17]. In Y3, many of these Ph.D. students had taken other interests or responsibilities, and we had two existing Ph.D. students along with two undergraduate students. We worked closely with these students to perform the conversion to online workshops by producing the Jupyter notebooks for learners. This process has allowed us to successfully work with short-term WTAs, who were able contribute remotely, even if they are from a different university. At the end of the fourth year, we have trained a WTA from the University of Virginia to help us complete the web-based lessons for final release. Within a couple of weeks, the student was able to meaningfully contribute to the lesson modules and be well versed in the lesson materials using Jupyter notebooks. To have student-contributors from other Virginia institutions was a forward-looking decision toward the expansion of the project activities, as described in Section 6.

4 ASSESSMENTS AND LESSONS LEARNED

Training assessments were conducted in all the training workshops conducted by the program, whether in-person (Y1 and Y2) or online (Y3) as well as in mixed mode during the Summer of 2022 (Y4, elaborated later). Collected assessment information includes demographic data, opinion questions (perception) about the workshops, and pre- (PRE) and post-workshop (POST) knowledge questions. The knowledge questions measured general, high-level knowledge on the CI topics, instead of focusing on toolkit-specific or programming issues. The questionnaires in both years were largely the same (some minor changes were implemented along the way to improve knowledge testing), which enabled us to compare the effectiveness of our mid-project changes. In this paper we will focus only on certain demographic data and learners' perception about the workshops. (Analysis and study on the knowledge questions will be a topic of an upcoming publication.) In particular, we examine two opinion questions asked of the learners in both the second and third years. This may give us an insight into the contrast between the in-person and virtual formats.

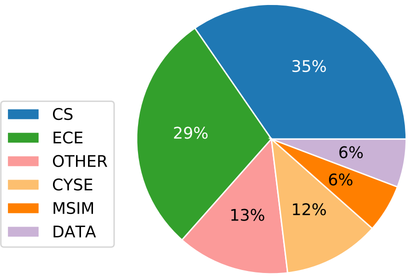

Figure 1: Distribution of Y3 workshop participants according to their academic majors. (Source: [7])

4.1 Learners' Profile

Figure 1 shows the distribution of the learners based on their academic majors in Y3. Not surprisingly, computer science (CS), electrical and computer engineering (ECE), and cybersecurity (CYSE) were the top three majors, collectively accounting for more than 75% of the learners. Other majors that are less prevalent include computational modeling and simulation engineering (MSIM) and data science (DATA). The OTHER category contains non-STEM majors and STEM majors representing less than 2% of participants per major, such as math and physics.

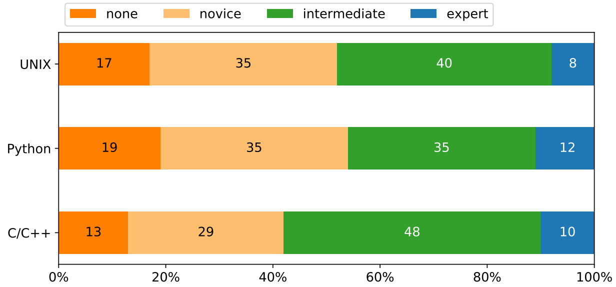

Figure 2: Distribution of programming skill levels in key programming languages (Unix shell, Python, and C/C++), self-assessed by the learners in Y3. (Source: [7])

During the registration process, learners were asked to self-identify their skill levels (none, novice, intermediate, or expert) on Unix, Python, and C/C++. (This question was asked because we were interested to see if this factor would have any bearings in their perception of the workshops and their learning effectiveness.) Figure 2 shows the results of this questionnaire in Y3. Many participants were novice or intermediate in each programming tool, but a considerable number of them self-identify as intermediates for C/C++; this is likely due to C++ being taught as a required course for Engineering and computer-related majors at ODU.

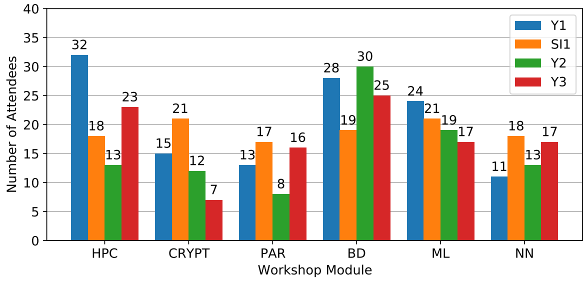

Figure 3: Number of learners attending individual workshops, reported for all the four complete rounds of DeapSECURE workshops (workshop series in Y1, Y2, Y3, as well as a summer institute [SI] at the end of Y1). (Source: [7])

4.2 Comparison of Workshops with In-person and Online Delivery

The attendance statistics of the DeapSECURE workshops is reported Fig. 3. We compare the attendance of the all-virtual workshops (Y3) with the other in-person training events (workshop series in Y1 and Y2, as well as a summer institute at the end of Y1). During Y3, where all workshops were delivered virtually, between 7 to 30 participants attended each workshop. Again, the workshop attendance

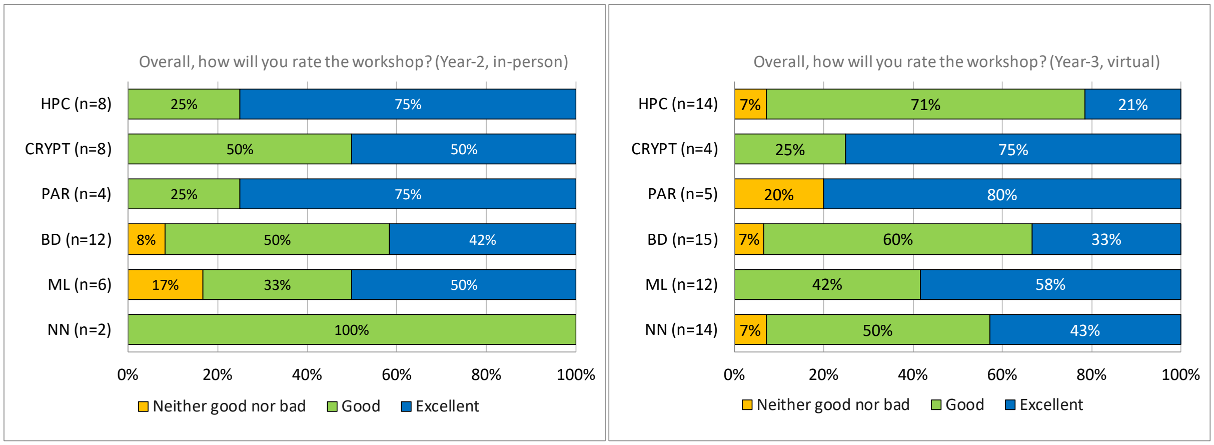

Figure 4: Percentage ratings of the six workshops given in Year 2 (left) and Year 3 (right) in response to the survey question "*Overall, how will you rate the workshop?*". The y-axis provides the workshop abbreviations (see Table 1, column 1) followed by the number n of opinions in parentheses.

was notably better and more consistent in the summer (the last three workshops in Y3) than during the academic year. The same attendance tendency was observed during the years of in-person delivery.

To get a comparative insight about the two delivery modes, online and in-person, we examine here an opinion question asked of the learners in *both* the second (Fig. 4, left) and third (Fig. 4, right) years. There were no radical differences in answers to the opinion and open-ended questions among the different years, where delivery changed from in-person to virtual (online). Note that, despite the low numbers of respondents, the comparisons in Fig. 4 are still fair since the low numbers are consistent across all the workshops (see the n-values) with slightly more responses received in Y3, which also corresponds to Y3 workshops having somewhat more attendees on average than those of Y2. From Fig. 4, note that, some workshops were consistently rated higher than others across the two years. For example, the ML rating was higher than that of BD in both years and PAR was higher than CRYPT. These relative ratings might correlate with (1) the perceived final applicability of the lessons to the cybersecurity task at hand, and (2) the continuity of the module materials. For example, the hands-on activities in the ML module led to the inferences for smartphone apps, whereas in the BD module, the activities mainly involved data handling and exploratory analysis. For the PAR module, the use of Python-paillier, the same tool to which the learners were introduced in CRYPT, might have contributed to the former's higher rating. The HPC module was rated lower in Y3; learners were split whether the lesson was too easy or too hard. The HPC module included a quick overview of UNIX shell commands, which topic is very hands-on in nature and require much practice to master. From the survey, we discovered the following: Because this module was taught using command-line interface, it might have been very challenging to learners who never used such a interface before, yet for others who had used shell for a period of time, this overview might have been considered a waste of time. This observation seems to support the notion that a command-line-based workshop is significantly harder to do virtually than in-person. The ML and NN modules received slightly higher ratings in Y3, which might have been due to the improved lessons in Y3.

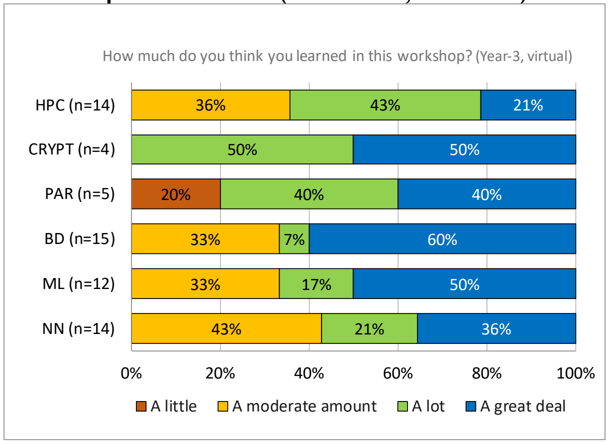

Figure 5: Percentage ratings of the six workshops given in Y3 in response to the survey question "*How much do you think you learned in this workshop?*". The y-axis provides the workshop abbreviations (see Table 1, column 1) followed by the number n of opinions in parentheses.

Learners' preferences are also reflected in Fig. 5, which shows Y3 outcomes to the opinion question: "How much do you think you learned in this workshop?" It is interesting to note that the BD module, which spent much time on the tedious handling of data in pandas, received a larger percentage of the highest ratings (compared with its *overall* rating in Fig. 4, right panel). In particular the highest ratings were 60% vs 33%, respectively. Of all the modules, the lowest rating was given for the PAR module by 20% of learners. We reckon that, in general, there may be two possible reasons for the perception of learning only little or moderately: (1) The concepts and hands-on material are very new, so that learners cannot keep up in absorbing exercises with respect to their applicability; (2) Conversely, the topics taught might have been quite familiar to learners, so that the concepts taught and exercises fill only small gaps in learner's knowledge and skill. In the case of the PAR module, the first reason is much more plausible because this module considers parallel programming with Message Passing Interface (MPI) (see Table 1), which is a very advanced topic typically taught only to upperclassmen and graduate students.

The responses from surveys and knowledge questions were instrumental in driving the iterative improvements of the training through its four years of development. We are aware that the responses to the opinion questions such as those reported in Figs. 4 and 5 do have their limitations; in particular, they are subject to respondents' biases, including their educational backgrounds, computing skills, etc. Nevertheless, they may give useful indicators on the areas needing improvement. In cases where improvements are needed, a focus group interview with the survey respondents might be valuable.

We found additional insights by analyzing responses to two open-ended questions: "What is most valuable about this training?" and "What is least valuable about this training?" (which will thereafter be abbreviated as "most valuable" and "least vaulable"). Responses from the open-ended questions in the post-workshop survey were analyzed by scanning for keywords (i.e. "hands-on") or themes (i.e. topic-related keywords like "encryption") and quantified. On Y2, there were 40 and 38 responses to the "most valuable" and "least valuable" open-ended questions, respectively. For Y3, the number of responses were 30 and 29. In general, learner's feedback consistently showed that participants enjoyed the hands-on component of the training, which evolved and was augmented over the project years. For both Y2 and Y3, 58% and 27% of respondents mentioned the hands-on training as the most valuable aspect of the training (with "Jupyter notebooks" repeatedly mentioned in Y3). On Y2, 38% of responses cited programming- or coding-related aspects (i.e. learning about different Python operations) as the most valuable. On Y3, most of the responses (40%) point to the topic or exposure to the training as the most valuable aspect. It should be mentioned that on Y3, 13% of responses indicated that the teaching assistants were the most valuable part of the training. Learners were generally happy with the training, as majority of respondents indicated that nothing was the "least valuable" part of the training in (73% in Y2, 68% in Y3). Upon further analysis on the "least valuable" responses, we found the following: Challenges with pace or insufficient time (14% in Y2, 7% in Y3); The material was difficult (10% in Y2, 3% in Y3). It is encouraging that the pace and level of materials seemed to have improved in Y3, based on learners' perception.

4.3 Lessons Learned

Through the four years of improving the training and conducting workshops, we have gained a number of important lessons. In terms of participation and attendance, there is no doubt that offering this training as a summer institute leads to the best level of engagement and learning, as students are completely focused on the training for a concentrated period of time. In the future, however, it might help to provide additional engagement opportunities in the year that follows the summer institute by offering seminars on cybersecurity research topics that leverage CI techniques, or small group meetups to work on specific challenges utilizing CI techniques. In general, unless there is a research driving needs, the students' participation and engagement will be somewhat limited to general literacy on CI.

Another important lesson learned is related to the timing of the workshop. It seems that devoting a whole-day workshop might be more appropriate for each DeapSECURE lesson module, to allow sufficient time to work through the notebooks. It is very important, however, to provide a way for learners to check-in at various stages, in order to keep up with their progress. This could be an important change that we will implement in the coming year.

Based on the level of materials presented in DeapSECURE lessons, the prerequisite for participation may need to be raised up so that learners will be able to engage with the presented CI techniques much more effectively. While currently we simply required participants to self-evaluate if they were able to write a 100-line code (or less), it may be better to require them to have command-line experience and Python programming experience. This can be satisfied, for example, by completing both the Software Carpentry's "Unix Shell" [8] and "Plotting and Programming with Python" [10] lessons prior to enrolling to the DeapSECURE training.

5 OPEN-SOURCE RELEASE AND COMMUNITY ADOPTION

The fourth and last year of DeapSECURE under the funding from NSF (**Y4, 2021–2022**) was spent completing all the lesson modules and hands-on materials, and releasing them open-source. The Big Data module has been completely released [15]; other modules are under review and refinement to become open-source. We expect to release all the data-intensive modules by the end of 2022. All the lessons will be released using CC-BY-4.0 license and all the codes with MIT license, compatible with lessons from the Carpentries.

We had two outreach activities to gauge the community interest in a training in the cross-cutting areas of cybersecurity and HPC. Firstly, a pilot workshop was conducted in the Fall of 2021, targeting students across Virginia, leveraging a blend of BD and ML lessons to teach students the basics of data analytics and machine learning. Secondly, we also conducted a small "community interest survey" gathering input and interest by faculty and researchers across Virginia on DeapSECURE training. From both the survey and the pilot workshop, we discovered great interest in adopting and leveraging DeapSECURE beyond ODU. We gathered nine responses from the community interest survey, with many indicating an interest in adopting DeapSECURE lessons for their own instruction. Some of the respondents would like to have hands-on workshops offered at their institutions, and/or customize DeapSECURE lessons for teaching.

In the Summer of 2022, a three-day summer institute was held to teach the DeapSECURE data-intensive modules to students in the Cybersecurity Research Experience for Undergraduate Students (REU) program at ODU. This institute was well received, and the lessons taught were instrumental in bringing the REU students up-to-speed with their summer research activities involving artificial intelligence and machine learning. During this institute, several Ph.D. students who were not part of the DeapSECURE team were quickly onboarded to teach the materials to the REU students. We were encouraged with how quickly the Ph.D. students were able to assimilate the lesson materials and step up to teach them. The success of this institute is an indicator that (1) the DeapSECURE lessons and teaching methodology have matured to the point that they are ready for a wide-range of instructors to take up and teach to others, and (2) graduate students may become quickly proficient in teaching the lessons.

6 SUMMARY AND FUTURE DIRECTION

Here we summarize major impacts of the DeapSECURE training on student future and carriers:

- Over the years DeapSECURE workshops had been offered, a number of students became interested in learning CI techniques in-depth and followed this up by taking formal HPC- and BD-related courses at ODU.
- At least one undergraduate student decided to pursue a M.S. degree in cybersecurity after attending the DeapSECURE workshop series (summer institute).
- DeapSECURE has been instrumental in augmenting REU students' interest in cybersecurity with HPC skills (Summer of 2019, 2021, 2022). This impact was evidenced by their final project posters, some of which embodied AI/ML work carried out on ODU's HPC cluster.
- A number of DeapSECURE learners (both undergraduate and graduate students) had continued their interest in CI/cybersecurity intersection by becoming WTAs in subsequent years. Their participation as WTAs afforded them intensive training in programming, in using state-of-the-art software development tools and methodologies, in team work, and in pedagogy (teaching) [7].

We plan to expand the current project both in-depth and in-breadth manners with the overarching goal to produce a community of practice (CoP) of next-generation cybersecurity researchers and scholars who are well-versed in leveraging CI technologies and methods—such as HPC, big data, AI, advanced cryptography and privacy protection, parallel computing. In particular, we plan to provide training for learners of different levels (depth) such as faculty, researchers, and graduate students by leveraging our initial experiences in training WTAs and expanding this to a full-fledged "train-the-trainer" program that is designed to be synergistic with research, teaching, and learning activities carried out by faculty, postdocs, and graduate students. We also plan to incorporate the the training modules into curriculum/instructional material fabric in various institutions in Virginia and beyond (thereby increasing the breadth of training application). In addition, we will closely engage and collaborate with Virginia Commonwealth Cyber Initiative (CCI) to strengthen, expand, and enrich the CI training program and scale up the effort and impact to state-wide and beyond. The project team will collaborate closely with CCI and its members from higher education institutions, industry, government, and non-governmental and economic development organizations.

ACKNOWLEDGMENTS

The development of DeapSECURE training program is supported by NSF CyberTraining grant #1829771. The workshops utilized Turing and Wahab HPC clusters provided by ODU Research Computing Services, part of Information Technology Services. Wahab cluster was acquired in part using NSF Major Research Instrumentation grant #1828593. We gratefully acknowledge our current and prior workshop teaching assistants: Liuwan Zhu, Qiao Zhang, Jacob Strother, Rosby Asiamah, and Orion Cohen for their excellent contribution to the training materials and to the teaching of the workshops in Y3 and Y4. We also acknowledge Rui Ning, Kazi Aminul Islam, Md. Shibly Sadique, and Joseph Zalameda for their guest contributions in teaching the DeapSECURE workshops in Y3 and Y4. We thank the ODU Distance Learning for their support in recording the workshop sessions, and the anonymous reviewers of the extended abstract, whose comments helped sharpen the presentation of this paper.

REFERENCES

[1] 2020. Kahoot! Game-based Learning Platform. https://kahoot.com

[2] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. https://www.tensorflow.org/ Software available from tensorflow.org.

[3] Erin Becker and Fran cois Michonneau. 2022. The Carpentries Curriculum Development Handbook. https://cdh.carpentries.org/

[4] François Chollet and Keras team. 2015. Keras. https://keras.io.

[5] CSIRO's Data61. 2013. Python Paillier Library. https://github.com/data61/python-paillier

[6] Lisandro D. Dalcin, Rodrigo R. Paz, Pablo A. Kler, and Alejandro Cosimo. 2011. Parallel distributed computing using Python. *Advances in Water Resources* 34, 9 (2011), 1124 – 1139. https://doi.org/10.1016/j.advwatres.2011.04.013

[7] Bahador Dodge, Jacob Strother, Rosby Asiamah, Karina Arcaute, Dr. Wirawan Purwanto, Dr. Masha Sosonkina, and Dr. Hongyi Wu. 2022. DeapSECURE Computational Training for Cybersecurity: Third Year Improvements and Impacts. http://www.modsimworld.org/papers/2022/MSVSCC_2022_InfrastructureSecurityMilitary.pdf

[8] Gabriel A. Devenyi (Ed.), Gerard Capes (Ed.), Colin Morris (Ed.), Will Pitchers (Ed.), Greg Wilson, Gerard Capes, Gabriel A. Devenyi, Christina Koch, Raniere Silva, Ashwin Srinath, and ... Vikram Chhatre. 2019. swcarpentry/shell-novice: Software Carpentry: the UNIX shell, June 2019 (Version v2019.06.1). (July 2019). http://doi.org/10.5281/zenodo.3266823

[9] David E. Hudak, Douglas Johnson, Jeremy Nicklas, Eric Franz, Brian McMichael, and Basil Gohar. 2016. Open OnDemand: Transforming Computational Science Through Omnidisciplinary Software Cyberinfrastructure. In *Proceedings of the XSEDE16 Conference on Diversity, Big Data, and Science at Scale (XSEDE16)*. ACM, New York, NY, USA, Article 43, 7 pages. https://doi.org/10.1145/2949550.2949644

[10] Allen Lee, Nathan Moore, Sourav Singh, and Olav Vahtras (eds). 2018. Software Carpentry: Plotting and Programming in Python. (2018). http://github.com/swcarpentry/python-novice-plotting

[11] Yisroel Mirsky, Asaf Shabtai, Lior Rokach, Bracha Shapira, and Yuval Elovici. 2016. SherLock vs Moriarty: A Smartphone Dataset for Cybersecurity Research. In *Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security (AISec '16)*. ACM, 1–12. https://doi.org/10.1145/2996758.2996764

[12] Alexander Nederbragt, Rayna Michelle Harris, Alison Presmanes Hill, and Greg Wilson. 2020. Ten quick tips for teaching with participatory live coding. *PLoS Comput. Biol.* 16 (2020), e1008090. Issue 9. https://doi.org/10.1371/journal.pcbi.1008090

[13] The pandas development team. 2020. pandas-dev/pandas: Pandas. https://doi.org/10.5281/zenodo.3630805

[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* 12 (2011), 2825–2830.

[15] Wirawan Purwanto and DeapSECURE Team. 2022. Open-Source Release of the DeapSECURE 'Big Data' Lesson Module to the Community. https://deapsecure.gitlab.io/posts/2022/02/release-big-data-lesson/.

[16] Wirawan Purwanto, Issakar Doude, Yuming He, Jewel Ossom, Qiao Zhang, Liwuan Zhu, Jacob Strother, Rosby Asiamah, Bahador Dodge, Orion Cohen, Masha Sosonkina, and Hongyi Wu. 2022. DeapSECURE Lesson Modules. https://deapsecure.gitlab.io/lessons/

[17] Wirawan Purwanto, Yuming He, Jewel Ossom, Qiao Zhang, Liuwan Zhu, Karina Arcaute, Masha Sosonkina, and Hongyi Wu. 2021. DeapSECURE Computational Training for Cybersecurity Students: Improvements, Mid-Stage Evaluation, and Lessons Learned. *The Journal of Computational Science Education* 12 (2021), 3–10. Issue 2. https://doi.org/10.22369/issn.2153-4136/12/2/1

[18] Wirawan Purwanto, Hongyi Wu, Masha Sosonkina, and Karina Arcaute. 2019. DeapSECURE: Empowering Students for Data- and Compute-Intensive Research in Cybersecurity through Training. In *Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (learning) (PEARC '19)*. ACM, New York, NY, USA, Article 81, 8 pages. https://doi.org/10.1145/3332186.3332247

[19] Bo Zhu. 2015. A pure Python implementation of AES. https://github.com/bozhu/AES-Python.git