Diverging assessments: What, Why, and Experiences

Amin Sakzad
Monash University, Melbourne, Victoria, Australia
amin.sakzad@monash.edu

David Paul
University of New England, Armidale, NSW, Australia
dpaul4@une.edu.au

Judithe Sheard
Monash University, Melbourne, Victoria, Australia
judy.sheard@monash.edu

Ljiljana Brankovic
University of New England, Armidale, NSW, Australia
ljiljana.brankovic@une.edu.au

Matthew P. Skerritt
RMIT, Melbourne, Victoria, Australia
matt.skerritt@rmit.edu.au

Nan Li
University of Wollongong
Wollongong, NSW, Australia
nanl@uow.edu.au

Sepehr Minagar
Monash University, Melbourne, Victoria, Australia
sepehr.minagar@monash.edu

Simon
Unaffiliated
Wadalba, Australia
simon.unshod@gmail.com

William Billingsly
University of New England, Armidale, NSW, Australia
wbilling@une.edu.au

ABSTRACT

In this experience paper, we introduce the concept of 'diverging assessments', process-based assessments designed so that they become unique for each student while all students see a common skeleton. We present experiences with diverging assessments in the contexts of computer networks, operating systems, ethical hacking, and software development. All the given examples allow the use of generative-AI-based tools, are authentic, and are designed to generate learning opportunities that foster students' meta-cognition. Finally, we reflect upon these experiences in five different courses across four universities, showing how diverging assessments enhance students' learning while respecting academic integrity.

CCS CONCEPTS

• **Social and professional topics → Computing education**.

KEYWORDS

diverging assessment, authentic, assessment-as-learning

ACM Reference Format:
Amin Sakzad, David Paul, Judithe Sheard, Ljiljana Brankovic, Matthew P. Skerritt, Nan Li, Sepehr Minagar, Simon, and William Billingsly. 2024. Diverging assessments: What, Why, and Experiences. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2024), March 20–23, 2024, Portland, OR, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3626252.3630832

1 INTRODUCTION

Traditional assessment tasks, such as exams and take-home assignments, are educator-focused in the sense that there is a set of questions unique to all students. These assessments are typically a form of assessment-of-learning [20], have low authenticity, and are not seen as learning opportunities for students. The educator is responsible for encouraging academic integrity by imposing constraints on the time, duration, resource accessibility, and the form of running the assessment.

With the trend to online assessment there are growing concerns about a rise in academic integrity violations with traditional assessments. Integrity issues with exams was recently demonstrated during and after Covid-19 restrictions, when students were forced to sit online exams with electronic and sometimes AI-assisted invigilation [5, 18]. The recent arrival of generative AI tools has put further pressure on maintaining integrity with traditional assessments.

One way to address concerns with assessment integrity is by randomizing the questions given to each student. This is possible in areas where a large pool of questions can be created, in which case the learning management system (LMS) is tasked with assigning a random subset of this pool to each student. Although this may relax the way an educator can run an assessment, there remain areas of concern such as scalability, fairness, and authenticity. In particular, such assessment tasks do not generate learning opportunities, although they can be used as either *assessment-of-learning* or *assessment-for-learning* [20].

To address these issues we propose 'diverging assessments', which emphasize authenticity, assessment-as-learning, and academic integrity. Rather than randomizing assessment questions for each student, we keep the question the same for all students but randomize the input. In this approach, the input data is usually generated automatically by the educator in a particular format, such as pcap files, VHD, websites, or compiled programs. Diverging assessments target the 'Apply/Analyse/Evaluate' level of Bloom's taxonomy [1], and are to be completed using industry-relevant tools and/or techniques such as Wireshark, Autopsy, XSS/SQL injection, and C debugger. This ensures authenticity.

The 'diverging assessments' approach may raise concerns about academic integrity in the light of generative AI tools. Most ChatGPT-era studies [2, 7, 16, 17] in computing education are about evaluating the effect of such tools on different computing courses. Our

SIGCSE 2024, March 20–23, 2024, Portland, OR, USA

ACM ISBN 979-8-4007-0423-9/24/03…$15.00
https://doi.org/10.1145/3626252.3630832

approach permits the use of such tools and even encourages students to work together in solving assessment tasks, with the intent that working on their own input data would activate *assessment-as-learning*. As the input data is usually not readily input to generative AI, and diverging assessments are process-driven, students can be shown some use of generative AI and its (im)proper prompts prior to the assessments in the class. Integration of generative AI tools into these assessments has not only enabled the creation of a unique learning experience for each student but has also opened up new avenues for fostering metacognition.

By applying this across diverse domains such as computer networks, operating systems, ethical hacking, and software development, we have experienced the versatility and efficacy of diverging assessments across five different disciplines in four universities. In this paper we describe our diverging assessments and our reflection on the application of diverging assessments, underscoring their positive impact on student learning outcomes and establishing diverging assessments as a valuable pedagogical framework.

2 FORMS OF ASSESSMENT

We now describe different forms of assessment, which lay the foundation for the development of the concept of *diverging assessments*.

Authentic assessment. Authentic assessment pertains to evaluations in which students engage with real-world tasks, or at least with tasks that mirror the real-world challenges awaiting students in the work force. Iverson et al. [11] proposed a framework to gauge the authenticity of an assessment using five key factors: frequency, fidelity, complexity, impact, and feed-forward. This framework defines a task as authentic to a professional endeavor if:

- A1 professionals regularly perform it within the field,
- A2 it accurately replicates an original environment,
- A3 it stimulates higher-order questioning and thinking through intricate challenges,
- A4 it encourages self-reflection, and/or
- A5 it contributes to subsequent assessment and/or learning tasks through feedback.

Assessment-as-Learning (AasL). Higher education is witnessing the emergence of a flourishing field of 'assessment as learning' (AasL) [20]. AasL involves students taking on the role of assessors for their own progress, actively overseeing their learning, formulating inquiries, and deploying various strategies to gauge their knowledge and skills. This self-assessment approach extends beyond the traditional notions of *assessment-of-learning*, which furnishes educators with evidence of student accomplishment according to predetermined outcomes and standards. AasL is also distinct from *assessment-for-learning*, in which students receive guidance from teachers during the learning journey.

Yan and Boud's definition [20] of AasL offers a comprehensive understanding of AasL in the context of higher education: it denotes an assessment process that inherently creates learning opportunities for students by engaging them in active quests for evidence, its interconnections, and its application. This definition underscores three fundamental aspects [20]:

- AasL1 An AasL activity represents a dual-purpose learning strategy and assessment, aimed not only at evaluating student performance but also at fostering the learning process.
- AasL2 It encourages and empowers students to glean knowledge from all elements intertwined with the assessment task itself, encompassing aspects such as deciphering assessment criteria and integrating feedback from peers or the teaching staff.
- AasL3 It compels students to assume an engaged and contemplative role, thus nurturing meta-cognition and self-regulation.

While assessment-for-learning can be characterized as 'assessment then learning' and assessment-of-learning as 'assessment after learning', AasL can be considered 'assessment while learning', thus laying the groundwork for both assessment-for-learning and assessment-of-learning to thrive.

3 DIVERGING ASSESSMENTS

We propose the concept of diverging assessments as process-based assessments designed in such a way that they are unique to each student but nonetheless easy to verify. Their process-based nature ties into their role in AasL – the assessment requires the student to engage in an authentic technical process where they are being asked to learn and experience. The goal in making the assessments unique is that it enables students to discuss and share knowledge about the process without risking sharing 'the answer' – without the risk that that discussion would obviate the need for each student to engage in the process.

We have developed the term diverging assessment partly to distinguish it from existing similar terms with somewhat different connotations. For example, the terms *personalized* [13, 19] and *individualized* [8, 9] often carry the connotation that the assessment is being tailored to the current developmental needs of individual students and therefore the learning goals may be different for different students. In diverging assessments, there is typically a common set of learning outcomes for all students as the educator asks students to engage in a particular authentic technical process.

Another similar term, *Parameterized assessment* [4], is often used to describe questions that take place within learning management systems or question platforms such as Mobius and Codio. However, 'parametrized assessment' is a specific means of making assessment items unique and as they usually take place within the question platform, they typically lack the requirement of there being an authentic external task that students engage with. Millar and Manohoran [14] use the term 'isomorphic questions', but only in the context of simple tests.

There are many ways in which a process-based assessment can be made diverging: there may be a *procedural* script that executes on demand for each student to generate a unique starting artifact to work with; there may be a large suite of *pre-generated* artifacts, in which case an allocation step would be used; the task may be engineered to be *self-diverging*, so that the step which makes their artifact unique is conducted by students themselves; or tasks might use a *matrix* approach – for example, allocating students to different combinations of competing software libraries to use in a common development task.

To demonstrate diverging assessments we have applied this concept to five different assignments in the areas of computer systems, networks, and security. These are described in the following section.

4 DIVERGING ASSESSMENT EXAMPLES

In this section, we describe five diverging assessments used in different disciplines in four Australian universities. Table 1 summarizes these diverging assessments.

4.1 Network Routing Configuration

At Monash University we designed diverging assessments for the networking module of an introductory computer architecture and networks course for postgraduate students. The module has the following learning outcome statement:

- Analyse and formulate the functions and architectures of (wireless) local area networks (LAN/WLAN), wide area networks (WAN) and the Internet;
- Examine networks using the underlying fundamental theories, models and protocols for data transmission;
- Use the fundamental concepts of cybersecurity on the Internet, including identifying common threats and applying countermeasures.

Students are required to configure the routing tables of multiple routers, a DHCP server, and a firewall using CORE Network Emulator (CNE) [6]. To marry all three learning outcomes to the goals of a diverging assessment we followed four design principles: providing an appropriate level of complexity, creating multiple instances of the problem, automating the instance creation, and automating or simplifying the marking process. Students must configure multiple routers where there are several subnet-to-subnet paths with different link bandwidths and delays. The assessment combines several concepts learned over several weeks to test students' depth of understanding. It elicits higher levels of cognitive performance in students than individual problems completed in the workshops.

To create multiple instances of the problem, we use several CNE configuration files as initial templates. The instances are then created by changing the subnet's IP addresses using randomly selected address ranges, which also affects all the network interfaces of all the routers and servers. Each student then receives one instance that is unique to that student. Students can discuss their approaches and collaborate in completing the assessment tasks without risking academic integrity. As the CNE configuration file is a text file, the problem generation is automated using Python. If required, finer changes can be made using Jinja2 templates [12].

At present the assessment is not automatically marked, but for each instance a marking script is generated that lists all the individual commands that need to be executed on each node within the emulation, and this can be used to assess students' work as well as generating the expected results for a correctly completed task.

The emulated nodes behave as Linux-based routers and firewalls, and in performing the task, students use commands that are used by professionals who use Linux, and are similar to those used in many other commercial routers and firewalls (authenticity criteria A1–A2). The assessment is complex enough to challenge students with a problem that requires a higher level of thinking than simply repeating their previous experience in the tutorial sessions (A3). As the assessment incorporates troubleshooting routing configurations and testing firewall rules, students need to reflect on their work and refine their process (A4).

4.2 Operating Systems – Debugging

In the University of New England's Operating Systems course, students are taught the concepts of threads and processes. To test this knowledge, one of the assessments has students write one C program that implements multi-threading and one that implements multi-processing. Concurrent programming requires different thought processes than the traditional single-threaded programming students experience before this course, and a combination of theory and practice is required [10]. To match professional activity (A1), students are expected to be familiar with C debugging tools, such as interactive debuggers, and this has been combined with the thread/process programming assignment.

To prepare for this task, students are guided through an AasL2 online quiz in the learning management system that presents a simpler version of the problem they will need to solve in the assignment. First, all students are presented with the same simple C program that uses a for-loop to sum the integer values returned by a function call. The called function takes a value as its parameter, and for this preparatory quiz, simply returns this parameter. For each iteration of the loop, the next value in an integer array is passed as the parameter to the called function, meaning that each iteration returns a different value. Students are encouraged to compile and execute this program and are required to enter the final output value (the sum of all array elements) in the quiz. The students are then given a compiled version of the program with a different set of values stored in the array, and guided to use an interactive debugger to determine the array values. A question in the quiz allows them to enter their discovered values to determine if they are correct. Finally, the preparation quiz guides the students on how to convert the iterative program into one that uses threads or processes to make the function calls in parallel.

The actual AasL1 assessment generates a newly compiled program for each student, with the source code unavailable. This binary format is currently not understood by public generative AI tools. The program uses the same for-loop structure as the program given in the preparatory quiz, where each value of an array is used as the parameter to a function call, the result of which is added to a total, and students are informed of this general structure. However, each student's instance has a randomly-sized array, and random integer values in that array, and the function called inside the for loop is replaced with a random linear function rather than just returning its parameter value. In this way, each student is given a different instance of the problem which then needs to be solved individually, and their response to the first task is fed forward to the remaining tasks in the assessment (A5), ensuring each student has unique values in all remaining tasks.

Thirty percent of the marks for this assessment are for a report where the student describes the process that they used to determine the array size, values, and linear function used for their instance. While it is expected that students will use an approach similar to that provided in the preparatory quiz, this is not required, and any valid approach is allowed. A further 10% is given for them to

Table 1: Comparing 5 diverging assessments in terms of type, authenticity (A1-A5), and assessment-as-learning (AasL1-AasL3).

Diverging Assessments	University	Diverging Type	Authenticity					AasL		
			A1	A2	A3	A4	A5	AasL1	AasL2	AasL3
Network Routing Configuration	Monash University	pre-generated	✓	✓	✓	✓	✗	✓	✓	✓
Operating Systems - Debugging	UNE	procedural	✓	✓	✓	✓	✓	✓	✓	✗
Software Development	UNE	self-diverging	✓	✓	✓	✓	✓	✓	✓	✓
Ethical Hacking	RMIT	procedural	✓	✓	✓	✓	✓	✓	✓	✓
System & Network Security	UoN	pre-generated	✓	✓	✓	✓	✗	✓	✗	✗

submit a C source file that compiles to the same program they were given. The tasks of converting the iterative program to ones that use threads and processes are each weighted at 30%. Students are informed that if they cannot determine any of the values to put in the report, they can just guess them, and those guesses can be used to complete the rest of the tasks. The students forgo marks on the report in this case, but it ensures that they don't get stuck at the first step of the assessment.

4.3 Software Development

In the earlier examples, we either used a randomization process or generated multiple starting artifacts to cause students' assignments to diverge from one another and enable helpful conversations. However, assignments can be designed so that the student's own actions cause their paths to diverge.

In a simple example, a software studio subject at the University of New England includes a low-stakes assignment that exercises students' basic skills with distributed version control and build systems (as a preparatory exercise for a later much larger group project). This is a process-oriented task, which asks students to push a (provided) local repository to a new remote, raise an issue for a known bug, write and commit tests for the bug, tag the commit where the tests are failing, fix the bug, push the fix, and close the issue. Rather than generate unique starting repositories for each student (although this would be feasible), we use the simple expedient of asking each team to make a unique change inside the code in their first commit. Although it would still be possible to replay another team's commits on top of the unique commit, students would need to use more advanced Git commands to do so (as well as varying the time and content of the replayed commits to mask their actions) – and in so doing would achieve the basic learning outcomes of the task anyway.

The group project, which we have run for some years [3], also uses students' own actions to make their work diverge. Each team is asked to complete different features of a single large class-wide project. It is not possible to make identical changes twice in one class-wide repository, so the teams are forced to diverge – and are encouraged to help others as software development teams do.

The preparatory assessment focuses on AasL1 (and authenticity aspects A1, A2, and A5), as the assessment prompts students to learn fundamental skills they will require to contribute to a distributed software development team. The group project then adds a significant focus on AasL2 and AasL3 (and authenticity aspects A3 and A4), as the open-ended nature of their contribution to a class-wide development team of approximately 90 students necessarily involves significant collaboration, feedback, and self-regulation, as well as the multi-faceted nature of such large development projects.

4.4 Ethical Hacking

The Ethical Hacking course at RMIT is offered as part of a master's level coursework program. Students learn the rudiments of ethical hacking, focusing on web app hacking guided by the OWASP top 10. Students are assessed by 'doing', and the assessment covers:

- Practical skills (performing hacking techniques)
- Demonstration (physically demonstrating skills from the practical assessment to teaching staff)
- Reporting (written communication of findings during the practical assessment).
- Presentation (a five-minute presentation of a vulnerability to a simulated non-technical stakeholder).

All aspects, apart from the presentation, are tied to the practical assessment. There are two practical assessment tasks, each of which has a corresponding reporting assessment. Originally, only the first practical assessment involved a demonstration, but this is being expanded in later offerings so that each practical assessment will have a corresponding demonstration.

Practical assessments are made divergent through randomization. Assessment is delivered through a website; each student receives a URL to a different randomly generated 'instance' of the assessment. Each instance contains randomly generated 'flags' (each encoded as a GUID for ease of recognition) in places that should only be accessible if a student can perform a particular hacking technique (or several techniques in concert). No two assessments have the same flags. As such, we use the flags as evidence of performing ethical hacking (noting that students can find flags in multiple ways, but hacking techniques are always required). Sample assignments were given for each assessment. In the samples, a subset of the possible vulnerabilities was given, and sometimes the vulnerabilities were less obfuscated than in the live (divergent) assessment.

Unique to the first practical assessment is a collection of cross-site scripting (XSS) challenges, in the form of eight random self-contained web pages for XSS testing. Each page contains a single vulnerable input, and the pages together cover a range of different XSS injection techniques. The first two XSS problems are common to all students and are considered basic. The remaining six pages are chosen randomly from a pool of eight possibilities. Additionally, each challenge has a vulnerable and invulnerable variant, and each student receives exactly one invulnerable variant among their six non-common challenges.

Common to both practical assessments is a simulated (limited in scope) banking site. The site differs between the two assessments, but provides a common theme and ostensibly represents the same logical site. Each instance of these assignments is an entirely self-contained (and randomized) copy of the simulated site. Furthermore, each instance has a separate database and database user/password combination to ensure that students see only the data for their instance. In addition to the randomized flags, other components of the site that are key for vulnerability exploitation (database table names, passwords, etc) are also randomized. Consequently, the exact code needed for exploitation differs between students.

In the first practical assessment, the simulated banking site contains only SQL injection vulnerabilities. It consists of only 2 pages, and there are 4 flags hidden on the site. Flags require a combination of observation, curiosity, and SQL injection techniques to find.

In the second practical assessment, the simulated banking site is extended into a larger and more varied site that has four main pages and a number of helper pages that are used in the background. Additionally, most of the SQL injection vulnerabilities from the first assessment are no longer present in the second practical assessment; one notable exception is a particularly difficult SQL injection vulnerability from the first assignment. Nine flags are hidden around the site, requiring a multitude of skills and techniques to find.

The second assessment covers authorization and authentication bugs (e.g., crackable passwords, JSON web tokens, directory traversal, and file upload vulnerabilities). Although most of the SQL injection vulnerabilities from Assessment 1 have been removed, some SQL injection techniques are still required. There are no cross-site scripting vulnerabilities, although server-side XSS checking, described above, would open the possibility of including XSS.

Further work will look into better compartmentalization of vulnerabilities, to allow multiple vulnerabilities for the same page such that the server can identify which was exploited (so as to present a correct flag). Great care will need to be taken to make sure that this does not allow students to use one vulnerability to find the flag for (and thus circumvent assessment of) a different vulnerability.

Both practical assessments were designed in collaboration with a research assistant who has experience designing capture-the-flag challenges for hacker conferences. The assessment is therefore authentic, meeting items A1–A2. Furthermore, by its very nature hacking requires higher-order questioning and thinking, as well as much trial and error (and thus self-reflection); and thus both A3 and A4 are met by this assessment, along with assessment-as-learning items AasL2 and AasL3. The assessment was designed from the start to both teach and assess, thus meeting AasL1.

4.5 System and Network Security

At the University of Newcastle the system and network security course delivers topics covering a wide range of system and network security fundamentals. Students learn security principles, mechanisms, and services, and demonstrate their knowledge in practical applications. In one assessment, we created scenarios to test their understanding of public key infrastructure (PKI), their understanding of X.509 certificate hierarchy, and their use of OpenSSL to generate public key certificates suitable for a given scenario.

As an objective of the assessment, we expect students to learn to use OpenSSL for a given X.509 certificate hierarchy implementation and verification, then extend to arbitrary structures. To achieve this, we applied a five-phase learning development structure:

- Learn: This phase introduces the underlying knowledge and tools required for the assessment. It is optional if the content has been delivered in lectures or computing laboratories.
- Prepare: The second phase provides students with simple questions to check if they are ready to start the assessment.
- Develop: Students work on their assessment with each with different data.
- Self-assess: Students can check their solutions by themselves.
- Redevelop: Students address issues from the above phases and extend their ability to more complex applications.

Self-assessing is integral to this assessment because OpenSSL can detect errors when checking a certificate (the student solution). In the course, we provide the necessary knowledge of OpenSSL in a computer laboratory as a prerequisite task. Therefore, the assessment development focuses on achieving the other four subtasks.

We first prepare a practice quiz that helps students to review their knowledge of both PKI and OpenSSL. The quiz provides practice for 'atomic' operations of public certificate generation using OpenSSL. Students are given a simple X.509 hierarchy and asked to create the required certificates. For example, the practice starts with generating a root Certificate Authority (CA) certificate. Students are required to use OpenSSL commands to generate a self-signed certificate on which they can create arbitrary information about a CA. Then the students generate certificates of other CAs in the X.509 hierarchy. In this part, students are required to understand the concept of cross-certification and to write correct OpenSSL commands for their implementation. The quiz also provides students with different ways to verify the result, followed by examples of the correct answers (from the input of various users). Eventually, students should be equipped to work on their actual assessment task after successfully accomplishing the practice quiz.

The real assessment questions and tasks are similar to those of the practice quiz, the main difference being the complexity level of the questions. In the actual assessment, a student is given a more complex X.509 hierarchy that requires them to perform more difficult analysis and implementation, such as bidirectional cross-certification and certificate verification crossing different PKI systems. For instance, the assessment hierarchy includes at least six CAs (which may be from different PKIs), while the practice quiz only contains two or three CAs. This new X.509 hierarchy is randomly generated for each student. These instances may have some overlaps but the solutions will be different. Nevertheless, students can use OpenSSL to detect, debug, and solve issues during the process. That is, they are allowed to engage in self-assessment rather than waiting until after submission to receive feedback. Finally, the assessment can be extended to implement more certificates under the given hierarchy, because the questions may not cover every aspect of the scenario. In this case, students can decide what to do and can check their own answers.

The designed assessment was divided into subtasks that each indicates a core component, such as root CA generation, cross-certification and verification of certificate in a given hierarchy. The

teaching staff can evaluate a student's submission by checking the output of the OpenSSL scripts.

This example shows an instantiation of authenticity criteria A1–A4. First, the students are required to use OpenSSL for developing cross-certification in PKI. Both OpenSSL and PKI are frequently used tools for security setup in practice (A1 and A2). Second, the assessment challenges students with more complex X.509 certificate hierarchies in their actual tasks, which are more realistic in applications (A3). Third, the assessment allows students to develop and verify their answers using OpenSSL. When students troubleshoot the commands, they will review their work and achieve self-reflection (A4). Finally, as a contribution to AasL1 we set a practice assessment starting with atomic operations and terms to help students understand and develop their knowledge, and to guide students in learning and working on the subsequent task.

5 REFLECTIONS

We now reflect on various aspects of the diverging assessments at the four universities.

Impact on student learning through authentic assessment-as-learning tasks. We conducted independent surveys on diverging assessments in the courses of three of the four universities. The general impression is that students valued these assessments as they closely mirror what they have practiced during weekly lab/tutorial sessions and with their fellow students and what is practiced by professionals in their day-to-day workforce lives. This mirroring covers multiple aspects of authenticity for diverging assessments.

Students are permitted and encouraged to discuss diverging assessment tasks, but are required to perform them independently on their own assigned input data. This lets students learn from each step of the assessment, self-monitor their own learning progression compared to their peers and against specific learning outcomes intended for the assessment task, ask questions freely of their peers and teachers, and apply a range of strategies to decide what they know and can do and how to use assessment information for new learning. This is what assessment-as-learning is about.

Generative-AI compatibility. All the diverging assessments presented in this paper share a few points with respect to generative AI. The educators can train their students on (1) how to use generative AI as an assistant, (2) what could potentially be an excellent prompt, and (3) how to read and verify the responses received from it. The educators can also freely let students use generative AI tools to do the diverging assessments (conditional on declaring its use). The main reason for this is that there is currently no tool that can accept the diverging assessment input data in the forms that we are using, such as VHD, pcap files, websites, and compiled programs. This may not remain true in the future, given the rate of AI advances in recent years. However, we do not currently envisage any publicly available AI tool capable of processing such input data in the next few years. Hence it is safe to say that the diverging assessment tasks presented here can be freely practiced by students before and while completing assessment tasks.

We recommend that educators ask students to record and explain the work that resulted in completing the assessment tasks for a small portion of the marks. This has two benefits: it may help prevent contract cheating, which is usually the hardest to catch among known academic integrity issues; and educators might learn additional approaches to solving tasks and adjust their marking guides accordingly for future uses of the same diverging assessment.

LMS integration and university rules. Many universities, including our four, generally require all assessments to be distributed and submitted via their learning management systems. However, it is integral to diverging assessments that each student gets a unique input instance, with the implanted artifacts known for (automatic) grading purposes; but LMSs, while they accommodate random assessment selection, can rarely guarantee that students will receive unique assessment data inputs. We have circumvented this by externally assigning data input instances to students, either manually or automatically. In some cases this has the potential to allow the use of a script to automatically grade students' work.

Other contexts and Bloom levels. Although we have presented different diverging assessment designs in the contexts of networking, software development, operating systems, and cybersecurity, it is not entirely clear if this approach can be adopted in some other computer science education areas. An example is introductory programming, where we are unsure whether it is possible to design meaningful diverging assessments. On the other hand, the requirements of diverging assessments can perfectly match any capture-the-flag style assessment, such as in digital forensics [15].

The primary target of diverging assessment is to generate learning opportunities by letting students *apply* suitable tools and techniques on randomised data input, *analyse* the results with their student peers and teaching team members, and finally *self-evaluate* their own work by repeating an assessment piece one or more times. We do not recommend the use of diverging assessments for lower levels of Bloom's taxonomy, but it might be promising to explore whether the *synthesize* level of Bloom's taxonomy can be targeted by crafting new ideas around diverging assessments.

6 CONCLUSION

We have presented the concept of 'diverging assessments', a novel and innovative approach to student assessment that transcends traditional methods and helps to address concerns about academic integrity. This experience paper has showcased the development and implementation of process-based diverging assessments that are tailored to individual students while retaining a common framework. By applying this approach, five different examples and experiences of diverging assessments from courses at four universities are presented across the diverse domains of computer networks and security, operating systems, ethical hacking, and software development. In essence, the experiences shared in this paper lay the foundation for a shift in how assessments can be conceptualized and executed in computing education.

7 ACKNOWLEDGEMENT

The authors were supported by 2022 ALTA small grants funded by the Australian Council of Deans of Information & Communications Technology (ACDICT). Matthew P. Skerritt acknowledges the substantial contributions of the Morgan Reed in developing their diverging assessment for RMIT.

REFERENCES

[1] Lorin W Anderson and David R Krathwohl. 2021. *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives*. Longman.

[2] Brett A. Becker, Paul Denny, James Finnie-Ansley, Andrew Luxton-Reilly, James Prather, and Eddie Antonio Santos. 2023. Programming Is Hard - Or at Least It Used to Be: Educational Opportunities and Challenges of AI Code Generation. In *54th ACM Technical Symposium on Computer Science Education V. 1* (Toronto ON, Canada) *(SIGCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 500–506. https://doi.org/10.1145/3545945.3569759

[3] William Billingsley, Rosemary Torbay, Peter R. Fletcher, Richard N. Thomas, Jim R. H. Steel, and Jörn Guy Süß. 2019. Taking a Studio Course in Distributed Software Engineering from a Large Local Cohort to a Small Global Cohort. *ACM Trans. Comput. Educ.* 19, 2, Article 13 (jan 2019), 27 pages. https://doi.org/10.1145/3218284

[4] Peter Brusilovsky and Sharad Pathak. 2002. Assessing student programming knowledge with web-based dynamic parameterized quizzes. In *ED-MEDIA 2002–World Conference on Educational Multimedia, Hypermedia & Telecommunications* (Denver, Colorado). Association for the Advancement of Computing in Education (AACE), 1548–1553. https://www.learntechlib.org/primary/p/9952/

[5] Ben Burgess, Avi Ginsberg, Edward W. Felten, and Shaanan Cohney. 2022. Watching the watchers: bias and vulnerability in remote proctoring software. In *31st USENIX Security Symposium, USENIX Security 2022, Boston, MA, USA, August 10-12, 2022*, Kevin R. B. Butler and Kurt Thomas (Eds.). USENIX Association, 571–588. https://www.usenix.org/conference/usenixsecurity22/presentation/burgess

[6] CORE. 2023. CORE Documentation. http://coreemu.github.io/core/, Last accessed 2 Aug 2022.

[7] Marian Daun and Jennifer Brings. 2023. How ChatGPT Will Change Software Engineering Education. In *2023 Conference on Innovation and Technology in Computer Science Education V.1* (Turku, Finland) *(ITiCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 110–116. https://doi.org/10.1145/3587102.3588815

[8] Meg Grigal, David W. Test, John Beattie, and Wendy M. Wood. 1997. An Evaluation of Transition Components of Individualized Education Programs. *Exceptional Children* 63, 3 (1997), 357–372. https://doi.org/10.1177/001440299706300305 arXiv:https://doi.org/10.1177/001440299706300305

[9] Robert M. Hashway. 1998. *Assessment and Evaluation of Developmental Learning: Qualitative Individual Assessment and Evaluation Models*. Bloomsbury Academic.

[10] Maurice Herlihy, Nir Shavit, Victor Luchangco, and Michael Spear. 2020. *The art of multiprocessor programming* (second ed.). Morgan Kaufmann. https://doi.org/10.1016/C2011-0-06993-4

[11] Heidi L. Iverson, Mark A Lewis, and Robert M. Talbot. 2008. Building a framework for determining the authenticity of instructional tasks within teacher education programs. *Teaching and Teacher Education* 24, 2 (2008), 290–302. https://doi.org/10.1016/j.tate.2007.09.003

[12] Jinja2. 2023. Template Designer Documentation. https://jinja.palletsprojects.com/en/3.0.x/templates/, Last accessed 15 Aug 2022.

[13] Sathiamoorthy Manoharan. 2017. Personalized Assessment as a Means to Mitigate Plagiarism. *IEEE Transactions on Education* 60, 2 (2017), 112–119. https://doi.org/10.1109/TE.2016.2604210

[14] Russell Millar and Sathiamoorthy Manoharan. 2021. Repeat individualized assessment using isomorphic questions: a novel approach to increase peer discussion and learning. *International Journal of Educational Technology in Higher Education* 18, 22 (2021). https://doi.org/doi.org/10.1186/s41239-021-00257-y

[15] Sepehr Minagar and Amin Sakzad. 2023. Automatic Problem Generation for CTF-Style Assessments in IT Forensics Courses. In *2023 Conference on Innovation and Technology in Computer Science Education V.1* (Turku, Finland) *(ITiCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 229–235. https://doi.org/10.1145/3587102.3588788

[16] Eng Lieh Ouh, Benjamin Kok Siew Gan, Kyong Jin Shim, and Swavek Wlodkowski. 2023. ChatGPT, Can You Generate Solutions for My Coding Exercises? An Evaluation on Its Effectiveness in an Undergraduate Java Programming Course.. In *2023 Conference on Innovation and Technology in Computer Science Education V.1* (Turku, Finland) *(ITiCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 54–60. https://doi.org/10.1145/3587102.3588794

[17] Jaromir Savelka, Arav Agarwal, Christopher Bogart, Yifan Song, and Majd Sakr. 2023. Can Generative Pre-Trained Transformers (GPT) Pass Assessments in Higher Education Programming Courses?. In *2023 Conference on Innovation and Technology in Computer Science Education V.1* (Turku, Finland) *(ITiCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 117–123. https://doi.org/10.1145/3587102.3588792

[18] Simon, Meena Jha, Sander J.J. Leemans, Regina Berretta, Ayse Aysin Bilgin, Lakmali Jayarathna, and Judy Sheard. 2022. Online Assessment and COVID: Opportunities and Challenges. In *Australasian Computing Education Conference* (Virtual Event, Australia) *(ACE '22)*. Association for Computing Machinery, New York, NY, USA, 27–35. https://doi.org/10.1145/3511861.3511865

[19] L. Tetzlaff, F. Schmiedek, and G. Brod. 2021. Developing Personalized Education: A Dynamic Framework. *Educational Psychology Review* 33 (2021), 863–882. https://doi.org/10.1007/s10648-020-09570-w

[20] Zi Yan and Lan Yang. 2021. *Assessment as Learning: Maximising Opportunities for Student Learning and Achievement*. Routledge.