Maestro: A Gamified Platform for Teaching AI Robustness

Margarita Geleta, Jiacen Xu, Manikanta Loya, Junlin Wang, Sameer Singh, Zhou Li, Sergio Gago-Masague

University of California, Irvine
Irvine, California 92697, USA
{mgeleta, jiacenx,manikanl,junliw1,sameer,zhou.li,sgagomas}@uci.edu

Abstract

Although the prevention of AI vulnerabilities is critical to preserve the safety and privacy of users and businesses, educational tools for *robust AI* are still underdeveloped worldwide. We present the design, implementation, and assessment of *Maestro*. *Maestro* is an effective open-source game-based platform that contributes to the advancement of *robust AI* education. *Maestro* provides *goal-based scenarios* where college students are exposed to challenging life-inspired assignments in a *competitive programming* environment. We assessed *Maestro*'s influence on students' engagement, motivation, and learning success in *robust AI*. This work also provides insights into the design features of online learning tools that promote active learning opportunities in the *robust AI* domain. We analyzed the reflection responses (measured with Likert scales) of 147 undergraduate students using *Maestro* in two quarterly college courses in AI. According to the results, students who felt the acquisition of new skills in robust AI tended to appreciate highly *Maestro* and scored highly on material consolidation, curiosity, and maestry in *robust AI*. Moreover, the leaderboard, our key gamification element in *Maestro*, has effectively contributed to students' engagement and learning. Results also indicate that *Maestro* can be effectively adapted to any course length and depth without losing its educational quality.

Introduction

AI is becoming the key technological asset for fueling automated decision-making systems, with significant societal impact and consequences upon many fields, including such sensitive ones as medicine, transportation and education (Hamon, Junklewitz, and Sanchez-Martin 2020). Traditionally, these systems have been designed and developed with the assumption that the environment is benign during both, their training and evaluation. However, a real environment can take adversarial shades. Consequential vulnerabilities in the shape of adversarial attacks may jeopardize AI-based systems and lead to unexpected outputs. Adversarial attacks have been extensively developed in computer vision applications and consist in altering input data to increase the odds of misclassification by the model (Goodfellow et al. 2017; Huang et al. 2017; Goodfellow, McDaniel, and Papernot 2018; Szegedy et al. 2014; Elsayed et al. 2018). Potential vulnerabilities in fast-spreading AI-based systems, likewise in the cybersecurity field, have created concerns and motivated new directions and strategies to educate the future AI workforce in countermeasures and the prevention of AI vulnerabilities. This aforementioned domain of expertise is known as *robust AI*. Despite *robust AI* being an active area of research, the advancement of educational tools specifically targeting this domain is still in its infancy. In fact, we found there were attempts to teach AI in general by reproducing the existing algorithms (Lucic et al. 2022), using explicit ethical agent (Green 2021), playing value cards (Shen et al. 2021), programming robots (Mataric, Koenig, and Feil-Seifer 2007), etc., but none of them touches the area of *robust AI*, not to mention building a learning platform for *robust AI*[1]. Hence, we make the *first* step towards bridging this gap. To facilitate other educators to teach *robust AI*, we will release the platform publicly in a GitHub repository.

There exist many examples of learning platforms and containerized environments in the cybersecurity domain, but very few in AI. Previously developed educational tools targeted to explore and simulate cybersecurity flaws and develop secure coding practices to train new cybersecurity professionals (Kjorveziroski, Mishev, and Filiposka 2020; Furfaro et al. 2018; Acosta et al. 2017; Kalyanam and Yang 2017; Mirkovic and Benzel 2012). A variety of open-source educational tools to teach cybersecurity have been confidently introduced into secondary and higher education around the world, such as the SEED project (Du 2002) or the OWASP Juice Shop project [2]. The common denominator of all these educational projects is the principle of *learning by doing*, i.e., providing a hands-on experience where users are presented to *goal-based scenarios* (GBSs) (Schank et al. 1994; Reigeluth 2013).

GBSs provide learning in a contextualized fashion, since they marry knowledge and skills with real-life scenarios,

[1] We found there are academic and industrial platforms to test and evaluate *robust AI* algorithms, like CleverHans (https://github.com/cleverhans-lab/cleverhans) and ART (https://github.com/Trusted-AI/adversarial-robustness-toolbox), but none of them are organized under *goal-based scenarios* (GBSs), as explained later.

[2] OWASP Juice Shop (https://owasp.org/www-project-juice-shop/) is an open-source project for demonstration and exercise in security risks in modern web applications.

supplying a more effective way of knowledge consolidation and skill development. Educational projects following GBSs optimize over the intended skills to learn by exposing students to opportunities where they can practise those in a task environment (Schank et al. 1994). Essentially, students become active participants in the learning scenario (Watson, Mong, and Harris 2011), which stimulates them to move towards the completion of the specified task. This encouragement, also referred to as *motivation*, is the key ingredient in pedagogical success (Uguroglu and Walberg 1979; Reigeluth 2013; Deci and Ryan 2013; Tohidi and Jabbari 2012; Ahmet and Fallows 2013). Games and competitions, as educational tools, have been shown to encourage curiosity and motivation for learning (Burguillo 2010; Lee and Hammer 2011; Lepper 1988; Watson, Mong, and Harris 2011), and have been actively used in schools, colleges and universities for computer science curricula (Adams 1998; Becker 2001; Ebner and Holzinger 2007; Lawrence 2004). In particular, in courses requiring programming, the concept of *competitive programming* has been developed (Lawrence 2004). *Competitive programming* allows students to engage with the programming topics set by the instructor. By improving and evaluating students' skills in a coding tournament, students can either try to beat a baseline set by the instructor or compete against other students. Examples of *competitive programming* implementations and communities include *Jutge.org* (Petit, Giménez, and Roura 2012), *CodeWars* (Doctor and Hoffner 2012), *CodeChef* (Turakhia 2009), *SPOJ* (SPLabs 2014), *LeetCode* (Chang et al. 2015), among others. *Competitive programming* has also been adapted in data science and machine learning contexts, with competitions where students compete not for a compile/no compile, best time execution and memory usage, but compete for attaining better supervised learning performance metrics, such as accuracy. Examples of such competition platforms include *Kaggle* (Goldbloom 2010) and *Driven Data* (Bull and Lipstein 2014).

Research Goals

The present project goal is to design, implement and evaluate an innovative open-source educational tool (*Maestro*) for training courses in *robust AI*. The goal of *Maestro* is to provide a friendly, *competitive programming* environment to engage undergraduate and graduate students in course assignments to acquire skills in *robust AI*. In our work, we investigate (i) whether *Maestro* provided an efficient learning platform for students to interact and acquire skills in *robust AI*, and (ii) whether *Maestro* provided students with enough autonomy for training on *robust AI* as a self-paced learning tool.

Approach

To encourage active student participation, we propose using game-based assignments (Lee and Hammer 2011), which include game stories, use cases and standard scenarios to reproduce settings covered by *Maestro*. The game is structured as an engaging project competition, where each student or team is assigned roles of attacker or defender, and *Maestro* assigns scores and ranks participants based on the quality of their attack or defense. These results are displayed in a user-friendly interface. Specifically, in the course described in this study, students are introduced to three different GBSs: one for adversarial attacks referred to as *Attack phase*; another one for adversarial defenses – *Defense phase*, and lastly an interactive scenario, referred to as *War phase*, where attacker and defender teams compete among each other, following the *Build it, break it, fix it* strategy (Parker et al. 2020).

Design

In order to maximize the instructional benefits of *Maestro*, it has been designed following the intrinsic motivational principles (Lepper 1988; Malone 1980). The basic proposition of these principles lies in the fact that interest and motivation can be stimulated by challenge in a contextualized environment (Deci and Ryan 2013). A continuously challenging activity can be given if (a) the student is uncertain about their success outcome and if (b) the student receives feedback about their progress towards the task goal. Within *Maestro*, students are enrolled into a *competitive programming* environment where they need to beat the baselines (i.e., hidden models) and their fellow students' solutions, contributing to the challenge of the game. Students also have access to the *Maestro* leaderboard, where their scores, ranks and/or (possible) errors are displayed, providing feedback and exposing the (possible) inconsistencies in their submitted solutions.

Interaction diagram The storyboard sequence diagram (Figure 1) represents the lifeline of *Maestro* usage. A student, once they developed their attack or defense according to the task defined in the GBS, submits their own solution to Gradescope[3]. Gradescope is a user-friendly platform with

[3]Gradescope (https://www.gradescope.com) is a suite of tools designed to accommodate the grading workflow.

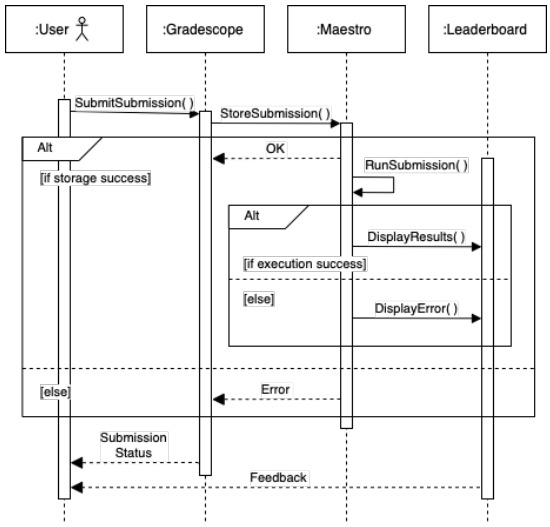

Figure 1: Interaction diagram of *Maestro*.

a submission recording system that documents each submission's timestamps, files' content, and user information. *Maestro* has been integrated with Gradescope in order to store the students' codes and provide an easy access to them for both, students and instructors, as well as join students into teams. After a solution has been sent to Gradescope, if it has been successfully stored, the autograder of Gradescope forwards the student's submission to the *Maestro* server. *Maestro* executes the submission and benchmarks the provided solution against several hidden models. Once the *Maestro* server finishes the evaluation, a comma-separated values (CSV) file with the results of these benchmarks is generated. That CSV file is displayed on the *Maestro* leaderboard for the corresponding phase (*Attack*, *Defense* or *War*), which students can access anytime. In case that the submission errors during runtime at *Maestro*, the caught exception is displayed at the error boards on the *Maestro* leaderboard.

System diagram Because a submission can be done by an individual student or a group of students, each submission is linked to a generalized entity referred to as *submitter* (Figure 2). Each *submitter* can have an unlimited amount of submissions for each phase prior to the submission deadline, and each phase has an associated *Maestro* leaderboard. Once a submission has been stored and sent to *Maestro* (see Figure 1), the *submitter*'s code is executed and tested against several hidden models. Depending on the phase, if the submission is for an attack, *Maestro* tests its effectiveness to downgrade the classification accuracy of hidden vision models, such as the convolutional neural network LeNet (LeCun et al. 1998). In the case that the submission is for a defense, such as adversarial training (Goodfellow, Shlens, and Szegedy 2014), *Maestro* tests its robustness against several white-box or black-box adversarial attacks, namely FGSM (Goodfellow, Shlens, and Szegedy 2014) and PGD (Madry et al. 2017). Finally, if the submission is for an attack or defense in the war phase, the provided solution is tested against the best attack or defense submissions from previous phases. At the end, *Maestro* associates a map with evaluation metrics to each submission. By default, the boards display only the last evaluated submission by each *submitter*. However, each board implements a set of operations, which includes filtering by *submitter*'s ID. This allows

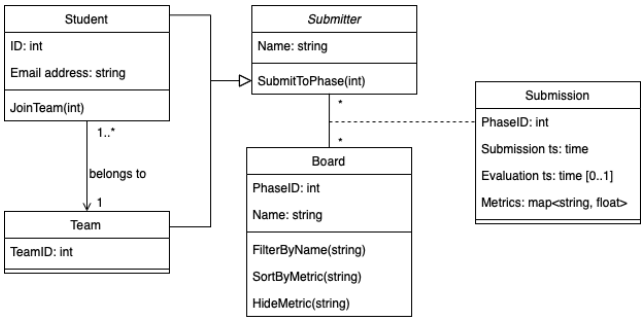

Figure 2: *Maestro* Unified Modeling Language (UML) entities.

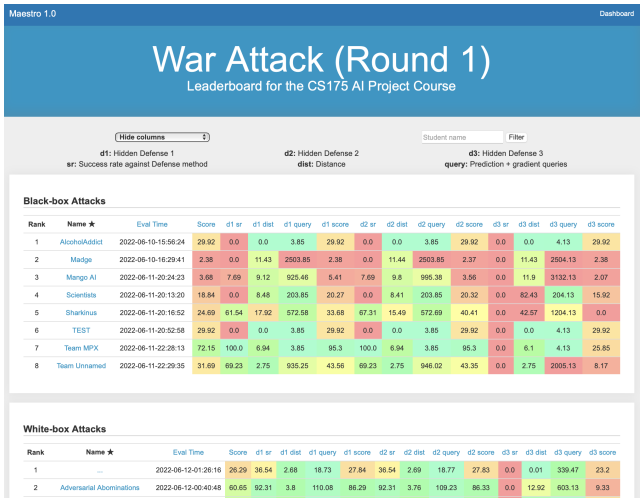

Figure 3: *Maestro* leaderboard showing the *War* phase board.

submitters to see their submission history and analyze the provided feedback to implement strategies to improve their following submission.

Methods The platform consists of code templates and essential libraries to let the students implement solutions for the attack homework (e.g., Genetic Algorithm (Chen et al. 2019)), defense homework (e.g., Adversarial Training (Goodfellow, Shlens, and Szegedy 2014)), attack project (where the student can implement any attack method against a selected defense method), defense project (where the student can implement any defense method against a selected attack method) and war phase (where the student plays attack and defense in turns). In further depth, the assignments are built under different scenarios such as evaluating the robustness of the rainforest monitor systems. Adversarial methods are used in the context of multi-class classification, using MNIST (LeCun 1998) and CIFAR-10 (Krizhevsky 2009) datasets, respectively.

Metrics We evaluate each submission with metrics that are widely adopted by the literature of robust AI:

- **Efficiency**: We consider the runtime overhead and resource consumption for the efficiency metric. For an attack submission, we encourage the implementation to trigger less interactions with the AI system (measured by the number of queries) and execution in shorter time. For a defense submission, the implementation needs to harden the AI model and we measure the extra time that is consumed for this process.
- **Effectiveness**: For an attack submission, we measure how it decreases the prediction accuracy of a targeted model, and how stealthy the attack sample is (e.g., measured by L_2 distance that compares the original image and the perturbed image (Carlini and Wagner 2017)). For a defense submission, we consider the change of the prediction accuracy; a good implementation should keep a similar level even under attack.

On top of these metrics, we also compute a weighted sum as the overall score for a submission. When multiple submissions are required (e.g., during the war phase), the overall score is a weighted sum of all the submissions.

Leaderboard To make the usage of *Maestro* compelling, a leaderboard is a key feature to include in order to gamify the learning environment, as it fulfills the learners' needs of connectedness (a tool driving to interaction), competence (a tool driving to practice and mastery), and autonomy (a tool for partial control) (Fotaris et al. 2016; Ortiz-Rojas, Chiluiza, and Valcke 2019; Sailer et al. 2017). Hence, the leaderboard becomes a game design element that provides a challenging environment for the students by creating social pressure to increase their level of engagement and participation. Within the *Maestro* leaderboard (Figure 3), we included the following features to ease the interaction with the data:

- **Submission history**: by default, each phase leaderboard displays one submission per each individual *submitter*. The main reason for this design decision is to keep the board not overflown with too many submissions. However, if desired, it is possible to access the submission history of each *submitter* by clicking (filtering) on their name.
- **Default & custom sorting by metrics**: the sorting of all the submissions by default is by evaluation time, which translates the submission history into a chronological order and allows tracking the improvement over different metrics. At anytime it is possible to sort the boards by any other metric by clicking on the metric name at the header.
- **Search by name**: the leaderboard includes a search input for fast retrieval of searched rows. The routine behind filters the submission data to display only the rows that match the search query.
- **Color indicators & thresholds**: each leaderboard has a configuration file where minimum and maximum values for each metric can be predefined. Additionally, it is possible to define a threshold value. Cells containing numbers above the threshold will be colored in green tones (suggesting an accomplishment regarding that metric), and cells with numbers below the threshold will be colored in red tones (suggesting further improvement of that metric). That color gradient allows for a fast visual evaluation of the submissions.
- **Metric selection**: leaderboards can display only the desired metrics which can be selected using a dropdown menu located at the top area of the leaderboard.

Evaluation

In order to assess the effectiveness of this new GBS-based approach for *robust AI* education, we implemented a course with *Maestro* for undergraduate students, which allowed the evaluation of its performance based on several metrics such as learning outcomes and students' engagement. For that aim, we designed a questionnaire, surveying upon the attitude and perceptions towards the *Maestro* platform as a *robust AI* educational tool, using well-established measures of course engagement and psychometric scales (Mandernach, Donnelli-Sallee, and Dailey-Hebert 2011; Handelsman et al. 2005). Indicator statements and questions have a restricted amount of answers measured on a Likert scale (1–5).

The main survey consisted of twelve questions. Each question has a prefix **Q**. Ten questions have five possible answers measured on a Likert scale (1 = Strongly disagree or **SD**, 2 = Disagree or **D**, 3 = Neutral or **N**, 4 = Agree or **A**, 5 = Strongly agree or **SA**). Two other questions have a binary and categorical response, respectively. To assess the quality and the perception of the leaderboard as a key gamification feature of *Maestro*, we included five additional questions in our survey specifically targeting the leaderboard features. These questions have a prefix **LQ**, and they are also measured on a Likert scale (1–5).

The full set of questions can be found in *Appendix A*. We can group the survey questions into four categories (**C1**–**C4**):

(C1) Gamification:

Competence: this set of questions aims to check the quality of *Maestro*'s game elements. **Q1** and **Q3** strive to examine the constructive effect of the leaderboards and game stories, respectively.

Autonomy: **Q2** examines the fulfilment of the psychological need of autonomy in regard to decision freedom.

Connectedness: **Q4** examines the fulfilment of the psychological need of social relatedness. Answers aim to score the level of sensed relevance among teammates and union towards a common goal.

(C2) Educational experience:

Curiosity: **Q5** raises a question about the perceptual level of interest to learn more about *robust AI* catalyzed by *Maestro*.

Consolidation: **Q6** examines the learning success and effectiveness by the *learning by doing* experience.

Maestry: **Q7** extrapolates **Q6** by surveying even further, not asking just about material consolidation but also about how students feel about transferring their new skills to other AI domains.

Effectiveness: **Q8** inquires about the perception of *Maestro* as an appropriate tool for communicating the course content.

Course organization: **Q11** inspects the preference between the classical way of instructions (evaluation via midterms) or our GBS-based instruction with *Maestro* (via contextualized assignments).

(C3) Academic challenge:

Material quantity: our course taxonomy can be divided into three phases (*Attack*, *Defense* and *War*). **Q9** questions the perceived challenge by the given amount of workload and material to be learned.

Time usage: **Q10** examines the perceived amount of spent time in order to finalize the assignments.

Tasks: **Q12** compares the perceived difficulty in material consolidation between adversarial attacks and defenses.

(C4) Leaderboard features: **LQ1**, **LQ2**, **LQ3**, **LQ4** and **LQ5** explore students' satisfaction with several features that were included in the *Maestro* leaderboard.

Results and Discussion

The first version of *Maestro* was successfully released in late 2021 and put into practice by 147 undergraduate students enrolled in *Projects in AI* course (with focus on *robust AI*) at the University of California, Irvine (UCI). The *Maestro* leaderboard was released in Spring 2022. There have been two course offerings; one in Winter (with 70 students enrolled) and a another in Spring quarter (with 77 students enrolled) both in 2022. All students had access to: (a) the private course Canvas[4] page (in the UCI intranet) including assignment descriptions and game stories, (b) the publicly available *Maestro* repository on GitHub[5], and (c) the private course Gradescope page, where students could submit their assignments solutions. Students enrolled in the Spring offering had also access to the Maestro leaderboard[6]. Students in the course were instructed to clone the *Maestro* repository from GitHub and practise locally before submitting their solutions to the remote *Maestro* server. At the conclusion of the courses, students participated in the survey anonymously regarding gamification, experience and challenge categories. Students from the Spring 2022 offering additionally answered questions regarding the leaderboard features category.

The results of the survey point to a significant efficacy of *Maestro* as it can be observed in the violin plots from Figure 4, where **C1**, **C2**, **C3** and **C4** are the aggregated responses of continuous-scale survey questions from each category (**C1** = Gamification, **C2** = Educational experience, **C3** = Academic challenge, **C4** = Leaderboard features). Observe that the median of the majority of questions lies at the *Agree* (**A**) answer. Regarding gamification questions, *Competence* **C1** had the highest rate in positive answers, signaling the effectiveness of the leaderboards and game stories. This is reinforced by the positive answers from the second category, educational experience **C2** (especially in **Q7** and **Q8**). As far as academic challenge **C3** is concerned, there is an unanimity that *Maestro* had provided a challenging learning environment.

The leaderboard features (**LQ1**–**LQ5**) had a very positive perception, highlighting the median of **LQ3** and **LQ4** as *Stronly Agree* (**SA**). According to **LQ3** and **LQ4**, the most appreaciated leaderboard features are the sorting options and the access to the submission history, while the feature of hiding columns (**LQ2**) had a perception biased towards neutral.

Given the partially categorical nature of the ordinal scale of our survey answers, we opted for indicator-matrix-based Multiple Correspondence Analysis (MCA) to analyze the association between the survey answers. In Figure 5 we represent questions **Q1**–**Q10** with answers measured on a Likert scale. For instance, **Q10:SA** represents the projection of the indicator matrix column corresponding to question **Q10** and answer **SA**. Each grey dot represents an individual that has responded the survey (147 points). Additionally, since the Likert scale can be interpreted as a continuous gradient of satisfaction, we performed Principal Component Analysis (PCA) on the survey answers, and projected both, rows and columns, in a bidimensional latent space resulting in a biplot (Figure 6), which is a powerful tool for the graphical exploration of multivariate data.

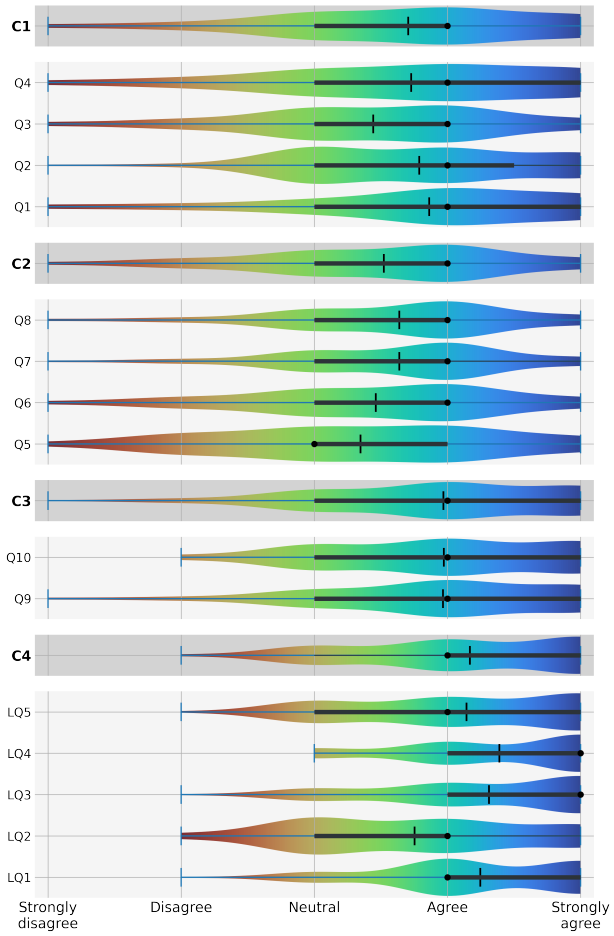

Figure 4: Violin plots featuring the responses of continuous-scale survey questions. The black rectangles in the boxplots represent the interquartile ranges, where the black dot represents the median. Black vertical lines represent the average response.

In Figure 5 we can observe 4 concentrations of categories: the blue cluster of **SA** categories, the yellow cluster of **SD** categories, the outspread sparse pink cluster of **SD** categories, and a densely-populated central cluster with **A** and **N** categories. We can affirm that there are no inversely correlated variables because there is no proximity between completely different categories and in Figure 6 there are no

[4]Canvas (https://canvaslms.com) is a learning management system (LMS).

[5]*Maestro* server can be found at https://github.com/ucinlp/maestro-public.

[6]*Maestro* leaderboard can be found at https://github.com/ucinlp/maestro-leaderboard.

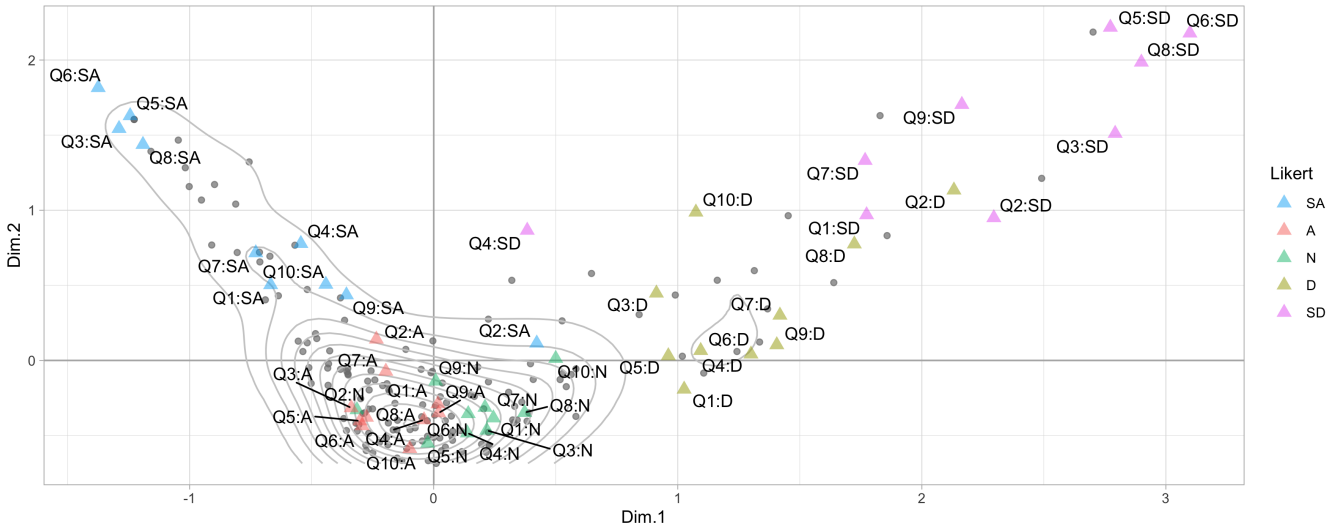

Figure 5: Multiple Correspondence Analysis (MCA) based on indicator matrix ran on the general survey results.

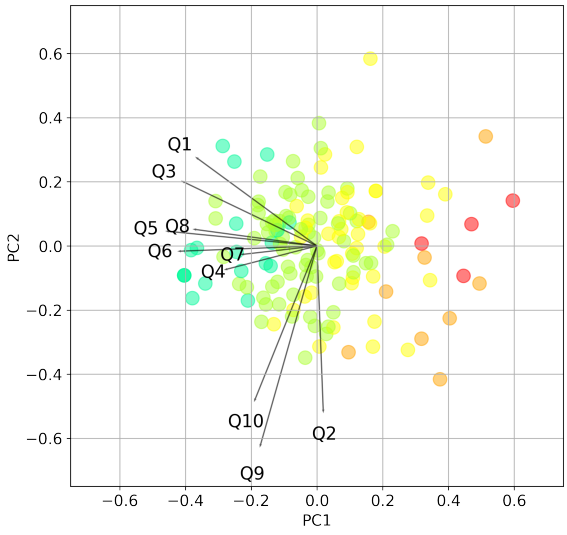

Figure 6: Biplot of Principal Component Analysis (PCA). The origin represents the answers' mean. The arrows are the directions of growth of each variable. Dots represent each projected individual and the color gradient corresponds to the *Effectiveness* variable **Q8** (with *red* = Strongly disagree or **SD**, *orange* = Disagree or **D**, *yellow* = Neutral or **N**, *green* = Agree or **A**, *turquoise* = Strongly agree or **SA**).

straight angles between variables. *Curiosity* **Q5**, *Consolidation* **Q6**, *Maestry* **Q7** and *Effectiveness* **Q8** appear to be highly correlated, which indicates that students who felt the acquisition of new skills in *robust AI* tended to appreciate highly *Maestro*. *Competence* **Q1**,**Q3** and *Maestry* **Q7** appear to have a clear association (see Figure 5 for **Q1**-**Q7** and Figure 6 for **Q1**-**Q3**) reinforcing the idea that game elements play a crucial role in effective learning and skill-development. Another interesting association is between *Connectedness* **Q4** and *Maestry* **Q7** – those individuals who frequently collaborated with others felt more successful in learning. Naturally, those who felt challenged by the amount of material to be learned spent a significant amount of time to delve into material, which is represented by the clear correlation of *Material quantity* **Q9** and *Time usage* **Q10**. Coincidentally, those two variables are also noticeably correlated with *Autonomy* **Q10**, signalling that those individuals who felt too much autonomy are the ones who required more time for material consolidation. In average, the perception of autonomy has been between being balanced and guided enough (Figure 4). *Gamification* variables (**Q1**,**Q3**) and *Educational experience* variables (**Q5**, **Q6**, **Q7**, **Q8**) are nearly perpendicular with respect to *Autonomy* **Q2** and *Academic challenge* variables **Q9**, **Q10** (Figure 6). This fact signifies that the course constraints (e.g., the sense of time usage and workload) uncorrelates with the assessment of *Maestro* as an educational tool, which, as matter of course, suggests that *Maestro* can be adapted to any course length and depth without losing its educational quality.

Conclusion

We successfully implemented and piloted *Maestro* – an innovative educational tool for teaching and learning AI robustness. We assessed the effectiveness of *Maestro* in two offerings of an undergraduate AI course, and collected students feedback with a thoroughly designed reflection survey. Based on the results, we were able to answer our initial research questions as follows:

(i) *Does* Maestro *provide an efficient learning platform for students to interact and acquire skills in* robust AI*?*

Our results suggest that *Maestro* has successfully gam-

ified the learning environment driving students to interaction, *robust AI* mastery and autonomy. The game elements provided in *Maestro* played a crucial role in students' effective learning and skill-development. The *Maestro* leaderboard was highly valued by students. Overall, students who felt the acquisition of new skills in *robust AI* tended to appreciate highly *Maestro*. Lastly, students who frequently collaborated with each other in *Maestro* felt more successful in learning.

(ii) *Does* Maestro *provide students with enough autonomy for training on* robust AI *as a self-paced learning tool?* Defining the teaching strategy on GBSs has allowed students to learn and practice their skills in an immersive experience at different paces. Students felt that autonomy has been between being balanced and enough autonomy. The results also suggest that *Maestro* can be adapted to any course length and depth without losing its educational quality as the course constraints uncorrelated with the assessment of Maestro as an effective educational tool.

Future Work

Our next step is to continue the development and assessment of *Maestro* by offering it as an open-source platform to other instructors, students and the broad *robust AI* community. Future users of *Maestro* can test it and contribute further to its implementation and maintenance. The authors plan to launch an updated version of *Maestro* during 2022-2023 academic year, and continue working on its effective integration in AI courses on and off campus.

Study Survey Questions

Q1 Leaderboards and competitiveness have helped me to work harder.

Q2 Do you consider that the project offered too much autonomy? (1 = *Too guided,* 2 = *Guided enough,* 3 = *Balanced,* 4 = *Enough autonomy,* 5 = *Too much autonomy*)

Q3 *Maestro* has helped me to reinforce the theory.

Q4 I frequently worked with other students to solve problems in class.

Q5 *Maestro* has developed enthusiasm and interest to learn more about course content.

Q6 I was able to understand class material because I could practice in *Maestro*.

Q7 I feel that I developed the ability to solve real AI problems.

Q8 I consider *Maestro* to be an appropriate technology tool which has been effectively used to communicate the course content.

Q9 I felt challenged by the overall amount of material to be learned?

Q10 To be successful, I have spent a significant amount of time on class material.

Q11 Did you like the course organization?

Q12 Which of the following topics did you find the most difficult?

LQ1 I feel that the color of the scores on the leaderboards has been useful / a good feature.

LQ2 I feel that the feature of hiding columns using a dropdown list on the leaderboard has been useful / a good feature.

LQ3 I feel that the feature of sorting the leaderboard by clicking at the name of the corresponding column has been useful / a good feature.

LQ4 I feel that the feature of displaying all my previous submissions on the leaderboard by clicking at my name has been useful / a good feature.

LQ5 I feel that the inclusion of the *filter* search input has been useful / a good feature.

Possible answers for **Q11**: *I prefer this format, just assignments and projects / I prefer evaluation via midterms*; possible answers for **Q12**: *Attacks / Defense / Equal difficulty*; all other questions had a continuous scale of answers: 1 = *Strongly disagree* / 2 = *Disagree* / 3 = *Neutral* / 4 = *Agree* / 5 = *Strongly agree*.

Acknowledgements

This work is supported by the National Science Foundation (NSF) under Grant DGE-2039634. We thank Yanqi Gu (UC Irvine) for assistance in the Spring 2022 course; we thank Ishana Patel (UC Irvine) for assistance in the Winter 2022 course, and we thank Hamza Errahmouni (UC Irvine) for assistance in the early stages of *Maestro* development.

References

Acosta, J. C.; McKee, J.; Fielder, A.; and Salamah, S. 2017. A platform for evaluator-centric cybersecurity training and data acquisition. In *MILCOM 2017-2017 IEEE Military Communications Conference (MILCOM)*, 394–399. IEEE.

Adams, J. C. 1998. Chance-It: An Object-Oriented Capstone Project for CS-1. *SIGCSE Bull.*, 30(1): 10–14.

Ahmet, K.; and Fallows, S. 2013. *Inspiring Students: Case Studies on Teaching Required Courses*. SEDA Series. Taylor & Francis. ISBN 9781135372262.

Becker, K. 2001. Teaching with games: the Minesweeper and Asteroids experience. *Journal of Computing Sciences in Colleges - JCSC*, 17.

Bull, P.; and Lipstein, G. 2014. DrivenData. Accessed on 08.19.2022.

Burguillo, J. C. 2010. Using game theory and Competition-based Learning to stimulate student motivation and performance. *Computers & Education*, 55(2): 566–575.

Carlini, N.; and Wagner, D. 2017. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, 39–57. Ieee.

Chang, H.; et al. 2015. LeetCode. Accessed on 08.19.2022.

Chen, J.; Su, M.; Shen, S.; Xiong, H.; and Zheng, H. 2019. POBA-GA: Perturbation optimized black-box adversarial attacks via genetic algorithm. *Computers & Security*, 85: 89–106.

Deci, E. L.; and Ryan, R. M. 2013. *Intrinsic motivation and self-determination in human behavior*. Springer Science & Business Media.

Doctor, N.; and Hoffner, J. 2012. Codewars. Accessed on 08.19.2022.

Du, W. 2002. Seed project. Accessed on 08.19.2022.

Ebner, M.; and Holzinger, A. 2007. Successful implementation of user-centered game based learning in higher education: An example from civil engineering. *Computers & education*, 49(3): 873–890.

Elsayed, G. F.; Shankar, S.; Cheung, B.; Papernot, N.; Kurakin, A.; Goodfellow, I.; and Sohl-Dickstein, J. 2018. Adversarial Examples that Fool both Computer Vision and Time-Limited Humans.

Fotaris, P.; Mastoras, T.; Leinfellner, R.; and Rosunally, Y. 2016. Climbing up the leaderboard: An empirical study of applying gamification techniques to a computer programming class. *Electronic Journal of e-learning*, 14(2): 94–110.

Furfaro, A.; Piccolo, A.; Parise, A.; Argento, L.; and Saccà, D. 2018. A Cloud-Based Platform for the Emulation of Complex Cybersecurity Scenarios. *Future Gener. Comput. Syst.*, 89(C): 791–803.

Goldbloom, A. 2010. Kaggle. Accessed on 08.19.2022.

Goodfellow, I.; McDaniel, P.; and Papernot, N. 2018. Making Machine Learning Robust against Adversarial Inputs. *Commun. ACM*, 61(7): 56–66.

Goodfellow, I.; Papernot, N.; Huang, S.; Duan, Y.; Abbeel, P.; and Clark, J. 2017. Attacking machine learning with adversarial examples. *OpenAI Blog*, 24.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Green, N. 2021. An AI Ethics Course Highlighting Explicit Ethical Agents. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, 519–524.

Hamon, R.; Junklewitz, H.; and Sanchez-Martin, J. 2020. Robustness and Explainability of Artificial Intelligence. (KJ-NA-30040-EN-N (online)).

Handelsman, M. M.; Briggs, W. L.; Sullivan, N.; and Towler, A. 2005. A measure of college student course engagement. *The Journal of Educational Research*, 98(3): 184–192.

Huang, S.; Papernot, N.; Goodfellow, I.; Duan, Y.; and Abbeel, P. 2017. Adversarial Attacks on Neural Network Policies.

Kalyanam, R.; and Yang, B. 2017. Try-cybsi: An extensible cybersecurity learning and demonstration platform. In *Proceedings of the 18th Annual Conference on Information Technology Education*, 41–46.

Kjorveziroski, V.; Mishev, A.; and Filiposka, S. 2020. Cybersecurity Training Platforms Assessment. In *International Conference on ICT Innovations*, 174–188. Springer.

Krizhevsky, A. 2009. *Learning multiple layers of features from tiny images*. Master's thesis.

Lawrence, R. 2004. Teaching data structures using competitive games. *IEEE Transactions on Education*, 47(4): 459–466.

LeCun, Y. 1998. The MNIST database of handwritten digits. *http://yann.lecun.com/exdb/mnist/*.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Lee, J.; and Hammer, J. 2011. Gamification in Education: What, How, Why Bother? *Academic Exchange Quarterly*, 15: 1–5.

Lepper, M. R. 1988. Motivational considerations in the study of instruction. *Cognition and instruction*, 5(4): 289–309.

Lucic, A.; Bleeker, M.; Jullien, S.; Bhargav, S.; and de Rijke, M. 2022. Reproducibility as a Mechanism for Teaching Fairness, Accountability, Confidentiality, and Transparency in Artificial Intelligence. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(11): 12792–12800.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Malone, T. W. 1980. What makes things fun to learn? Heuristics for designing instructional computer games. In *Proceedings of the 3rd ACM SIGSMALL symposium and the first SIGPC symposium on Small systems*, 162–169.

Mandernach, B. J.; Donnelli-Sallee, E.; and Dailey-Hebert, A. 2011. Assessing course student engagement. *Promoting student engagement*, 1: 277–281.

Mataric, M. J.; Koenig, N. P.; and Feil-Seifer, D. 2007. Materials for Enabling Hands-On Robotics and STEM Education. In *AAAI spring symposium: Semantic scientific knowledge integration*, 99–102.

Mirkovic, J.; and Benzel, T. 2012. Teaching cybersecurity with DeterLab. *IEEE Security & Privacy*, 10(1): 73–76.

Ortiz-Rojas, M.; Chiluiza, K.; and Valcke, M. 2019. Gamification through leaderboards: An empirical study in engineering education. *Computer Applications in Engineering Education*, 27(4): 777–788.

Parker, J.; Hicks, M.; Ruef, A.; Mazurek, M. L.; Levin, D.; Votipka, D.; Mardziel, P.; and Fulton, K. R. 2020. Build it, break it, fix it: Contesting secure development. *ACM Transactions on Privacy and Security (TOPS)*, 23(2): 1–36.

Petit, J.; Giménez, O.; and Roura, S. 2012. Jutge.Org: An Educational Programming Judge. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, SIGCSE '12, 445–450. New York, NY, USA: Association for Computing Machinery. ISBN 9781450310987.

Reigeluth, C. 2013. *Instructional-design Theories and Models: A New Paradigm of Instructional Theory*. v. 2. Taylor & Francis. ISBN 9781135706678.

Sailer, M.; Hense, J. U.; Mayr, S. K.; and Mandl, H. 2017. How gamification motivates: An experimental study of the effects of specific game design elements on psychological need satisfaction. *Computers in human behavior*, 69: 371–380.

Schank, R. C.; Fano, A.; Bell, B.; and Jona, M. 1994. The Design of Goal-Based Scenarios. *Journal of the Learning Sciences*, 3(4): 305–345.

Shen, H.; Deng, W. H.; Chattopadhyay, A.; Wu, Z. S.; Wang, X.; and Zhu, H. 2021. Value cards: An educational toolkit for teaching social impacts of machine learning through deliberation. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, 850–861.

SPLabs, S. R. L. 2014. Sphere Online Jutge. Accessed on 08.19.2022.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2014. Intriguing properties of neural networks. In *International Conference on Learning Representations*.

Tohidi, H.; and Jabbari, M. M. 2012. The effects of motivation in education. *Procedia - Social and Behavioral Sciences*, 31: 820–824. World Conference on Learning, Teaching & Administration - 2011.

Turakhia, B. 2009. CodeChef: Competitive Programming. Accessed on 08.19.2022.

Uguroglu, M. E.; and Walberg, H. J. 1979. Motivation and achievement: A quantitative synthesis. *American educational research journal*, 16(4): 375–389.

Watson, W. R.; Mong, C. J.; and Harris, C. A. 2011. A case study of the in-class use of a video game for teaching high school history. *Computers & Education*, 56(2): 466–474.