Machine vs Machine: Large Language Models (LLMs) in Applied Machine Learning High-Stakes Open-Book Exams

Máquina contra Máquina: Modelos de Lenguaje de Gran Escala (LLM) en Exámenes de Alto Riesgo de Aprendizaje Automático Aplicado con apuntes

Keith Quille
TU Dublin, Dublin, Ireland
Keith.Quille@TUDublin.ie

Csanad Alattyanyi
TU Dublin, Dublin, Ireland
x00164689@mytudublin.ie

Brett A. Becker
University College Dublin, Dublin, Ireland
Brett.Becker@UCD.ie

Róisín Faherty
TU Dublin, Dublin, Ireland
Roisin.Faherty@TUDublin.ie

Damian Gordon
TU Dublin, Dublin, Ireland
Damian.X.Gordon@TUDublin.ie

Miriam Harte
TU Dublin, Dublin, Ireland
Miriam.Harte@TUDublin.ie

Svetlana Hensman
TU Dublin, Dublin, Ireland
Svetlana.Hensman@TUDublin.ie

Markus Hofmann
TU Dublin, Dublin, Ireland
Markus.Hofmann@TUDublin.ie

Jorge Jiménez García
TU Dublin, Dublin, Ireland
x00193937@mytudublin.ie

Anthony Kuznetsov
TU Dublin, Dublin, Ireland
x00162229@mytudublin.ie

Conrad Marais
TU Dublin, Dublin, Ireland
x00170725@mytudublin.ie

Keith Nolan
TU Dublin, Dublin, Ireland
Keith.Nolan@TUDublin.ie

Cianan Nicolai
TU Dublin, Dublin, Ireland
x00160711@mytudublin.ie

RED. Revista de Educación a Distancia. Núm. 78, Vol. 24. Artíc. 4, 30-mayo-2024
DOI: http://dx.doi.org/10.6018/red.603001

Ciarán O'Leary
TU Dublin, Dublin, Ireland
Ciaran.OLeary@TUDublin.ie

Andrzej Zero
TU Dublin, Dublin, Ireland
x00166801@mytudublin.ie

Abstract
There is a significant gap in Computing Education Research (CER) concerning the impact of Large Language Models (LLMs) in advanced stages of degree programmes. This study aims to address this gap by investigating the effectiveness of LLMs in answering exam questions within an applied machine learning final-year undergraduate course.
The research examines the performance of LLMs in responding to a range of exam questions, including proctored closed-book and open-book questions spanning various levels of Bloom's Taxonomy. Question formats encompassed open-ended, tabular data-based, and figure-based inquiries.
To achieve this aim, the study has the following objectives:
Comparative Analysis: To compare LLM-generated exam answers with actual student submissions to assess LLM performance.
Detector Evaluation: To evaluate the efficacy of LLM detectors by directly inputting LLM-generated responses into these detectors. Additionally, assess detector performance on tampered LLM outputs designed to conceal their AI-generated origin.
The research methodology used for this paper incorporates a staff-student partnership model involving eight academic staff and six students. Students play integral roles in shaping the project's direction, particularly in areas unfamiliar to academic staff, such as specific tools to avoid LLM detection.
This study contributes to the understanding of LLMs' role in advanced education settings, with implications for future curriculum design and assessment methodologies.
***Keywords*—** Applied Machine Learning, AI, LLMs, ChatGPT, Transformers, Detection, Performance.

Resumen
Existe un importante vacío en la Investigación de Educación en Computación (CER) sobre el impacto de Modelos de Lenguaje de Gran Escala (LLM) en etapas avanzadas de estudios de grado. Este artículo trata de cubrir este vacío investigando la efectividad de las LLM respondiendo preguntas de examen de Aprendizaje Automático Aplicado en último curso de Grado.
El estudio examina el desempeño de las LLM al responder a una variedad de preguntas de examen, que incluyen modelos de examen diseñados con y sin apuntes, a varios niveles de la Taxonomía de Bloom. Los formatos de pregunta incluyen de respuesta abierta, basadas en tablas, o en figuras.
Para conseguir esta meta, este estudio tiene los siguientes objetivos:
Análisis Comparativo: Comparar respuestas generadas por LLM y por estudiantes para juzgar el desempeño de las LLM.
Evaluación de Detectores: Evaluar la eficacia de diferentes detectores de LLM. Además, juzgar la eficacia de los detectores sobre texto alterado por alumnos con el objetivo de engañar a los detectores.

El método investigador de este artículo incorpora una relación entre seis alumnos y ocho profesores. Los estudiantes juegan un rol integral para determinar la dirección del proyecto, en especial en áreas poco conocidas para el profesorado, como el uso de herramientas de detección de LLM.

Este estudio contribuye a entender el rol de las LLM en el ámbito de la educación universitaria, con implicaciones para el diseño de futuros curriculums y técnicas de evaluación.

Palabras clave: Aprendizaje Automático Aplicado, IA, LLM, ChatGPT, Transformers, Detección, Rendimiento.

1. Introduction

Artificial intelligence (AI), in particular Large Language Models (LLMs), has made remarkable advancements in recent years. Recently, the number of publications about LLMs in Computing Education Research (CER) has increased dramatically. The focus of the research is predominately based on introductory programming, (Becker et al., 2023; Finnie-Ansley et al., 2023; Leinonen, Denny, et al., 2023; Leinonen, Hellas, et al., 2023; MacNeil et al., 2023; Prather, Denny, Leinonen, Becker, Albluwi, Caspersen, et al., 2023; Prather, Reeves, et al., 2023; Shields, 2023; Wermelinger, 2023), where LLMs' performance in examinations and assessment (Finnie-Ansley et al., 2023) and related academic integrity matters (Biderman and Raff, 2022) are the predominant areas of inquiry.

There is a scarcity of research in the CER community on the impact of LLMs in subjects at a more advanced stage than introductory programming, such as machine learning domains. This paper contributes to this by analysing the outputs of LLMs for six years of final year undergraduate examinations in applied machine learning. The examination format for the first three years was proctored closed-book examinations. The format for the recent three years was open-book examinations (Quille et al., 2021; Quille et al., 2020). The exam questions varied (Quille et al., 2020) in the level of Bloom's Taxonomy, and also employed Socratic questions. The format of the questions also varied with open-ended questions, tabular data-based questions, and questions based on figures.

The authors investigated the performance of several LLMs in answering these questions, comparing open- and closed-book questions and finally, identifying at a micro level the examination questions and types for which LLMs' efficacy was strongest and weakest. This level of analysis is rarely, if ever, reported in the literature. The paper then examines LLM detection tools, exploring their efficacy in identifying answers that were entirely generated by an LLM. It considers both original LLM outputs and LLM outputs tampered with by students, focusing on the perspective of academic integrity. This study is designed as a staff-student research partnership (Harrington et al., 2014), with contributions from eight academic staff and six students. Partnership with students involves going beyond engagement, ensuring that students occupy leadership positions within projects, and determine the direction of the project.

This paper focuses on the application of LLMs in answering examination questions in the domain of machine learning. This work provides valuable insights for practitioners who are designing high-stakes examinations for machine learning courses. The examination questions cover a diverse range of topics within the machine learning domain, spanning theoretical concepts, algorithmic implementations, and practical applications. This

study assesses the depth of knowledge and reasoning abilities exhibited by the LLMs in high-stakes examinations. This study helps inform examiners of the question types for which LLMs perform strongly and weakly. This is cross-referenced with student performance on each question to identify questions that are fair to students and that LLMs struggle with. An innovative aspect of this study is the partnership with undergraduate students. Six undergraduate students actively participated in the testing of the language models and collaborated closely with the academic staff to design the experimental setup, collect data, and analyse the results. This collaboration provides valuable insights into the student perspective, contributes to bridging the gap in academia between staff and students, and introduces students to the practices of formal research. This study explores the potential for collusion detection by allocating a fixed time window for tampering with the LLMs' responses to make them indistinguishable from genuine student work. The effectiveness of the collusion detection software is assessed by comparing the tampered LLM answers with the original answers and evaluating the software's ability to identify traces of tampering. The findings, while based upon machine learning examinations, may have value in other fields as the variation in question types and styles (over the six years and two exam formats) may be transferable to other disciplines.

1.1 Research Goals

The methodology for this study was a collaborative staff-student partnership, involving eight academic staff and six students. Unlike traditional LLM evaluations which focus solely on overall accuracy, this approach aimed to delve deeper, to provide actionable insights for educators and students regarding LLM performance across different exam formats and question types.

The goal of this research is to address and understand the effectiveness and use of LLMs for the answering of open- and closed-book exam questions. Criteria for evaluating research goals were established during the team meetings. The collaborative process between students and academic staff involved experimenting with various LLMs and different inputs to define evaluation criteria and identify suitable LLM detectors. A dataset spanning six years of applied machine learning exams, from a BSc. (Honours) Computing programme, was then utilised to compare LLM performance across different exam formats and question types, providing a comprehensive basis for analysis and discussion.

The study addresses two research goals:

1. **Research Goal 1: Evaluate the performance of LLMs at answering exam papers, and different question types (for open- and closed- book exam questions), in applied machine learning courses.** This goal aims to assess and evaluate the effectiveness of LLMs in responding to examination papers and various question types, including both open- and closed-book formats, within the context of an applied machine learning course. The questions from six years of exams on applied machine learning course were fed into the LLM and the generated answers are recorded. Several LLMs and LLM versions were used for this part of the study, including ChatGPT 3.5, ChatGPT 4, and Bing Chat.

2. **Research Goal 2: Assess the effectiveness of LLM detectors, for raw LLM output and student tampered LLM output.** This goal aims to examine the efficacy of LLM detectors in evaluating both, the original, unaltered output, of LLMs and tampered output. The tampered output is where the students modified the LLM output answer to a question, in an effort to disguise this from detection. The study evaluates the performance of these detectors on direct LLM output as well as the student-manipulated language model-generated content. In addition to this, the study examines the LLM performance in distinguishing between authentic and manipulated language model-generated content.

2 Literature Review

2.1 Large Language Models in Programming Education

Large Language Models have been shown to be proficient at many programming tasks at many levels Denny et al., 2024. Further, this is not a very recent development for basic tasks – LLMs were shown to score in the top quartile of human students on real introductory programming (CS1) exams over two years ago (Finnie-Ansley et al., 2022) and similarly for data structures (CS2) exams over one year ago (Finnie-Ansley et al., 2023). However, LLMs are capable of producing more advanced code. Kazemitabaar et al., 2023 showed that LLMs routinely produce code that is too advanced for novices to understand, and Becker et al., 2023 questioned the appropriateness of LLMs for novices, particularly given that much of the code they have been trained on is more advanced. Prather, Denny, Leinonen, Becker, Albluwi, Craig, et al., 2023 interviewed over 20 CS1 instructors and at least one theme emerged supporting this questioning, with several instructors reporting that students simply copy and paste LLM-generated code containing advanced approaches that were not taught in the course they were in. Li et al., 2022 showed that LLMs can be proficient on advanced programming competition challenges – at least when trained on such data as DeepMind AlphaCode. Such a task should not be underestimated, given that successfully solving such problems requires complicated natural language descriptions, reasoning about novel tasks – not just memorising code snippets – dealing with diverse data structures and algorithms, and implementing programs that can run into hundreds of lines of code (Li et al., 2022). Savelka et al., 2023 tested GPT-4 on complicated coding projects including multiple-file code bases, noting ”the rate of improvement across the recent generations of GPT models strongly suggests their potential to handle almost any type of assessment widely used in higher education programming courses” (Savelka et al., 2023). MacNeil et al., 2023 used LLMs to generate multiple types of code explanations, integrating these into a web software development e-book with success.

LLMs are not only proficient in generating code, but helping programmers in working with code. Ribeiro et al., 2023 developed a tool which leverages GPT-3 to automatically repair type errors in OCaml programs, outperforming two other techniques – demonstrating proficiency in less-common languages, and in advanced programming capability outside code generation. Leinonen, Hellas, et al., 2023 showed that LLMs are proficient in providing more helpful programming error messages than standard compilers, a known barrier for those learning to program (Karvelas et al., 2020) as well as professionals

(Becker et al., 2019). Santos et al., 2023 showed that when the code context is provided to GPT-4 along with the code context from which the error occurs, a correct fix is provided 100% of the time for the errors they tested.

Prather, Denny, Leinonen, Becker, Albluwi, Craig, et al., 2023 found that LLMs have affected forums more than web searches due to the fact that they are currently used more by upper-level students, while lower-level students continue to rely on peers for help more. They also identified a trend in their instructor interviewees revealing a sentiment that LLMs should not be used in lower-level (or basic) courses, but should be allowed in upper-level (or more complicated) courses. All of these factors lead to the possibility that LLMs are capable, both in terms of upper-level course content including exams, and that upper-level students are capable of using LLMs for such.

2.2 Open-Book Exams

Although there seems to be conflicting evidence in the literature on the advantages of open-book assessments vs closed-book exams, open-book assessments are often viewed positively by students. It is well recognised that computing students are prone, like other students, to mental health problems during their coursework, with stress and anxiety—including exam anxiety—becoming more prevalent concerns recently Dooley et al., 2019; Nolan and Bergin, 2016; Nolan, Bergin, and Mooney, 2019; Nolan et al., 2015, 2019a, 2019b; Quille and Bergin, 2015; Quille, 2019; Quille et al., 2019; Quille et al., 2022. Using a survey designed to gauge students' views of exam anxiety related to both online and paper-based exams, Deloatch et al. examined the relationship between exam modality and students' perceptions of exam anxiety and performance during programming exams (Deloatch et al., 2016). Out of the 391 students that took part, 22% (n=61, $\bar{x}$=4.26, SD=1.51) felt they had high test anxiety for examinations that were conducted on paper, and 23% (n=64, $\bar{x}$=4.15, SD=1.67) felt they had high test anxiety for exams that were conducted online. De Raadt (2012) suggested a way to let pupils make "cheat sheets" for tests. Exam scores did somewhat improve for the 89 students who took part and were given permission to use a cheat sheet, and those who did so reported feeling less anxious both before and during the test. Making cheat sheets could also help with retention by reinforcing previously taught information. Green et al. conducted research on the use of open-book evaluation to improve student performance and discovered that this technique can help to deepen comprehension in both cooperative learning and standard classroom settings (Eilertsen & Valdermo, 2000). Students stated after the exam that their stress and anxiety levels were lowered by having access to reference materials during the test. However, they did note that there was a problem with timing.

3 Overview of AML Course and the Open- Book Assessment

Our work focused on Applied Machine Learning (AML), a final year undergraduate module with six student cohorts. The reasoning for this module selection was that it included final-year high-stakes examinations. Other modules were considered, such as CS1, Advanced Routing and Switching and Enterprise Database Technologies. However, these had large components of continuous assessment which reduced the value and the stakes of an examination component, if any existed. The students sitting open-book exams within this

module consisted of both part-time (PT) and full-time (FT) students. In this instance, full-time students were majoring in Computing, with minors in Software Development or AI and Machine Learning. All student cohorts were enrolled in a 4-year Honours Bachelor Degree at Level 8 on the National Framework of Qualifications (NFQ) in Ireland (of Ireland, 2009) (Honours Bachelor's Degree Level).

3.1 The Applied Machine Learning Course

The AML course descriptor (Quille et al., 2020) is based on the following learning outcomes:

1. Apply data pre-processing and data exploration techniques in the context of the machine learning process.
2. Demonstrate knowledge of machine learning techniques, their methods and application.
3. Determine the machine learning techniques and methods for particular scenarios.
4. Evaluate the models produced, using relevant performance metrics.

The programming language used is Python, with an in-house Jupyter Lab server as the IDE, which is a cloud-based Jupyter notebook. Some students also used local Anaconda installs, as the computation power required for traditional machine learning was not significant. Assessment is broken down into two in-class assignments, calling for students to investigate a dataset and use suitable data pre-processing methods in order to facilitate machine learning. Students also receive a pen-and-paper assessment in which they are given a specific dataset or issue, along with model outputs and performance measurements (confusion matrices). They have to evaluate the results and produce conclusions, including statistical testing. The two assessments have a combined weighting of 50% with the end-of-term exam worth the remaining 50%.

3.2 The Proctored Closed-Book Exam

Proctored closed-book exams are assessments which are conducted under the supervision of a proctor/invigilator. These exams are designed to be time-limited and are usually short in duration; at our university, they are typically 2 hours in length. The exam takes place in a large hall, with multiple other exams taking place at the same time.

Closed-book exams require more memorisation and rote learning, which means that closed-book exams are better suited for students to be able to recall specific information, whereas open-book exams allow students to elaborate and explore deeper meaning within the content of the exam or course. Proctored closed-book exams are reassuring to examiners and external accrediting bodies as they ensure a high degree of academic integrity.

3.3 The Open-Book Exam Format

The format and method of delivery of the open-book exam were top priorities. We used a student-centred approach, consulting openly with the students before presenting our format for approval to the university. Since this was a first for our university, different instructors utilised different strategies, such as using a quiz or making the traditional closed-book exam an elapsed piece of work (Moodle is the Virtual Learning Environment used in our university). All of the approaches implemented had to be submitted and approved by the Head of Department and/or the Academic Council.

The method of delivery that we selected for the AML course was an open-book exam carried out on a computer (submitted via a Microsoft Word/PDF file). Based on student concerns voiced during the open discussion phase, the duration of the exam was extended by 30 minutes to allow for uploading and navigating the exam itself. Additionally, we reduced each question from a typical structure of four parts per question to three parts per question (to further alleviate time concerns). This was in addition to the regular format of choosing any three of four questions from the traditional exam. An exemplar paper was also created using an open-book online exam format (Quille et al., 2021) for students to use as a revision aide (this was adapted from a previous sample closed-book paper, where both papers were presented to the students). The Newcastle guide was also provided to students and the mapping was explained, so students could map historical (traditional) exam paper questions for additional revision examples. In anticipation of issues/concerns, several study/revision sessions were provided to demonstrate the mapping process using the sample paper and to allow students to practice the open-book questions. The examination was provided in PDF format, which could be downloaded in case of any internet or technology issues. To ensure there was no technical discrimination, the exam paper was opened for viewing 10 minutes prior to the exam. During these ten minutes, each question from the exam paper was read out to the students via the lecturer. While this was primarily to aid students with additional needs (including the printing of the paper), the overall feedback was very positive on this, from all students. Students with additional needs had an additional 20 minutes for the examination. In addition, an optional session was provided before the exams on using Microsoft Office's accessibility tools such as screen readers and dictation to further prepare students. The majority of students availed of this.

Plagiarism on open-book online tests was one of the institutions' main concerns regarding academic integrity. An initial strategy which was created was based on the ideas of academic integrity. Students signed the university-wide plagiarism policy, and each revision class included a discussion on the policy's meaning and its implications (both in terms of ethical implications and potential disciplinary consequences). In several instances, it was discovered that highlighting institutional policies actually decreased the quantity of plagiarism in computer science courses Quille et al. 2021. An innovative, proactive (and perhaps also reactive) approach was the inclusion of a viva after the examination. This consisted of a ten-minute viva-style session with 20% of each exam cohort. Students were randomly pre-selected for this viva prior to the exam. These students were not informed that they were pre-selected to avoid any undue stress or anxiety. Students were pre-selected to avoid any selection bias based on any events which may have arisen during the exam. The pre-selected students took part in the viva straight after the exam. Most importantly, the students were not assessed on the correctness of their

answers. They were asked Socratic questions such as, ”Where did you get that idea?” or ”By what reasoning did you come to that conclusion?”. All students, regardless of if they were selected for the viva or not, were provided with details about the process prior to the exam. Students were asked for their consent to record their viva. 100% of selected students agreed to this and the recording was within GDPR compliance. No students were flagged as plagiarism concerns based on the viva responses. Finally, Urkund/Turnitin (https://www.urkund.com/), the university-selected plagiarism detection software was used for the final student exam uploads which were in word or PDF format. Urkund/Turnitin reports the percentage of plagiarism and for each instance, the source of the plagiarism. In some cases, the tool reported a relatively high amount of plagiarism, but on further investigation, several students copied the exam questions (perhaps for a placeholder or guide) and this resulted in a higher plagiarism score. In Urkund/Turnitin it is possible to manually remove these cases from the scores. No students had a concerning plagiarism score (greater than 25%) after manual processing. This result, coupled with the viva also finding no plagiarism concerns, was a positive outcome.

4. Methodology

The methodology for this paper adopted a staff-student partnership approach, involving contributions from eight academic staff and six students. The large number of contributors was to capture all facets that may emerge when developing the methodology, to try to ensure that the experiment design was deeper than just reporting the performance of LLMs overall in an exam. While it may be useful to determine the overall accuracy of LLMs compared to student results, it does not tell the full story. Just like reporting accuracy in a machine learning model, it does not provide deeper details (for example Sensitivity or Specificity) that may help in developing guidelines or to identify which questions or exam types (proctored or open book) LLMs perform better or worse on. Thus, providing actionable guidance for educators and students alike.

The six students were final year undergraduate students. The students were selected from the cohort who in the previous semester, took the AML module. Initially we aimed to recruit four students, as there was funding to pay students the university student helper rates for the project. Six students applied, thus, to avoid selection bias we recruited all six students who applied. The rationale for selecting students who had just completed the module was their familiarity with the content and assessment. The students in total completed 62 hours each for this project. This was arrived at between co-design meetings, face validity meetings and finally the work with submitting questions into LLMs, recording the raw outputs, running the raw outputs through the LLM detectors and finally tampering with the LLM raw outputs and then rerunning the LLM detectors.

For each Research Goal, the staff-student partnership was a co-design approach, where we adopted a validity measure to ensure that the methodology approaches were developed and agreed upon at multiple stages. Face validity determines whether or not the instrument appears to measure what it is intended to measure (Trochim, 2006). The 16-person team met six times between March and June of 2023. Following this the data collection phase for Research Goals 1 and 2 commenced. Initially, at the first meeting, a

provisional set of criteria was proposed for evaluating Research Goals 1 and 2. The subsequent meetings involved the students and academic staff trialling various LLMs with trial inputs to develop a final set of criteria for Research Goal 1 that would be used to generate the solutions from the LLMs. The meetings also included an iterative and student-guided (as they are on the ground using tools) identification and a trail of LLM detectors. It should be noted that many of the tools identified were previously unknown to the academic staff (such as Sapling).

To address Research Goals 1 and 2, six years of applied machine learning exams were selected. The rationale for this was that for the prior three years of exams, the Applied Machine Learning course exam was an open book exam, and the three years before that the exam was a closed book pen and paper proctored exam. This allowed for the comparison of LLMs not only across three years of each type of exam format but also a large sample of questions from each exam format to facilitate a deeper dive into the question types and LLM performance for Research Goal 1. This also left a large body of questions for the detector component of Research Goal 2. The exam papers and marking schemes can be found at http://tiny.cc/RED24.

4.1 Research Goal 1: Evaluate the performance of LLMs at answering exam papers, and different question types (for open and closed book exam questions), in applied machine learning courses.

During the six group meetings, the criteria for the inputs to the LLMs as well as the LLMs themselves were agreed between academics and students. The LLMs selected were:

- Microsoft's Bing Chat
- Open AI's ChatGPT 3.5 (free tier)
- Open AI's ChatGPT 4 (paid version)

The rationale for the selected LLMs was that both, Bing Chat and ChatGPT 3.5, were free and openly available in Ireland at the time of the research. ChatGPT 4 was not free, but the hypothesis was that due to the claims of significant performance improvements, students may opt to purchase and use this version. Bing Chat, while based on ChatGPT 4, was also free and came with an addition that the user could select one of three options for the text generation, that is: more creative, more balanced, and more precise. For this study "more creative" was selected. Finally, Bard (from Google) was unavailable in Ireland at the time of the data collection. Based on the trial inputs and face-validity iterative meetings, the following criteria were identified and used to feed exam questions into the LLMs. This work was conducted by the students. The criteria were:

- No images were used as inputs.
- Raw verbatim word input, thus no student rewording of the exam questions.
- The conversation context will be kept, that is all questions will be in one chat.
- Only text, no helper text, no descriptions, or no LaTeX (to help with formula, etc).

- Table data if applicable in the exam question is just copied from the exam PDF, with some basic formatting, such as spaces, and tabs only (no rewriting in LaTeX).
- The first output from the LLM was used, with no additional regenerating.
- Screenshot and timestamps for all questions and responses were recorded.

The resultant question responses from the LLMs for each of the six exams were stored and they are available at http://tiny.cc/RED24. Each question was graded individually thus a comparison per question type was also possible. A sub-question example used in the face validity phases included (Question 1, part C from the January 2023 open book exam paper where all papers can be found at http://tiny.cc/RED24):

(C) "The target class should be examined prior to running a classification model".

i. Explain in your own words why it may be useful to examine the target class prior to running a classification model.

ii. Discuss what issues may arise if this exercise is not conducted, specifically in relation to Sensitivity and Specificity.

The output using ChatGPT 4.0 was:

(i) Importance of pre-examination of class distribution: Pre-examination of the class distribution is an essential step before developing classification models for several reasons: 1. Imbalanced class distribution: In many real-world datasets, the class distribution is often imbalanced, with one or more classes having significantly fewer instances than others. Imbalanced class distribution can lead to biased models that favor the majority class, resulting in poor performance on minority classes. By examining the class distribution before model development, you can identify potential imbalances.....

This was deemed an appropriate response (acknowledging that this is the purpose of these LLMs) during the face validity phase. This grading was done separately by two of the academics per LLM, with additional comments added for each response. One of the authors was the academic who delivered the AML module, and for validity purposes, two additional authors/academics graded the LLM response, with support in the form of marking schemes and sample answers, and validation from the AML lecturer. The rationale for the additional comments was to identify any patterns that may have existed when they were grading the LLM outputs. The comments were organised per LLM type, for example: *GPT3.5 is good at descriptive answers, but weak at giving rationales compared to Bing. It also presents lists when it could do paragraphs.* This additional anecdotal feedback from the graders will help support the grades identified and the discussion following the results. Finally, the findings of this will be used to compare the grades of each LLM to that of the student's grades. Performance per question will be assessed to identify LLMs strengths in question types and styles. Questions with notably strong or weak performance will be highlighted. All questions and grades can be found at http://tiny.cc/RED24.

4.2 Research Goal 2: Assess the effectiveness of LLM detectors, for raw LLM output and student tampered LLM output.

Initially, the raw outputs from the LLMs were fed into AI detection tools. The tools selected using the iterative face validity meetings were: Urkund/Turnitin, GPTZero, Writer.com, and Sapling.ai. Each of the selected LLM detection tools offers some functionality that the students and academic staff identified as useful for detection, though many of the tools differ in their approaches internally. These outputs were recorded and used for the grading of the exams by the two graders to investigate Research Goal 1. The detector results were recorded for each question response and are available at http://tiny.cc/RED24.

Following this, to support the hypothesis that minimum student tampering could alter the AI detectors' scores the students tampered with the raw LLM outputs and resubmitted the responses back into an AI detector. Students, when tampering with the raw LLM output, could manually modify the output or use tools like QuillBot[1]. The students then fed tampered answers into the detection tools where responses were recorded to identify the performance of the tools. To identify a suitable time allowed per question for tampering (irrespective of the method used), the students timed themselves taking an applied machine learning exam and identified an average time to complete the exam. It was agreed that in the remaining time, students could try and tamper with the LLM responses. The students agreed on a ten-minute window, which covered 3 minutes per question and 1 minute per sub-part of the question. A sub-question may look like this (Question 1, part C from the January 2023 open book exam paper where all papers can be found at http://tiny.cc/RED24):

> *(C) "The target class should be examined prior to running a classification model".*
>
> *i. Explain in your own words why it may be useful to examine the target class prior to running a classification model.*
>
> *ii. Discuss what issues may arise if this exercise is not conducted, specifically in relation to Sensitivity and Specificity.*

Thus for *(C)* part *i* and *ii* the students had a total of one minute to modify the LLM output. This was a relatively small amount of time to allocate to this task, however, after several meetings and face validity (which included discussions and test runs per question) this time was agreed. For example, one such trial using the tampering time of one minute resulted in:

Raw LLM output:

(i) Importance of pre-examination of class distribution: Pre-examination of the class distribution is an essential step before developing classification models for several reasons: 1. Imbalanced class distribution: In many real-world datasets, the class distribution is often imbalanced, with one or more classes having significantly fewer instances than others. Imbalanced class distribution can lead to biased models that favor the majority class, resulting in poor performance on minority classes. By examining the class distribution before model development, you can identify potential imbalances.....

[1] https://quillbot.com

Tampered LLM Output:

(i) Importance of pre-examination of classification distribution: Pre-examination of the category distribution is a crucial step earlier than creating classification fashions for quite a few reasons: 1. Imbalanced type distribution: In many real-world datasets, the classification distribution is frequently imbalanced, with one or extra training having substantially fewer situations than others. Imbalanced classification distribution can lead to biased fashions that want the majority class, resulting in negative overall performance on minority classes. By inspecting the classification distribution earlier than mannequin development, you can discover possible imbalances.....

Interestingly, using one detector for the trial (Sapling) the raw LLM output was identified as AI generated with a certainty of 99.9% whereas the tampered output resulted in an AI-generated score of 0.2%. The students felt that this was an appropriate amount of time to tamper with the LLM-generated answers and noted the significant impact on the AI detectors score. All of the LLM tampered outputs can be found at http://tiny.cc/RED24, as well as AI detectors scores.

5. Results

The following sections (Section 5.1 and 5.2) discuss the results, which are framed per research goal (RG1 and RG2), as discussed in Section 1.1.

5.1 RG1: Evaluate the performance of LLMs at answering exam papers, and different question types (for open- and closed- book exam questions), in applied machine learning courses.

This section investigates the performance of LLMs on open and closed-book applied machine learning exams through three lenses, first a comparison of the performance of LLMs compared to that of students for each of the six years of exams investigated in this paper (this included open and closed book exam formats). Second, we compare the LLM's performance on open and closed-book exams. Finally, we identify any question types that the LLMs perform well in and where LLMs struggle. The following subsections present the investigation of Research Goal 1 through the three lenses. First, a comparison between LLM and student grades (5.1.1), then performance per question types (5.1.2), finally RG1 discusses performance per question types (5.1.3).

5.1.1 Comparison between LLM and Student Grades

Table 1 presents the grades for each exam for the LLMs and for the full-time students who sat the exams for each year. Both the LLM grades (that is for Bing, ChatGPT 3.5, and ChatGPT 4) and the student grades are averaged. The sample size for the LLMs and students is also reported for each year's exam.

1: Comparison of LLMs to student grades. Quantitative measure of the magnitude of a phenomenon (Cohen's *d*), that a *p*-value may not truly represent (Lortie-Forgues & Inglis, 2019). The effect size observed (Cohen's *d*) was large at 4.05. This indicates that the magnitude of the difference between the average of the differences and the expected average of the differences is large. Thus confirming that the LLMs significantly outperformed student performances in the applied machine learning course. Interestingly

the worst performance for both students and LLMs was in the 2022 open-book exam, where this will warrant further investigations.

Table 1

Comparison of LLMs to student grades.

Exam	LLM Avg	LLM N	Student Avg	Student N
2019 Closed Book	87%	3	61%	34
2020 (Jan) Closed Book	97%	3	54%	22
2020 (May) Closed Book	79%	3	54%	17
2021 Open Book	78%	3	47%	28
2022 Open Book	67%	3	43%	18
2023 Open Book	88%	3	62%	25
Average	**83%**		**54%**	

quantitative measure of the magnitude of a phenomenon (Cohen's *d*), that a *p*-value may not truly represent (Lortie-Forgues & Inglis, 2019). The effect size observed (Cohen's *d*) was large at 4.05. This indicates that the magnitude of the difference between the average of the differences and the expected average of the differences is large. Thus confirming that the LLMs significantly outperformed student performances in the applied machine learning course. Interestingly the worst performance for both students and LLMs was in the 2022 open-book exam, where this will warrant further investigations.

5.1.2 Performance in open and closed book exams

Table 2 presents the performance per LLM per exam (overall grade per exam). The exams and grades assigned by the two graders can be found in http://tiny.cc/RED24.

Table 2

Comparison of LLM Grades for open and closed-book exams.

Exam	Bing	GPT 3.5	GPT 4	Average
2019 Closed Book Exam	80%	83%	97%	**87%**
2020 (Jan) Closed Book Exam	93%	97%	100%	**97%**
2020 (May) Closed Book Exam	68%	72%	98%	**79%**
2021 Open Book Exam	64%	83%	88%	**78%**
2022 Open Book Exam	48%	67%	85%	**67%**
2023 Open Book Exam	73%	90%	100%	**88%**

There is variability across LLM types and yearly exams. This ranges from 48% in an exam to 100% in an exam. Next, we will examine the overall difference per LLM type for open and closed-book exams to identify if the differences were statistically significant (thus investigating if LLMs are better, worse, or indifferent per exam type). Table 3

presents the average percentage per LLM for both open and closed-book exams. A student's *t* -test was conducted for each LLM comparing the performance of the LLM on open and closed-book exams. For each LLM the *p*-value was greater than 0.05, reporting that the differences for each LLM for the two exam modes were not statistically significant.

Table 3

Comparison of LLMs grouped per exam type.

Exam	Bing	GPT 3.5	GPT 4	Average
Closed Book Exams Avg	80%	84%	98%	**88%**
Open Book Exams Avg	62%	80%	91%	**78%**
Closed Book Exams STd	10%	10%	1%	**7%**
Open Book Exams STd	10%	10%	6%	**9%**
***p*-Value**	0.3276	0.8020	0.2786	
Effect Size	0.7400	0.0160	0.8500	

However, *p*-values are not always informative especially when sample sizes are small or the context is not traditional. In educational contexts, an effect size of 0.2 was identified as a threshold for educational evaluation trials (Lortie-Forgues & Inglis, 2019). If this threshold is applied, both Bing and ChatGPT 4 perform worse for open-book examinations reporting an effect size greater than 0.2. Finally, if we compare all three LLM's performances across open and closed-book exams, we find a similar finding. The *p*-value reports no statistically significant (*p*-value = 0.1511) however the effect size (even outside of an educational context) was large (d = 1.31). This confirms that the LLMs do not perform as well on open-book exams.

5.1.3 Performance per question types

For this Research Goal, all of the grades per question and the LLM responses can be found at http://tiny.cc/RED24. This section also presents the findings through three lenses. That is, questions that were common to the open and closed book exams, questions that were specific to the closed book exams, and finally questions that were specific to the open book exams. In addition, based on the findings from Table 2 the LLMs on average scored very well in the exams, and while there were differences (for example Bing scoring significantly worse than ChatGPT variants) we only looked at questions where all three LLMs scored 50% or less for that question. This criterion was based on the evidence that in the majority of cases, at least one LLM (mostly ChatGPT 4) scored full marks for each question, and while we had initially intended to report on questions that the LLMs scored poorly on as well as strongly on, we will only report on questions where each LLM scored poorly on or gave generic answers rather than specific. The LLM solutions and complete grading sheets from the two graders can be found at http://tiny.cc/RED24.

Questions common to both exam formats

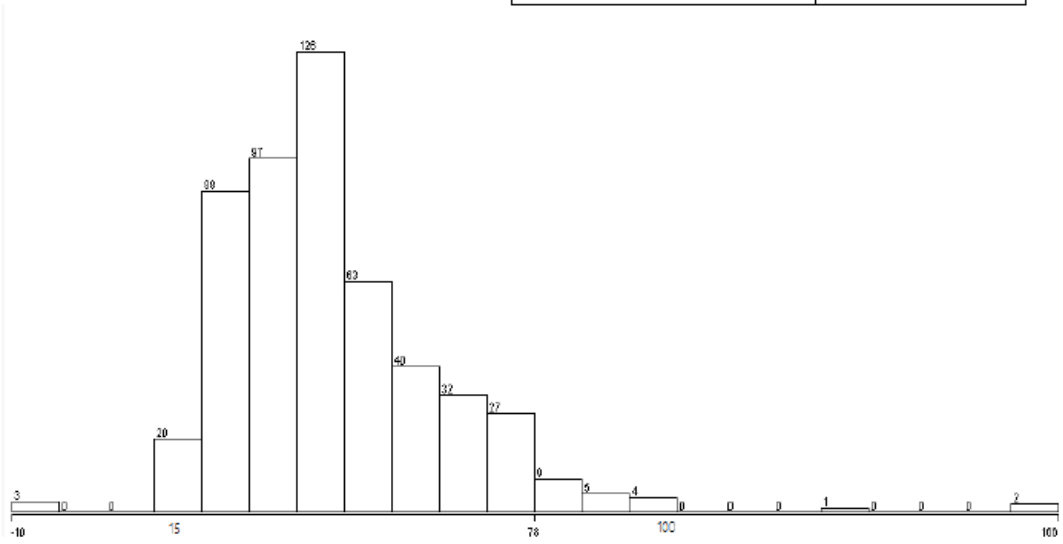
Interestingly, there were very few questions that were the same for both open and closed-book exams, and the question that was verbatim to each exam format was Question 1 part b (the questions were the same but the dataset varied, thus the figure and corresponding data values differed year on year). An example from the 2019 closed-book exam is presented in Figure 1, which shows a typical Q1 part b question in both closed and open-book exams.

The weighting of this question is 14 marks from a possible 24 marks per question and a total of 72 marks for the paper. In later years the marks changed but the approximate weightings are the same. The goal of this question is to evaluate the students on identifying outliers and missing data. To do this they must look at the figure and use the statistics presented. In addition, Table 4 presents the grades for each LLM on the 2019 Q1 part b question, the question is divided into sub-parts with the marks allocated also being presented.

Figure 1
2019 Closed Book exam, question 1 part b

(B) The Histogram presented, represents the number of individual days (y axis) and the relative humidity in percentage (normal range of 15% to 100%, x axis). Describe this histogram, and what values if any, would you identify as concerns (consider for marking). Finally, what steps (and the rationale for your choice) would you take for each identified concern?

Mean	44.63
Standard Deviation	18.66
Minimum	-10
Maximum	166
Number of Instances	517



(14 Marks)

Table 4

Comparison of LLMs on 2019 Q1 part b

	Marks Available	Bing	ChatGPT 3.5	ChatGPT 4
Describe the skew	2	1	2	2
Identify and create filter for missing	3	0	2	2
Examine STDev (for outliers)	5	0	2	2
Create filters for outlier	4	1	2	4

In addition, for a deeper dive, the reviewer comments are also presented for each response for each LLM, Table 5. This was interesting, as each of the LLMs was able to answer the questions generically, there were some exceptions for the LLMs (both strengths and weaknesses of the LLMs):

- **Strengths:** Both ChatGPT 3.5 and ChatGPT 4, while not being able to use the figure (histogram) in the question, were to use the accompanying statistical summary (that did not contain information about skew or kurtosis for example) to correctly discuss the skew and shape of the histogram. Both ChatGPT 3.5 and ChatGPT 4 were also able to discuss filters and solutions to what to do with the outliers, this is presumably from the interpretation of the statistical summary. This could be removed in the future.
- **Weaknesses:** All models scored less than 50% for the component of the question that asked the students to examine the figure for outliers or missing data. The models at best described a generic response or responded that they can not provide an answer.

Closed Book

For question types that were only available for closed book format, there were very few that the LLMs struggled to answer. One question type however LLMs struggled with, when they had access to all of the questions, was questions involving calculations, specifically calculating confusion matrices. An example of this closed-book question type can be found in Figure 2. The responses and errors were also common in the majority of other years with closed-book exams.

Figure 2
2020 Closed Book exam, question 2 part a

(A) Given the following two confusion matrices for classification models ***a*** and ***b***, calculate: Accuracy, Sensitivity and Specificity. Compare the two models explaining which model is most appropriate and why (outlining any concerns), considering that both models are trying to find if a shop is running low on stock, as represented by the output class "**0**".

a	b	<- classified as
2490	458	a = 0
859	820	b = 1

Model a

a	b	<- classified as
589	2359	a = 0
335	1344	b = 1

Model b

Table 5

Grader Comments for the 2019 Q1 part b question

	Bing	**ChatGPT 3.5**	**ChatGPT 4**
Describe the skew	Can't answer , but gives general advise	Made good use of accompanying table to describe skew	Made good use of accompanying table to describe skew
Identify and create filter for missing	Can't answer	Good answer	Good answer
Examine STDev (for outliers)	Can't answer	Generic answer	Generic answer
Create filters for outlier	Cant' answer, but does say "some steps that could be taken include removing outliers or transforming the data to make it more normally distributed"	Good answer	Excellent answer, includes formulae

The rationale for this question was not only to evaluate students on their confusion matrices calculations but also their evaluation of which model is the best (which means the students must take into account not just the accuracy but also the sensitivity and specificity, where in some cases these may be below that of chance, 50%). Table 6 presents the grades for Question 2, from the 2020 closed book exam question.

Table 6

Grades for the 2020 Q2 part b question

	Available Marks	Bing	ChatGPT 3.5	ChatGPT 4
Calc for Model A	4	0	4	1
Calc for Model B	4	1	2	1
Model A is answer	2	0	0	1

Interestingly, and this pattern was repeated for other closed book exams, the LLMs struggled both with the calculations of each model's confusion matrices and the selection of the most suitable model. It also varied for which calculation the LLM got incorrect, but for each LLM it got at least one of the three metrics (Accuracy, Sensitivity and Specificity) incorrect for at least one model's calculations. In addition, each LLM struggled to identify which model was the best from the calculations it just completed. Table 7 presents the grader comments. The LLM responses and full grader grading sheets can be found at http://tiny.cc/RED24.

Table 7

Grader comments for the 2020 Q2 part b question

	Bing	ChatGPT 3.5	ChatGPT 4
Calc for Model A	All Wrong	All Right	Only Accuracy Right
Calc for Model B	Only Specificity Right	All but Accuracy Right	Only Accuracy Right
Model A is answer	Wrong Answer	Wrong Answer	Partially Right Answer

Open Book

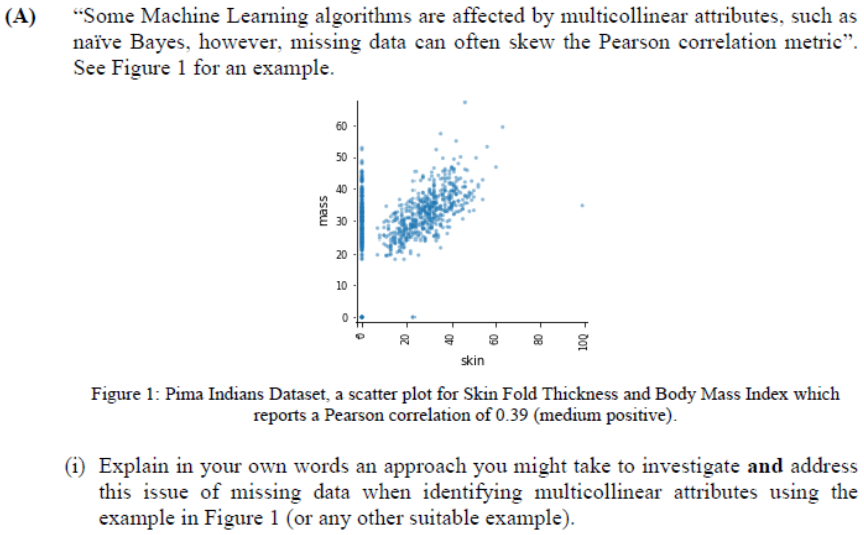
While for the closed-book examinations, the LLMs did exceptionally well for all of the questions (except the common question to open and closed-book exams and the calculation questions), the LLMs for open-book questions struggled with several. In fact if you take the the closed book exams, the LLMs struggled with two such examples (including the common question), however in open book exams, they struggled with five, including the common question. On a deep dive, however, it was clear that there was a theme consistent with all four questions identified as difficult for LLMs to answer in the open book format. These were questions based on ML or AI bias. The questions identified were:

- 2021, Question 2 part ci.
- 2022, Question 1 part a.

- 2022, Question 1 part c.
- 2023, Question 1 part a.

For the purposes of this study and space constraints, we will look at in detail one question, and for open-book papers that was 2023 Q1 part a. Figure 3 presents the question.

Figure 3
2023 Open-book exam, question 1 part a.



(A) "Some Machine Learning algorithms are affected by multicollinear attributes, such as naïve Bayes, however, missing data can often skew the Pearson correlation metric". See Figure 1 for an example.

Figure 1: Pima Indians Dataset, a scatter plot for Skin Fold Thickness and Body Mass Index which reports a Pearson correlation of 0.39 (medium positive).

(i) Explain in your own words an approach you might take to investigate **and** address this issue of missing data when identifying multicollinear attributes using the example in Figure 1 (or any other suitable example).

It should be noted that while the question allowed candidates to use the figure in the question as a discussion point, it also allowed for the use of any suitable data example. Table 8 presents the grades.

Table 8

2023 Open book exam, question 1 part a grades.

	Available Marks	**Bing**	**ChatGPT 3.5**	**ChatGPT 4**
Investigate missing data	2	1	1	1
Address missing data	4	1	1	1

For each response in the question, the LLMs grader comments were *Generic answer, no real link to the data or figure.* This was a common theme across the LLMs for each of the four questions around bias in open-book exams.

5.2 RG2: Assess the effectiveness of LLM detectors, for raw LLM output and student tampered LLM output.

This section aims to identify the effectiveness of AI detection tools using two approaches, that is the raw output for each LLM and for the tampered output where students had a fixed set of time to tamper with the output to reduce the AI detector scores. The tools selected by the students were: Turnitin[2], GPTZero, Writer.com, and Sapling.ai. The full set of LLM outputs (both raw and tampered) as well as the AI detector scores can be found in http://tiny.cc/RED24. When presenting the results it will be difficult to display all of this content in the paper, thus our approach will be to compare the AI Tool's performance on raw output data for both open and closed-book exams, where the average score (typically between 0 and 100%) for the detector is presented. This of course means that some detectors can do better or worse for specific types of questions, and this can be investigated further in http://tiny.cc/RED24. Finally, each of the outputs was AI-generated, thus all scores, if the tool was performing well should be predicting 100% AI-generated for each question.

It should be noted that while GPTZero was employed for this investigation and the results were recorded (http://tiny.cc/RED24), the output of GPTZero was a dichotomous decision in text and a perplexity score[3]. This metric has no specific threshold (there is a guide value) or no upper limit, thus it was decided for comparison purposes to exclude GPTZero from the investigation. In addition, Turnitin was only used for the raw LLM output detection as due to the very poor performance it was decided not to include it in the investigation for the tampered LLM output. Finally, there are some NA values in the raw and tampered investigation, which were due to time constraints and student availability. Future work will include these missing detector scores. The following two sections (5.2.1 and 5.2.2) report the findings for the raw LLM output and the tampered student output respectively.

5.2.1 Raw output

Table 9 presents the detector scores for the raw LLM outputs. The most note-worthy point is that Turnitin had the lowest average identification of the LLM outputs. The overage score for Turnitin was only 6.01%. This was very concerning as every output it was tested on was AI-generated with no tampering. Between the two detectors Writer and Sapling, the difference in performance is statistically significant (students *t* -test) with a *p*-value is 0.0022. The effect size is also very large with Cohen's d = 128.2991. Interestingly Writer seems to perform better on Bing and ChatGPT 4 (where Bing chat is based on GPT 4), as perhaps the detector was trained only on GPT 4 variants. Comparing open and closed book performance, there was no statistically significant difference for Writer or Sapling between the two exam formats with a *p*-value = 0.3557 and a *p*-value = 0.1455 respectively.

[2] At time of testing the univeristy used Urkund which was powered by Turnitin. All references to Turnitin are where the tool Urkund was used.
[3] https://support.gptzero.me/hc/en-us/articles/15130070230551-\\How-do-I-interpret-burstiness-or-perplexity

5.2.2 Tampered output

9: AI detector average AI identification (in percentages) on each LLM raw output for both open and closed book exams across all questions.outputs, it performed significantly worse on the tampered outputs and was very statistically significantly different in performance comparing raw and tampered LLM outputs, reporting a *p*-value < 0.0001.

6. Threats to Validity

While all efforts were made to ensure the results presented were as indicative as possible for both research questions, we acknowledge several possible threats to validity. These include:

- For Research Question 1, we employed two graders, and while their expertise was in this area (applied machine learning), neither were the original grader. Even with detailed marking schemes (available at http://tiny.cc/RED24), there could be some discrepancies in the grades.
- We did not include GPTZero due to its reporting differences with Writer, Ukrund/Turnitin and Sapling. GPTZero used burstiness and perplexity to identify AI content and was not easily comparable to the other percentage metrics. Future work would involve unpacking these scores, and the scores themselves can be found in http://tiny.cc/RED24.
- Students generated the detector outcomes, and due to time constraints, not all of the detector scores are available at present. Future work will include these.
- Students tampered with the raw LLM outputs in different ways, some manually and some with online tools, where this was felt to be a more authentic approach, however, some differences did emerge in the performance of the detectors that could be attributed to the varying student approaches.

Table 9

AI detector average AI identification (in percentages) on each LLM raw output for both open and closed book exams across all questions.

Year	LLM	Writer	TurnitIn	Sapling
2019	Bing	74.54%	9.00%	84.53%
	ChatGPT 3.5	59.61%	5.00%	97.62%
	ChatGPT 4	92.00%	1.00%	NA
2020	Bing	51.92%	1.00%	82.57%
	ChatGPT 3.5	65.21%	4.00%	77.09%
	ChatGPT 4	76.00%	2.00%	NA
2020 (May)	Bing	64.06%	6.00%	89.90%
	ChatGPT 3.5	66.26%	1.00%	72.56%
	ChatGPT 4	86.00%	3.00%	NA

2021	Bing	75.00%	18.00%	NA
	ChatGPT 3.5	44.33%	4.00%	94.60%
	ChatGPT 4	85.00%	4.00%	79.95%
2022	Bing	70.84%	32.20%	NA
	ChatGPT 3.5	42.00%	9.00%	98.43%
	ChatGPT 4	85.00%	6.00%	83.67%
2023	Bing	83.20%	0.05%	NA
	ChatGPT 3.5	25.63%	3.00%	97.20%
	ChatGPT 4	94.00%	0.00%	83.38%
Averages		**68.92%**	**6.01%**	**86.79%**

Table 10

AI detector average AI identification (in percentages) on each tampered output for both open- and closed-book exams across all questions.

Year	**LLM**	**Writer**	**Sapling**
2019	Bing	94.50%	8.34%
	ChatGPT 3.5	59.61%	48.01%
	ChatGPT 4	NA	NA
2020	Bing	90.71%	29.26%
	ChatGPT 3.5	65.21%	43.06%
	ChatGPT 4	NA	NA
2020 (May)	Bing	98.06%	21.10%
	ChatGPT 3.5	66.26%	32.34%
	ChatGPT 4	NA	NA
2021	Bing	87.00%	51.16%
	ChatGPT 3.5	16.88%	50.47%
	ChatGPT 4	NA	0.50%
2022	Bing	81.83%	50.50%
	ChatGPT 3.5	12.04%	54.54%
	ChatGPT 4	NA	0.38%
2023	Bing	92.19%	39.04%
	ChatGPT 3.5	4.04%	66.94%
	ChatGPT 4	NA	0.15%
Averages		**64.03%**	**33.05%**

7. Discussion & Conclusions

The Results section presented above explored a combination of Open Book and Closed Book exams to review the effectiveness of the three LLMs under investigation; this measure of effectiveness was both in terms of their independent performance and also in terms of their comparative performance. A total of six (6) exam papers were used (3 Open Book and 3 Closed book exams), and solutions were generated for each of these using the three LLMs, producing a total of 18 exam scripts.

The results presented above explored three key questions: (1) how does the performance of the three LLMs compare with the students' performance, (2) does the performance of

the LLMs differ between the open and closed book exams, and (3) what types of questions does the LLMs perform best at.

In terms of the first key question, the results showed that there was a statistically significant difference between the performance of the students and the LLMs in the exams (p-value = 0.0001), with the LLMs significantly outperforming the students. This analysis was performed over six years of exam papers, both open and closed-book exams. This results agrees well with existing research mentioned previously (Finnie-Ansley et al., 2022; Finnie-Ansley et al., 2023; Kazemitabaar et al., 2023) that indicates that LLMs are capable of answering exam papers to a very high level of quality.

The second key question was to explore if the LLMs perform better in either the open or closed-book exams. Two statistical approaches were undertaken to explore this question, with a t-test showing no statistically significant difference (p-value = 0.1511), however the effect size (even outside of an educational context) was large (d = 1.31) indicating that there is a difference, and that the LLMs do not perform as well on open-book exams. This presents a striking contrast to literature (as mentioned in Section 2.2) that indicates that for students, open-book exams can deepen their comprehension of the topics in both cooperative learning and standard classroom settings (Eilertsen & Valdermo, 2000).

The third key question looked at the question types that the LLMs perform well on and where LLMs struggle. The types of exam questions which required definitions were the ones that the LLMs did best with, whereas those that included images, or those that required a calculation to be done were the ones that the LLMs struggled most with. To improve the results for questions that require calculations, recent research (Schrier, 2024; Sonkar, et al., 2024) suggests that it is possible to restate the questions (or provide additional information) to significantly improve the outputs of those type of exam questions.

An additional question explored was to determine if the ability of software tools to detect the use of LLMs could be impacted significantly when the outputs of the LLMs were altered in various ways by students. The results indicated that one of the alteration tools, Sapling, produced a statistically significantly difference in performance comparing raw and altered output. This agrees well with existing research (Bernabei, et al., 2023; Nicks, et al., 2023).

This analysis indicates that LLMs provide a significant challenge to examiners, it is clear that the detection of their use depends on the type of exam being given, and the type of exam questions being asked. It also suggests that if students are allowed to alter the output of the LLMs, it significantly increases the challenge of detecting their use.

Acknowledgements

Funded by the European Union
NextGenerationEU

This research was partly funded by the N-TUTORR programme funded by the funded by the European Union and NextGenerationEU fund, where the students who developed the LLM components were funded to conduct this research.

Article Presentation: January 31, 2024
Approval Date: May 4, 2024
Published Date: May 30, 2024

Quille, K., et al. (2024). Machine vs Machine: Large Language Models (LLMs) in Applied Machine Learning. High-Stakes Open-Book Exams. *RED. Revista de Educación a Distancia, 24*(78). http://dx.doi.org/10.6018/red.603001

Funding

This work has not received any specific grants from funding agencies in the public, commercial, or non-profit sectors.

Authors' statement on the use of LLMs

This article has not used texts from (or generated) from an LLM (ChatGPT or others) for its writing.

References

Becker, B. A., Denny, P., Finnie-Ansley, J., Luxton-Reilly, A., Prather, J., & Santos, E. A. (2023). *Programming is hard - or at least it used to be: Educational opportunities and challenges of AI code generation.* Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1, 500–506. https://doi.org/10.1145/3545945.35697599

Becker, B. A., Denny, P., Pettit, R., Bouchard, D., Bouvier, D. J., Harrington, B., Kamil, A., Karkare, A., McDonald, C., Osera, P.-M., Pearce, J. L., & Prather, J. (2019). *Compiler error messages considered unhelpful: The landscape of text-based programming error message research.* Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education, 177–210. https://doi.org/10.1145/3344429.33725088

Bernabei, M., Colabianchi, S., Falegnami, A., & Costantino, F. (2023). *Students' use of large language models in engineering education: A case study on technology acceptance, perceptions, efficacy, and detection chances.* Computers and Education: Artificial Intelligence, *5*, 100172.

Biderman, S., & Raff, E. (2022). *Fooling moss detection with pretrained language models.* Proceedings of the 31st ACM International Conference on Information & Knowledge Management, 2933–2943. https://doi.org/10.1145/3511808.35570799

Deloatch, R., Bailey, B. P., & Kirlik, A. (2016). *Measuring effects of modality on perceived test anxiety for computer programming exams.* SIGCSE '16 Proceedings of the 47th ACM Technical Symposium on Computing Science Education, 291–296. https://doi.org/10.1145/2839509.28446044

Denny, P., Prather, J., Becker, B. A., Finnie-Ansley, J., Hellas, A., Leinonen, J., Luxton-Reilly, A., Reeves, B. N., Santos, E. A., & Sarsa, S. (2024). *Computing education in the era of generative AI*. Communications of the ACM, 67(2), 56–67. https://doi.org/10.1145/36247200

de Raadt, M. (2012). *Student created cheat-sheets in examinations: Impact on student outcomes.* Proceedings of the Fourteenth Australasian Computing Education Conference, 71–76.

http://dl.acm.org/citation.cfm?id=2483716.24837255

Dooley, B., O'Connor Cliodhna, C., Fitzgerald, A., & O'Reilly, A. (2019). *My world survey 2: The national study of youth mental health in Ireland.*

Eilertsen, T. V., & Valdermo, O. (2000). *Open-book assessment: A contribution to improved learning?* Studies in Educational Evaluation, 26(2), 91–103. https://doi.org/10.1016/S0191-491X(00)00010-9

Finnie-Ansley, J., Denny, P., Becker, B. A., Luxton-Reilly, A., & Prather, J. (2022). *The robots are coming: Exploring the implications of OpenAI Codex on introductory programming.* Proceedings of the 24th Australasian Computing Education Conference, 10–19. https://doi.org/10.1145/3511861.35118633

Finnie-Ansley, J., Denny, P., Luxton-Reilly, A., Santos, E. A., Prather, J., & Becker, B. A. (2023). *My AI wants to know if this will be on the exam: Testing OpenAI's Codex on CS2 programming exercises.* Proceedings of the 25th Australasian Computing Education Conference, 97–104. https://doi.org/10.1145/3576123.35761344

Harrington, K., Flint, A., Healey, M., et al. (2014). *Engagement through partnership: Students as partners in learning and teaching in higher education.* Higher Education Academy

Karvelas, I., Li, A., & Becker, B. A. (2020). *The effects of compilation mechanisms and error message presentation on novice programmer behavior.* Proceedings of the 51st ACM Technical Symposium on Computer Science Education, 759–765. https://doi.org/10.1145/3328778.33668822

Kazemitabaar, M., Hou, X., Henley, A., Ericson, B. J., Weintrop, D., & Grossman, T. (2023). *How novices use LLM-based code generators to solve CS1 coding tasks in a self-paced learning environment.* Proceedings of the 23rd Koli Calling Conference on Computing Education Research.

Leinonen, J., Denny, P., MacNeil, S., Sarsa, S., Bernstein, S., Kim, J., Tran, A., & Hellas, A. (2023). *Comparing code explanations created by students and large language models.* Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1

Leinonen, J., Hellas, A., Sarsa, S., Reeves, B., Denny, P., Prather, J., & Becker, B. A. (2023). *Using large language models to enhance programming error messages.* Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1, 563–569. https://doi.org/10.1145/3545945.35697700

Li, Y., Choi, D., Chung, J., Kushman, N., Schrittwieser, J., Leblond, R., Eccles, T., Keeling, J., Gimeno, F., Lago, A. D., Hubert, T., Choy, P., de Masson d'Autume, C., Babuschkin, I., Chen, X., Huang, P.-S., Welbl, J., Gowal, S., Cherepanov, A., ... Vinyals, O. (2022). *Competition-level code generation with AlphaCode.* Science, *378*(6624), 1092–1097. https://doi.org/10.1126/science.abq11588

Lortie-Forgues, H., & Inglis, M. (2019). *Rigorous large-scale educational RCTs are often uninformative: Should we be concerned?* Educational Researcher, *48*(3), 158–166.

MacNeil, S., Tran, A., Hellas, A., Kim, J., Sarsa, S., Denny, P., Bernstein, S., & Leinonen, J. (2023). *Experiences from using code explanations generated by large language models in a web software development e-book.* Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1, 931–937. https://doi.org/10.1145/3545945.35697855

Nicks, C., Mitchell, E., Rafailov, R., Sharma, A., Manning, C. D., Finn, C., & Ermon, S. (2023, October). *Language model detectors are easily optimized against.* In The Twelfth International Conference on Learning Representations.

Nolan, K., & Bergin, S. (2016). *The role of anxiety when learning to program: A systematic review of the literature.* Proceedings of the 16th Koli Calling International Conference on Computing Education Research, 61–70. https://doi.org/10.1145/2999541.29995577

Nolan, K., Bergin, S., & Mooney, A. (2019). *An insight into the relationship between confidence, self-efficacy, anxiety and physiological responses in a CS1 exam-like scenario.* Proceedings of the 1st UK & Ireland Computing Education Research Conference, 8(1-8), 1–7. https://doi.org/10.1145/3351287.33512966

Nolan, K., Mooney, A., & Bergin, S. (2015). *Facilitating student learning in computer science: Large class sizes and interventions.* International Conference on Engaging Pedagogy.

Nolan, K., Mooney, A., & Bergin, S. (2019a). *An investigation of gender differences in computer science using physiological, psychological and behavioural metrics.* Proceedings of the Twenty-First Australasian Computing Education Conference, 47–55. https://doi.org/10.1145/3286960.32869666

Nolan, K., Mooney, A., & Bergin, S. (2019b). *A picture of mental health in first year computer science.* Proceedings of the 10th International Conference on Computer Science Education: Innovation and Technology.

Prather, J., Denny, P., Leinonen, J., Becker, B. A., Albluwi, I., Caspersen, M. E., Craig, M., Keuning, H., Kiesler, N., Kohn, T., Luxton-Reilly, A., MacNeil, S., Petersen, A., Pettit, R., Reeves, B. N., & Savelka, J. (2023). *Transformed by transformers: Navigating the AI coding revolution for computing education: An ITiCSE Working Group conducted by humans.* Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 2, 561–562. https://doi.org/10.1145/3587103.35942066

Prather, J., Denny, P., Leinonen, J., Becker, B. A., Albluwi, I., Craig, M., Keuning, H., Kiesler, N., Kohn, T., Luxton-Reilly, A., MacNeil, S., Petersen, A., Pettit, R., Reeves, B. N., & Savelka, J. (2023). *The robots are here: Navigating the generative AI revolution in computing education.* Proceedings of the 2023 Working Group Reports on Innovation and Technology in Computer Science Education, 108–159.

Prather, J., Reeves, B. N., Denny, P., Becker, B. A., Leinonen, J., Luxton-Reilly, A., Powell, G., Finnie-Ansley, J., & Santos, E. A. (2023). *"It's weird that it knows what I want": Usability and interactions with copilot for novice programmers.* ACM Transactions on Computer-Human Interaction, 31(1). https://doi.org/10.1145/3617367

Quille, K., & Bergin, S. (2015). *Programming: Factors that influence success revisited and expanded.* International Conference on Engaging Pedagogy (ICEP), 3rd and 4th December, College of Computing Technology, Dublin, Ireland.

Quille, K. (2019). *Predicting and improving performance on introductory programming courses (CS1) (Doctoral dissertation).* National University of Ireland Maynooth.

Quille, K., Bergin, S., & Quille, K. (2019). *CS1: how will they do? How can we help? A decade of research and practice research and practice.* Computer Science Education, 29(2-3), 254–282. https://doi.org/10.1080/08993408.2019.16126799

Quille, K., Nolan, K., Becker, B. A., & McHugh, S. (2021). *Developing an open-book online exam for final year students.* Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1, 338–344. https://doi.org/10.1145/3430665.34563733

Quille, K., Nam Liao, S., Costelloe, E., Nolan, K., Mooney, A., & Shah, K. (2022). *PreSS: Predicting Student Success Early in CS1. A Pilot International Replication and Generalization Study.* Proceedings of the 27th ACM Conference on on Innovation and Technology in

Computer Science Education Vol. 1 (ITiCSE '22). Association for Computing Machinery, New York, NY, USA, 54–60. https://doi.org/10.1145/3502718.3524755

Quille, K., Nolan, K., McHugh, S., & Becker, B. A. (2020). Associated exam papers and module descriptors. Available at: http://tiny.cc/ITiCSE21OpenBook

Ribeiro, F., de Macedo, J. N. C., Tsushima, K., Abreu, R., & Saraiva, J. (2023). *GPT-3-Powered type error debugging: Investigating the use of large language models for code repair.* Proceedings of the 16th ACM SIGPLAN International Conference on Software Language Engineering (pp. 111–124). https://doi.org/10.1145/3623476.3623522

Santos, E. A., Prasad, P., & Becker, B. A. (2023). *Always provide context: The effects of code context on programming error message enhancement.* Proceedings of the ACM Conference on Global Computing Education Vol 1 (pp. 147–153).

Savelka, J., Agarwal, A., An, M., Bogart, C., & Sakr, M. (2023). *Thrilled by your progress! Large language models (GPT-4) no longer struggle to pass assessments in higher education programming courses.* Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 1 (pp. 78–92).

Schrier, J. (2024). *Comment on "Comparing the Performance of College Chemistry Students with ChatGPT for Calculations Involving Acids and Bases".* Journal of Chemical Education.

Shields, C. (2023). *ChatGPT for teachers and students*. Ingram Content Goup UK

Sonkar, S., Chen, X., Le, M., Liu, N., Basu Mallick, D., & Baraniuk, R. (2024). *Code soliloquies for accurate calculations in large language models.* In Proceedings of the 14th Learning Analytics and Knowledge Conference (pp. 828-835).

Trochim, W. M. (2006). *Types of reliability. research methods knowledge base.* Web Center for Social Research Methods. Retrieved from: http://www.socialresearchmethods.net/kb/reltypes.php

Wermelinger, M. (2023). *Using GitHub Copilot to solve simple programming problems.* Proceedings of the 54th ACM Technical Symposium on Computer Science Education (pp. 172–178). https://doi.org/10.1145/3545945.3569830