Extending Fraud Detection in Students Exams Using AI

Georgi Cholakov * and Asya Stoyanova-Doycheva

Posted Date: 22 February 2024

doi: 10.20944/preprints202402.1204.v1

Keywords: e-learning; software agents; fraud detection; artificial intelligence; ChatGPT

 doi:10.20944/preprints202402.1204.v1

Article

Extending Fraud Detection in Students Exams Using AI

Georgi Cholakov * and Asya Stoyanova-Doycheva

Department of Computer Systems, Plovdiv University "Paisii Hilendarski", Bulgaria;
gcholakov@uni-plovdiv.bg (G.C.); astoyanova@uni-plovdiv.bg (A.S.-D.)
* Correspondence: gcholakov@uni-plovdiv.bg; Tel.: +359 887987995.

Abstract: The Distributed eLearning Center (DeLC) is a portal, providing extensive support in the day-to-day work when it comes to e-learning content – it helps students and teachers organize their learning materials, fill the gaps in knowledge (for students) and educational approaches (for teachers), organize and conduct exams, and overall, with providing proactive and personalized e-learning environment. The scope of DeLC as a project involves many extensions, covering the aspects of learning, teaching, exams, and collecting statistical information. Such extension is an agent-oriented environment, which enriches the functionalities with intelligent components, that are reactive and proactive, referred to as agents or assistants. This paper is focused on presenting the latest step in the evolution of FraudDetector software agent, which started with base functionality for fraud detection, and now it aims at usage of AI to accomplish its tasks, taking advantage not only from its knowledgebase, but from a much larger one, used by ChatGPT – through integration with it, which is the main contribution of this research, although the real results from production environment are still pending. In this process, agent's architecture should stay open for collaboration with another external AI provider if necessary, trying to decouple the components, responsible for integration. As of now, the experiments show that involving ChatGPT in FraudDetector's functionality enriches it and the agent's precision could be improved this way.

Keywords: e-learning; software agents; fraud detection; artificial intelligence; ChatGPT

1. Introduction

Since the e-learning became a standard for distance learning, almost every school and university try to enrich its education processes with some educational platform that provides many and different kinds of help in learning process. Among most popular choices are Google Classroom [1], Moodle [2], to name a few. Making the right choice rely on a large range of arguments – price, functionalities, support efforts, friendliness of interface, and much more. But when it comes to having partial or overall control over developed components, investing in a custom solution starts to sound much more attractive. It would give the ability to implement only needed functionality, and in the way it would be used in the institution, not making the institution change its processes to fit the business logic of an out-of-the-box system. Furthermore, when the choice was made, open-source systems had not enough functionality or were lacking features that our faculty needed. These arguments prevailed in making the decision in many institutions, among which was Faculty of Mathematics and Informatics in Plovdiv University "Paisii Hilendarski", Bulgaria.

The concept emerged many years ago, and the developed system has been serving our needs for over a decade. Initially established as the Distributed e-Learning Center (DeLC) [3], it began as a research project dedicated to creation of new context-oriented and adaptive architecture. Its objectives encompassed catering to our requirements for distance learning, exams, and various educational and organizational activities. Additionally, a significant goal was to engage in the development and experimentation of diverse prototypes within the e-learning domain. Through a series of iterations, a hybrid service and an agent-oriented environment was fashioned to provide educational materials and electronic services in the field. Opting for an in-house customized solution carried the added advantage of making the codebase accessible to researchers within the institution.

 doi:10.20944/preprints202402.1204.v1

This accessibility facilitated development, reengineering, and enhancement of most features, allowing internal exploration of the system's workings, stored data, and the execution of analytical processes to extract valuable information and knowledge. Over time, the user base expanded beyond our institution to include other universities, transforming the system into a comprehensive platform that merges functionalities and data from multiple satellite systems, thereby broadening its capabilities. This brought additional problems to take care of – how to organize the conditions for confidentiality, integrity, accessibility, and security of information, as it's crucial to protect sensitive data and ensure a smooth learning experience. As a result, there is currently ongoing process for implementing some key principles (shortly mentioned here as this is not the focus of this article):

- Risk management – to identify potential risks and vulnerabilities in the e-learning system; conduct regular risk assessments to stay ahead of emerging threats;
- **Access controls – revision of currently implemented authentication mechanisms and role-based access;**
- **Regular software updates –** apply security updates on a regular basis to minimize vulnerabilities;
- **Data backups – r**egularly backup critical data to prevent data loss in case of system failures, cyber-attacks, or other emergencies;
- **Incident response plan – to d**evelop a comprehensive incident response plan to address security incidents promptly;
- **Monitoring and auditing – to i**mplement monitoring tools to track user activities, system logs, and potential security incidents;
- **User training and awareness – e**ducate users about security best practices and the importance of confidentiality, integrity, and accessibility. Promote strong password policies and ensure users understand the risks associated with sharing login credentials.

The primary objective of this system was to enhance the quality of the educational process by providing interactive and proactive personalized services for students, fostering creative thinking. This response was driven by the escalating standards in university education, technological advancements, and heightened expectations of students regarding the quality of their education. The system aimed to engage students in a personalized, creative, and flexible approach to self-education, promoting their activity and collaboration. Moreover, the system was designed to be open for extensions and experiments with prototypes, such as service and agent-oriented architectures, aligning it with the interactive, reactive, and proactive educational processes seen in other systems [4–6].

Throughout its lifecycle, the architecture underwent expansion with the inclusion of various subsystems, including IntelliDeLC, which ensures the provision of a personalized e-learning environment with reactive and proactive behavior [7]. Proactivity, enhancing usability and friendliness, is achieved through the reinforcement of the service-oriented architecture with intelligent components – in essence, software agents. This agent-oriented extension creates an environment housing these software agents, constantly developed and improved. Their functionalities, behavior, and the latest results are discussed in [8]. Recently, in the era of pervasive artificial intelligence (AI), an exploration was undertaken to determine how these agents could benefit from the integration of AI.

The main goal of this research is to improve the functionality of one of the components in IntelliDeLC – software agent called FraudDetector, which purpose is to track attempts to cheat during exams, made by the students. Now it's knowledge base (dictionary) could leverage from the intelligence, provided by modern AI systems out-of-the-box, like ChatGPT, using so called large language models.

 doi:10.20944/preprints202402.1204.v1

2. Related works

Current state-of-the-art reveals that using AI in education and particularly in e-learning recently becomes a trend, a modern approach, a must-have solution if the adopting institution wants to keep extending and improving its e-learning systems and quality of education. It even looks like an approach that everyone is trying to benefit from, and the fields to apply it to vary in many aspects. Without precise statistic we would roughly assume that AI is mostly used for helping learners in their learning path, but not only. Here are some areas in e-learning where AI is applied to:

- Generally, for improving learning process, for example, making it adaptive [9];
- Using chatbots to accelerate learning process [10] and later measure the effectiveness of each feature [11];
- Improving accessibility of e-learning and academic connectivity [12];
- Personalized learning pathways [13] and adaptive assessment [14];
- For conversational agents for classroom use [15];
- For creating virtual assistants/teachers, even trying to replace teachers when there is deficit of such [16];
- And more, like automating learning processes by building teaching materials, curriculum, training, evaluating student performance [17], and so on.

On the other hand, the problem with academic dishonesty is a matter that every university should consider. Usage of different approaches and systems is observed:

- For prevention of cheating, plagiarism and collusion [31];
- For online detection for learning time in distance learning systems [32];
- For detecting cheating in electronic exams [33].

Of course, some of the features enlisted above, that fit perfectly in one system, wouldn't fit well in another. This is true also for DeLC portal – some of the fields mentioned above are also under consideration, but the current study focuses on another possibly useful way of AI application – fraud detection in students' exams. In DeLC we have such an internal implementation that fits our needs, developed as a software agent.

3. Materials and Methods

The goal of the experiment is to enrich the existing functionality for fraud detection – it's the purpose of FraudDetector agent on Figure 1. This software component has its own data dictionary, formed by lexemes from each test question, keywords, answers and more sources, described in [7,8] and further in the article. Since this data is structured, it's stored in relational database (currently MySQL [29]). Using external AI, we could leverage from its huge knowledge base and functionalities, adding additional skills to our agent and measuring its performance after that. Expected results are to increase the number of detected frauds during exams – if students use general phrases, generated by AI like ChatGPT, the FraudDetector would detect them mark the answer as suspicious.

3.1. Current Architecture

The part of the architecture regarding agent-oriented extension called IntelliDeLC is presented on Figure 1. It shows simplified picture of interactions between software agents and the components from DeLC portal, and its purpose is to illustrate what the existing components are (subject to this article) and how they interact with each other. The agents (also called assistants) „live" and operate in the back-end agent-oriented server – build with JADE framework [18]. Their environment is referred to as Agent Village (AV).

 doi:10.20944/preprints202402.1204.v1

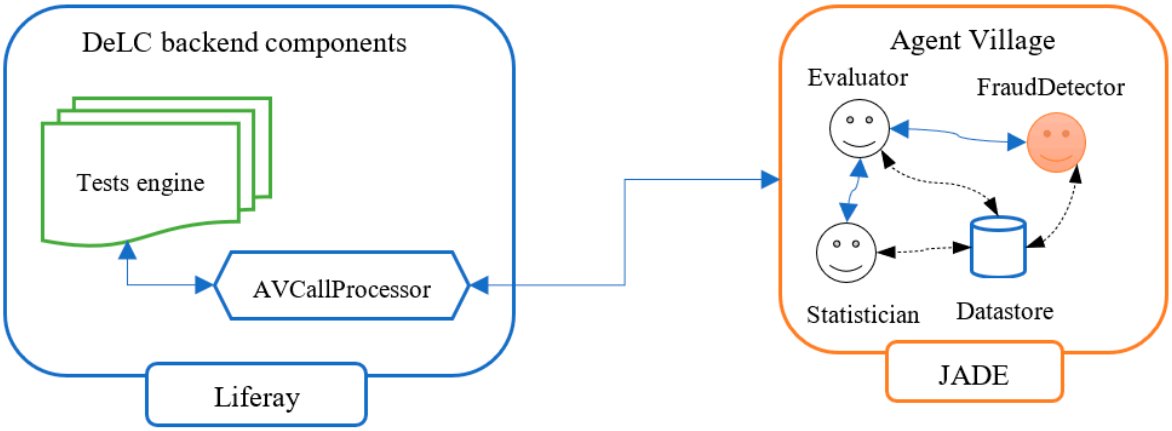

Figure 1. IntelliDeLC extension of DeLC.

DeLC is running on Liferay portal [19] and its architecture consists of set of components and satellite extensions [20], developed by different researchers from our team and omitted on the figure for readability and because they are not related to the current topic, in contrast with tests engine, which is the component that contains functionality and is responsible for students' exams. Simply put, when there are tests questions with free-text answers, depicted workflow comes into play.

When DeLC needs to take advantage of the assistants' functionality, it calls their services using AVCallProcessor, which plays the role of adapter, making the technical details of the call transparent, providing single point of implementation of the real calls. It functions as an intermediary in the communication between DeLC and Agent Village. Positioned within the portal as a system service, it serves as a mediator for all requests from the portal to the agents. The communication process operates in the following manner: when DeLC requires a service from Agent Village, it calls a specific method from AVCallProcessor. This action triggers the generation of a SOAP request to Agent Village by AVCallProcessor. Subsequently, this request undergoes transformation into an ACL message [21], a format comprehensible to the agents. On the way back, the corresponding agent formulates the result as an ACL message. Agent Village then transforms this message into a SOAP response, which is transmitted to AVCallProcessor. AVCallProcessor, in turn, parses the response and generates the result in the expected format for the portal.

There are some workflows that require agents to cooperate, for example:

- During answers' estimation, Evaluator communicates with Statistician in order to select the algorithm that best matches the test creator's (teacher) approach and behavior, in order to simulate her/his manual estimation;
- When FraudDetector tries to check a suspicious activity during exams, it also communicates with Statistician to compare the similarities with such previous cases in order to decide to investigate further or not.

These internal communications between agents rely on standard ACL communication messages, which is default mechanism for it in Agent Village (JADE) environment.

When the agents need to access the datastore, they use JDBC (Java Database Connectivity [30]) driver, since it's currently a relational database (MySQL).

3.1.1. Agent Village

This architectural component is actually the physical environment, where the software agents (assistants) are operating. The environment is established and operational on the JADE framework. Entire internal communication among agents relies on ACL messages. Outside Agent Village, the services provided by the agents are encapsulated and accessible as SOAP web services [22]. Additional information on this subject is extensively covered in [7].

 doi:10.20944/preprints202402.1204.v1

3.1.2. Evaluator

The Evaluator Agent (EA) functions as an expert, aiding the teacher in the assessment of electronic tests. While DeLC's test engine includes a system service for the automated assessment of "multiple choice questions", EA contributes by analyzing responses to short free-text questions, providing a rating for each answer and deferring the final decision to the teacher. When external assessment is needed, the test engine sends a request for expert assistance to EA. It then leverages its knowledge base to search for matches generated from keywords and phrases associated with each test question. Typically, these keywords and phrases are supplied by the test maker. The precision of the test maker in specifying keywords directly impacts the quality of results produced by the agent. Essentially, the agent needs to be appropriately "educated" to be effective. The keywords in the knowledge base carry no priorities; they are considered equal in searches. Currently, EA employs two distinct algorithms to calculate points, as previously detailed [7]. In the evaluation process for each answer, EA also considers the points (estimations) previously assigned by the teacher for that specific answer in prior exam runs. This approach allows EA to refine its estimation based on the teacher's style and approach. Post-evaluation, EA stores data about each answer, including the awarded points. Further details on the latest validation of this agent can be found in [8].

The application of this agent is during exams with students from Faculty of mathematics and informatics, Plovdiv University "Paisii Hilendarski", primarily in the subjects "Database management systems", "Software engineering", and "Design patterns".

3.1.3. Statistician

The Statistician is responsible for storing comprehensive information on all processed answers, maintaining a complete history of details from all calculating methods employed by the Evaluator Agent. Currently, two methods, namely pessimistic and optimistic, are utilized as described in [7]. This assistant actively seeks feedback on the final points assigned by the teacher for each answer. Consequently, it accumulates a knowledge base for each teacher, enabling it to determine which method aligns best with the assessment style of the current evaluating teacher. Upon the Evaluator Agent returning its results, the Statistician plays a crucial role in determining which results are deemed suitable for presentation to the teacher. It identifies eligible results, while the remaining outcomes from other methods are presented as alternative options. This approach aims to provide the teacher with a comprehensive view of the assessment, incorporating alternative perspectives from different calculation methods.

3.1.4. FraudDetector

FraudDetector's purpose is to help recognizing attempts for cheating in the answers, given by the students. Among mostly used ways for cheating are:

1. Using portal's integrated chat system to share answers – easy to trace and react to, and it's already covered by existing functionality;
2. Copy/paste results from Internet search engines, e.g., Google, Bing – harder to recognize, most common so far;
3. Copy/paste results from ChatGPT – not so rare recently. This is something that we could work on and is undergoing preliminary analysis.

The first point has been successfully implemented and rigorously tested over the years, with recent results detailed in [8]. This assistant primarily relies on its own knowledge base, which is derived from several sources: the words constituting each question, the keywords specified for each question (provided by the test creator), the words found in students' answers, and messages exchanged between students in the portal's chat system – particularly those flagged by an operator as potentially involving cheating. This process results in the creation of a substantial dictionary that expands with each examination, as both the number of answers and chat messages increase. This necessitates another key functionality for the agent—its ability to be "self-learning". This crucial feature ensures the agent stays abreast of new potential cheating trends. An interesting side effect of

 doi:10.20944/preprints202402.1204.v1

this self-learning capability is that the agent may become somewhat stringent, flagging messages and answers as suspicious even when they are unrelated to the subject. However, this behavior was anticipated and discussed in [8].

The second point is large to cover with custom functionality, but since Google and Bing provide APIs for searching programmatically, there are some ideas in this direction, so far just in plans and analysis/investigation phase.

The third point is very interesting now, as ChatGPT also provides API for accessing its functions. In fact, this is the subject of the current study – can we take advantage of using out-of-the-box AI solution for our needs?

4. Implementation of fraud detection improvement

Recently we observed in our university increasing number of cases when students try to use ChatGPT during exams to help them passing the tests. As this way of cheating is becoming more and more popular, detecting fraud from this direction becomes important goal to stay ahead of emerging new ways of cheating. The idea is to extend FraudDetector's dictionary by adding functionality for collaborating with ChatGPT – the agent, in addition to its current functionality, will ask the AI the same question, answered by each student, and compare result with the answer for similarities – if the percentage is higher than a parameterized value, the agent marks that answer as suspicious. Of course, there are many aspects here for improvement, e.g., selecting correct percentage threshold requires education of the agent, based on many runs over the same data until final results are reliable enough and could be used in real environment, eventually; using multiple answers from AI to compare student's one with, and so on. But as a starting point, for first iteration the explained functionality is considered as sufficient. Such an extension of this agent would bring indirect access to an enormous knowledge base, because the scale of data amount, used by ChatGPT, is in fact incomparable to current one, used by the agent, increasing the expected reliability of fraud detection in times. The topic of the proper threshold is not subject of interest in this article and selecting it is still in development and experiments continue.

Integrating with ChatGPT was not the only solution, when it comes to selecting third party provider – because developing own solution would be more expensive and would take efforts in direction, which is not among our primary goals now. Thinking in this direction, we did some research on how most popular AI providers could be used for our educational purposes, and particularly in fraud detection during exams. Among intriguing ones were ChatGPT [24], Perplexity [24] and Google Bard [25] – a good comparison could be found in [26,27]. Different arguments were taken into account when the choice was made, but in summary:

- Google Bard is still in development. In the near future, it could be a very good alterative. It produces also additional information, which currently doesn't bring additional value for our experiment. Nevertheless, we will keep an eye on it, because it tends to be very good in answering questions with up-to-date information, with real-time access to Google. In contrast, ChatGPT (with GPT 3.x) sometimes returns a bit outdated information. Also, Bard so far is free, while new version of GPT 4 is paid. There is another feature here, that attracts – Bard can produce multiple answers (variations) to a single question. It makes it very interesting for experiments in our area, because we could investigate multiple answers within a single roundtrip – that would perform better and also would return more results to search within;
- Perplexity would be actually a very good choice, its advanced answer engine considers the entire conversation history and uses predictive text algorithms to generate concise responses from multiple sources, which is helpful for generating answers bound to a specific context. It provides real-time information from multiple sources, just like Bard and in contrast with ChatGPT. This should produce more topic-oriented results, and it would be our second choice for the experiment;
- ChatGPT tries to mimic human conversation and its training methodology involves learning from human feedback. Communication with it provides a possibility to tune up the search for answer – more deterministic or more creative, which was interesting for our experiment, that's why the accent fell on ChatGPT, because of its simplicity of use, good overall support and recent

 doi:10.20944/preprints202402.1204.v1

progress. As a downside we can outline that new version is available with paid subscription, as opposed to Bard and Perplexity, but since the price is affordable and we are only interested in correctness of the results, this fact doesn't play substantial role.

To summarize this comparison and as final talks to choice's arguments, any of the mentioned AI machines above would do the job for our case – just because we use only a small part of their possibilities.

Integration architecture is presented on Figure 2. The communication with ChatGPT is achieved by REST calls, since the existing ChatGPT API exposes its functions in this way. But implementing integration with external systems in existing and working component (FraudDetector) directly could be error prone, that's why here we made it by developing another adapter, to incorporate the entire communication in it. Thus, if we need to change some technical details in REST calls, or even replace external system with another one, no changes will affect the FraudDetector assistant, only the adapter will accommodate the implementation change.

Furthermore, such decoupling would provide a possibility to use more than one external system as source for answers comparison, and this could be transparent to the agent, like shown on Figure 2.

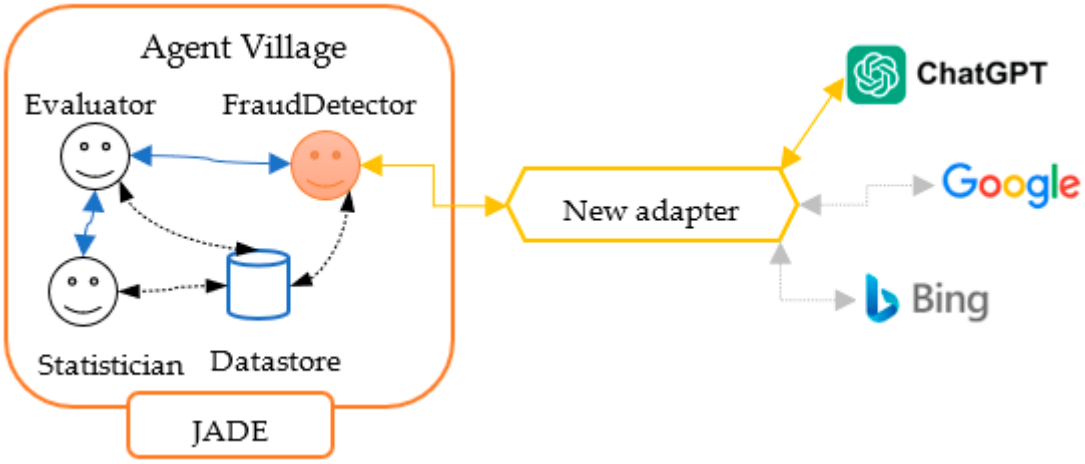

Figure 2. Improved architecture for accessing external systems e.g., ChatGPT, Google, Bing. .

As of now, introducing integration with multiple systems is scheduled for future analysis and development, current implementation is focused on integration with ChatGPT only – there are many parameters to tune up for getting reliable results, that's why adding integrations will be accomplished iteratively, keeping control over incrementing complexity.

As Figure 3 shows, request and response look quite simple and even human friendly. Indeed, produced result is accurate, but let's have a closer look at request attribute *temperature*. It ranges from 0 to 2, as the higher the temperature, the higher the randomness of the result, and vice-versa. If we look for more focused and deterministic responses, its value should not exceed 0.7, as our tests revealed, otherwise the response could be nonsensical. According to the documentation, higher values produce more creative responses, but it doesn't seem to work well for sciences. On the other hand, going too low tends to generate formula-like responses, which are unusual in answers for tests questions with short free-text answers. Of course, choosing the right value depends heavily on context, and setting it up for our needs relies on many experiments and statistical analysis, not finished yet.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 February 2024 doi:10.20944/preprints202402.1204.v1

REQUEST

```
{
  "model": "gpt-3.5-turbo",
  "temperature": 0.7,
  "messages": [
    {
      "role": "user",
      "content": "Which functional dependency third normal form deals with"
    }
  ]
}
```

RESPONSE


```
{
  "id": "chatcmpl-8Huvm2wXhBoIjd2G4161g75tepKqm",
  "object": "chat.completion",
  "created": 1699281110,
  "model": "gpt-3.5-turbo-0613",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "The third normal form (3NF) deals with functional dependencies that are
          transitive in nature. \n\nIn 3NF, a table is considered to be in third normal form if
          it meets the following conditions:\n\n1. It is in second normal form (2NF).\n2. There
          are no transitive dependencies between non-key attributes.\n\nA transitive dependency
```


Figure 3. Raw result from ChatGPT to a particular question.

5. Results

5.1. Tests for ChatGPT's temperature parameter

The tests for temperature parameter were conducted with simulation, including only the new adapter and ChatGPT. The adapter is implemented as separate microservice with Spring Framework 6 [28]. In the simulation we used 89 questions, given to students in the tests in exams for one particular subject – Database Management Systems, which subject is intended for pilot version for real environment. The appropriate accuracy in results, returned by ChatGPT, is summarized on Figure 4. It's important here to declare, that the correctness below is not a general or overall estimation of ChatGPT functionality, it actually works surprisingly good for our purposes – and our opinion is subjective, because we use this AI in very specific domain and its success is estimated by humans, each one with his own opinion on the topics.

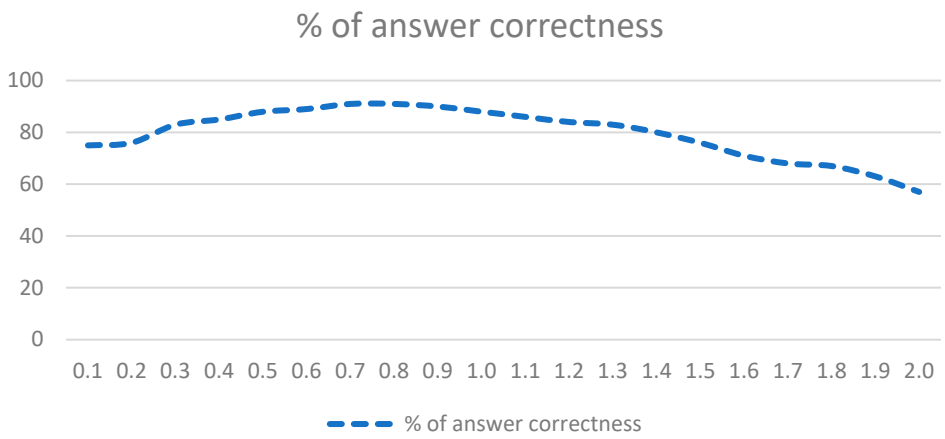

Figure 4. Results accuracy of the answers by ChatGPT.

The results above were produced by the new adapter, iterating over those 89 questions, each one sent to ChatGPT with all possible values for the temperature with step 0.1. As it seems from the graphic, somewhere in the middle we received the most accurate results for our purposes, that's why currently this parameter's value is set to 0.7. At some point, even after getting first results in the real environment, this value could be amended, to compare different behaviors and results in empirical way.

 doi:10.20944/preprints202402.1204.v1

5.2. Tests for fraud detection over existing data

This simulation was conducted over existing data from previous periods (kept in portal's database), consisting of selected top suspicious 5000 messages from the integrated chat messaging system in DeLC portal. This system is often used by the students for helping each other during exams, and it was the first target of FraudDetector for investigation.

Having the results from FraudDetector before the new functionality to take place, we compared the results now with the new functionality, running a simulation over the same data. The results are summarized in Table 1.

Table 1. Comparison between the results from FraudDetector prior and after the new changes over the same dataset.

Suspected messages	Reasonably suspected	Without reason
Before: 235	167 (71%)	68 (29%)
After: 258	179 (69%)	79 (31%)

From the results above it looks like after integration with ChatGPT the correctness is dropping, and the suspiciousness is rising, but it's only percentage – after the integration the agent suspected more messages, and it found 12 messages reasonably suspected more than it did it previously, which is a progress. The percentage of its suspiciousness increased, as well, and that reveals another room for improvement – the agent's functionality should be revised further to find its weaknesses in marking a message as suspicious.

As of now, the results are very few, because this implementation is not put in production environment yet. When it does its first approbation, all outcomes will be monitored and considered for the next development iteration. Production results will give us also better understanding of what should be revised in more details regarding the algorithms of the FraudDetector.

6. Conclusions

Keeping the quality of education high relies on many parameters, going far beyond the teaching skills of the personnel and students' awareness that building their skills needs dedication and self-motivation. As the technologies go further, entire process of university education needs to keep up with recent standards and trends – not only to attract students' attention, but to provide a healthy environment for building professionals from students. It includes also the quality of examination, part of which is to guarantee fair process of assessment – which is subject of the current study. Improvement of software components, that work in background and support in fact the entire educational process, is a major part of building a strong foundation of education nowadays.

Extending fraud detection with artificial intelligence will provide contemporary functionalities to the activities that FraudDetector is responsible for. Having its own knowledge base extended is not an easy task, that why integration with external systems is justified, leaving focus on precision of fraud detection methods instead of building and extending its own knowledge base.

If the integration with ChatGPT turns into a successful one, as we could assume even from the modest improvement in numbers, shown in Table 1, Evaluator agent would be the next candidate for such integration as well, enriching its answers estimation algorithms with another source of truth for building larger knowledge base for its purposes. Furthermore, Evaluator could use another AI provider, for example Google Bard, which is able to produce multiple results. These results could be used to form several evaluations, depending on the results number, providing the teacher with different estimations for the same answer – and when the teacher makes a choice, Evaluator could be educated what answer is preferred.

Author Contributions: Conception, G.C. and A.S.; software implementation, G.C.; evaluation and tests creation, A.S.; solutions investigation, G.C.; state-of-the-art, A.S. and G.C.; formal analysis, A.S.; writing—original draft preparation, G.C.; writing—review and editing, A.S.; funding acquisition, A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This study is financed by the European Union – NextGenerationEU, through the National Recovery and Resilience Plan of the Republic of Bulgaria, project № BGRRP-2.004-0001-C01.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.
Data Availability Statement: Not applicable.
Acknowledgments: we would like to thank the system and network administrators in Faculty of mathematics and informatics, Plovdiv university "Paisii Hilendarski", who supported our hardware and software through all these years of experiments. Also, thanks to people involved in the project European Union – NextGenerationEU, through the National Recovery and Resilience Plan of the Republic of Bulgaria, project № BGRRP-2.004-0001-C01, which made our study possible.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Google Classroom. Available online: https://edu.google.com/workspace-for-education/classroom/ (accessed on 13 November 2023).
2. Moodle: Online Learning With The World's Most Popular LMS. Available online: https://moodle.com/ (accessed on 13 November 2023).
3. DeLC (Distibuted e-Learning Center). Available online: http://delc.fmi.uni-plovdiv.bg/ (accessed on 13 November 2023).
4. Goyal, M.; Krishnamurthi, R.; Goyal, M.; Krishnamurthi, R. Pedagogical Software Agents for Personalized E-Learning Using Soft Computing Techniques. In *Nature-Inspired Algorithms for Big Data Frameworks*; IGI Global, 2019 ISBN 978-1-5225-5852-1.
5. Farzaneh, M.; Raeesi Vanani, I.; Sohrabi, B. Utilization of Intelligent Software Agent Features for Improving E-Learning Efforts. *IJVPLE* **2012**, *3*, 55–68, doi:10.4018/jvple.2012010104.
6. Rani, M.; Vyas, O. An Ontology-Based Adaptive Personalized E-Learning System, Assisted by Software Agents on Cloud Storage. *Knowledge-Based Systems* **2015**, doi:10.1016/j.knosys.2015.10.002.
7. Cholakov, G. Hybrid Architecture for Building Distributed Center for e-Learning. PhD, Plovdiv University "Paisii Hilendarski": Plovdiv, Bulgaria, 2013.
8. Cholakov, G. Approbation of Software Agent Evaluator in a Nonspecific Environment for Extension of Its Purpose. In Proceedings of the 2020 International Conference Automatics and Informatics (ICAI); October 2020; pp. 1–5.
9. Benkhalfallah, F.; Laouar, M. Artificial Intelligence-Based Adaptive E-Learning Environments. In; 2023; pp. 62–66 ISBN 978-3-031-44096-0.
10. Benachour, P.; Emran, M.; Alshaflut, A. Assistive Technology and Secure Communication for AI-Based E-Learning. In; 2023; pp. 1–21 ISBN 978-1-66848-938-3.
11. S, M.; VS, A.; H, A.; R, M. E-Learning Management System with AI Assistance. *International Journal for Research in Applied Science and Engineering Technology* **2023**, *11*, 1233–1238, doi:10.22214/ijraset.2023.56730.
12. Sinha, M.; Fukey, L.; Sinha, A. AI in E-Learning. In; 2021; pp. 126–150 ISBN 978-1-83953-120-0.
13. Tapalova, O.; Zhiyenbayeva, N.; Gura, D. Artificial Intelligence in Education: AIEd for Personalised Learning Pathways. *Electronic Journal of e-Learning* **2022**, *20*, 639–653, doi:10.34190/ejel.20.5.2597.
14. Tanjga, M. E-Learning and the Use of AI: A Review of Current Practices and Future Directions. *Qeios* **2023**, doi:10.32388/AP0208.2.
15. Alfehaid, A.; Hammami, M. Artificial Intelligence in Education: Literature Review on The Role of Conversational Agents in Improving Learning Experience. *International Journal of Membrane Science and Technology* **2023**, *10*, 3121–3129, doi:10.15379/ijmst.v10i3.3045.
16. Muzurura, O.; Mzikamwi, T.; Rebanowako, T.; Mpini, D. APPLICATION OF ARTIFICIAL INTELLIGENCE FOR VIRTUAL TEACHING ASSISTANCE (Case Study: Introduction to Information Technology). **2023**.
17. Udayakumar, A.; Ganesan, M.; Gobhinath, S. A Review on Artificial Intelligence Based E-Learning System. In; 2022; pp. 659–671 ISBN 978-981-19283-9-0.
18. JADE – Java Agent DEvelopment Framework.
19. Liferay Available online: https://www.liferay.com/ (accessed on 14 November 2023).
20. Doychev, E. Environment for Provision of eLearning Services. PhD, Plovdiv University "Paisii Hilendarski": Plovdiv, Bulgaria, 2013.
21. FIPA Agent Communication Language Specifications Available online: http://www.fipa.org/repository/aclspecs.html (accessed on 14 November 2023).

 doi:10.20944/preprints202402.1204.v1

22. W3C SOAP Version 1.2 Part 1: Messaging Framework (Second Edition) Available online: https://www.w3.org/TR/soap12/ (accessed on 14 November 2023).
23. ChatGPT Available online: https://chat.openai.com (accessed on 13 November 2023).
24. Perplexity Available online: https://www.perplexity.ai/ (accessed on 13 November 2023).
25. Bard – Chat Based AI Tool from Google, Powered by PaLM 2 Available online: https://bard.google.com (accessed on 13 November 2023).
26. Java, W. at W. by Battle of the AI's: Chat GPT vs. Perplexity AI vs. Google Bard. *Medium* 2023.
27. Horsey, J. Perplexity vs Bard vs ChatGPT Available online: https://www.geeky-gadgets.com/perplexity-vs-bard-vs-chatgpt/ (accessed on 19 November 2023).
28. Spring Framework Microservices Available online: https://spring.io/microservices (accessed on 19 November 2023).
29. MySQL Available online: https://www.mysql.com/ (accessed on 5 February 2024).
30. Java Database Connectivity. *Wikipedia* 2024.
31. Bylieva, D.; Lobatyuk, V.; Tolpygin, S.; Rubtsova, A. Academic Dishonesty Prevention in E-Learning University System. In Proceedings of the Trends and Innovations in Information Systems and Technologies; Rocha, Á., Adeli, H., Reis, L.P., Costanzo, S., Orovic, I., Moreira, F., Eds.; Springer International Publishing: Cham, 2020; pp. 225–234.
32. Ueno, M. Online Outlier Detection System for Learning Time Data in E-Learning and Its Evaluation. In Proceedings of the Proceedings of the 7th IASTED International Conference on Computers and Advanced Technology in Education; Kauai, Hawaii, USA, August 16 2004; p. 253.
33. Bawarith, R.; Basuhail, D.A.; Fattouh, D.A.; Gamalel-Din, P.D.S. E-Exam Cheating Detection System. *International Journal of Advanced Computer Science and Applications (IJACSA)* **2017**, *8*, doi:10.14569/IJACSA.2017.080425.