Cheating Detection: Identifying Fraud in Digital Exams

Bastian Küppers[1], Julia Opgen-Rhein[2], Thomas Eifert[3], Ulrik Schroeder[4]

[1]IT Center / Learning Technologies Research Group, RWTH Aachen University, Seffenter Weg 23, 52074 Aachen, kueppers@itc.rwth-aachen.de
[2]IT Center, RWTH Aachen University, Seffenter Weg 23, 52074 Aachen, opgen-rhein@itc.rwth-aachen.de
[3]IT Center, RWTH Aachen University, Seffenter Weg 23, 52074 Aachen, eifert@itc.rwth-aachen.de
[4]Learning Technologies Research Group, RWTH Aachen University, Ahornstraße 55, 52074 Aachen, schroeder@informatik.rwth-aachen.de

Keywords
Computer based examinations, e-Assessment, Digital Examinations, Paper-based Examinations

1. SUMMARY

Digital exams are more and more adapted in institutions of higher education, but the problem of preventing cheating in those examinations is not yet solved completely. Electronic exams potentially allow for fraud beyond plagiarism, because the usage of computers during the exam technically enables the students to communicate with each other or to access online resources. Therefore, possibilities to detect impersonation and prohibited communication between the students in-situ and a-posteriori are desirable. To implement these measures, several techniques from the fields of artificial intelligence and statistical analysis can be used. This paper describes the required conditions for both, in-situ and a-posteriori detection, as well as the techniques that can be applied.

2. ABSTRACT

The introduction of digital exams into the examination systems of institutes of higher education equips the students with a powerful tool during the exam, the computer, which they can use to solve the exam's assignments in adequate ways. However, the computer and the possibilities it brings with it can also be used to cheat during an examination by communicating with fellow students or accessing online resources to get help while solving the assignments. This problem gets worse if a BYOD approach is taken, where students are allowed to use their own devices to solve the assignments, because the students' devices are *untrusted devices* per se from the examiner's point of view. There are software solutions, so-called *lockdown applications*, which can be used to lock the computer in a way that only white-listed applications can be launched or only white-listed webpages can be visited. However, these tools cannot reliably guarantee that no cheating takes place on the devices (Søgaard, 2016) (Heintz, 2017). One obvious solution would be backing off from BYOD solutions, but this approach would cancel all benefits of this approach. To be able to give meaningful marks based on the solutions that are handed in by the students, it has to be possible to reliably determine authorship and integrity of those. This can be accomplished by using digital signatures, which are commonly used to provide security measures for online applications, for example banking or shopping. These measures, however, only work reliably if the so-called private key can be kept secret, but this, in turn, is only possible if the students use their own devices. Otherwise the students' private keys had to be copied to foreign computers which would compromise the security of these keys. Following this chain of reasoning, a BYOD is inevitable, but has to be secured reliably to prevent cheating. In addition, even with tightly configured PCs for the exam, traditional cheating methods are still available to the students.

In this paper we propose approaches to in-situ and a-posteriori detection of cheating. The in-situ part does not try to lock the computer, but only monitors the actions that are carried out on the computer. There are generic events that can be related to cheating, for example if software other than the exam client is launched. However, even not so obvious ways of cheating are possible. To detect these ways, it has to be ensured that the students are really doing input themselves and the computer is not

remotely controlled. Therefore, student-related patterns in the log of events have to be identified, for example typing patterns (Peltola et al., 2017).

For the a-posteriori approach, we resort to the fact that working on the exam is a continuous process, which can be logged and interpreted as a time series. When not cheating, it can be assumed that students enter snippets of their work in rather short time intervals. In contrast to that, a cheat can be expected to show a different pattern, e.g., a period of inactivity followed by larger chunks of work in short time. By analyzing the time series of these events and comparing it to the average series of the particular part of the exam, we expect to find deviations of statistical significance and thus strong indications of a fraud attempt. Various techniques for analyzing these timelines can be applied, for example Process Mining (Van der Aalst, 2016) or Wavelet Analysis (Karel, 2018). The paper describes which techniques are applicable and what attributes a time series has to have in order to work for a particular technique. Additionally, the final submissions of the students can be analyzed and compared with previous work from assignments and tutorials. This can be done with written texts as well as with source code and depends on the assumption that everyone has a unique writing style. That programmers have a unique coding style was demonstrated recently by (Caliskan-Islam et al., 2015) while author verification for written texts is better researched (Koppel & Schler, 2004). These styles can be extracted and compared by using machine learning techniques like Support Vector Machines and Artificial Neural Networks. We describe which techniques can be used for fraud detection in the context of a digital exam and the conditions that have to be met for this approach to work.

Both of the described approaches, however, can only indicate a cheating attempt, but not prove it. The decision whether a student has actually cheated or not is a decision that has to be made by the examiner.

3. REFERENCES

Caliskan-Islam, A., Harang, R., Liu, A., Narayanan, A., Voss, C., Yamaguchi, F., Greenstadt, R. (2015). De-anonymizing Programmers via Code Stylometry. *24th USENIX Security Symposium (USENIX Security 15). USENIX Association, Washington, D.C.,* pp. 255–270.

Heintz, A. (2017). Cheating at Digital Exams – Vulnerabilities and Countermeasures. *Master's Thesis (NTNU, Norway)*.

Karel, J., Peeters, R. (2018). Orthogonal Matched Wavelets with Vanishing Moments: A Sparsity Design Approach. *Circuits, Systems, and Signal Processing, 37(8),* pp. 3487-3514.

Koppel, M., Schler, J. (2004). Authorship verification as a one-class classification problem. *Proceedings of the twenty-first international conference on Machine learning (ICML '04)*. ACM. pp. 62-68.

Peltola, P., Kangas, V., Pirttinen, N., Nygren, H., Leinonen, J. (2017) *Identification based on typing patterns between programming and free text*. Koli, Finnland: ACM.

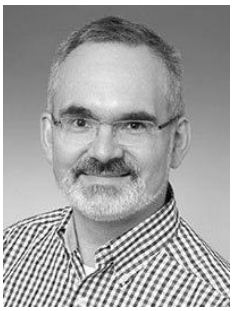
Søgaard, T.M. (2016). Mitigation of Cheating Threats in Digital BYOD exams. *Master's Thesis (NTNU, Norway)*.

Van der Aalst, W.M.P. (2016). *Process Mining: Data Science in Action*. Berlin, Germany: Springer-Verlag.

4. AUTHORS' BIOGRAPHIES

Bastian Küppers, M.Sc. is research associate at the IT Center of RWTH Aachen University. His research focusses on e-Learning and e-Assessment technologies. He received his M.Sc. cum laude in Artificial Intelligence from Maastricht University in 2012. Since 2010 he works at IT Center as a software developer and later as a teacher for parallel programming, robotics and other topics in computer science.

Julia Opgen-Rhein, B.Sc. is a student assistant at the IT Center of RWTH Aachen University. She received her B. Sc. in Scientific Programming from FH Aachen in 2017 and is currently pursuing a M.Sc. in Data Science for Decision Making at Maastricht University.



Dr. rer. nat. Thomas Eifert is Chief Technology Officer of the IT Center of RWTH Aachen University and as such responsible for the strategy for technological development and the corresponding third-party funding of the IT Center. The focus in this function includes concepts for research-oriented storage infrastructures. He is also a lecturer for Calculus in the study program "Applied Mathematics and Computer Science" at the FH Aachen University of Applied Sciences, where he is involved in the processing and digitalization of traditional teaching content.

Prof. Dr-Ing. Ulrik Schroeder received his Diploma degree as well as his PhD in Computer Science from Technische Universität (TU) Darmstadt. Since 2002 he heads the Learning Technologies Research Group in the computer science department at RWTH Aachen University. His research interests include assessment and intelligent feedback with a focus on learning processes, Web 2.0 applications and social software in education, mobile Internet and learning, gender mainstreaming in education, and Computer Science didactics.