# 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.den
[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. ABSTRACT

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. As a result, we show methods with very high detection probability of cheating and thus a credibility of e-Assessments at least at the level of paper based exams.

## 2. INTRODUCTION/PROBLEM STATEMENT

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 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 whitelisted applications can be launched or only whitelisted web pages can be visited. However, these tools cannot reliably guarantee that no cheating takes place on the devices (Søgaard, 2016) (Heintz, 2017). Additionally, there is no *lockdown application* available that supports Windows, macOS, iOS and Linux at the same time, which makes this approach unsuitable for BYOD exams. One obvious solution would be backing off from BYOD solutions, but this approach would cancel all benefits of this approach. We do not repeat the benefits of e-Assessments in general. One main benefit of a BYOD approach is the use of the students' personal and thus well-known devices. The second main benefit is the ease for taking e-exams. Without BYOD, one has to maintain lots of dedicated rooms with appropriate equipment or needs a logistic to hand out large numbers of laptops at the beginning of the exam and re-collect all of them afterwards. With some exams at our university carried out with up to 1700 students simultaneously and 2.5 kg for an average laptop one thus would need to handle more than 4 tons. 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.

# 3. TECHNIQUES

## 3.1. In-Situ

In-situ detection of fraud in an exam means that a violation of the guidelines is detected while an examination is still in progress. In a paper-based examination this means e.g. observing a student using a crib sheet or another forbidden aid or hearing two students talking during the test. With an exam taking place on the computer, cheating becomes in some cases less obvious, because it is not easy for an invigilator to distinguish between regular behaviour and banned activities. In contrast to a piece of paper, an electronic equivalent of a crib sheet can be opened and closed within a few moments. Communication between students can take place silently by using a chat program and solutions can be exchanged even without an internet connection with Bluetooth. Hence, there is a need for countermeasures that are able to detect such forms of cheating.

Since, as described above, *lockdown* solutions are not feasible, there is a need for tools that detect illicit activities on the students devices and report them to the exam invigilator, who is still needed as 'classic' cheating behaviour might still take place. One solution to this problem is to monitor the actions that are carried out on the computer instead of locking it. 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).

## 3.2. A-Posteriori

An a-posteriori analysis of the exam makes sure that cases that were not uncovered in-situ are detected afterwards. This analysis can be based on the log data and intermediate solutions collected during the exam and the submitted final solution or a combination of those.

For one 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 even without a-priori knowledge about the "normal" activity pattern. Various techniques for analyzing these timelines can be applied, for example Process Mining (Van der Aalst, 2016) or Wavelet Analysis (Karel, 2018).

Process Mining is a set of techniques that (Van der Aalst, 2016) come from the field of process management. It can be used to discover processes, check the conformance with a process model or improve an existing process by analysing event logs. A process can be passed through in different ways, e.g. tasks can be solved one after the other or in an individual order, they can can paused and finished later etc. Such an individual pass of the process that occurs in reality is called a case. Each case consists of an ordered chain of events that are described by attributes. For the purpose of cheating detection the process that shall be monitored is the solving of the exam by a student using the examination software. The actions of each student are the events that form a case. To apply Process Mining techniques, an event log has to be created that records relevant events. Regarding the individual execution order this are the beginning and end of the interaction with the examination software, the start and submission time of a task, potential interruptions by work on other tasks and subsequent alterations on solutions. At a more detailed level, general interactions with the software like entering text, compiling a piece of source code or switching between tasks can also be logged. Essential attributes of the events are an event ID to keep track of the sequence of events, the ID of the student/case, a timestamp and a descriptive name of the activity that occurred.

The easiest Process Mining technique for extracting a process from log data is the α-algorithm (van der Aalst et al., 2004). It results in a model of the process as a petri net that can be used for a graphical representation of the process. Although all cases are used to construct the model and thus are represented as valid paths, it is nonetheless possible to detect deviations by searching for rare paths or unusually long or short durations for solving a task. Besides being applicable for the detection of fraud, these analysis also provides valuable insight into the way, students go through their exams.

Wavelet transformations deal with linear time-frequency functions. It is possible to take the process of solving an examination as a time-frequency function. To do that the amount of answers that a student has entered into the has to be interpreted as a frequency. In that interpretation a low amount of answers relates to a low frequency and a high amount of answers relates to a high frequency. And a change of the amount of answers leads to a change in the frequency. Again, it is assumed that there are significant differences in the change of frequencies during the examination if a regular student is compared to a cheating student.

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.

## 4. CONDITIONS

Every cheating detection technique requires a sufficient amount and type of data to base its decision on. Therefore, it is necessary to monitor all relevant activities surrounding the exam. This includes not only the final submission of each student but also intermediate results, network activity, programs launched on the students devices etc. These data should be saved with a timestamp so that the exam software is able to represent the continuous interaction between the students and their computers. In the case of a programming exam with an integrated development environment, a version of the current solution could be stored every time the examinee compiles his or her code and produces a new version of the result file(s).

Event logs for the application of Process Mining need to be created during exam in way that does not interfere with the overall performance of the software and the events must be determinable on each type of device regardless of the operating system and its configuration.

Author verification techniques are often dependent on a number of reference material like typing behaviour or coding examples. These have to be obtained in advance under preferably similar conditions and the examinees have to consent to their usage. This means that, in addition to the actual examination software or as an extension of it, programs for gaining such material need to be installed on the student's devices. This may already limit the measures that can be taken in a BYOD context to those which work with data that can be obtained during the semester without extensive monitoring. While most students will agree to hand in work for grading that they did at home or during a tutorial, they might not accept a software on their devices that records their typing behaviour over a long period of time.

## 5. CONCLUSIONS AND FUTURE WORK

Cheating detection for digital exams requires different measures than for paper-based exams. While examinations on the computer allow for new ways of cheating they also offer new possibilities for the detection of fraud. Especially in the context of BYOD, work samples that are created during the semester form well suited reference material for author verification.

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 still has to be made by the examiner.

Next steps include the prototypical implementation of the proposed ways of a-posteriori cheating detection. This will happen in our ongoing project called FLEX. It will be, however, a very tricky task to gather data to verify the implemented prototypes' correct functioning. Gathering test data is not easy, because it can not be artificially created in a sound way. There are plenty of ways how cheating can take place and even if students are asked to simulate a cheating attempt, it is questionable whether this will lead to the desired results.

## 6. 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., Weijters, A.J.M.M., Maruster, L. (2004). *Workflow Mining: Discovering Process Models from Event Logs.* IEEE Transactions on Knowledge and Data Engineering , 16 (9):1128–1142.

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


## 7. AUTHORS' BIOGRAPHIES

**Bastian Küppers, M.Sc.** is research associate at the IT Center of RWTH Aachen University. His research focuses 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 for the study program "Applied Mathematics and Computer Science" at FH Aachen University of Applied Sciences.

**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.