**Koli Calling**
International Conference on Computing Education Research

# An Application to Discover Cheating in Digital Exams

Julia Opgen-Rhein, Bastian Küppers, Ulrik Schroeder

RWTH AACHEN UNIVERSITY

# An Application to Discover Cheating in Digital Exams

## Motivation

- Cheating is a problem in examinations and can have many forms [1]
- Electronic exams come with an increased danger of impersonation and illegal communication between students [2]
- A-posteriori cheating detection that goes beyond plagiarism detection is required
- Code de-anonymization has already be successfully been performed with code from experienced programmers [3]

Goal of the presented work:

1. Show that it is possible to apply machine learning for cheating detection in electronic exams
2. Find the machine learning algorithms that give the best results in this setting

# An Application to Discover Cheating in Digital Exams

## Our Project: FLEX


**FLEX (Framework for FLExible Electronic EXaminations)**

# Individual Programming Style

## Assumptions

- Every student has a unique programming style
- This style is reflected in the layout of the code as well as in the way classes, methods and variables are named and which elements of the programming language are frequently used
- These personal styles are distinguishable, even if students work on the same tasks and have the same lecturer

## Example:

**Write a method that sums up all natural numbers that are smaller than a parameter n.**

```
private int sumTo(int n){
  int ret=0;
  for (int i=0;i<=n;++i){
    ret=ret+i;
  }
  return ret;
}
```

```
public int sum_to(int n)
  {
    int ret = 0;
    int i = 0;
    while (i < n)
    {
      i++;
      ret += i;
    }

    return ret;
}
```

# Data

I) 12 on-campus assignments from 13 students of a first semester programming course in Java

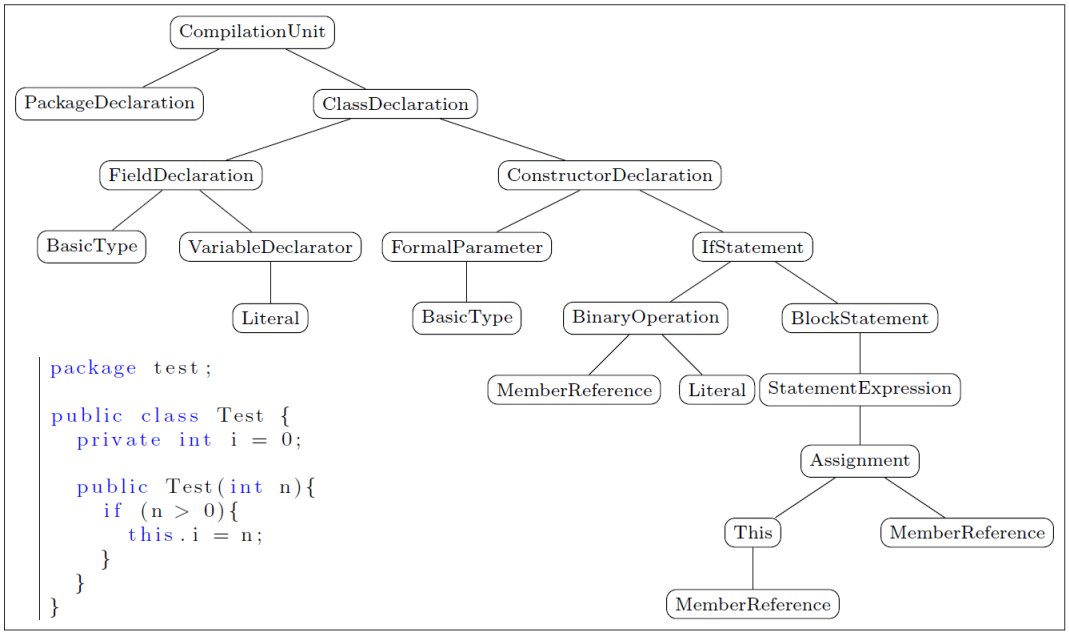
- Very strict tasks
- no electronic exams exist for this course

II) A set of files from the best 40 (Java) programmers that participated in Google Code Jam 2017 contest

- Participants had to hand in a single file for each task
- Large and deeply nested code files
- Up to 2500 lines of code
- Experienced programmers

# Features

- Feature vector for every file
- Layout and syntax features
- Varying number of features




Abstract Syntax Tree

# Features

## Examples

| Layout Features | Syntactic Features |
|---|---|
| Number of inline comments | Number of underscores/method name |
| Number of block comments | Number of prefix operators |
| min/max/avg number of consecutive empty lines | Number of consecutive calls (e.g. do().doNext()) |
| Number of  spaces | Number of short notations (e.g. +=) |
| Number of { in new line | Number of classes |
| Number of { in same line | Number of chars/variable name |

# Implementation

- Python program that works with Java source code


- Javalang library for parsing of source code

- Scikit-learn was used for the machine learning part
  - (Linear Support Vector Machine)
  - Neural Network
  - Random Forest

# Results

| Aachen Student Data Set | | | |
|---|---|---|---|
| | Support Vector Machine | Neural Network | Random Forest |
| Top 1 | 44.54% | 34.69% | 58.36% |
| Top 3 | 57.69% | 37.78% | 67.15% |

~ 93.8% accuracy on Google Code Jam Data Set

# Recent Progress

- Additional Data Sets
- Improved Normalization
- Improved Feature Set
- Feature Reduction

# Recent Progress

## Results

| | Aachen Student Data Set | | |
|---|---|---|---|
| | Support Vector Machine | Neural Network | Random Forest |
| Before | 44.54% | 34.69% | 58.36% |
| Now | 80.15% | 78.49% | 89.15% |

| Google Code Jam Data Set | | |
|---|---|---|
| Support Vector Machine | Neural Network | Random Forest |
| 100.00% | 99.69% | 100.00% |

# Summary & Outlook

**Code de-anonymization is a promising approach for cheating detection in electronic exams**

- A weighted scheme to combine SVM, RF and NN into an ensemble classifier needs to be calculated
- Expand the application to work with other common programming languages like Python and C++
- Final tests with assignments and an exam have still to be carried out

**Thanks for your attention!** ☺
**Kiitos huomionne!** ☺

Are there any questions or comments?

# Sources

[1] Martin Dick, Judy Sheard, Cathy Bareiss, Janet Carter, Donald Joyce, Trevor Harding, and Cary Laxer. 2003. Addressing student cheating: definitions and solutions. ACM SIGCSE Bulletin 35, 2 (June 2003), 182–196. http://www.cs.kent.ac.uk/pubs/2003/1645

[2] Guttorm Sindre, Aparna Vegendla. 2015. E-exams versus paper exams: A comparative analysis of cheating-related security threats and countermeasures. Norsk Informasjonssikkerhetskonferanse (NISK) 2015.

[3] Aylin Caliskan-Islam, Richard Harang, Andrew Liu, Arvind Narayanan, Clare Voss, Fabian Yamaguchi, and Rachel Greenstadt. 2015. De-anonymizing Programmers via Code Stylometry. In 24th USENIX Security Symposium (USENIX Security 15). USENIX Association, Washington, D.C., 255–270. https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/caliskan-islam