The Academic Integrity Violations Detection System for Data Science Course on the MOOC-platform

P.Armila Devi, B.Manju Bhargavi, M. Raghu Kedarnath Kumar

[1,2,3] *Asst. Professor in Computer Science, St.Joseph's Degree College, Kurnool*

Received: May 01, 2018 Accepted: June 10, 2018

Abstract *Academic integrity violations (as plagiarism, answers sharing, etc) are major challenges for many high education institutions around the world. These challenges had become more urgent after wide adoption of e-learning approach by education community in form of MOOCs or SPOCs. Therefore, the usage of automatic academic integrity violation detection systems is highly advisable in case of courses with large enrollment and large amounts of machine- and peer-graded assignments. In this work, we share our experience how to overcome these challenges in such course - Introduction to Data Science. The brief description of the course and assignments structure will be given, after that we will highlight some features of the assignments for Data Science course that requires a bit more complex approach to plagiarism detection and we will discuss architecture of proposed system for academic integrity violations detection.*

Keywords: *academic integrity; plagiarism; data science; massive open online course; small private online course; big data analysis*

INTRODUCTION

Academic integrity violations can take different forms, e.g. plagiarism, answers sharing, exam cheating, tests dumping, etc. This challenge is known from ancient times, and many institutions face this problem – among them there are not only educational organizations such as schools, colleges and universities, but also hi-tech companies with programs for external professional certification, and ever corporations with internal training and staff assessment programs. Some of educational institutions try to overcome these challenges by introduction and enforcement of the Honor Code, other try to educate their students by promotion a culture of academic integrity through special ethic courses or by introducing freshmen to these concepts as soon as possible in other courses. But the challenge became more urgent in recent years after wide adoption of e-learning methods in education, where course grades or credentials are based, partially or entirely, on auto-grading or peer-grading. Therefore, some institutions are invested in development of special tools – academic integrity violation detection systems that can help to detect academic misconducts in fully automatic manner or raise the red flags to attract attention to suspicious behavior.

In this work, we will give brief information about works related to academic integrity and we will discuss MOOC and SPOC phenomena in this context. After that we will give example of the MOOC/SPOC curriculum for Introduction to Data science course with focus on assignments' structure and we will share our concern and some statistics about violations of academic integrity in these assignments. We will conclude this paper by introduction of the architecture of academic integrity violations detection systems that can be used with this type of on-line or blended courses.

RELATED WORKS

As we mentioned above, the violation of academic integrity is a major challenge for educational organizations around the world. A good review about academic dishonesty can be found in [1]. Some recommendations for educators about cheating awareness and how to handle this issue are provided in [2]. Interesting results were shown in [3], the author shared statistics about Honor Code violation in Stanford University according to which majority of the cases was originated from computer science department.

Many authors provide comprehensive analysis of academic misconducts in different settings, sometimes their researches are focused on particular regions or countries. For example, in authors provide analysis of fraudulent behavior among the students in setting of Portuguese high education institutions with major in Engineering. They highlight that more than 94% of the students recognize of the fraud existence and more than 20% of them responded that it occurred on a regular basis. Authors propose a theoretical model which divides academic fraud into four main types – appropriation, simulation, facilitation and concealment; also they share statistical data about frequency of different fraud types occurrence and perception of its seriousness among Portuguese students. Another example is [5] where

authors tried to understand reasons behind plagiarism among students in Malaysian public universities, they highlighted three major categories of plagiarism causes – historical baggage, institutional demands, individual attributes and perceptions. In the [6], authors shared their point-of-view on plagiarism causes in setting of Indian educational institutions, they draw attention to how policy making can affect ethical issues in academic processes, also they provide approach to changing learning habits and behavior of students including reviving the syllabus in a way to became more oriented on practical activities.

In general, there are two ways to approach the issue of academic integrity violations in high education institution – promote ethical behavior and detect (and punish) fraudulent one. There are several works that suggest to introduce concept of ethic and academic integrity into the curricula as soon as possible. The good example of such initiative is [7] where authors describe the academic integrity module for freshman students developed and adopted in University of Puerto Rico-Mayaguez. In context of the module, authors formulated and promoted Academic Integrity as "a fundamental condition that makes possible the mission of the University." Another interesting approach to promote academic integrity was described in [8]. During their research, authors provided on-line access to results of code similarity analysis obtained through special software for the students taking programming class to raise awareness about plagiarism. As a result, majority of the students prefer to change their habits, as code or idea sharing, in case of the tool adoption.

A. General Methods for Plagiarism Detection

Scientific community struggled with the task of plagiarism detection for decades, as a result nowadays, there are several well-known approaches that can be used to compare documents in case of text or ever idea coping detection. Some methods are working in general, but some created for plagiarism detecting in special types of documents, as program code, spoken language or images. In this brief review we will highlight some recent works relevant to this topic.

In [9], several algorithms for plagiarism detection in natural language documents are listed, among them there are fingerprinting, n-gram overlap analysis, usage of word frequency metrics, different features extraction as syntactic patterns, stop words, etc. Although some of the listed methods can perform well in general, authors raise awareness that the testing for nearly all published results of such systems performance are based on artificially created examples of plagiarism. The review of some methods and comparison research of on-line plagiarism detection systems are published in [10].

The fingerprinting is very popular approach of solving problem of document comparison through some stages of plagiarism detection, it is used in several tools as [11] or [12]. First tool is well-known software for plagiarism detection in computer programs source codes using by several Universities, second is the one of recent research projects in this field, which highlights some limitations of current approaches such as the implementation is memory and bandwidth consuming. In [13] authors had achieved six time speed up for plagiarism detection tasks inside large corpus of documents by using GPUs for sequence matching algorithm.

Unfortunately, fully automated plagiarism detection at current state of researches is not possible in case of highly-obfuscated texts or for detection plagiarism of ideas. Authors of [14] proposed usage of the special visualization tools for documents comparisons based on compressed bitmaps text representation as sets of words, and they showed feasibility of this approach for rather complex cases as cross-languages or translation plagiarism and theft of ideas. Another approach to solve this task is shown in [15] where authors tried to combine via logical regression model different well -known similarity metrics using for text comparison as lexical (Dice and Jaccard Coefficients; Jaro, Levenshtein, Manhatan, Ngram and Soundex Distances), syntactic (POS N-gram Distance and Noun Ratio), semantic (semantic similarity distance) and structural (Stopword N-gram Distance, Word Pair Order, String Length Ratio) features.

Another complication for plagiarism detection is necessity to detect plagiarized images (or even parts of image) in compound documents as this task usually has huge dimensionality. The research described in [16] is devoted to solve this task by applying higher degree F -transformation to the image that can drastically reduce further images processing complexity in favor of the images plagiarism detection.

B. Plagiarism Detection for Source Code

The review of a state of art on source code plagiarism detection is given in [17], where authors distinguished plagiarism detection algorithms and systems on textual and source code where the second ones can work on string, token, parse-tree or program dependencies graph levels or can use feature metrics for code as count of loops, variables, etc.

As mentioned in [18], tree-based algorithms for source code copy detection can outperform string- or token-based in more complex cases. Moreover, authors of this paper introduced the algorithm for source code plagiarism detection based on abstract syntax trees transformation - linearization and subtrees regrouping with special care of false-positives in case of syntax trees for arithmetic operations. As mentioned in [19], fingerprinting is a common technique for abstract syntax trees comparison, but right choice of the hash function is crucial.

For relatively simple cases (non-obfuscated source code plagiarism or code snippets) combinations of string - and token-based algorithms can be helpful. JPlag [20] is well-known open-source software that is successfully used by academic community for code plagiarism detection for more than decade, some examples are given in [21]. It supports variety of programming languages (Java, Python 3, C, C++, C#, Scheme) and plain text analysis. Moreover, it is easily expandable to new one by providing a parser for the language and a few lines of code that send the tokens to JPlag.

MOOCS AND SPOCS

Massive Open Online Courses (MOOCs) and Small Private Online Courses (SPOCs) are relatively new learning paradigms, but they are already highly adopted by academic institutions around the world. The reviews of some works at the field and how these approaches can be used in high education can be found in [22]. Thus, combining SPOCs, MOOCs, vendor and/or OEMs trainings with on-site classes can help to create powerful learning environment with rich learning experience for the students as shown in [23]. Detailed example of how the SPOC can help to flip classroom with careful analysis of students' interaction with on-line learning content is given in [24], it also shows that this classroom model improves student involvement, satisfaction and grades.

Some authors, as [25], already talk about post-MOOC era in education, that is characterized by wide adoption of SPOC learning environments. As they provide more personalized learning experience, it leads to higher completion rate and can be used in blended classroom environments for teaching for-credit courses. Also, students show high degree of satisfaction using SPOC environments, as shown in [26].

In practice, usage of SPOC-platforms can raise some challenges, thus researchers from [27] highlight three of them: professor should spend more time to course planning and curriculum preparation; some learners have lack of self-study ability that is crucial in flipped classrooms; shortage of information system environment (hardware, software, IT staff) to facilitate that kind of learning support systems. Moreover, authors from [28] showed that students tend to lose interest in usage learning technologies during the course in SPOC-based classroom and willing to participate in learning activities only if they can be directly translated to grades and tend to skip non-graded assignments.

The plagiarism in MOOC/SPOC environments is another challenge, thus author of [29] shares results that the substantial amount of students from SPOC-supported Data Structures course tend to copy other students' programs. The authors of

[30] are also aware of plagiarism in their Programming course, where they tried to prevent this type of misconduct by usage of large pool with time-restricted programming assignments. As mentioned in [31], major MOOCs providers try to address this issue by enforcing Honor Code, typing style control and web photo verification due assignments. But it is not enough in blended learning classrooms that provide training for credits, as students know each other and are more willing to participate in unlawful results sharing.

INTRODUCTION TO DATA SCIENCE COURSE CURRICULUM AND EXPERIMENT SETUP

Our course curriculum is based on [32], that was substantially enriched as described in [23]. Main topics of the course are covered in seven modules:

- introduction to Big Data analytics;
- data analytics lifecycle;
- review of data storage and data processing infrastructure;
- review of basic data analytic methods using R;
- theory and methods of data analysis;
- technology and tools for data analysis;
- data visualization and creating final deliverables.

As this class is very intensive, part of learning materials is provided through SPOC-environment and substantial part of assignments are in form of peer-graded assignments and quizzes that facilitated by standard Open edX components [33] Also student should learn some concepts, as data formats and data

querying techniques, through external MOOC-systems, as [34], to obtain grades for these assignments they should upload certificates or transcripts obtained from these external systems into special forms in our SPOC-environment.

To obtain hand-on expertise students usually work in lab environment to learn new concepts and instruments under guidance of professors or teacher assistants, and interact with our SPOC-environment in form "In -lab additional exercises" where they receive the dataset and few analytical questions and guided through analysis by quizzes with automatically graded responses in conclusion, they should compose a brief report usually in R Markdown format that incorporates data, code, visualization and analytical conclusions. They should upload these reports for grading by course staff to obtain full grade on assignments.

The course "Introduction to Data Science" was launched in September 2016 on our departmental MOOC/SPOC-platform [35] The 33 of 76 participated students were from MEPhI, others were from MIPT. Registration and participation in learning activities through the platform was mandatory in both cases. Students from MEPhI were already exposed to the platform usage in course "Computer Networks" with analogous assignment structure, but usage of the platform in that case was optional, and no severe academic misconduct were detected among that group of students (only small part of them had active interaction with the platform for extra credits as these students were highly motivated to do assignments for themselves). Some of the students from both Universities were exposed to the MOOCs on major public platforms, but no further analysis was made due to lack of data about behavioral habits of those students on public platforms.

The course includes 4 peer-graded open response assignments, 9 in-lab additional exercises (7 of them were quiz-guided analytical mini-researches with requirements to upload reports in form of R Markdown document, 2 of them were analytical mini-researches without quizzes), also there were three programming assignments on SQL, Java and HQL for Greenplum [36] and Hadoop ecosystem [37].

The students were aware that some sort of plagiarism detection system will be deployed but no reports about plagiarism (or other forms of misconducts) detection were made to the student until the end of the course, as it seems doubtful with pedagogical point-of- view, but it is justifiable for data collection about academic integrity violations. All students' submissions were graded by course staff members (professors or TA) to obtain training data and after that by different automatic methods were described below.

ACADEMIC MISCONDUCT DETECTION SYSTEM FOR DATA SCIENCE COURSE

From our previous experience with courses that use e-learning components in form of external or internal platform usage from MEPhI, MIPT and MISiS Universities we encountered several forms of academic integrity violation in scale that should be addressed by automatic or semi-automatic system that can help course staff to detect these misconducts and act according to University policy:

- external certificate forgery,
- plagiarism in programming or analytical assignments,
- unlawful collaboration on quizzes and other automatically graded assignments,
- unfair grading in peer-graded assignments.

This work is focused on the first two of these misconducts.

Hence, we need a compound system for academic misconduct detection that addresses all these issues and allowed to gather information of students' misbehavior in the 'umbrella' model that covers all aspects because it is supported by our observations that misconduct in one aspect is raised probability of unlawful behavior in others. Moreover, frequently tendency of academic integrity violations for particular individual is persistent between courses in case of usage e-learning environments, therefore in best case scenario this information should be shared in multi-course environments.

The architecture of proposed system is depicted at the figure 1. Learning Management System (*LMS)* is providing access to the rich learning content for different SPOCs or MOOCs including automatically and semi-automatically graded assignments and generates internal logs (*int. logs*) with information about who, when and how (with what outcome) accessed the content. These logs are crucial for the unlawful collaboration detection system (*UCDS*) that can utilize process mining [38] techniques to detect "leader - followers", "collaborative guessing" and other patterns [39], but further discussion of those techniques is out of scope of this work.

Depending on the type of assignments different systems for academic misconduct detection (*AMDS*) can be used, consideration about two of them – external certificate checker (*ECC*) and plagiarism detector for R Markdown assignments (*PD-R*) will be further discussed in a bit more details. These modules can use

compound models to detect misconducts as will be discussed in plagiarism detection for Data Science course subsection.

Sometimes academic misconduct detection is straightforward (for example, if some of the submitted works are exactly the same), but sometimes we can't say with full confidence that the some sort of misconduct had been happened, in that case probabilistic models can be useful, for example, we can train Bayesian Network that incorporates information from different misconduct detection modules and information about history of academic misconducts (*AM logs*) from current or past running of the course or ever cross-course information for particular learners or learner groups [40].

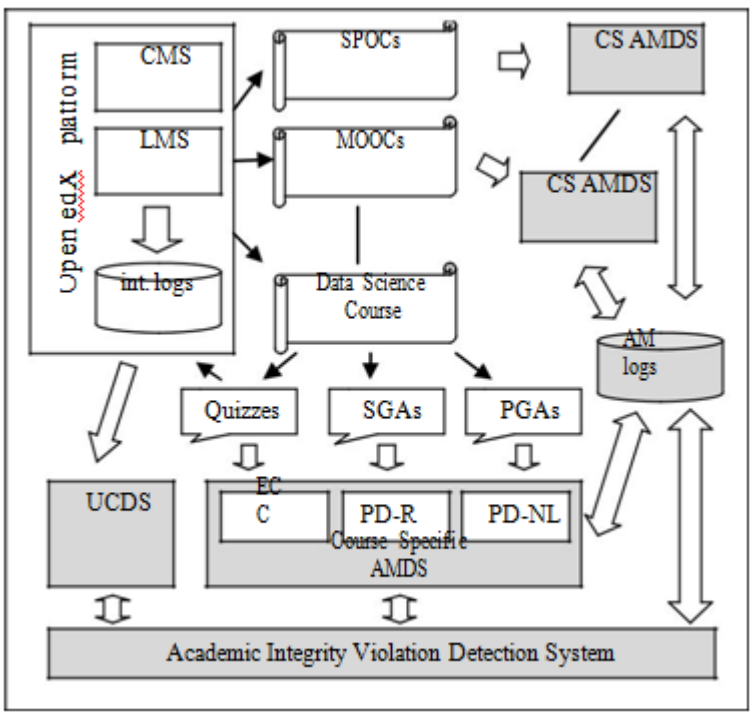

Fig. 1. The architecure of academic integrity violation detection system

A. External Certificates Auto-checking System

From our previous experience of using external MOOCs as sources for supplemental learning and additional assignments, several cases of certificate forgery were reported where student tried to change name or date on certificate. Hence, to have system that can check certificate validity in course with high enrollment is highly desirable.

We accept two forms of external certificates – one of them are certificates issued by Open edX platform from Stanford University [34], others are in form of badges [41]. In both cases external certificate checking is straightforward as certificates (or link to badge itself) contains verification link to external repository that contains record about student name, credential name, date of credential issuance (it is not in case of Open edX certificate, but the date is on the certificate itself), etc. So we can easily load the batch of uploaded certificates after deadline and run checking process by parsing certificates and linked pages of web-repositories with verifications.

There are only few false positives are reported when students used different name spelling in our and external e-learning systems, that should be resolved through course staff assistance.

B. Plagiarism Detection in R Markdown Assignments

Unfortunately, there is no known solution for plagiarism detection in R or R Markdown documents. In case of R Markdown documents, the complexity of analysis is increasing because the document has the compound structure, and we should analyze R code, for example, in accordance with R language grammar, text part as a text on natural language (English or Russian in our case) and markdown part in accordance with markdown grammar. Also we should consider that R-scripts in most of our assignments are relatively short, they have linear structure, and often they are based on examples given in textbook. But as we will show it is possible to use several well-known lightweight approaches that are usually used for monolingual

documents either for natural languages, or different programming languages and heuristics to detect plagiarism in that kind of assignments. In this work we examined the following approaches:

- the "punctuation anomalies" heuristic,
- JPlag with natural text analysis,
- JPlag with R tokenizer analysis,
- bag-of-words model with use R functions as keywords.

The "punctuation anomalies" heuristic is our extension of the method based on stopword n-gram distance [42] that is based on clever observation that students tend to copy punctuation even they had renamed variables and rearranged pieces of the code; and if there is an anomaly, for example, in the function call where most of parameters are separated by comma and one space, but some of them are separated by only comma or comma with double space, this anomaly will be copied with high probability if the piece of that code had been plagiarized.

In this work, we perform the experiments with Jplag in two forms – when the R Markdown is treated as a text and with frontend with slightly modified R grammar obtained from [43] that can anticipate R Markdown format. In first case, the false positives rate is low, but we miss too much plagiarized assignments where R code is essentially the same, but comments are different. In second case, false negatives rate is substantially elevated, but the method showed almost best performance in catching plagiarized assignments.

Also, we had investigated bag-of- word model that can help to solve task of plagiarism detection, in that case we only used subset of R standard function names as keywords and several similarity metrics: cosine similarity and two domain specific heuristic metrics – average rate of similar functions usage (AVR) and rate of exact amount function usage with adaptive threshold (MAX). As correlation analysis shows, the introduced metrics perform well in term of accuracy in different cases and they allow to catch different types of plagiarism depending on code complexity and using plagiarism obfuscation techniques. The results of these methods in terms of percentage false positives and false negatives are shown in figure 2.

The introduced method can perform as "weak learners" in the compound models, but description of these models is beyond the scope of this work as we are still gathering data from current version of the course.

CONCLUSIONS AND FUTURE WORKS

MOOCs and SPOCs are great (self-)education tools for highly motivated persons seeking for new knowledge or skills that have potential to provide access to high quality learning content from leading educational organizations worldwide for free or for affordable fees. They also have provided ability to enrich and intensify learning process in traditional educational setting too.

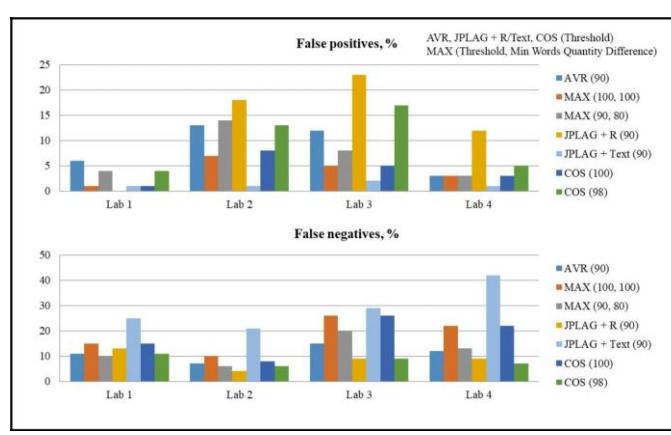

Fig. 2. False positives and false negatives rates for plagiarism detection methods in R Markdown assignments

But there is a bunch of challenges that will arise when we try to use these educational models as credit granting education. The violation of academic integrity by some learners are among them, hence the ability detection of such misconducts is a valued part of credits granting educational environment.

In this work, we share our experience with academic integrity violation by students using supplemental learning materials and assignments on MOOC platforms in cross-university settings. As a result, we had justified need of the system that can help to detect and enforce prosecution of such misconducts. The architecture of the system was proposed. It includes course-specific academic misconduct detection system (with assignment-specific modules), cross-course unlawful collaboration detection system, academic misconduct detection logs that can be used with high-level academic integrity violation detection models based on probabilistic approach. Also need and usefulness of hierarchical models for plagiarism detection in case of analytical assignments in R Markdown format are shown.

Unfortunately, training high-level Bayesian models for academic misconducts detection are required a lot of data, and we gather additional data from current run of the courses on our platform, so comprehensive description of these models is beyond this paper and will be given in the future works.

Academic misconduct detection in fully fledged MOOC environments due amount of generated data and much higher probability of false positives is another challenge that should be addressed in the following projects.

REFERENCES

[1] T. S. Harding, D. D. Carpenter, S. M. Montgomery, and N. H. Steneck, "The current state of research on academic dishonesty among engineering students," in *31st Annual Frontiers in Education Conference*, Oct 2001, vol. 3, pp. F4A13–F4A18.

[2] V. Maier-Speredelozzi, "Promoting academic integrity in your first classroom," in *34th Annual Frontiers in Education*, 2004. FIE 2004., Oct 2004, vol. 3, pp. S1C9–S1C14 .

[3] E. Roberts, "Strategies for promoting academic integrity in cs courses," in *32nd Annual Frontiers in Education*, 2002, vol. 2, pp. F3G14–F3G19.

[4] P. Gama, F. Almeida, A. Seixas, P. Peixoto, and D. Esteves, "Ethics and academic fraud among higher education engineering students in portugal," in 2013 *1st International Conference of the Portuguese Society for Engineering Education (CISPEE)*, Oct 2013, pp. 1–7.

[5] C. M. Chan, A. S. M. Seman, and A. Shamsuddin, "Plagiarisim: A review of why malaysian students commit the academic dishonour," in *2014 International Symposium on Technology Management and Emerging Technologies*, May 2014, pp. 119–122.

[6] Pallela and Talari, "Plagiarism: a serious ethical issue for indian students," in *2016 IEEE International Symposium on Technology and Society (ISTAS)*, Oct 2016, pp. 1–6.

[7] L. O. Jimnez, E. O'Neill-Carrillo, and M. Rodrguez, "An introductory learning module on ethics and academic integrity for freshman engineering students," in 2009 *39th IEEE Frontiers in Education Conference*, Oct 2009, pp. 1–6.

[8] T. Le, A. Carbone, J. Sheard, M. Schuhmacher, M. de Raath, and C. Johnson, "Educating computer programming students about plagiarism through use of a code similarity detection tool," in *2013 Learning and Teaching in Computing and Engineering*, March 2013, pp. 98–105.

[9] X. Wang, K. Evanini, J. Bruno, and M. Mulholland, "Automatic plagiarism detection for spoken responses in an assessment of english language proficiency," in *2016 IEEE Spoken Language Technology Workshop (SLT)*, Dec 2016, pp. 121–128.

[10] S. Krizkova, H. Tomaskova, and M. Gavalec, "Preference comparison for plagiarism detection systems," in *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, July 2016, pp. 1760–1767.

[11] S. Schleimer, D. S. Wilkerson, and A. Aiken, "Winnowing: Local algorithms for document fingerprinting," in *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '03. New York, NY, USA: ACM, 2003, pp. 76–85. [Online]. Available: http://doi.acm.org/10.1145/872757.872770

[12] M. Elkhidir, M. M. Ibrahim, T. A. Khalid, S. Ibrahim, and M. Awadalla, "Plagiarism detection using free-text fingerprint analysis," in *2015 World Symposium on Computer Networks and Information Security (WSCNIS)*, Sept 2015, pp. 1–4.

[13] M. Jiffriya, M. A. Jahan, H. Gamaarachchi, and R. G. Ragel, "Accelerating text-based plagiarism detection using gpus," in *2015 IEEE 10th International Conference on Industrial and Information Systems (ICIIS)*, Dec 2015, pp. 395–400.

[14] A. Schmidt, S. Bhler, R. Senger, S. Scholz, and M. Dickerhof, "Detection and visual inspection of highly obfuscated plagiarisms," in *2016 49th Hawaii International Conference on System Sciences (HICSS)*, Jan 2016, pp. 4113–4122.

[15] L. Kong, Z. Lu, H. Qi, and Z. Han, "High obfuscation plagiarism detection using multi-feature fusion based on logical regression model," in *2015 4th International Conference on Computer Science and Network Technology (ICCSNT)*, vol. 01, Dec 2015, pp. 355–359.

[16] P. Hurtik and P. Hodakova, "Ftip: A tool for an image plagiarism detection," in *2015 7th International Conference of Soft Computing and Pattern Recognition (SoCPaR)*, Nov 2015, pp. 42–47.
[17] M. Agrawal and D. K. Sharma, "A state of art on source code plagiarism detection," in *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, Oct 2016, pp. 236–241.
[18] J. Zhao, K. Xia, Y. Fu, and B. Cui, "An ast-based code plagiarism detection algorithm," in *2015 10th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA)*, Nov 2015, pp. 178–182.
[19] M. Chilowicz, E. Duris, and G. Roussel, "Syntax tree fingerprinting for source code similarity detection," in *2009 IEEE 17th International Conference on Program Comprehension*, May 2009, pp. 243–247.
[20] "Jplag," Sep 2017. [Online]. Available: https://github.com/jplag/jplag
[21] M. Novak and M. Binas, "Automated testing of case studies in programming courses," in *2011 9th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, Oct 2011,pp.157–162.
[22] S. V. Andrianova, A. A. Dyumin, and L. I. Shustova, "The intensification of the education on telecommunications using mooc-platforms," in *2015 International Conference on Engineering and Telecommunication (EnT)*, Nov 2015, pp. 87–92.
[23] A. A. Dyumin and S. V. Andrianova, "Moocs and vendor trainings in academic curriculum: Yet another step towards global university," in *2016 International Conference on Engineering and Telecommunication (EnT)*, Nov 2016, pp. 39–44.
[24] G. Martnez-Munoz and E. Pulido, "Using a spoc to flip the classroom," in *2015 IEEE Global Engineering Education Conference (EDUCON)*, March 2015, pp. 431–436.
[25] L. H. Zhang and F. Li, "Application of the spoc teaching mode in courses of computer network in the post-mooc period," in *2016 8th International Conference on Information Technology in Medicine and Education (ITME)*, Dec 2016, pp. 436–440.
[26] G. Shouchao, Y. Chunyan, and Z. Shenghui, "Survey of satisfaction of small private online course (spoc): A case of chuzhou university students, china," in *2016 International Conference on Educational Innovation through Technology (EITT)*, Sept 2016, pp. 193–197.
[27] J. Zhou, H. Yu, B. Chen, C. Mai, and L. Yu, "The construction of teaching interaction platform and teaching practice based on spoc mode," in *2016 11th International Conference on Computer Science Education (ICCSE)*, Aug 2016, pp. 293–298.
[28] S. Bansal and P. Singh, "Blending active learning in a modified spoc based classroom," in *2015 IEEE 3rd International Conference on MOOCs, Innovation and Technology in Education (MITE)*, Oct 2015, pp. 251–256.
[29] Q. Huang, "Reveal the key factors in affacting the spoc-supported course: Data and survey analysis for data structures course in ustb," in *2016 IEEE 16th International Conference on Advanced Learning Technologies (ICALT)*, July 2016, pp. 300–301.
[30] X. Su, T. Wang, J. Qiu, and L. Zhao, "Motivating students with new mechanisms of online assignments and examination to meet the mooc challenges for programming," in *2015 IEEE Frontiers in Education Conference (FIE)*, Oct 2015, pp. 1–6.
[31] R. Tsoni and A. Lionarakis, "Plagiarism in higher education: The academics' perceptions," in *2014 International Conference on Interactive Mobile Communication Technologies and Learning (IMCL2014)*, Nov 2014, pp. 296–300.
[32] EMC Education Services, *Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data*. John Wiley & Sons, 2015.
[33] "Open edX." [Online]. Available: https://open.edx.org/
[34] "Stanford Lagunita." [Online]. Available: https://lagunita.stanford.edu/
[35] "Hyper MEPhI." [Online]. Available: https://hyper.mephi.ru/
[36] "Introduction to greenplum in-database analytics." [Online]. Available: http://greenplum.org/gpdb-sandbox-tutorials/introduction-greenplum-database-analytics/
[37] T. E. White, *Hadoop: The Definitive Guide (4th Edition)*. OReilly Media, 2015.
[38] W.M.P. van der Aalst, *Process Mining: Discovery, Conformance and Enhancement of Business Processes,* 1st ed. Springer Publishing Company, Incorporated, 2011.
[39] A. A. Dyumin, "Process mining in e-earning systems: collaboration detection", unpublished.
[40] A. A. Dyumin, "About Bayesian models for academic integrity violation detection", unpublished.
[41] "Digital badges are unlocking the global job economy." [Online]. Available: https://www.youracclaim.com/
[42] E. Stamatatos, "Plagiarism detection using stopword n-grams," *Journal of the American Society for Information Science and Technology*, vol. 62, no. 12, pp. 2512–2527, 2011. [Online]. Available: http://dx.doi.org/10.1002/asi.21630
[43] T. Parr, "R grammar." [Online]. Available: https://github.com/antlr/ grammars-v4/blob/master/r/R.g4