Plagiarism Detection in Learning Management System

Arkan Kh. Shakr SABONCHI
Mathematics and Computer Science Department
Cankaya University
Ankara, Turkey
arkankhaleel@gmail.com

Abdül Kadir GÖRÜR
Computer Engineering Department
Cankaya University
Ankara, Turkey
agorur@cankaya.edu.tr

***Abstract*—Learning Management Systems has become one of the teaching tool same as the pen, pencil, blackboard, book, and notebook. Learning Management System (LMS) is defined and types that are derived from this system such as those focused only on the curriculum and those focuses on the management of the education system, and the combination of both are explained. The properties and features that must exist are identified and showed so that it can be called a learning management system. The name and the experiences of countries in terms of the use of this system as well as the benefits resulting from the user experiences have motivated schools and educational institutions to start the establishment of such systems. The main purpose of this study is, an online plagiarism detection system will be developed and integrated into Moodle so that it can be used by teachers to detect cheating and plagiarism cases in students submitted answers**

Keywords— Learning Management System; Moodle; plagiarism detection algorithms

I. INTRODUCTION

In this study, Moodle system was selected since it is free and open source. This gives an opportunity for schools to work without restrictions or wage. The development and widespread of such systems, programs, and the Internet that use by students in all education levels have become the main reason that students depend on the Internet to get solutions to their homework's and share it between them, so it leads to the negative use of learning management systems. Besides benefits and functionalities of these systems, the major problem regarding LMS is plagiarism in student homework and school duties in their academic stages. Since the plagiarism detection and controlling process by a human are a slow process, need for an electronic plagiarism detection system that has been integrated into LMS must be built in order to grade and evaluate electronic student submissions faster and more accurately. Although there are systems and algorithms to detect plagiarism, these systems are not open source and they are not integrated into LMS. In this study, plagiarism detection was developed and integrated into Moodle system. Finally, the test case is prepared and applied to the developed system in order to observe the effectiveness of proposed plagiarism detection approach. The results and error-free application show that the work of proposed plagiarism system is compatible with Moodle system. In this study, our main goal is to demonstrate the applicability of plagiarism detection system in Moodle environment. Therefore, the performance of plagiarism detection algorithm in terms of accuracy and error write are out of the scope of this study and left as future work.

II. MODEL

Moodle, is a Learning Management System (LMS) or a Virtual Learning Environment (VLE) or a Course Management System (CMS). It is also known as a free web application with which educators can benefit to create effective online learning websites [1]. In 2002, in Australia, a computer science graduate named Martin Dougiamas was testing a web tool which he developed to help teachers to create lessons online. Inspired by his own experiences with "School of the Air", Martin's Modular Object-Oriented Dynamic Learning Environment (MOODLE) offered tutors a way to connect remotely with their students in a collaborative and supportive workspace [1,2]. Moodle has truly changed the face of learning and now used by millions of students as shown in table I, [3].

TABLE I. MOODLE STATISTICS

Field	Statistics
Registered sites	75,943
Countries	231
Courses	11,770,878
Users	99,945,680
Enrolments	335,287,549
Forum posts	208,306,908
Resources	104,385,541
Quiz questions	561,376,595

978-1-5090-6332-1/17/$31.00 ©2017 IEEE

Moodle also used in thousands of websites in many countries, we display top 10 from registered sites in 231 countries, as shown in table II, [3]. With improved access to the internet, and with commercial companies which are a fast spot for a potential money-earner, many such Learning Management Systems have arisen since then [1,2]. The main purpose of this study is to clarify the idea of learning management systems and the selection of criteria for this system as well as the benefits of using this system and to get to know the experiences of countries that use this system. In addition, an online plagiarism detection system will be developed and integrated into Moodle so that it can be used by teachers to detect cheating and plagiarism cases in students submitted answers.

TABLE II. TOP 10 FROM REGISTERED SITES IN 231COUNTRIES

Country	Registrations
United States	10,104
Spain	7,323
Brazil	4,431
Mexico	3,545
United Kingdom	3,427
Italy	2,647
Germany	2,529
Australia	2,284
India	2,270
Colombia	2,119

III. THE ADVANTAGES OF AN LMS

The five particular advantages of an LMS for the works are listed in the following [4,5].

A. Centralized Learning Environment to Ensure Consistency

Naturally, an LMS has the ability to make all kind of educational content, constructive content, and practice content accessible to persons all the time from any place online. Various members are always able to connect the LMS. The LMS guarantees flexibility in transfer and estimation because every member can access the certain material in a certain manner and can be figured out with the help of shared pre-testing and/or post-testing ways. An LMS enables the members to plan and set up individual educational modules. This component is particularly significant when new material is brought out, the standing material is upgraded or determined performances are importantly corrected. Furthermore, this component carries out upgrades to guidelines and performances. Workers cannot deny that they see them.

B. Tracking and Reporting for Enhanced Performance

The LMS enables executive members to check a needed learning way, carry on the process against the learning way, take a look at again the documents, and sign up for extra courses. Workers can suggest these courses via different media such as instructor-led education, online education or webinars. Administration can see the same documents and can examine the data to Figure out the parts of success and the part that should be improved.

C. Immediate Capabilities Evaluation

The LMS enables the members to be assessed before taking a course while attending the course and ending up a course. Workers are able to evaluate hindrance by routinely managing determined evaluations with the LMS. They can check the documents of the results to decide success levels and real time to finish every course and its parts.

D. Continuous Product and Service Proficiency for Employees who Interact with Customers and Clients

The LMS brings the main opportunity for corporations to alter product portrayal, details, needs, shapes, and to enable to uploading of new product or information about the transferring. Member will see the same educational courses, and the same materials to evaluate. Members customize pre-determined course accomplishment dates and observe the total number of participators accomplishing the course at any time. Moreover, an LMS enables works to manage upgrades and analysis in web and help information levels and capabilities.

E. Regulatory and Legal Compliance

Most corporations should meet some allowable and supervisory needs. The medical area and the pharmaceutical area are the manufacturing with very tough needs. Some allowable and supervisory needs are met by one time incidents whereas the others need routinely check or reconfirmation. Industry, generally, can be needed to meet training regulations assorted by corporate or standard agencies such as OSHA and the EEOC. An LMS guarantees constant transfer of regulated course suggestions, for example, HIPPA, Lockout/Tagout, Sexual Harassment… etc. The LMS, via its kept system, guarantees that the termination of those who took the needed courses can be seen in any time. Corporations can define fields of non-conformance and show remedial actions to decrease the risk of non-compliance with needs.

IV. PLAGIARISM DETECTION ALGORITHMS IN LEARNING MANAGEMENT SYSTEMS

Student plagiarism is, in fact, huge problem facing academic institutions. Plagiarism is often interpreted as copying or duplicating another's work from the originator such as students or from sources such as textbooks, without indicating the proper acknowledgment of the source. Automatic and computer-based plagiarism detection systems are presented to detect plagiarism in student works. The effectiveness of detection in such systems relies on what types of plagiarism they can detect. Such systems offer precious benefits regarding to saving time and efforts of academics to conduct the detection process by themselves. Computer-based plagiarism detection has been interesting to academics in the last two decades due to the fact that using such tools decreases academics workload via automating the comparison procedure and detecting the similarity of student works quickly, which the academics need to check for doubtful similarity [6,7,8,9]. The use of computer-aided plagiarism detection furthermore concerns a set of ethical and legal issues. These issues are caused both by technical imperfectness of plagiarism detection algorithms (for example, a system might incorrectly suspect a student's work as plagiarized) and by misunderstanding the role of plagiarism detection software in the educational process. Due to the importance and the rising interest in ethics of automated plagiarism detection [10,11]. There is an increasing and common issue in the scholar process. The plagiarism in the exams. The classic, not automated disclosure of such sort of plagiarism is hard, unreliable, and slow technique. This task addresses to build up a plagiarism detection program which is accessible by the Internet and which is able to assist college or school instructors to get a more improved awareness for a learner's task. In present, there are lots of standing detection algorithms. Since In this study our main goal is to demonstrate the applicability of plagiarism detection system in Moodle AC algorithm is consider to be most appropriate algorithm because of it is simplicity. Since Moodle is developed in PHP environment, in this study the selected algorithm is implemented with PHP. On the contrary, the program which is introduced here is unpaid, open-source and can be associated with Moodle. Existing Cheating Integrated Algorithms and Open Source Software will be introduced in this study. There are many algorithms for plagiarism detection programs which have just been improved. This section presents such algorithms and defines their characteristics. These algorithms may be practically categorized into two parts [12]. Attribute-counting method and Structure-metric method.

A. Attribute-Counting Program

This is the first kind of plagiarism detection algorithm. Such programs figure out the degree of the likeness between a pair of works by benefiting from four basic program statistics: [7].

- Estimated values of distinct operators.
- Estimated values of distinct operands.
- Total estimated values of operator occurrences, complete distinct types.
- Total estimated values of operand occurrences, complete distinct types.

B. Structure-Metric Programs

This kind of plagiarism detection algorithms presents more abundant number of metrics and awareness of likeness for the arising characteristic vector to develop a presentation (depending on the structure and metric match). Such algorithms generally depend on transforming the program to a flood of tokens (so not caring simply unstable knowledge for example blank, border gaps, explanations… etc.) and then matching these token floods to discover likeness between them. The most improved programs in this part (regarding plagiarism detection process) are SIM, MOSS, JPlag, AC and Code Match. Here is a short definition of these programs [12].

1) Software Similarity Tester (SIM): It was developed by Gitchell and Tran in 1999 as a program for figuring out the likeness between the text which is written in C, Java, Pascal and general language.

2) Measure of Software Similarity (MOSS): It was improved by Alex Aiken at Berkeley in 1994 as a program to calculate the likeness of source code written in C, C+, Java or Pascal. MOSS analyzes the source code in the file, indicating it and carrying out the matched algorithm to the indicated shape of the code. Then it matches it with the source code in other files [12,13,14].

3) JPlag: The total knowledge transferred about JPlag is so rare. JPlag does not match the internet. It is planned to discover likeness between the student homework, which is generally adequate for computer programs. But its basic feature is to transfer the programs to shown rows and matching these rows [12,15,16]. The valid JPlag website sums up its study by explaining that likeness of 0% or 5% can be shown by the likeness value only, so it is definitely a plagiarism, however, if there is a 40% likeness, these circumstances generally ought to be examined manually for a final decision.

4) Software Integrity Detection (SID): It determines the likeness between source codes by calculating the accessible information among them. SID is simple to benefit from the program to find plagiarism in a source code and has indicated to be the most efficient at detecting the ones who cheats. SID presently can work with Java and C++ source codes. Looking at these two programs, SID calculates the accessible information among these two [12,17,18].

5) Code Match: It matches each file in sole guide with each file in another guide, containing all subguides if offered. CodeMatch generates an information bank that can be conveyed to an HTML main report that shows the most suitable integrated pairs of files. You can on any specific pair shown in the HTML main report to examine an HTML elaborated report which presents the particular parts in the files (explanations, definitions, instruction orders or comments) that lead to the high integration [12].

6) Anti Copias (AC): It introduces a web page to find the likeness between the works or projects and any member can freely benefit from it. This site gives statistical tests and a plenty of graphical illustrations assist in the judgment of test results. AC functions are accessible for research and improvement at http://tangow.ii.uam.es/ac [12,19,20,21,22].

V. THE PROPOSED ALGORITHM

Since the algorithm is selected, it will be presented here. Implementation of the algorithm so as to facilitate the process of integrating this algorithm with the Moodle system will be explained in the next. The algorithm consists of three sections A, B and C.

A) Start for each document in the collection repeat

1) Clear all the blanks as an example of the extra spaces between words and clear the lines so that the text on one line as an example.
2) Removal of
    a. All the words confined between the A- Z, as well as figures confined between 0-9.
    b. All the symbols and signals, except for the words confined between the A- Z, as well as numbers confined between 0-9.
3) Conversion all text and symbols to-array.
4) Order of the words in the array in alphabetical order to facilitate the comparison process.

B) Part

1) Dictionary creation: Calculate the number of repeated words and numbers, ignore the repeating tokens.

Note: Steps A, B, applied to the answers (texts) of all students. End Repeat.

C) Comparison of two documents (Create an array for each document)

1) Count the number of words and symbols in each array and call this array as a token array.
2) Merge token arrays (token array contains words that appear in text)
3) Clear the duplicate words or symbols by keeping only one copy of each duplicate in the merged array.
4) Find total words and symbols in the array resulting from merge and duplicate removal operations.
5) Choose the least number as a basis for comparison.
6) Find the total of similar words and symbols by comparing merged (step 4) array with a token array with fewer tokens(step 5).
7) Find similarity is calculated dividing the number of tokens of an array with less number of token (obtained in step 5) by a number of tokens of a merged array (obtained in step 4).
8) Show results.

To find the percentage of similarity between first text and second text, we use the following “Eq. (1)”:

$$(X_i, X_{i+1}) = \frac{X_i}{(X_I + X_{i+1}) - K} *100 \qquad (1)$$

X_i = Number of tokens of i^{th} student answer.
K = Number of duplicate tokens in a resultant merged array of student’s *i* and *i+1* answers.

VI. PRACTICAL SECTION

In this part, the features of the code are explained, which includes Moodle program that is chosen as one of the main learning management programs in which it is used to combine code plagiarism detection and identify the variety of the folders in the Moodle program and feature of the way of each one of them. Furthermore, the stages of combining the algorithm, which is mentioned in the fourth quarter with program Model upping stages code are explained. In addition to that, the transforming them that will cause mod/quiz in the folders, particularly, in the renderer.php and reviewquestion.php. Remember that this code is combined with Moodle program for a school or for colleges and education. To make deeper research for the download and configure it to a school or a college, check this link https://moodle.org/. Since PHP is a translated language, the Moodle code is saved as source code files on the web. The PHP translator separates the code on the fly and the output is sent by the web server program when a specific file is offered on the server. The “M” in the Moodle means “Modular”, and its control structure shows that every one of the top-level folders indicates a general part of Moodle as explained before. Lots of the basic parts work with plugin modules. Every plugin owns its own folder inside the other part’s folder. In some circumstances, modules can work with additional plugins, as well. For example, from the last member’s thought, modules are put by copying the module to the suitable folder place on the web. Moodle finds the new module and later, when a manager signs in the program, places the module’s SQL code, runs it and lastly, shows the results. Updates are run same as Moddle, by following the database and updates the database if it is needed accordingly. This simple integration for last users may be difficult for the developer. This screenshot is a illustrate list of folders and files after installation of Moodle list of current Moodle installation. All guides in the basic folder will not be covered not, but the features of some significant folders benefited by developers who modifies Moodle will be explored. Moodle benefits from a basic terminology for modules, in which all modules are covered in their folder. Furthermore, the folder name is the name which Moodle shows in its integration when showing the module in “Fig. 1,” [23,24].

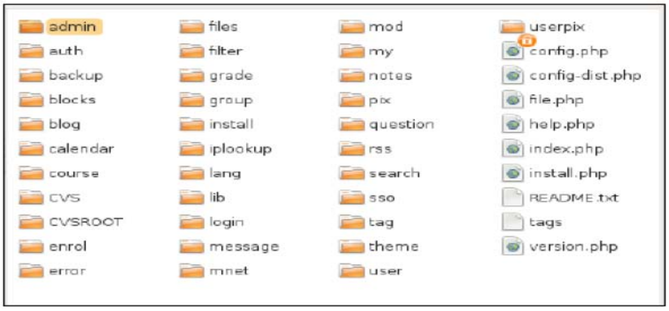


Fig. 1. Moodle Folders. *(23)*

VII. THE STAGES TO COMBINE THE CODE WITH THE PROGRAM

The algorithm is converted to php code. After that, the code is stored as similar.class.php file format and saved in the C:\xampp\htdocs\moodle\mod\quiz, as show in “Fig. 2,”

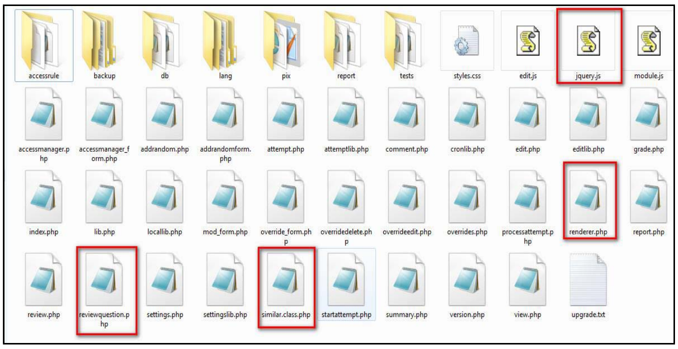


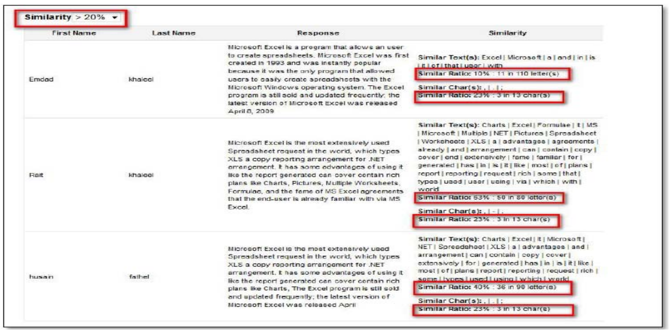
Fig. 2. The Contents of the File Quiz

The following is the changes that are applied on file renderer.php and reviewquestion.php.

- Write Code to bring information from the database.
- Writing code to get student’s answer.
- Writing code for the first and last name of the student's and the answers by analyzing the student‘s user name who has completed the exam.
- Write Code to find a similarity ratio.
- Write Code to create a table. The table is created with four columns, the first and second columns represent the student’s first and last name respectively, and the third one represents the answer of student and the last one for the similarity of the words and symbols.
- Adding jquery.js function in the path C: \ xampp \ htdocs \ moodle \ moodle \ mod \ quiz.It is one of the special functions in JavaScript. In order to control the rate of the similarity, it depends on the highest percentage of similarity between the words and symbols.
- Adding code that calls similar.class.

VIII. TESTING

This part demonstrates in which way all program visuals can be benefited by an instructor. The succession and shown results indicated that all basic program visuals are performing normally. Firstly, the instructor opens the Moodle Website, as an instance, all schools or other colleges benefit from it to discover plagiarism in the program. Then he/she encounters the home page of Moodle that includes transferable items like the names of the courses or the organization and the definition of the web page. To see the program, the instructor writes his/her own ID and password to sign. Then alternatively the user may click on the login button on the top of the page, he/she will be redirected to the login page; on this page, the teacher should enter his/her own username and password correctly. When the username and password are entered correctly, Moodle will redirect teacher to the main page which contains the names of the courses that he/she is teaching. When the teacher clicks on a course name, he/she will be redirected to the main page of the selected course. After clicking on the name of the selected course, the main page of that course will be displayed. This page contains course name, course participants, course quiz and other activities. Then the teacher will be able to click on the name of the available quiz (if any). The quiz page contains quiz description, the number of students who submitted quiz and the link to preview quiz. In order to view the submitted quiz, the teacher will be able to click on the link (Attempts). After clicking on the (Attempts), the teacher will be redirected to the quiz page that contains submitted the quiz, this page contains the names of students who submitted the quiz and some other information. When the (Requires grading) is clicked the developed plagiarism detection module will be triggered and the report will be generated so that the teacher will be able to view and detect plagiarism.



Similarity > 20% ▾

<table>
<tr><th>First Name</th><th>Last Name</th><th>Response</th><th>Similarity</th></tr>
<tr><td>Emdad</td><td>khaleel</td><td>Microsoft Excel is a program that allows an user to create spreadsheets. Microsoft Excel was first created in 1993 and was instantly popular because it was the only program that allowed users to easily create spreadsheets with the Microsoft Windows operating system. The Excel program is still sold and updated frequently; the latest version of Microsoft Excel was released April 8, 2009</td><td>Similar Text(s): Excel | Microsoft | a | and | in | is | it | of | that | user | with
Similar Ratio: 10% : 11 in 110 letter(s)
Similar Char(s): , | . | ;
Similar Ratio: 23% : 3 in 13 char(s)</td></tr>
<tr><td>Raft</td><td>khaleel</td><td>Microsoft Excel is the most extensively used Spreadsheet request in the world, which types XLS a copy reporting arrangement for .NET arrangement. It has some advantages of using it like the report generated can cover contain rich plans like Charts, Pictures, Multiple Worksheets, Formulae, and the fame of MS Excel agreements that the end-user is already familiar with via MS Excel.</td><td>Similar Text(s): Charts | Excel | Formulae | It | MS | Microsoft | Multiple | NET | Pictures | Spreadsheet | Worksheets | XLS | a | advantages | agreements | already | and | arrangement | can | contain | copy | cover | end | extensively | fame | familiar | for | generated | has | in | is | it | like | most | of | plans | report | reporting | request | rich | some | that | types | used | user | using | via | which | with | world
Similar Ratio: 63% : 50 in 80 letter(s)
Similar Char(s): , | - | .
Similar Ratio: 23% : 3 in 13 char(s)</td></tr>
<tr><td>husain</td><td>faihel</td><td>Microsoft Excel is the most extensively used Spreadsheet request in the world, which types XLS a copy reporting arrangement for .NET arrangement. It has some advantages of using it like the report generated can cover contain rich plans like Charts, The Excel program is still sold and updated frequently; the latest version of Microsoft Excel was released April</td><td>Similar Text(s): Charts | Excel | It | Microsoft | NET | Spreadsheet | XLS | a | advantages | and | arrangement | can | contain | copy | cover | extensively | for | generated | has | in | is | it | like | most | of | plans | report | reporting | request | rich | some | types | used | using | which | world
Similar Ratio: 40% : 36 in 90 letter(s)
Similar Char(s): , | . | ;
Similar Ratio: 23% : 3 in 13 char(s)</td></tr>
</table>

Fig. 3. Detect Plagiarism Page

Plagiarism detection page as show in “Fig.3,” consists of a table, which has four columns, the first and second columns contain student’s first and last name respectively, the third column contains the student’s answer and the last one for the similarity of the words and symbols. In addition, there is a drop-down list on the right top of the table used to control the ratio of similarity. After ratios of similarities is displayed to the teacher, the system also enables the instructors to write a review or grade about student’s submission currently being inspected. Then the user has to click on make comment or override mark link it will move to another page which contains boxes to put the degree and comment and then click on Save button. After writing a degree and clicking on save button, the student’s information page will be displayed. We notice that the paragraph (Requires grading) will be changed to a degree that has given by the teacher.

CONCLUSION

In this study, we aimed to enhance features of the Moodle. In this sense, plagiarism detection was developed and integrated into Moodle system. The results and error free application show that the work of proposed plagiarism system is compatible with Moodle system. An added feature with the Moodle lets the instructors find out the plagiarism between student’s replies. In this study, our main goal was to demonstrate the applicability of plagiarism detection system in Moodle environment. Therefore, the performance of plagiarism detection algorithm in terms of accuracy and error write are out of the scope of this study and left as future work.

REFERENCES

[1] J. Cole and H. Foster, Using Moodle: Teaching with the popular open source course management system: " O'Reilly Media, Inc.", 2007.

[2] M. Mostert and J. D. Snowball, "Where angels fear to tread: Online peer-assessment in a large first-year class," Assessment & Evaluation in Higher Education, vol. 38, pp. 674-686, 2013.

[3] Moodle homepage, https://moodle.net/stats/ , visited in January 2017.

[4] A. Brown and J. Johnson, "Five advantages of using a learning management system," Book Five Advantages of Using a Learning Management System, Series Five Advantages of Using a Learning Management System, Microburst Learning, 2007.

[5] A. J. Swart, "Student usage of a learning management system at an open distance learning institute: A case study in electrical engineering," International Journal of Electrical Engineering Education, vol. 52, pp. 142-154, 2015.

[6] P. Clough, "Plagiarism in natural and programming languages: an overview of current tools and technologies," 2000.

[7] J. D. Velásquez, Y. Covacevich, F. Molina, E. Marrese-Taylor, C. Rodríguez, and F. Bravo-Marquez, "DOCODE 3.0 (DOcument COpy DEtector): A system for plagiarism detection by applying an information fusion process from multiple documental data sources," Information Fusion, vol. 27, pp. 64-75, 2016.

[8] T. Lancaster and F. Culwin, "Using freely available tools to produce a partially automated plagiarism detection process," in Beyond the comfort zone: Proceedings of the 21st ASCILITE Conference, 2004, pp. 520-529.

[9] D. Kermek and M. Novak, "Process model improvement for source code plagiarism detection in student programming assignments," Informatics in Education-An International Journal, pp. 103-126, 2016.

[10] A. L. Foster, "Plagiarism-detection tool creates legal quandary," The chronicle of higher education, vol. 48, pp. A37-A38, 2002.

[11] I. Kostka and V. Maliborska, "Using Turnitin to Provide Feedback on L2 Writers' Texts," TESL-EJ, vol. 20, p. n2, 2016.

[12] D. Tresnawati and A. Syaichu, "Plagiarism Detection System Design for Programming Assignment in Virtual Classroom based on Moodle," Procedia-Social and Behavioral Sciences, vol. 67, pp. 114-122, 2012.

[13] K. W. Bowyer and L. O. Hall, "Experience using" MOSS" to detect cheating on programming assignments," in Frontiers in Education Conference, 1999. FIE'99. 29th Annual, 1999, pp. 13B3/18-13B3/22 vol. 3.

[14] J. Pierce and C. Zilles, "Investigating Student Plagiarism Patterns and Correlations to Grades," in Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, 2017, pp. 471-476.

[15] L. Prechelt, G. Malpohl, and M. Philippsen, "Finding plagiarisms among a set of programs with JPlag," J. UCS, vol. 8, pp. 1016-, 2002.

[16] Z. Tian, Q. Zheng, T. Liu, M. Fan, E. Zhuang, and Z. Yang, "Software plagiarism detection with birthmarks based on dynamic key instruction sequences," IEEE Transactions on Software Engineering, vol. 41, pp. 1217-1235, 2015.

[17] X. Chen, B. Francia, M. Li, B. Mckinnon, and A. Seker, "Shared information and program plagiarism detection," IEEE Transactions on Information Theory, vol. 50, pp. 1545-1551, 2004.

[18] A. Burkovski, S. Klenk, and G. Heidemann, "Similarity calculation with length delimiting dictionary distance," in Tools with Artificial Intelligence (ICTAI), 2011 23rd IEEE International Conference on, 2011, pp. 856-864.

[19] M. Freire, M. Cebrian, and E. del Rosal, "Uncovering plagiarism networks," arXiv preprint cs/0703136, 2007.

[20] P. Modiba, V. Pieterse, and B. Haskins, "Evaluating plagiarism detection software for introductory programming assignments," in Proceedings of the Computer Science Education Research Conference 2016, 2016, pp. 37-46.

[21] M. Freire, "Visualizing program similarity in the AC plagiarism detection system," in Proceedings of the working conference on Advanced visual interfaces, 2008, pp. 404-407.

[22] A. Abdul□Rahman, G. Roe, M. Olsen, C. Gladstone, R. Whaling, N. Cronk, et al., "Constructive visual analytics for text similarity detection," in Computer Graphics Forum, 2017, pp. 237-248.

[23] J. Moore, Moodle 1. 9 Multimedia Extension Development: Customize and Extend Moodle by Using Its Robust Plugin Systems: Packt Publishing Ltd, 2010.

[24] A. Marengo, A. Pagano, and A. Barbone, "Evaluation of Student Performance in Adaptive E-Learning Processes with Active Tutorship," International Journal of Technology Diffusion (IJTD), vol. 5, pp. 35-49, 2014.