WeCheat: Algorithm for E-learning Smart Cheating Detection Using Mean-Shift Clustering

Hussein Hazimeh[1, 4], Hasan Fayyad-Kazan[2], Hussin J. Hejase[3], Karl Daher[4], Elena Mugellini[4], and Omar Abou Khaled[4]

[1] Lebanese University, Faculty of Sciences, Beirut, Lebanon
hussein.hazimeh@ul.edu.lb
[2] Al Maaref University, Faculty of Business Administration, Information Technology, Beirut, Lebanon
hasan.kazan@mu.edu.lb
[3] IEEE Senior Member, Senior Researcher, Professor of Business Administration, Beirut, Lebanon
[4] HumanTech Institute, University of Applied Sciences of Western Switzerland, Fribourg, Switzerland
firstname.lastname@hes-so.ch

Abstract. During the COVID-19 pandemic, most people around the world started using internet and online technologies for multiple purposes. A very common digital transformation is online learning or e-learning. Despite its importance, online learning has several issues compared to standard learning. One of the most issues of concern is cheating between students during online exams. This paper aims to address cheating problem by proposing a novel method called WeCheat. It automates the cheating detection process on e-learning platforms, such as Google Forms. The target for the detection of cheating-groups are online exams. To the best of our knowledge, this is the first time someone proposes such a systemthat detects cheating, particularly during the COVID-19 pandemic. To comply with the set goal, a clustering-based solution is proposed. This Method employs Mean-Shift, a non-parametric and density-based clustering algorithm for this task.Validation of this approach is evidenced by the application which performs excellently on an arbitrary numberof features in the cheating detection problem in terms of accuracy. This application helps educational institutions to address cheating cases and at the same time offers the opportunity to focus more on the use of anti-cheating logistics.

Keywords: E-Learning, cheating detection, machine learning, clustering, mean-shift, COVID-19

1. Introduction

E-learning is another approach for delivering knowledge to students. According to [16], "*E-learning is part of the new dynamic that characterizes educational systems at the start of the 21st century.*" Despite the typical learning approaches that require students to attend their classes and exams inside schools or universities. E-learning has been emerged in the 1990s upon the emergence of computer and internet technologies, to be adopted in parallel with classical learning. Recently, during the Covid-19 pandemic that struck the whole world, e-learning has a considerable impact on all aspects especially the field of education [8, 10]. Some schools that have used information technology as learning media are required to have high creativity and adequate knowledge of information technology. Testing as an assessment of students' learning during a pandemic is conducted remotely so that online self-exams and web-based are required. The classical on-campus test system was changed to online. In this context, online learning has several issues compared to typical learning. Generally, students use social and instant messaging applications to circulate exam answers across each other. Recently, a number of solutions have been proposed in the literature to address this problem [15] and [9]. To detect this problem, instructors usually use the manual comparison of exam answers, which is a time-consuming process and need a lot of effort. In this context, this research introduces a new method called 'WeCheat' that automates the cheating detection process between university students who submit their exams to

Google Forms (http://forms.google.com).
The process starts by collecting data from Google forms (i.e., answers from students), then a number of preprocessing steps are applied to clean and normalize the data on hand; mainly stop-words and special characters removal. The aforementioned is a very important step before running any supervised/unsupervised learning algorithm. The next step is to develop a new clustering-based method to find the cheating groups in a certain exam. Finally, an evaluation of the proposed method is carried out on real-world datasets to assess its high accuracy. The contribution of this paper is evidenced by introducing a novel approach to address the cheating detection problem on e-learning platforms, mainly Google Forms.
The rest of this paper is organized as follows: Section 2, a literature review is carried out to create a theoretical background on this domain. Section 3 defines the problem. While section 4 introduces the 'WeCheat' system's architecture. Section 5 presents the experimental results of the proposed method, and finally, section 6 depicts the conclusion and mentions the limitations and future work.

2. Literature Review

In this section, we introduce an overview of the recent literature work in the context of cheating detection systems and educational data mining. In addition, we list a number of papers that use Mean-Shift algorithm in theirclustering problem.

2.1 Automatic cheating detection systems

Kamalov, Sulieman, & Santandreu [14] examined a new approach to detect cheating on the final exam using machine learning techniques. The authors proposed the use of a post-exam analysis of the student grades. The proposed method took into "account student grades prior to the final exam, grades on the final exam, and the overall performance of the class to make a decision" (p. 12). Actually, to identify the potential cases of cheating, the researchers employed long short-term memory networks (LSTMs) in combination with a Kernel density estimation (KDE)-based outlier detection technique. Results showed significant performance as compared to the benchmark methods used in the experiments. Nevertheless, "the use of baseline algorithms have the added burden of learning the importance of the final exam scores from the data. So the comparison may not be entirely adequate" (p. 13).
Chen & Chen [5] introduced an approach to find cheating groups in multiple-choice questions. They leveraged principle component analysis and hierarchal clustering to reach their goal. However, this approach considered only multiple-choice questions, without involving open-answers in their study.
Li, Zhu, & Yang [15] proposed a cheating-detection system to identify potential cheating students, based on Expectation-Maximization (EM) algorithm and neural network. They achieved accuracy results 79.4%. However, this paper as well did not consider the correlation between students during exam cheating.
Balderas & Caballero-Hernández [1] proposed a system called 'Py-Cheat' to detect potential cheating amongstudents during the online examination. They collected their data from 103 students' exams and analyzed the cheating using Python. However, the

authorsdid not mention which algorithm they used to reach this goal. In the results, the authors compare the students' marks before and after online examinations.

Cavalcanti, et al. [3] analyzed the cheating between students using document similarity. They used cosine similarity to find the closeness of answers between students. Finally, they employed decision-tree supervised-based machine learning algorithm to classify if a student is cheating or not. However, the authors relied on labelling if the student is cheating or not, without focusing on whom is this student cheating.

Best & Shelley [2] studied the contribution of social media platforms like Facebook, Twitter, Snap Chat/Instagram, Texting and various smartphone applications on academic dishonesty in higher education. The authors found that although "students report utilizing these forms of social media to assist with their studies most do not use these applications for cheating or any form of academic dishonesty. Nevertheless, there was an increased willingness to use texting, screenshots, video and audio recordings to cheat on exams and other academic requirements" (p. 1).

Hogan & Jaska [11] proposed a survey-based analysis to assess the ethical behavior of junior and senior-level Computer Information Systems students. They collected data from 300 students. "The results are analyzed based upon student classification, grade point average, and gender. Indications are that seniors, students with lower grade point averages, and males have a higher propensity to engage in academic dishonest behavior" (p. 169).

Chau, Loc, & Tran [4] proposed a mean-shift based clustering algorithm named 'iMS_nps' using the nearest prototype strategy to solve the problem of the incompleteness of the educational data gathered in an academic credit system. The authors' solution was characterized with better cluster quality when compared to other approaches.

Esnaashari, Gardner, & Watters [9] mainly studied the correlation between in-class participation and its contribution to the overall class performance. The authors combined "in-class and out-of-class data with a range of qualitative and quantitative self-report measures. Then used a range of data mining (DM) algorithms to predict final course outcomes" (p. 313). Results show that more in-class participation leads to better performance.

Various researchers have widely used Mean-Shift as density-based clustering algorithms mainlyin the domain of object detection and tracking in videos [13].

2.2 Limitations of related work

Despite all the aforementioned approaches and systems in the context of cheating detection or academic dishonesty analysis. None of these works tackled the problem of finding groups of students who cheat together in an exam. Moreover, none of them addressed the cheating problem on exams submitted fully online (electronic submissions such as Google Forms). In the next section, we introduce WeCheat, a new method to overcome the shortcomings of the related work.

3. Materials and Methods

3.1 Problem formulation

An exam is represented by *E*, while a student submits the exam by *S*. Moreover, a student's identity is denoted by *Si*, whereby the first name and last name define the identity of the student. Furthermore, the student' answer to an exam is denoted by *Aj*, such that, the question number is *j*. Let the student's overall exam answers be grouped in a vector *Vi*. Then, a student *x* with answers of an exam E is represented by the vector:

$$V^E = [A_1, A_2, \ldots, A_j].$$

Next, the assumption of cheating is represented by groups of students who likely cheated on each other. Certainly, one student alone is not considered a cheating case, but cheating groups start with 2 or more students. Consequently, the cheating detection process deals with groups of cheating students. Then, R_c denotes the cheating vector,

$$R_c = [\text{"X, Y, Z","U","W"}].$$

Where, "X, Y, & Z" denotes a group of three studentscheating with each other, whereas U & W are non-cheating individual students.

'WeCheat' System Architecture

Fig. 1 depicts an illustration of the proposed 'WeCheat' system architecture.

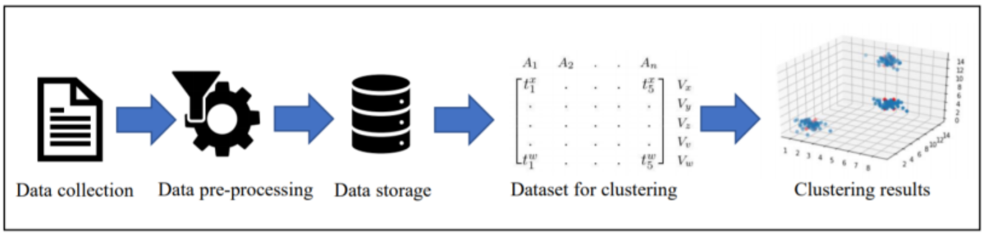

Fig. 1. WeCheat system architecture

'WeCheat' system has five (5) stages as follows:
Stage 1. Data are collected from 'Google Forms' and stored into a database.
Stage 2. Pre-processing tasks are carried out to prepare data for analysis (data cleaning).
Stage 3. Dataset is stored
Stage 4.The dataset is treated with 'WeCheat'. Whereby a set of rules are applied using machine learning algorithms. The aim is to detect the cheating cases in an exam.
Stage 5. Cheating detection results are obtained

3.2 Data Collection

'WeCheat' aims to analyze and detect cheating cases in students' responses collected from a University e-learning exam. As a matter of fact, in a selected university, exams are administered via Google Classroom, mainly using Google Forms format. When all students finish their exams, these are collected and stored in a database. There are two methods to collect the responses from Google Forms, these are as follows:

1. Having an application programming interface like 'Google API' which is based on Restful calls and returns array data. However, a subscription is required for such API.
2. Accessing the web pages from browsers using a web crawler tool.

3.3 Data Pre-Processing

Following the collection task, data retention is performed in a database repository, in this case, inside the MySQL database. Students' submitted exams are designed using different questions formats, including: True or False (T/F), Multiple Choice (MCQ), and Open Answers (OA). Responses pertaining to the MCQ and T/F formats do not undergo cleaning. While, the responses to OA questions do require pre- processing such as: stop word removal, stemming, lemmatization, and special character removal.

Table 1. Exam answers for different students composed of two categories.

	(1) Rule-based Approach		**(2) String Similarity Approach**
	Option-Based Answers		**Open-Based Answers**
Students	**MCQ Answers**	T/F Answers	**Open/Free text answers (Showing *the first answer only*)**
X	A, A, B, C, A, B	T, T, T, T, F, F	"Computer 1, due to its newer core, has more RAM and a solid drive that is stable"
Y	A, C, B, C, B, B	T, F, F, T, F, F	"Computer 2, even though computer 1 has a faster and more reliable SSD. For the main purpose of storing large amount of data a 1TB HDD PC with lower specifications can fulfill this particular task more than a 240GB SSD."
Z	C, A, B, C, B, B	T, T, F, T, F, F	"Computer 1. Its CPU is quad core, with RAM capacity that is bigger than Computer 2 – SSD isbetter than HDD"

3.5 Analytical Modelling

Matrix Representation

This next step is to use linear algebra, matrices, to model the gathered students' answers. In fact, there is a need for two matrices, M_a and M_b. In matrix M_a, all the 'option-based' answers are represented such as multiple-choice and true or false questions. In matrix M_b, all the 'open-answer' questions are set. Then, matrix M_a depicts 'single-valued' data, for example: a, b, c, t, or f, such that ‘a, b, and c’ are answers from a multiple-choice question, and ‘t & f’ are answers from true or false questions.

$$
\mathbf{M_a} = \begin{array}{c} \begin{array}{ccccc} A_1 & A_2 & . & . & A_n \end{array} \\ \begin{bmatrix} a & b & c & t & f \\ a & b & c & t & f \\ a & b & c & t & f \\ b & a & c & t & t \\ b & a & c & t & t \end{bmatrix} \end{array} \begin{array}{c} \\ V_x \\ V_y \\ V_z \\ V_v \\ V_w \end{array} \quad \mathbf{M_b} = \begin{array}{c} \begin{array}{ccccc} A_1 & A_2 & . & . & A_n \end{array} \\ \begin{bmatrix} t_1^x & . & . & . & t_5^x \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ t_1^w & . & . & . & t_5^w \end{bmatrix} \end{array} \begin{array}{c} \\ V_x \\ V_y \\ V_z \\ V_v \\ V_w \end{array}
$$

Matrix M_a (1) includes five students’ answers. In fact, the following observations may be inferred from it: (a) students ‘x, y, and z’ answered rightfully the ‘true or false’ questions. (b) Students ‘v & w’ had exactly the same false answers, that is, having the same mistakes. The mistakes are common across all the five questions. Further, Matrix M_b (2), depicts the same cases, that is five students and answers to five exams, reported in Matrix M_a but in textual form. Actually, Matrix M_b includes the responses of the five ‘open-answer’ questions. The term t_n^x depicts the answer r of student x in question n. Moreover, it is necessary to transform the textual data to numerical format in order to handle the textual data in the analysis phase. All Matrix M_b (2) data is textual i.e., ‘open-answer’ questions. Consequently, we propose a text encoding method for textual data as shown next.

Text encoding algorithm

The clustering phase necessitates that the data is normalized before its use. Then a common format will be used for data transformation. That is, textual to numerical and this happens by encoding the text by numbers in M_b. As a matter of fact, all the words in the matrix are stored in a ‘term-code’ vector. For example, assume there are two open-answers for question x for two students, t_1^x = “A Monitor Is Classified As Hardware” & t_2^x = “Mouse Is A Hardware”. Fig. 2 shows the ‘term-code’ vector for the two questions. Reading the encoded data in Fig. 2, the answers are t_1^x= 123456 & t_2^x = 7316.

Index terms

a	mouse	is	example	of	hardware	keyboard

Term-code vector

1	2	3	4	5	6	7

Fig. 2. Exhibit of a term-code vectors

Encoded Open-answers Matrix

t_1^x	1	2	3	4	5	6
t_2^x	7	3	1	6	0	0
t_3^x	5	4	8	1	5	2
t_4^x	8	3	1	7	0	0

Encoded Open-answers Matrix (Ordered)

1	2	3	4	5	6	0	0
1	0	3	0	0	6	7	0
1	2	0	4	5	0	0	8
1	0	3	0	0	0	7	8

$$
M_b = \begin{array}{c} \begin{array}{ccccc} A_1 & A_2 & . & . & A_n \end{array} \\ \begin{bmatrix} t_1^x & . & . & . & t_5^x \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ t_1^w & . & . & . & t_5^w \end{bmatrix} \end{array} \begin{bmatrix} V_x \\ V_y \\ V_z \\ V_u \\ V_w \end{bmatrix}
$$

Fig. 3. Term-Swapping and final matrix production

All the ‘open-answers’ for every student have corresponding ‘term-vector’, which are later depicted in a matrix created to handle them. Fig. 3 shows the resultant encoded ‘open-answers’ matrix. This matrix consists of elements of unordered values. For example, the first two rows, depicting the first two answers contain common numbers namely, ‘1, 3, & 6’. Nevertheless, data analysis requires that data should be in a clustering model which is based on a distance comparison metric, though swapping of the codes is performed before being analyzed. In order to deal with this issue, the following steps are carried out: (1) Common numbers are swapped according to their occurrence in columns, that is, if a student's answer t_1^x has common 'term-codes' with another students' answer t_2^x, these common terms are organized as depicted in Figure 4 in the encoded 'open-answers' matrix (Ordered). (2) The next step is replacing the missed codes with zeros. (3) Finally, filling these values in M_b is carried out, however, the size of the matrix will change commensurate with the number of terms in the 'open-answer' set. That is, if eight (8) represents the maximum number of terms in the answer t_1^x, then there should be eight (8) columns for t_2^x in M_b. The remaining answers follow suit.

3.6 Why we apply Mean-Shift?

This study aims to detect cheating groups in a certain academic course/class by using a clustering approach. This task is actually an unsupervised-learning problem. There is no need to have a training set as well as labels (classes). The objective is to divide students into groups (clusters). Before selecting the clustering algorithm, the following characteristics of this study task are defined:
(1) The existing dataset is non-parametric because the attribute's size increased/decreased according to the number of questions in a certain exam.
(2) The number of clusters is unknown in this task. For instance, each answer-form is considered as an individual cluster.
(3) As mentioned in (2), initially each answer is treated as an individual cluster. Then as a result, (i) The number of clusters is unknown, (ii) The attributes are non-parametric, and (iii) Each answer is considered as a single group. We considered using the Mean-Shift clustering algorithm for the cheating detection problem.
(4) Despite its limitation of scalability on high datasets, Mean-Shift is adopted in this work due to the small-scale data size, typically 10-50 samples (students). Mean-Shift is a non-parametric clustering algorithm [6, 7], i.e., it can be fully implemented in the current task with arbitrary parameters in multi-dimensional feature space. Mean-Shift works by shifting the mean of high-density clusters towards the mode [6], iteratively, until finding the centroid of each cluster.
In the cheating detection problem, the Mean-Shift algorithm will iterate to seek for the highest Gaussian Kernels, i.e., Kernel Density Estimation (KDE). Each cheating group will be estimated as a kernel by KDE, and therefore the number of clusters will be equal to the number of estimated kernels. The data entered into the clustering model is composed of the two integrated matrices M_a and M_b. In M_b, each answer is replaced by its term-vector as described in the 'Matrix Representation' / Text encoding algorithm. The eventual clustering dataset is illustrated in Figure 3.

4. Experimental Results

All experiments are conducted on Google Classroom (GCR) forms. The experiments are conducted on different courses, such as computer science, management information systems, and humanities.
The dataset used in this paper is analyzed in Table 2.

Table 2. Dataset statistics

Number of exams analysed	120
Number of courses	3
Total number of answers analyzed	7043

The source code of 'WeCheat' is available in the following link
https://github.com/HusseinHazimeh/Cheating

4.1 Data Collection

All the data were collected from GCR. Fully online learning took place at the University during the academic semesters of spring 2020 and fall 2021. All students were registered in GCR classes. The partial and final exams, as well as quizzes, were conducted on Google Forms. All answers data were collected using Selenium Web Crawler [12].

4.2 Bandwidth comparison

The next step is to compare the research's results in terms of the size of the bandwidth parameter. Mean-

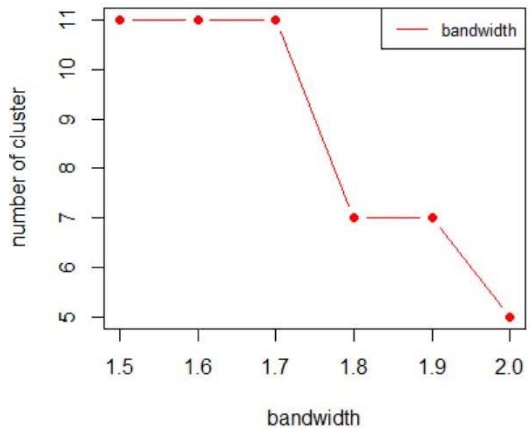

Fig. 4. Number of clusters obtained by different bandwidth values.

Shift algorithm is run on different bandwidth values. The objective of this analysis is to optimize the algorithm used in selecting a better value for the task on hand. Mainly, the number of data elements in this research is not big, i.e., between 10 to 40 students roughly. Six different bandwidth values were chosen between 1.5 and 2.0. As a matter of fact, Fig. 4 concludes that when the bandwidth value increases, the number of clusters decreases.

4.3 Analysis per course

Table 3 depicts the clustering results in three courses. Each course has a specific number of students who participated in the examination ('number of students' row in the table). The results are analyzed on different bandwidth values. According to the results, if the number of clusters for a specific course, take for example "Intro to Computing", is 23, this means that there are 23 different estimated densities.

Table 3. The number of clusters in each class given the value of bandwidth

		Course Name		
		Intro to Computing	Intro to Programming 1	Business Software Applications
No. of students		40	19	40
		No. of clusters		
Bandwidth	1.5	11	8	15
	1.7	11	8	14
	1.9	7	8	14
	2.0	5	7	9

However, a particular cluster, in fact, may include a minimum and maximum students between

1 and n. If the cluster comprises one student, this indicates that this student is not cheating, i.e., it does not have any similar answers with the rest of the students.

4.4 Analysis per cluster

Based on Fig. 5. It is concluded in this study that the average number of students cheating with each other is two.

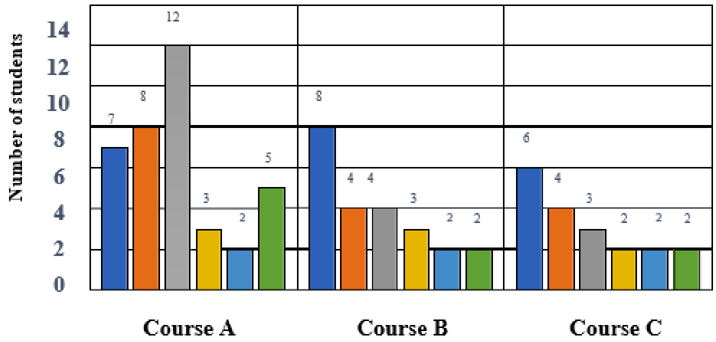

Fig. 5. The number of cheating students inside top clusters in three courses (band- width=1.0).

4.5 Baseline comparison

In this part, we evaluated the results according to a specific baseline. The objective is to find the accuracy of the proposed method. To reach this goal, the pre-available cheating results in each course are compared with the results of 'WeCheat'. The pre-available results are found by comparing the exams manually in each course by every instructor. Therefore, each instructor is aware of the cheating group(s) in his/her class. The results of this analysis shown in Table 4 are very promising. Actually, 'WeCheat' algorithm can successfully find the cheating groups in a certain course, with an accuracy of 91.5% based on the results analyzed of three courses.

Table 4. Accuracy per cluster obtained by 'WeCheat'

	Actual number of top clusters in each course		
	A	B	C
'WeCheat' :number of cheating students in each course	3	4	3
	3	3	3
Accuracy	1.0	0.75	1.0
Average accuracy		**0.916**	

All the results involved in the baseline benchmarking are related to the partial exams of all the three courses.

5. Conclusions and Future Work

In this paper, a novel detection method is introduced that addresses the problem of cheating among students in online exams conducted on E-learning platforms, such as Google Forms. Due to the COVID-19 pandemic, most universities and schools around the world were obliged to go digital and therefore using E-learning tools. One of the most popular E-learning academic tools is Google Classroom (GCR). However, despite its importance and simplicity, GCR does not have a feature that allows instructors to detect cheating among students when doing an online exam via a google form. In this context, this research introduced a novel method "WeCheat" that uses a density-based clustering algorithm.
Several evaluation experiments were carried out on different courses taking as samples the partial exams collected from Google Forms. Findings of this paper support that the proposed

method performed excellently with very high accuracy values. However, this approach still has a number of limitations that will be addressed in future work. These limitations include (1) the topic of the exam can affect the overall accuracy, because answers in mathematics, physics, and other calculation-based exams, contain a lot of formulas and calculations that need a specific model to manipulate them, (2) we hypothesized that the exam structure is organized similarly to Google Forms structure, i.e., based on multiple-choice, true/false, and open-answer questions. In this context, we propose to extend this method to address the aforementioned limitations as future work.

References

1. Balderas, A., & Caballero-Hernández, J.A. (2020).“Analysis of Learning Records to Detect Student Cheating on Online Exams: Case Study during COVID-19 Pandemic.” *International Conference Proceeding Series (ICPS): TEEM'20* (pp. 752–757). https://doi.org/10.1145/3434780.3436662
2. Best, L.M., & Shelley, D.J. (2018). “Academic Dishonesty: Does Social Media Allow for Increased and More Sophisticated Levels of Student Cheating?” *Int. J. Inf. Commun. Technol. Educ., 14*(3), 1-14.
3. Cavalcanti, E.R., et al. (2012). “Detection and Evaluation of Cheating on College Exams using Supervised Classification”. *Informatics Educ., 11*(2), 169-190. DOI:10.15388/INFEDU.2012.09
4. Chau, V.T.N, Loc, P.H., & Tran, V.T.N. (2015). “A Robust Mean Shift-Based Approach to Effectively Clustering Incomplete Educational Data.” ACOMP (pp. 12-19), Nov. 23-25, 2015. Ho Chi Minh City, Vietnam.
5. Chen, M., & Chen, C. (2017). “Detect Exam Cheating Pattern by Data Mining.” In: Antonio J. Tallón-Ballesteros, Kaicheng Li (Eds.). Fuzzy Systems and Data Mining III - Proceedings of FSDM 2017, Hualien, Taiwan, November 2017. *Frontiers in Artificial Intelligence and Applications* (pp. 25-34). IOS Press 2017, ISBN 978-1-61499-827-3.
6. Comaniciu, D., & Meer, P. (1999). “Mean shift analysis and applications”, *Proceedings of the Seventh IEEE International Conference on Computer Vision* (ICCV), Vol 2, (pp. 1197-1203). September 20-27, 1999. Kerkyra, Greece. DOI: 10.1109/ICCV.1999.790416.
7. Comaniciu, D., & Meer, P. (2002). “Mean Shift: A Robust Approach Toward Feature Space Analysis.” IEEE Trans. *Pattern Anal. Mach. Intell., 24*, 603-619. DOI:10.1109/34.1000236
8. EL Zein, Ahmad, et al. (2021). “Distance learning emergence as an outcome of Pandemic: Case Study for Lebanese Universities.” *International Journal of Education and Research, 9*(4), 35-48.
9. Esnaashari, S., Gardner, L.A., & Watters, P.A. (2018). “Clustering Student Participation: Implications for Education.” *WAINA* (pp. 313-318), May 16 - 18 2018. Krakow, Poland. DOI: 10.1109/WAINA.2018.00104
10. Hejase, H.J., & Chehimi, G.M. (2020). “E-learning: What to look for amid the Pandemic.” *Journal of Economics and Economic Education Research, 21*(Special Issue 1), 1-4. DOI: 10.13140/RG.2.2.17011.35369.
11. Hogan, P. T. & Jaska, P. V. (2000). “Academic Dishonesty: A Study of CIS Student Cheating Behavior.” *Journal of Information Systems Education, 11*(3-4), 169-178.
12. Holmes, A., & Kellogg, M. (2006). “Automating functional tests using Selenium.” *Proceedings of the Agile Development Conference* (AGILE'06), July 23-28, 2006. Minneapolis, MN, USA (pp. 275-287). DOI: 10.1109/AGILE.2006.19.
13. Hui, Q. (2019). “Motion video tracking technology in sports training based on Mean-Shift algorithm.” *J. Supercomputing, 75*(9), 6021-6037. DOI: 10.1007/s11227-019-02898-3. https://doi.org/10.1007/s11227-019-02898-3
14. Kamalov, F., Sulieman, H., &Santandreu, Calonge D. (2021). “Machine learning based approach to exam cheating detection.” *PLoS ONE 16*(8): e0254340. https://doi.org/10.1371/journal.pone.0254340
15. Li, Zhizhuang, Zhu, Zhengzhou, & Yang, Teng (2019). “A Multi-index Examination Cheating Detection Method Based on Neural Network.” *IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, November 4-6, 2019. Portland, OR, USA. DOI: 10.1109/ICTAI.2019.00086
16. Sangrà, Albert, Vlachopoulos, Dimitrios, & Cabrera, Nati (2012). “Building an Inclusive Definition of E-Learning: An Approach to the Conceptual Framework.” International Review of Research in Open and Distributed Learning, 13(2), 145–159. DOI: 10.19173/irrodl.v13i2.1161