

Trazabilidad de actividades de aprendizaje en cursos de programación de computadores usando un juez automático en línea dentro de un LMS

Andrés Felipe Pineda¹ and Julián Moreno Cadavid²

¹⁻²DDepartamento de Ciencias de la Computación y de la Decisión, Universidad Nacional de Colombia
{afpinedac, jmoreno1}@unal.edu.co

Abstract. As many teachers may know, the only way of actually learning computer programming is through practice. Online judges may help in this task because students can practice as much as they want and receive automatic feedback about their codes correctness. Teachers, on the other side, may spend less time checking codes, and more on tutoring and advising. When embedded into a LMS, those tools might become even more helpful because the data they provide can be used for assessment purposes but also to feed learning analytics processes. Considering this scenario, we propose in this paper a process mining approach to use those data and identify student behavioral patterns. In particular, we used the Disco tool to identify interaction patterns using codes submission data as well as LMS resources access records. In order to validate our proposal, a case study was conducted in a data structures course with 96 engineering students at the National University of Colombia, in Medellín. We found well-differentiated patterns, from those that exhibit a sequential or incremental pattern, to those that reveal trial-error tactics.

Resumen. Como muchos docentes deben saber, la única forma de aprender verdaderamente programación de computadoras es a través de la práctica. Los jueces en línea pueden ayudar en esta tarea porque los estudiantes pueden practicar todo lo que quieran y recibir retroalimentación automática sobre la corrección de sus códigos. Los docentes, por otro lado, pueden pasar menos tiempo revisando los códigos y más en tutorías y ayuda personalizada. Cuando se integran en un LMS, esas herramientas pueden ser aún más útiles porque los datos que proporcionan pueden utilizarse para fines de evaluación, pero también para alimentar los procesos de analítica de aprendizaje. Teniendo en cuenta este escenario, proponemos en este documento un enfoque de minería de procesos para utilizar esos datos e identificar los patrones de comportamiento de los estudiantes. En particular, utilizamos la herramienta Disco para identificar patrones de interacción utilizando datos de envío de códigos, así como registros de acceso a recursos dentro del LMS. Para validar nuestra propuesta, se realizó un estudio de caso en un curso de estructuras de datos con 96 estudiantes de ingeniería en la Universidad Nacional de Colombia en Medellín. Encontramos patrones bien diferenciados, desde aquellos que exhiben un patrón secuencial o incremental, hasta aquellos que revelan tácticas de prueba-error.

Palabras clave: Analíticas de aprendizaje, Cursos de programación, Minería de procesos, E-Learning, ACM-ICPC

1 Introducción

La enseñanza y el aprendizaje de la programación es considerado uno de los grandes desafíos en las ciencias de la computación [1], aun así, existe una gran cantidad de evidencia acerca del bajo rendimiento que los estudiantes universitarios pueden tener en los cursos básicos de programación durante sus primeros años [2].

Distintos autores mencionan que el único modo para mejorar las habilidades de programación es a través de la práctica [3][4], y una de las principales herramientas usadas en la actualidad para fomentarla son los jueces en línea, los cuales son programas que permiten evaluar los algoritmos realizados por los estudiantes brindando información en tiempo real.

Son varias las ventajas que puede tener el adoptar un sistema de estos como apoyo dentro de un curso de programación. Por parte del profesor, la calificación automática le permite ahorrar tiempo y así invertirlo en otras tareas dentro de la clase [5]. Mientras que por parte del estudiante le permite recibir retroalimentación inmediata, mantener estadísticas de su progreso a lo largo del curso, al tiempo que lo exige a ser mucho más riguroso al momento de diseñar los algoritmos [6].

Considerando este panorama, en este artículo se explora el potencial del uso de un juez en línea embebido dentro de un LMS, pero sobre todo, cómo la información generada puede servir para realizar una analítica del aprendizaje. En particular, se busca estudiar la trazabilidad de las actividades realizadas por los estudiantes y la incidencia que sobre estas tienen las realimentaciones brindadas por el juez. Siendo así, el resto del artículo está organizado de la siguiente manera: En la sección 2 se presentan trabajos relacionados con la temática de investigación, en la sección 3 se presenta la propuesta realizada. En la sección 4 y 5 se presentan los materiales y el método de experimentación. Por último, se presentan las conclusiones del trabajo.

2 Trabajos relacionados

Al hacer una búsqueda de investigaciones relacionadas con procesos de analítica de aprendizaje en cursos de programación que incorporen juez en línea, no se encontraron trabajos que consideraran esos dos elementos de manera integrada. En cambio, si se encontraron varios que los abordan de manera aislada.

En el caso de la analítica de aprendizaje por ejemplo, en [7] hacen uso de técnicas de minería de datos, específicamente árboles de decisión, en un curso de estructuras de datos con 70 estudiantes, con el fin de clasificar su rendimiento en 'Bueno', 'Promedio', y 'Bajo'. Para ello, recolectan información generada por los de los estudiantes a través de un LMS y luego es analizada en el software Weka con el fin de crear el modelo que permita realizar dicha clasificación.

En [8] presentan los resultados que obtuvieron luego de aplicar un proceso de analítica de aprendizaje en dos cursos, uno de introducción a la programación y otro de estructuras de datos. El objetivo de la investigación era encontrar relaciones entre variables que hacen más efectivo el rendimiento de los estudiantes en estos cursos, para ello, hicieron uso de la información generada por el LMS utilizado y realizaron un análisis basado en reglas de asociación. En la investigación concluyen que el horario de estudio, el número de accesos y el número total de ejercicios realizados de practica son los principales indicadores del rendimiento de los estudiantes.

Ya en el caso del uso de jueces en línea, en [9] presentan una aproximación llamada OJPOT (Online Judge & Practice Oriented Teaching) el cual es aplicado en un curso de fundamentos de programación, y que consiste en hacer uso de este tipo de herramientas con el fin de promover las habilidades de diseño y verificación de algoritmos. Se realiza un estudio empírico con 84 estudiantes mediante un grupo de intervención y un grupo de control. El grupo de intervenido usó el juez mientras que el otro hizo siguió el esquema de enseñanza “tradicional”. Luego de realizar un pre y post test para evaluar el efecto de la intervención, los autores concluyen que el rendimiento del grupo intervenido fue muy superior con respecto al grupo de control.

En [6] afirman que hay dos principales razones por las cuales los estudiantes no terminan sus carreras profesionales en el área de la computación, primero por la complejidad de esta y la segunda es la falta de motivación en los estudiantes. Intentando resolver esta problemática, los autores muestran la experiencia en la Universidad de Murcia en España haciendo uso de un juez en línea y concluyen que con ello lograron bajar la tasa de deserción de un 70% a 40% en dos años.

3 Propuesta

Retomando lo mencionado en la introducción, lo que busca esta propuesta es estudiar la trazabilidad de las actividades de aprendizaje realizadas por estudiantes en cursos de programación y la incidencia que sobre estas tienen las realimentaciones brindadas por un juez automático de códigos. Para ello, lo primero que hay que entender es el funcionamiento y las respuestas brindadas por dicho juez. Cuando se sigue el estándar de ACM-ICPC, un ejercicio de programación se estructura tal como aparece en la Fig. 1. De esta manera, lo que se espera del estudiante es que diseñe un algoritmo y posteriormente produzca un código en un lenguaje de programación específico que le dé solución al problema planteado.

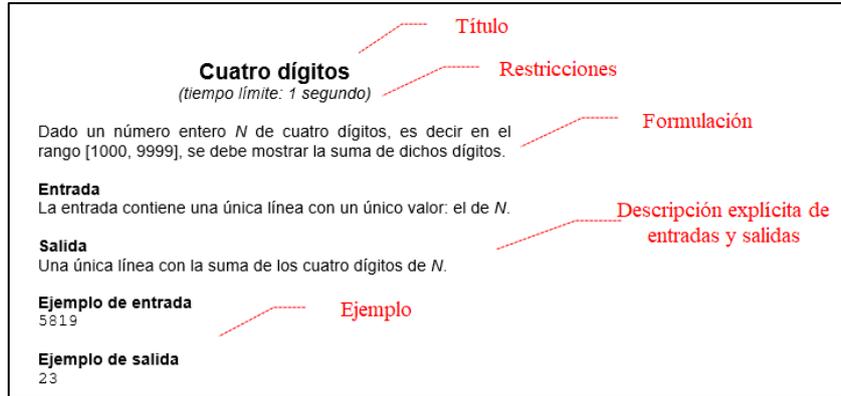


Fig. 1. Ejemplo de ejercicio de tipo ACM-ICPC

Al recibir un código, el juez lleva a cabo un proceso secuencial de la siguiente manera. Primero, intenta compilar el código. Si la compilación no es exitosa, la realimentación brindada será “*Compilation Error*” seguido de la información correspondiente (falta de un punto y coma, por ejemplo), en caso contrario procederá a ejecutarlo. Si tal ejecución genera un error, la realimentación será ‘*Runtime Error*’ seguida igualmente por la información complementaria (división por cero, por ejemplo), en caso contrario procederá a evaluar la respuesta obtenida. Para ello, el juez internamente tiene un *dataset* de entradas y salidas que cumple con el formato establecidos, pero que generalmente es diferente, la mayoría de las veces mucho menos trivial, que el ejemplo presentado en el ejercicio. En caso que el tiempo al ejecutar tal *dataset* sea mayor al permitido, la realimentación brindada será “*Time Limit Exceeded*”, en caso contrario será “*Wrong Answer*” o “*Accepted*” dependiendo si resuelve el problema o no de manera exitosa. En el primer caso, se suele acompañar el mensaje con el tiempo de ejecución para que el estudiante tenga como referencia la eficiencia algorítmica de su solución. Un resumen de dicho proceso se esquematiza en la Fig. 2.

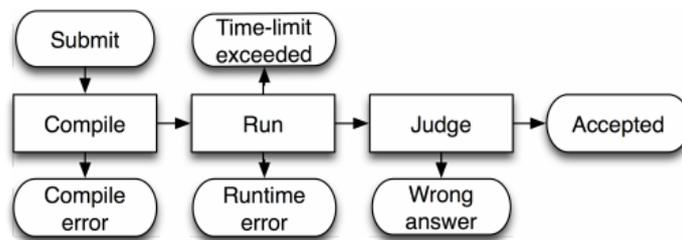


Fig. 2. Proceso de evaluación de un ejercicio en el juez en línea (tomado de [10])

Considerando lo anterior, definimos en este contexto una actividad de aprendizaje dentro del LMS como cualquier interacción, o secuencia de interacciones, bien sea con los materiales de estudio (PDFs y otros archivos provistos por el profesor), con

los ejercicios (los problemas ACM-ICPC), o con las entradas en los foros de discusión.

Siendo así, un comportamiento esperado “típico” para una sesión dentro del LMS sería el de estudiar un material, intentar resolver los ejercicios correspondientes y, dependiendo de las realimentaciones recibidas, resolver las dudas bien sea mediante los materiales o mediante los foros de discusión. Qué tanto los estudiantes coinciden o difieren de dicho comportamiento, es precisamente la pregunta que guía este trabajo.

4 Materiales y métodos

Para dar respuesta a la pregunta recién planteada, se llevó a cabo un estudio piloto con estudiantes de primer año de la Universidad Nacional de Colombia, en la ciudad de Medellín. En total participaron 101 estudiantes del curso “Estructuras de datos” durante parte de un semestre académico.

Tabla 1. Número total de materiales y ejercicios en cada módulo

Módulo	Descripción	Materiales	Ejercicios ACM-ICPC
1	Introducción eficiencia algorítmica	5	14
2	Arreglos estáticos y dinámicos	5	18
3	Listas, pilas, colas	9	20
4	Árboles binarios de búsqueda y AVL, montículos binarios	12	20
5	Tablas hash	6	14

El curso estaba dividido en 5 módulos tal como se muestra en la Tabla 1 y fue montado a un LMS propio denominado TICademia (www.ticademia.com) tal como se muestra en la Fig. 3.

Fig. 3. Visualización del curso dentro del LMS

Dicho LMS permitió capturar una bitácora de las acciones realizadas por los estudiantes, siendo de interés las relacionadas con las actividades de aprendizaje descritas previamente. A partir de tal bitácora se procedió a realizar un análisis de minería de procesos mediante el software DISCO para entender el patrón de comportamiento de los estudiantes. Un resumen de los datos utilizados se presenta en la Tabla 2.

Tabla 2. Datos recopilados del LMS

Variable	Descripción	Cantidad de ocurrencias
RESOURCE (R)	Visualización de un material	4771
ATTEMPT (A)	Envío de un ejercicio	2625
NEW_ATTEMPT (NA)	Envío de un ejercicio diferente al anterior	2706
CE	Error de compilación en un envío	354
RTE	Error de ejecución en un envío	608
TLE	Tiempo límite excedido en un envío	180
WA	Respuesta incorrecta en un envío	1117
CORRECT (C)	Respuesta correcta en un envío	3042
FORUM (F)	Interacción con una entrada de un foro de discusión	30

5 Resultados

Como resultado de aplicar el algoritmo de minería de procesos provisto por el software DISCO se obtuvo en primera instancia el modelo de procesos ‘espagueti’ presentado parcialmente en la Fig. 4. Se denomina de ese modo porque tiene tantos arcos y cruces que resulta difícil de entender u observar patrones. Contiene un punto de inicio y un punto final representados por un triángulo y un cuadrado dentro de un círculo respectivamente. Los arcos representan las secuencias de acciones llevadas a cabo por diferentes estudiantes. Así, por ejemplo, una secuencia de interacción consiste en 1) ver un material, luego 2) realizar el envío de un ejercicio, 3) recibir una realimentación de que está correcto, y 4) terminar la sesión, siendo esta una de las muchas secuencias posibles.

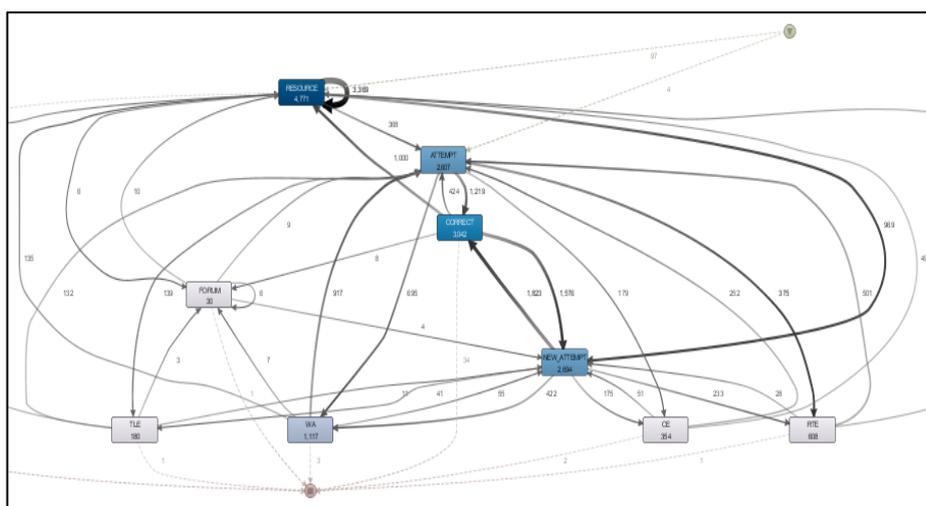


Fig. 4. Extracto del modelo de procesos completo

Para llevar a cabo un análisis de la información contenida en tal modelo se le puede “vaciar” en términos porcentuales a una matriz de interacciones como se presenta en la Tabla 3. Para facilitar su visualización, por cada fila se presenta en gris oscuro la acción siguiente más frecuente, en gris medio la segunda y en gris claro la tercera.

Al mirar en conjunto la información de ambas, es posible evidenciar algunos patrones de comportamiento interesantes:

En primera instancia se nota que el punto de ingreso en la gran mayoría de los casos es la visualización de materiales, incluso no uno solo, si no varios. En otras palabras, fueron pocos los casos en que los estudiantes se “arriesgaron” a realizar los ejercicios sin antes consultar estos recursos de apoyo.

Cuando un envío no es correcto, claramente se nota que la acción más frecuente, por arriba del 70% en todos los casos, es intentarlo de nuevo sin antes revisar materiales o foros de discusión. En cambio, cuando si lo hacen, se nota que es mucho más común el uso de los materiales que de los foros. De hecho, una conclusión importante es que los foros son poco utilizados. Incluso, una vez se accede al foro, la acción más

frecuente no es un envío, estando este en segundo lugar, sino la visualización de un material.

Este hallazgo referente al foro sugiere que dentro del curso quizá haga falta hacerle más difusión. Sin embargo, puede que el fenómeno presentado quizá se explique porque los estudiantes acudan a herramientas de comunicación síncronas, como es el chat, pero en cuyo caso, y por razones de privacidad, no existe una bitácora.

Dentro de las realimentaciones brindadas por el juez en línea, y empleando también los datos de la Tabla 2, se evidencia que la mayoría de los envíos de códigos producen una respuesta correcta con el 57,4% de los casos. Esto indica que los estudiantes se esfuerzan, no solo por realizar buenos algoritmos, sino por codificarlos bien. En segundo lugar, están las respuestas incorrectas con 21,1% mientras que las otras tres posibles realimentaciones, relacionadas a errores de compilación, de ejecución, o de ineficiencia, suman el restante 21,5%. Esto puede explicarse porque los estudiantes de la prueba tienen experiencia previa en programación, dado que el curso de “Estructuras de datos” tiene como prerrequisito un curso previo de lógica de programación.

Por último, cuando un envío es correcto, la siguiente acción suele ser o el envío de otro ejercicio o la visualización de un recurso. Esto puede ser fácilmente explicado por la relación 1:muchos entre materiales y ejercicios. Es decir, un mismo material de estudio puede servir para resolver varios ejercicios. Esto explica porque los estudiantes en ciertos casos continúan con el siguiente ejercicio y en otros se devuelven a los materiales.

Tabla 3. Matriz de interacciones

	R	F	A	NA	CE	RTE	TLE	WA	C
R	71,5%	0,1%	7,8%	20,6%	*	*	*	*	*
F	34,5%	20,7%	31,0%	13,8%	*	*	*	*	*
A	*	*	*	*	6,9%	14,4%	5,3%	26,7%	46,8%
NA	*	*	*	*	6,5%	8,6%	1,5%	15,7%	67,7%
CE	13,9%	0,0%	71,6%	14,5%	*	*	*	*	*
RTE	12,9%	0,0%	82,5%	4,6%	*	*	*	*	*
TLE	18,4%	1,7%	73,7%	6,1%	*	*	*	*	*
WA	12,1%	0,6%	82,3%	4,9%	*	*	*	*	*
C	33,2%	0,3%	14,1%	52,4%	*	*	*	*	*

* No aplica, pues toda realimentación se debe a un envío y todo envío conlleva a una realimentación

Al analizar, ya no todas las acciones de todos los estudiantes sobre todo el curso, sino al enfocarse en un ejercicio específico se puede obtener un diagrama como el presentado en la Fig. 5 y el cual es provisto directamente por el LMS utilizado. En él pueden apreciarse la presencia de los patrones descritos. Desde los estudiantes que resuelven el ejercicio al primer intento, así como aquellos que, ante el fracaso, o bien intentan de nuevo sin recurrir ni a revisar ni materiales ni foros, o bien los que si lo hacen primero.

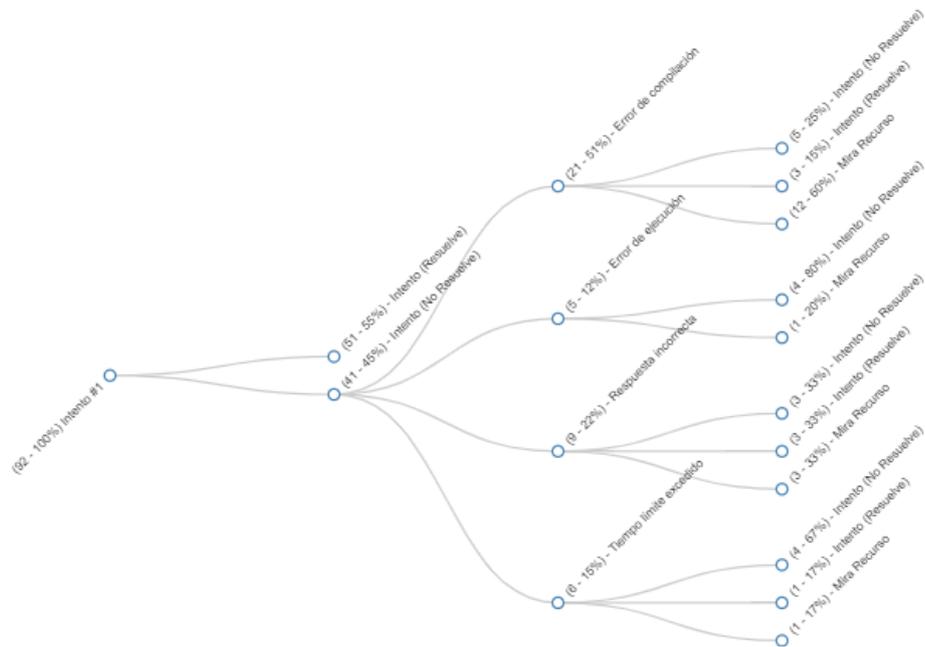


Fig. 5. Diagrama de secuencias de un ejercicio

Ahora, si lo que se desea es un análisis de las realimentaciones recibidas por los estudiantes mucho más resumido, pero también a nivel de un ejercicio particular, el LMS utilizado también ofrece un diagrama como el presentado en la Fig. 6. Algo que resulta interesante al analizarlo es que se puede evidenciar casos en los que estudiantes, aun obteniendo una respuesta correcta, envían de nuevo el ejercicio, probablemente con el fin de obtener un menor tiempo de ejecución y así mejorar la eficiencia algorítmica de su solución.



Fig. 6. Diagrama de resumen de realimentaciones de un ejercicio

6 Conclusiones

El presente estudio tenía como objetivo determinar el uso de la información generada por un juez en línea embebido dentro de un LMS en un curso de programación. Dentro de la experimentación realizada en un entorno real, y gracias a los resultados brindados por procesos de analítica de aprendizaje, generados a partir del LMS utilizado así como por el software DISCO, se logró obtener información valiosa sobre las actividades de aprendizaje llevadas a cabo por los estudiantes.

En particular, y gracias a procesos de minería de procesos, se analizar de forma objetiva los patrones de comportamiento de los estudiantes. Quizá el hallazgo más importante fue que, si bien se presentan casos en que los estudiantes quieren resolver los problemas con una estrategia típica de ensayo y error, en la mayoría de los casos lo que se ve es un proceso sistemático de análisis de material de estudio - intento - realimentación - acción correctiva en caso de ser necesaria.

Como trabajo futuro, y a partir de las cifras obtenidas, haría falta analizar el uso de herramientas de comunicación, ya no asíncronas como el foro, sino síncronas como el chat. Lo anterior entendiéndolo que pese a no poder almacenarse como tal las conversaciones, si se podría por lo menos generar registros, sea vacíos o cifrados, sobre los mensajes compartidos por este medio. Si ello no fuera posible, por lo menos se debería realizar un análisis a-posteriori a manera de encuesta sobre el uso de este mecanismo para la resolución de dudas.

Otro trabajo futuro interesante sería contrastar los resultados obtenidos en un curso de programación, pero más introductorio. Ojalá uno sobre el primer contacto de los estudiantes con la programación. Esto para determinar que tanto varían los patrones encontrados.

Referencias

- [1] J. Bennedsen and M. E. Caspersen, "Persistence of elementary programming skills," *Comput. Sci. Educ.*, vol. 22, no. 2, pp. 81–107, 2012.
- [2] D. A. Kranch, "Teaching the novice programmer: A study of instructional sequences and perception," *Educ. Inf. Technol.*, vol. 16, no. 3, pp. 291–313, 2012.
- [3] V. G. Renumol, S. Jayaprakash, and D. Janakiram, "Classification of Cognitive Difficulties of Students to Learn Computer Programming," 2009.
- [4] A. F. P. Corcho and J. M. Cadavid, "Courses design based on competitive programming and gamification as strategy to improve student performance."
- [5] G. Garcia-mateos and J. L. Fernandez-aleman, "Make Learning Fun with Programming Contests," in *Transactions on Edutainment II, Lecture Notes in Computer Science 5660*, Springer-Verlag, 2009, pp. 246–257.
- [6] G. García-mateos and J. L. Fernández-alemán, "A Course on Algorithms and Data Structures Using On-line Judging," *ITiCSE*, 2009.
- [7] A. A. Pathan, M. Hasan, M. F. Ahmed, and D. M. Farid, "Educational data

- mining: A mining model for developing students' programming skills," *Ski. 2014 - 8th Int. Conf. Software, Knowledge, Inf. Manag. Appl.*, 2014.
- [8] M. Ayub, H. Toba, S. Yong, and M. C. Wijanto, "Modelling students' activities in programming subjects through educational data mining," *Glob. J. Eng. Educ.*, vol. 19, no. 3, pp. 249–255, 2017.
- [9] G. P. Wang, S. Y. Chen, X. Yang, and R. Feng, "OJPOT: online judge & practice oriented teaching idea in programming courses," *Eur. J. Eng. Educ.*, vol. 41, no. 3, pp. 304–319, 2016.
- [10] E. Enström, G. Kreitz, F. Niemelä, P. Söderman, and V. Kann, "Five Years with Kattis – Using an Automated Assessment System in Teaching," *FIE '11 Proc. 2011 Front. Educ. Conf.*, pp. 1–6, 2011.