

## **Inteligencia artificial en la prueba de software: una revisión sistemática de la literatura**

Alfonso Robles-Aguilar<sup>1</sup>, Angel Juan Sánchez-García,  
Jorge Octavio Ocharán-Hernández

Facultad de Estadística e Informática,  
Universidad Veracruzana,  
México

alfonsoroag8@gmail.com, {angesanchez, jocharan}@uv.mx

**Resumen.** Este estudio presenta el método y el resultado de una revisión sistemática de la literatura para identificar las técnicas de Inteligencia Artificial (IA) usadas en la prueba de software (PS). Cincuenta y un estudios fueron seleccionados y analizados para encontrar las técnicas que han sido utilizadas para asistir actividades de PS e identificar nichos de oportunidad para investigaciones futuras. En estos estudios 18 actividades de la PS fueron abordadas con 20 técnicas de IA. Como resultado se obtuvo que las técnicas de IA más utilizadas en la PS son la Inteligencia de Enjambre, los Algoritmos Genéticos y las Redes Neuronales Artificiales. Por otro lado, las actividades de PS más abordadas con IA fueron la generación de casos o datos de prueba y localización de fallas. Por último, los resultados muestran que la colaboración de estas dos disciplinas ha ido en aumento, lo que confirma que la relación de estas dos áreas es activa y prometedora.

**Palabras clave:** Inteligencia artificial, prueba de software, revisión sistemática de la literatura, calidad de software.

### **Artificial Intelligence in Software Testing: A Systematic Literature Review**

**Abstract.** This paper presents the method and the result of a systematic literature review to identify the Artificial Intelligence (AI) techniques used in the Software Testing (ST). Fifty-one studies were selected and analyzed to find the techniques that have been used to assist ST activities and identify niches of opportunity for future research in this area. In these papers, 18 ST activities were approached with 20 AI techniques. As a result, it was obtained that the AI techniques most used in the ST are Swarm Intelligence, Genetic Algorithms, and Artificial Neural Networks. On the other hand, the activities of ST most approached with AI were test cases or test data generation and fault location. Finally, the results show that

the collaboration of these two disciplines has been increasing, confirming that the relationship between these two areas is active and promising.

**Keywords:** Artificial intelligence, software testing, systematic literature review, software quality.

## 1. Introducción

En años recientes, la Inteligencia Artificial (IA) ha ayudado a encontrar soluciones en problemas de diferentes disciplinas, desde ciencias como la arqueología [1] o la biología [2], hasta ingenierías como la eléctrica [3] o la ambiental [4]. El apoyo que ha representado la IA a muchas disciplinas se ha visto reflejado con mayor investigación y búsqueda de soluciones, llegando a campos como la Ingeniería de Software.

La Ingeniería de Software (IS) es una disciplina que comprende todos los aspectos de producción de software útil y de calidad [5]. Para lograr esto, se debe seguir un conjunto de actividades entre las que se encuentra la elicitación, el diseño, la construcción y la prueba de software (PS). En ocasiones, la prueba de software toma hasta el 50% de los recursos totales asignados para un proyecto [6]. La PS hace la diferencia entre un producto de calidad o uno sin garantías, pues el aseguramiento de la calidad se define en esta fase [7]. Esto se logra aplicando casos de prueba para la comprobación de la correcta funcionalidad del sistema y detectando fallas cuando existan casos de prueba exitosos.

La importancia de esta fase y la necesidad de optimizarla se ha reflejado en la búsqueda de soluciones a problemas de actividades en la PS. En esta búsqueda, algunas de esas soluciones se han encontrado en la IA [8]. El objetivo de esta Revisión Sistemática de la Literatura es tener un panorama general de las técnicas de la IA que se han aplicado en actividades de la PS y los resultados de su aplicación.

El trabajo está organizado de la siguiente manera: en la Sección 2 se muestra el trabajo relacionado. En la Sección 3 se presenta el método de investigación. En la Sección 4 se exponen los resultados obtenidos. Por último, en la Sección 5, se extraen conclusiones y se describe el trabajo futuro.

## 2. Trabajos relacionados

Para la realización de este trabajo no se encontraron revisiones se encontraron tres trabajos relacionados con el tema de la IA y la PS.

Sorte *et al.* [8] divulgaron una revisión del estado del arte del uso de IA en todo el ciclo de vida del desarrollo de software, en el cual abordaron a la PS pero con alcance muy acotado, ya que solamente cuenta con dos secciones que exponen el uso que se le ha dado a la IA en dos actividades de la PS.

Bunke *et al.* [9] publicaron un libro donde exponen algunos usos que se le ha dado a la IA en actividades de la PS, como por ejemplo: generación y reducción de sets de prueba, pruebas de caja negra y pruebas automatizadas de interfaces de usuario. Esta publicación muestra un panorama del uso de varias técnicas de IA en diferentes

actividades de PS, así como las tendencias que existían en este campo en los primeros años del siglo XXI. Al ser una publicación de 2004, es viable una actualización de la información con los resultados más recientes en el área.

Por último, Hourani *et al.* [10] publicaron un artículo en el cual explican el impacto de la IA en la PS, donde llevaron a cabo una revisión de la literatura. En esta revisión encontraron el uso de técnicas de IA en PS como Algoritmos Genéticos, Redes Neuronales Artificiales, Árboles Aleatorios, entre otros. Una oportunidad de mejora de este trabajo es exponer la metodología que se usó para realizar la revisión de la literatura, y conocer el cómo se llegó a estos resultados.

### 3. Método de investigación

Esta investigación fue basada en la guía provista por Kitchenham y Charters [11] y se condujo en tres fases: la planeación, la ejecución de la RSL y el reporte. En estas tres fases se cubrió lo siguiente: definir términos de búsqueda, construir una cadena de búsqueda utilizando estos términos, establecer las fuentes de búsqueda y determinar los criterios para la selección de los estudios primarios. También, se definió el proceso para la extracción de datos y su análisis.

#### 3.1. Planeación

**Preguntas de investigación.** En esta sección se formularon las preguntas de investigación (PI) que fueron de interés en esta RSL. A través de ellas se pretende exponer trabajo previo sobre técnicas de la IA en la PS, así como los resultados que se han obtenido al usarlas y encontrar nichos de oportunidad para futuras investigaciones en este campo. Las PI formuladas y su motivación se muestran en la Tabla 1.

Tabla 1. Preguntas de investigación.

Pregunta	Motivación
<b>PI1.-</b> ¿Cuáles son las técnicas de la IA que se han aplicado en la PS?	Conocer las técnicas de la IA que se han aplicado en la prueba de software.
<b>PI2.-</b> ¿Qué actividades de la PS han sido abordados con estas técnicas?	Identificar las actividades de la PS que se han intentado solucionar o mejorar con IA.
<b>PI3.-</b> ¿Cuáles han sido los resultados de las técnicas en su aplicación?	Conocer las ventajas y desventajas que representa el uso de estas técnicas.
<b>PI4.-</b> ¿Qué áreas de oportunidad o trabajo futuro se mencionan en las publicaciones?	Identificar áreas de investigación prometedoras en este ámbito.

**Selección de fuentes y estrategia de búsqueda.** Para responder las PI se identificaron los términos de búsqueda ubicados en la Tabla 2. Estos términos surgieron de las mismas PI. De la PI1 se ubicaron las palabras “técnicas” e “Inteligencia Artificial” como los conceptos más importantes de esta pregunta. En cuanto a la PI2, “Pruebas de Software” es la palabra que se consideró necesaria para responder esta pregunta. De los tres conceptos, se establecieron cuatro palabras clave de esta revisión.

**Tabla 2.** Términos de búsqueda.

Concepto	Palabras clave
Inteligencia Artificial	Artificial intelligence
Técnica	Technique, techniques
Prueba de software	Software testing

**Tabla 3.** Base de datos electrónicas seleccionadas.

Base de datos electrónica	Enlace
ScienceDirect	<a href="https://www.sciencedirect.com/">https://www.sciencedirect.com/</a>
SpringerLink	<a href="https://link.springer.com/">https://link.springer.com/</a>
IEEE Xplore	<a href="https://ieeexplore.ieee.org/">https://ieeexplore.ieee.org/</a>
ACM Digital Library	<a href="https://dl.acm.org/">https://dl.acm.org/</a>

**Tabla 4.** Criterios de selección de estudios primarios

Criterios de inclusión	Criterios de exclusión
CI1.- Publicados entre el año 2015 y 2019.	CE1.- Publicado en un idioma diferente a inglés.
CI2.- Es un artículo de un journal o memoria de congreso.	CE2.- Es reporte técnico, poster, diapositivas, opinión, resumen de workshop o Keynote.
CI3.- En el resumen o palabras clave dan indicios de que el artículo contesta por lo menos una pregunta de investigación.	CE3.- Habla de pruebas a sistemas que usan Inteligencia Artificial pero no aplican técnicas de IA en el proceso de prueba
CI4.- Contesta al menos una pregunta de investigación.	CE4.- Artículos duplicados.

Tomando las palabras clave, se construyó la siguiente cadena de búsqueda:

*(“software testing”) AND (“technique” OR “techniques”) AND (“artificial intelligence”).*

A partir de los términos de búsqueda que se encuentran en la cadena, debido a que las dos más importantes (Artificial intelligence y Software testing) son del área de la Informática y la Computación, se optó como fuentes las bases de datos de la Tabla 3. La elección de estas fue debido a que tienen una gran colección de artículos de esta área, y para ser publicados en estas bases de datos, los artículos pasan por un proceso de arbitraje por expertos en el área.

**Criterios de selección de estudios primarios.** Para seleccionar los estudios a revisar, se eligieron las publicaciones que contaron con los cuatro criterios de inclusión (CI) y que no cumplieron con ningún criterio de exclusión (CE) de la Tabla 4.

**Proceso de selección de estudios primarios.** Para incluir a los estudios como primarios se llevó a cabo un proceso de selección en diferentes fases, en las cuales se aplicaron los CI y CE en el orden de la Fig. 1.

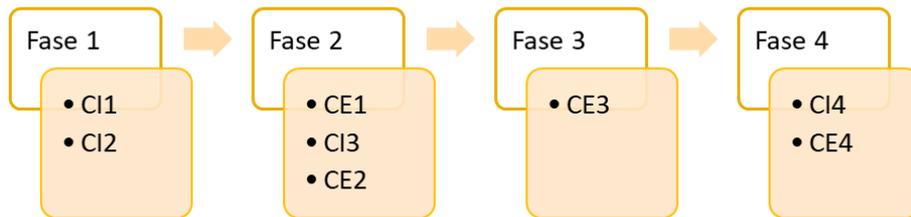


Fig. 1. Criterios de selección por fase.

Tabla 5. Formato para la extracción de datos.

Título	Autores	Año	Fuente	Palabras clave	Técnica de la IA usada	Actividad de PS	Resultados	Trabajo futuro
--------	---------	-----	--------	----------------	------------------------	-----------------	------------	----------------

**Estrategia de extracción de datos.** Se definieron nueve datos de importancia para responder las PI de este estudio. Estos datos son mostrados en la Tabla 5.

Las primeras cinco características son para conocer las generalidades del artículo. Las siguientes cuatro son las secciones del estudio que contestan directamente cada una de las PI. Estos datos se extrajeron llenando una plantilla en una hoja de cálculo, donde cada columna correspondía a un dato de la Tabla 5 y cada fila era un artículo, y los espacios se llenaron con los fragmentos de texto que respondían a cada dato.

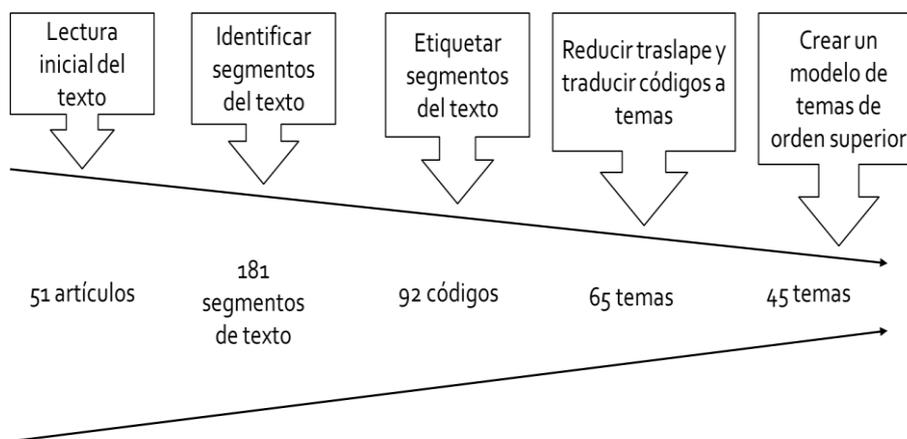
**Método de síntesis de datos.** Se siguió la Síntesis Temática propuesta por Cruzes y Dyba [12] para el análisis de datos cualitativos. Se optó por este método ya que se ha demostrado que es útil en revisiones con datos cualitativos del área.

### 3.2. Ejecución de la búsqueda

Para la selección de los estudios primarios, en la Fase 1 se aplicó la cadena de búsqueda en cada una de las bibliotecas digitales y se utilizó el filtro automático con la finalidad de que se mostraran artículos de *journals* y congresos publicados entre los años 2015 y 2019. En la Fase 2 se verificó que estuvieran escritos en idioma inglés, que no fueran diferentes a artículo de *journal* o congreso y se leyeron los resúmenes y palabras clave del artículo. Si estos dieron algún indicio de que pudieran contestar al menos una pregunta de investigación (no tenía que incluir un término de búsqueda obligatoriamente, ya que en algunos estudios nunca se menciona IA, pero si se menciona alguna técnica de IA) fueron incluidos para la siguiente fase. La Fase 3 consistió en eliminar a todos los estudios que expusieran sistemas que usaran IA pero que no se usó IA en la PS. La última fase fue la lectura de cada uno de los estudios, donde fueron incluidos como estudios primarios si contestaron al menos una pregunta de investigación y se excluyeron los artículos repetidos. La cantidad de estudios seleccionados por fase de cada una de las bibliotecas se muestra en la Tabla 6.

**Tabla 6.** Número de estudios por fase.

Biblioteca digital	Fase 1	Fase 2	Fase 3	Fase 4
SpringerLink	149	44	15	10
ScienceDirect	336	43	17	12
ACM Digital Library	146	43	14	10
IEEE Xplore	83	30	22	19
Totales	714	160	68	51



**Fig. 2.** Proceso de síntesis temática de datos.

## 4. Resultados

### 4.1. Resultados del análisis de datos

El análisis de datos cualitativos, se siguió el proceso que se muestra en la Fig. 2. La lectura inicial fue de 51 artículos considerados como Estudios Primarios. De ellos se extrajo 181 segmentos de texto, donde se encontraban respuestas a las PI. Después de codificar y tematizar los segmentos de texto, se obtuvieron 45 temas de orden superior repartidos de la siguiente manera: 20 temas en las técnicas de IA (las técnicas de IA se agruparon según su taxonomía), 18 temas en las actividades de PS (se determinó el tema de orden superior según a qué actividad exactamente atacaba la técnica de IA), 2 temas en resultados y 5 temas para trabajo futuro.

### 4.2. Respuestas de las preguntas de investigación

Para responder las PI, en los 51 artículos se encontraron 65 casos del uso de técnicas de IA en actividades de PS, ya que algunos artículos reportaban más de un caso.

**Tabla 7.** Técnicas de IA reportadas y frecuencia de aparición.

Técnica (Identificador)	Frecuencia	Referencias
Inteligencia de Enjambre (8)	18	[6, 13–23]
Algoritmos Genéticos (1)	15	[17, 22–34]
Redes Neuronales (4)	11	[17, 35–44]
Redes Bayesianas (2)	2	[28, 45]
Búsqueda de Monte Carlo (3)	2	[28, 46]
Bosques Aleatorios (7)	2	[47, 48]
K Vecinos Más Cercanos (12)	2	[49, 50]
Búsqueda Hiperheurística (5)	1	[51]
PRINCE (6)	1	[52]
Lógica Difusa (9)	1	[23]
Regresión de Huber (10)	1	[38]
Regresión de Vectores de Soporte (11)	1	[38]
LOUD (13)	1	[53]
Exploración Basada en <i>Q-Learning</i> (14)	1	[54]
Prueba por Comité (15)	1	[55]
Algoritmo de Agrupamiento (16)	1	[56]
Clasificador Bayesiano Ingenuo (17)	1	[57]
Sistema de Inferencia Difusa Mamdani (18)	1	[58]
Algoritmo K-medias (19)	1	[59]
Agentes (20)	1	[60]

En respuesta a la **PI1**, las técnicas de IA que se usan en la PS reportadas en los Estudios Primarios se muestran en la Tabla 7. Con respecto a la frecuencia de aparición, se encontró que la técnica de IA que más se ha utilizado en la PS es la Inteligencia de Enjambre, abarcando todos los algoritmos que pertenecen a ella.

Mientras que la segunda técnica más utilizada son los Algoritmos Genéticos, y la tercera más utilizada son las Redes Neuronales. El hecho de que estas tres técnicas se hayan utilizado en el 68% de los 65 casos registrados, indica que hay un relativo abandono generalizado en las otras 17 técnicas. Algoritmos clásicos como KNN y enfoques como los Agentes o las Redes Bayesianas no tienen mucha presencia en esta área, por lo que hacer investigaciones aplicándolos a la PS puede resultar en un estudio novedoso.

En respuesta a la **PI2**, las actividades de PS en donde se han aplicado técnicas se muestran la Tabla 8, en donde se expone a la generación de casos de prueba y la localización de fallas como las dos actividades más abordadas con IA. Estas dos son las únicas que representan verdaderos repuntes donde se busca mejoras para sus procesos con IA en comparación con las otras.

**Tabla 8.** Actividades de la PS y frecuencia de aparición.

Actividades (Identificador)	Frecuencia	Referencias
Generación de casos/datos de prueba (6)	21	[6, 14, 20–24, 26, 27, 29–31, 33, 34, 55, 58, 60]
Localización de fallas (5)	11	[17, 32, 40, 41, 47, 48, 52, 53]
Priorización de casos de prueba (11)	6	[15, 23, 57, 59]
Cobertura de ramas (7)	5	[13, 37, 38]
Pruebas de Software basada en búsqueda (1)	3	[28]
Pruebas de interfaz de usuario (3)	3	[42, 43, 54]
Exploración de caminos (10)	2	[19, 46]
Clasificación de resultados de prueba (13)	2	[49, 56]
<i>Covering Arrays</i> (15)	2	[16, 25]
Optimización de casos de prueba (18)	2	[14]
Pruebas automáticas (2)	1	[35]
Prueba de interacción combinatoria (4)	1	[51]
Identificación de casos correctos de prueba (8)	1	[50]
Prueba de mutación (9)	1	[18]
Pruebas a Sistemas Orientados a Servicios (12)	1	[44]
Detección de código maligno (14)	1	[45]
Entrenamiento de datos (16)	1	[39]
Oráculos (17)	1	[36]

Entre ambas representan el 34% de los casos, un porcentaje muy alto tomando en cuenta que la tercera actividad de PS más frecuente (priorización de casos de prueba) representa al 7% de los casos. Considerando la existencia de otras 16 actividades registradas y su poca exploración, muchos estudios de investigación pueden surgir a partir de ellas. Además, es muy útil para los Ingenieros de Software y *Testers* conocer la existencia de una amplia gama de estrategias para realizar generación de casos de prueba y localización de fallas de una forma óptima utilizando IA.

Para identificar cuáles fueron las técnicas más aplicadas en las actividades de la PS, se usó el gráfico de burbuja de la Fig. 3, en el cual se representa a cada una de las técnicas de IA y actividades de PS con el identificador que les fue asignado en la Tabla 7 y la Tabla 8, respectivamente. En la Fig. 3 el tamaño de las burbujas representa la frecuencia con la que una actividad de PS es abordada con una técnica de IA. El número que se encuentra en el centro de la burbuja es la magnitud que representa.

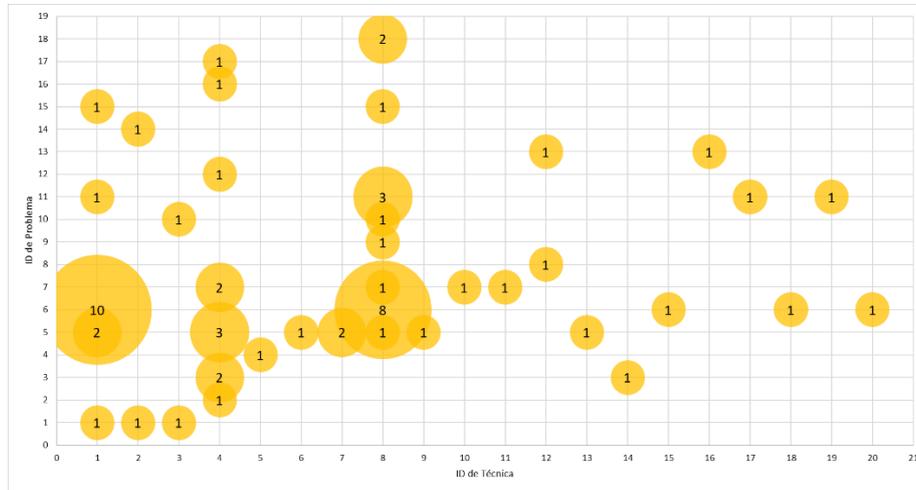


Fig. 3. Técnicas de IA utilizadas (horizontal) en actividades de la PS (vertical).

En la Fig. 3 se puede distinguir que los Algoritmos Genéticos, representados por el número 1 horizontal, se han utilizado para diversas actividades, pero principalmente para la generación de casos de prueba, representado con el número 6 vertical, sumando en total 10 reportes de esta aplicación. Otro punto para destacar es el amplio uso de la Inteligencia de Enjambres, representada por el 8 horizontal, siendo esta técnica la más utilizada en diferentes actividades de software, estando presente en ocho diferentes de las reportadas en este trabajo. El número 4 horizontal, que son las Redes Neuronales, también se distribuye en la aplicación de diferentes actividades de PS ya que, de sus 11 apariciones, se han aplicado a siete actividades.

Por el lado de las actividades de la PS, como se muestra en las frecuencias de la Tabla 8, la generación de casos de prueba, representada por el identificador 6 vertical, ha sido la más estudiada, principalmente con Algoritmos Genéticos (1 horizontal) e Inteligencia de Enjambre (8 horizontal). En cambio, la localización de fallas, representada por el número 5 vertical, ha sido tratada con 7 diferentes técnicas de la IA.

La Fig. 3 es muy útil para la identificación de áreas muy estudiadas, en las que un Ingeniero de Software o *Tester* pudiera buscar información para aplicarlas en sus actividades, así como para los investigadores, en donde fácilmente se pueden distinguir nichos de oportunidad.

En respuesta a la **PI3**, se encontró que de los 65 reportes de uso de técnicas de IA en actividades de PS todos lograron realizar la tarea para la que fueron diseñadas, y no sólo eso, ya que en 26 (40%) de los 65 casos lograron mejorar el proceso de la actividad de PS usando la técnica de IA. Como se puede observar, existe un gran potencial en el uso de IA en PS debido a los buenos resultados obtenidos por los diferentes estudios.

Para responder la **PI4** se muestra en la Fig. 4, que, de los 31 trabajos, 29 continuarán con algún tipo de investigación, mientras que los dos restantes tienen planeado implementar una herramienta que utilice las técnicas de la IA que aplicaron, para que los

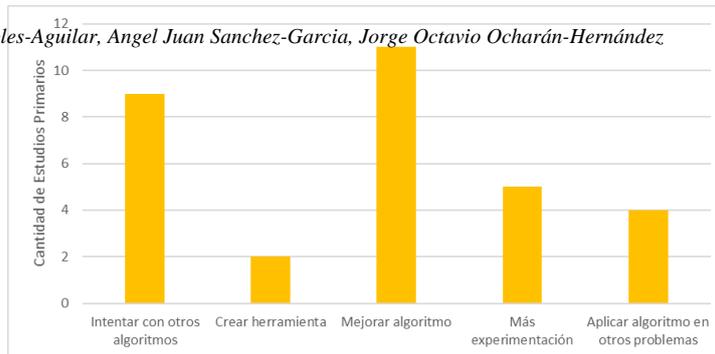


Fig. 4. Trabajo futuro reportado.

Ingenieros puedan consumir el conocimiento a través de una herramienta, mientras que una gran parte sigue en investigaciones.

## 5. Conclusiones y trabajo a futuro

Este documento presenta la planeación, ejecución y reporte de una RSL, enfocada en conocer la aplicación de la IA en la fase de PS. Reconocer las áreas con poca cobertura de pruebas nos deja nichos de oportunidad para hacer nuevas investigaciones, como por ejemplo la aplicación Lógica Difusa, Agentes o Algoritmos de Agrupamiento, o bien buscar soluciones con IA en entrenamiento de datos, prueba de mutación o en la identificación de casos correctos de prueba, por citar algunos ejemplos. Igualmente, al ver los resultados que se han obtenido al usar IA, donde todos los estudios reportan que tienen la misma o mejor eficiencia comparando con las técnicas actuales y sin IA, nos explica por qué la tendencia de investigaciones en esta área ha aumentado durante los últimos cinco años. Como trabajo futuro de este estudio, al ser parte de un trabajo más grande, se busca conocer la aplicación de la IA en otras áreas y fases de la IS, como Requisitos, Diseño y Construcción de Software, con la finalidad de encontrar más nichos de oportunidad para la colaboración entre la Inteligencia Artificial y la Ingeniería de Software.

## Referencias

1. Bewes, J., Low, A., Morphett, A., Pate, F.D., Henneberg, M.: Artificial intelligence for sex determination of skeletal remains: Application of a deep learning artificial neural network to human skulls. *J. Forensic Leg. Med.* 62, pp. 40–43 (2019)
2. Turan, E., Çetin, G.: Using artificial intelligence for modeling of the realistic animal behaviors in a virtual island. *Comput. Stand. Interfaces.* 66, 103361 (2019)
3. Mayr, A., Weigelt, M., Masuch, M., Meiners, M., Hüttel, F., Franke, J.: Application Scenarios of Artificial Intelligence in Electric Drives Production. *Procedia Manuf.* 24, pp. 40–47 (2018)
4. Ye, Z., Yang, J., Zhong, N., Tu, X., Jia, J., Wang, J.: Tackle environmental challenges in pollution controls using artificial intelligence: A review. *Sci. Total Environ.* 134279 (2019)
5. Sommerville, I., Galipienso, M.I.A.: Ingeniería del software. Pearson Educación (2005)

6. Rabbi, K., Mamun, Q., Islam, M.R.: An efficient particle swarm intelligence based strategy to generate optimum test data in t-way testing. In: 2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA), pp. 123–128 (2015)
7. Bourque, P., Fairley, R.E.: SWEBOK: guide to the software engineering body of knowledge. IEEE Computer Society (2014)
8. Sorte, B., Joshi, P., Jagtap, V.: Use of Artificial Intelligence in Software Development Life Cycle: A state of the Art Review. *Int. J. Technol. Manag.* 03, pp. 2309–4893 (2015)
9. Bunke, H., Last, M., Kandel, A.: *Artificial Intelligence Methods in Software Testing*. World Scientific (2004)
10. Hourani, H., Hammad, A., Lafi, M.: The Impact of Artificial Intelligence on Software Testing. In: 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), pp. 565–570 (2019)
11. Kitchenham, B., S.C.: Guidelines for performing systematic literature reviews in software engineering (2007)
12. Cruzes, D.S., Dyba, T.: Recommended Steps for Thematic Synthesis in Software Engineering. In: 2011 International Symposium on Empirical Software Engineering and Measurement, pp. 275–284 (2011)
13. Boopathi, M., Sujatha, R., Senthil Kumar, C., Narasimman, S.: Quantification of Software Code Coverage Using Artificial Bee Colony Optimization Based on Markov Approach. *Arab. J. Sci. Eng.* 42, pp. 3503–3519 (2017)
14. Khari, M., Kumar, P., Burgos, D., Crespo, R.G.: Optimized test suites for automated testing using different optimization techniques. *Soft Comput.* 22, pp. 8341–8352 (2018)
15. Khatibsyarhini, M., Isa, M.A., Jawawi, D.N.A., Hamed, H.N.A., Mohamed Suffian, M.D.: Test Case Prioritization Using Firefly Algorithm for Software Testing. *IEEE Access.* 7, pp. 132360–132373 (2019)
16. Mahmoud, T., Ahmed, B.S.: An efficient strategy for covering array construction with fuzzy logic-based adaptive swarm optimization for software testing use. *Expert Syst. Appl.* 42, pp. 8753–8765 (2015)
17. Singh, P.K., Garg, S., Kaur, M., Bajwa, M.S., Kumar, Y.: Fault localization in software testing using soft computing approaches. In: 2017 4th International Conference on Signal Processing, Computing and Control (ISPCC), pp. 627–631 (2017)
18. Sadath, L., Nair, R.: Oncological Inspired Techniques for Intelligent Software Testing. In: 2019 Amity International Conference on Artificial Intelligence (AICAI) pp. 327–333 (2019)
19. Popentiu-Vladicescu, F., Albeanu, G.: Nature-inspired Approaches in Software Faults Identification and Debugging. *Procedia Comput. Sci.* 92, pp. 6–12 (2016)
20. Cai, L., Zhang, Y., Ji, W.: Variable Strength Combinatorial Test Data Generation Using Enhanced Bird Swarm Algorithm. In: 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), pp. 391–398 (2018)
21. Sharifipour, H., Shakeri, M., Haghghi, H.: Structural test data generation using a memetic ant colony optimization based on evolution strategies. *Swarm Evol. Comput.* 40, pp. 76–91 (2018)
22. Chawla, P., Chana, I., Rana, A.: A novel strategy for automatic test data generation using soft computing technique. *Front. Comput. Sci.* 9, pp. 346–363 (2015)
23. Mann, M., Tomar, P., Sangwan, O.P.: Bio-inspired metaheuristics: evolving and prioritizing software test data. *Appl. Intell.* 48, pp. 687–702 (2018)
24. Sabharwal, S., Bansal, P., Mittal, N., Malik, S.: Construction of Mixed Covering Arrays for Pair-wise Testing Using Probabilistic Approach in Genetic Algorithm. *Arab. J. Sci. Eng.* 41, pp. 2821–2835 (2016)
25. Sabharwal, S., Bansal, P., Mittal, N.: Construction of t-way covering arrays using genetic algorithm. *Int. J. Syst. Assur. Eng. Manag.* 8, pp. 264–274 (2017)

26. Li, X., Wong, W.E., Gao, R., Hu, L., Hosono, S.: Genetic Algorithm-based Test Generation for Software Product Line with the Integration of Fault Localization Techniques. *Empir. Softw. Eng.* pp. 23, 1–51 (2018)
27. Manikumar, T., John Sanjeev Kumar, A., Maruthamuthu, R.: Automated test data generation for branch testing using incremental genetic algorithm. *S{\={a}}dhan{\={a}}*. 41, pp. 959–976 (2016)
28. Feldt, R., Poulding, S.: Broadening the Search in Search-based Software Testing: It Need Not Be Evolutionary. In: *Proceedings of the Eighth International Workshop on Search-Based Software Testing*. pp. 1–7. IEEE Press, Piscataway (2015)
29. Shamshiri, S., Rojas, J.M., Fraser, G., McMinn, P.: Random or Genetic Algorithm Search for Object-Oriented Test Suite Generation? In: *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*. pp. 1367–1374. ACM (2015)
30. Rodrigues, D.S., Delamaro, M.E., Corrêa, C.G., Nunes, F.L.S.: Using Genetic Algorithms in Test Data Generation: A Critical Systematic Mapping. *ACM Comput. Surv.* 51(41), pp. 1–41 (2018)
31. Esfandyari, S., Rafe, V.: A tuned version of genetic algorithm for efficient test suite generation in interactive t-way testing strategy. *Inf. Softw. Technol.* 94, pp. 165–185 (2018)
32. Arasteh, B., Bouyer, A., Pirahesh, S.: An efficient vulnerability-driven method for hardening a program against soft-error using genetic algorithm. *Comput. Electr. Eng.* 48, pp. 25–43 (2015)
33. Aleti, A., Grunske, L.: Test data generation with a Kalman filter-based adaptive genetic algorithm. *J. Syst. Softw.* 103, pp. 343–352 (2015)
34. Qi, R.-Z., Wang, Z.-J., Li, S.-Y.: A Parallel Genetic Algorithm Based on Spark for Pairwise Test Suite Generation. *J. Comput. Sci. Technol.* 31, pp. 417–427 (2016)
35. Zhang, M., Zhang, Y., Zhang, L., Liu, C., Khurshid, S.: DeepRoad: GAN-based Metamorphic Testing and Input Validation Framework for Autonomous Driving Systems. In: *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*. pp. 132–142. ACM (2018)
36. Singhal, A., Bansal, A., Kumar, A.: An approach to design test oracle for aspect oriented software systems using soft computing approach. *Int. J. Syst. Assur. Eng. Manag.* 7, pp. 1–5 (2016)
37. Jitsunari, Y., Arahori, Y.: Coverage-Guided Learning-Assisted Grammar-Based Fuzzing. In: *IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pp. 275–280 (2019)
38. Grano, G., Titov, T. V., Panichella, S., Gall, H.C.: How high will it be? Using machine learning models to predict branch coverage in automated testing. In: *2018 IEEE Workshop on Machine Learning Techniques for Software Quality Evaluation (MaLTeSQuE)* pp. 19–24 (2018)
39. Karmore, S.P., Mahajan, A.R.: New Approach for Testing and providing Security Mechanism for Embedded Systems. *Procedia Comput. Sci.* 78, pp. 851–858 (2016)
40. Kahles, J., Törrönen, J., Huuhtanen, T., Jung, A.: Automating Root Cause Analysis via Machine Learning in Agile Software Testing Environments. In: *2019 12th IEEE Conference on Software Testing, Validation and Verification (ICST)*, pp. 379–390 (2019)
41. Heris, S.R., Keyvanpour, M.: Effectiveness of Weighted Neural Network on Accuracy of Software Fault Localization. In *5th International Conference on Web Research (ICWR)*, pp. 100–104 (2019)
42. Kamal, M.M., Darwish, S.M., Elfatraty, A.: Enhancing the Automation of GUI Testing. In: *Proceedings of the 8th International Conference on Software and Information Engineering*. 66–70. ACM (2019)

43. White, T.D., Fraser, G., Brown, G.J.: Improving Random GUI Testing with Image-based Widget Detection. In: Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis, pp. 307–317, ACM (2019)
44. Eniser, H.F., Sen, A.: Testing Service Oriented Architectures Using Stateful Service Virtualization via Machine Learning. In: IEEE/ACM 13th International Workshop on Automation of Software Test (AST), pp. 9–15 (2018)
45. Zein, S., Salleh, N., Grundy, J.: A systematic mapping study of mobile application testing techniques. *J. Syst. Softw.* 117, pp 334–356 (2016)
46. Yeh, C., Lu, H., Yeh, J., Huang, S.: Path Exploration Based on Monte Carlo Tree Search for Symbolic Execution. In: 2017 Conference on Technologies and Applications of Artificial Intelligence (TAAI), pp. 33–37 (2017)
47. Raman, M., Abdallah, N., Dunoyer, J.: An Artificial Intelligence Approach to EDA Software Testing: Application to Net Delay Algorithms in FPGAs. In: 20th International Symposium on Quality Electronic Design (ISQED), pp. 311–316 (2019)
48. Gupta, L., Salman, T., Zolanvari, M., Erbad, A., Jain, R.: Fault and performance management in multi-cloud virtual network services using AI: A tutorial and a case study. *Comput. Networks.* 165, 106950 (2019)
49. Liu, Y., Li, M., Wu, Y., Li, Z.: A weighted fuzzy classification approach to identify and manipulate coincidental correct test cases for fault localization. *J. Syst. Softw.* 151, pp. 20–37 (2019)
50. Li, Z., Li, M., Liu, Y., Geng, J.: Identify Coincidental Correct Test Cases Based on Fuzzy Classification. In: 2016 International Conference on Software Analysis, Testing and Evolution (SATE), pp. 72–77 (2016)
51. Jia, Y., Cohen, M.B., Harman, M., Petke, J.: Learning Combinatorial Interaction Test Generation Strategies Using Hyperheuristic Search. In: Proceedings of the 37th International Conference on Software Engineering - Volume 1. 540–550. IEEE Press, Piscataway (2015)
52. Kim, Y., Mun, S., Yoo, S., Kim, M.: Precise Learn-to-Rank Fault Localization Using Dynamic and Static Features of Target Programs. *ACM Trans. Softw. Eng. Methodol.* 28(23), pp.1–23:34 (2019)
53. Mariani, L., Monni, C., Pezzé, M., Riganelli, O., Xin, R.: Localizing Faults in Cloud Systems. In: IEEE 11th International Conference on Software Testing, Verification and Validation (ICST), pp. 262–273 (2018)
54. Koroglu, Y., Sen, A., Muslu, O., Mete, Y., Ulker, C., Tanriverdi, T., Donmez, Y.: QBE: QLearning-Based Exploration of Android Applications. In: IEEE 11th International Conference on Software Testing, Verification and Validation (ICST), pp. 105–115 (2018)
55. Walkinshaw, N., Fraser, G.: Uncertainty-Driven Black-Box Test Data Generation. In: 2017 IEEE International Conference on Software Testing, Verification and Validation (ICST), pp. 253–263 (2017)
56. Roper, M.: Using Machine Learning to Classify Test Outcomes. In: 2019 IEEE International Conference On Artificial Intelligence Testing (AITest), pp. 99–100 (2019)
57. Butgereit, L.: Using Machine Learning to Prioritize Automated Testing in an Agile Environment. In: 2019 Conference on Information Communications Technology and Society (ICTAS), pp. 1–6 (2019)
58. Zamli, K.Z., Din, F., Baharom, S., Ahmed, B.S.: Fuzzy adaptive teaching learning-based optimization strategy for the problem of generating mixed strength t-way test suites. *Eng. Appl. Artif. Intell.* 59, pp. 35–50 (2017)
59. Wang, S., Huang, K.: Improving the efficiency of functional verification based on test prioritization. *Microprocess. Microsyst.* 41, pp. 1–11 (2016)
60. Arora, P.K., Bhatia, R.: A Systematic Review of Agent-Based Test Case Generation for Regression Testing. *Arab. J. Sci. Eng.* 43, pp. 447–470 (2018)