

Temática: V Taller Internacional de Ingeniería y Calidad de Software

Redes Neuronales Artificiales para estimar el esfuerzo en proyectos de software. Un estudio comparativo.

Artificial Neural Networks for estimating effort in software projects. A comparative study.

Wisleidys Campos Wright ^{1*}, Alejandro Machado Paredes ²

¹ Universidad de las Ciencias Informáticas. Cuba, Carretera San Antonio Km 2 ½. wcamposw@uci.cu

² Universidad de las Ciencias Informáticas. Cuba, Carretera San Antonio Km 2 ½. amachadop@estudiantes.uci.cu

* Autor para correspondencia: wcamposw@uci.cu

Resumen

Definir el cronograma de un proyecto se ha convertido en una tarea de vital interés para la industria del software. Existen diferentes métodos para realizar la estimación, clasificados en las siguientes categorías: Juicio de Expertos, Modelos Algorítmicos y Aprendizaje Automático, siendo este último, el que ha generado interés en la comunidad científica en los últimos años, debido a la eficiencia en el resultado de las estimaciones y la capacidad de resolver problemas complejos. En esta categoría, se destaca el uso de las redes neuronales artificiales en conjunto con algoritmos de optimización, que apoyan el aprendizaje eficaz de las mismas. En este artículo se realizó una comparación de las redes neuronales artificiales consideradas de orden superior y de aprendizaje profundo, basada en el estudio de diversos artículos e investigaciones, con el objetivo de definir la más óptima en la obtención de la estimación del esfuerzo en el proceso de desarrollo de proyectos de software, arrojando como resultado que las redes neuronales artificiales de orden superior suelen obtener mayor velocidad de respuesta y consumir menor tiempo computacional.

Palabras clave: desarrollo de software, estimación del esfuerzo, aprendizaje automático, redes neuronales artificiales

Abstract

Defining a project schedule has become a task of vital interest for the software industry. There are different methods to perform the estimation, classified in the following categories: Expert Judgment, Algorithmic Models and Machine

Learning, the latter being the one that has generated interest in the scientific community in recent years, due to the efficiency in the estimation results and the ability to solve complex problems. In this category, the use of artificial neural networks in conjunction with optimization algorithms, which support their efficient learning, stands out. In this article, a comparison of artificial neural networks considered of higher order and deep learning was made, based on the study of various articles and research, with the aim of defining the most optimal in obtaining the estimation of effort in the development process of software projects, resulting that higher order artificial neural networks tend to obtain higher response speed and consume less computational time.

Keywords: *software development, effort estimation, machine learning, artificial neural networks*

Introducción

En el negocio del desarrollo de software, la capacidad de entregar proyectos según el cronograma, el presupuesto y la alineación con los objetivos comerciales es fundamental para determinar el límite de las condiciones productivas globales altamente competitivas. La organización debe entregar un producto de calidad, mantener los costos por debajo del presupuesto del cliente y entregar el proyecto según lo planeado. Por lo que se requiere que la gestión de proyectos de software incluya las necesidades del usuario y las limitaciones de presupuesto y tiempo (HEZAM, 2021).

La gestión de proyectos de software es una parte esencial de la ingeniería de software (Sommerville, 2016). Implica la planificación, el seguimiento y la coordinación de personas, procesos y eventos que se producen a medida que el software evoluciona desde un concepto preliminar hasta una implementación operativa completa (Roger S. Pressman & Bruce R. Maxim, 2015). El trabajo del jefe de proyecto es garantizar que el producto de software cumpla y supere estas limitaciones, además de que este sea de alta calidad. Una buena gestión no puede garantizar el éxito del proyecto. Sin embargo, la mala gestión generalmente resulta en el fracaso del mismo, ocasionando que el software pueda entregarse tarde, costar más de lo estimado originalmente o no cumplir con las expectativas de los clientes (Sommerville, 2016). Este proceso comienza con un conjunto de actividades que de manera colectiva se llaman planificación de proyecto. Antes de comenzar un proyecto, el equipo de software debe estimar el trabajo que se va a realizar, los recursos que se requerirán y el tiempo de duración del mismo (Roger S. Pressman & Bruce R.

Maxim, 2015). La planificación en la industria del software, se ha convertido en uno de los principales retos para la gestión de proyectos y una actividad fundamental para desarrollar software de alta calidad. La misma requiere estimaciones que garanticen la eficacia del producto. Para planificar es imprescindible la aplicación de buenos métodos de estimación por la creciente influencia que ejercen en el control preciso y predecible, los procesos de producción y los productos de software (Wright & Casañola, 2021a). Pressman define la estimación como la necesidad real que tiene el equipo del proyecto como intento para determinar costo, esfuerzo recursos y tiempo que tomará en construir un sistema o un producto específico basado en software (Roger S. Pressman & Bruce R. Maxim, 2015). Por otro lado, el esfuerzo es la combinación de persona y tiempo y se refiere a la cantidad de tiempo que una persona necesitaría para completar cierto trabajo productivo (Trendowicz & Jeffery, 2014). Según un estudio de (Huang & Chiu, 2006) y (Solimán, 2018), uno de los pilares básicos para una correcta estimación de un proyecto software es la estimación del esfuerzo asociado a su desarrollo y constituye el principal soporte para la toma de decisiones respecto a factores como costos, tamaño y tiempo.

Las estimaciones certeras constituyen unos de los factores críticos de éxitos para evitar proyectos fracasados. Según estadísticas publicadas por el Standish Group en un estudio a diferentes empresas desarrolladoras de software desde el año 2017 al 2020, informan que una de las principales causas de los excesos de tiempo y costos son los reinicios. Por cada 100 proyectos iniciados, hay 94 reinicios. Los resultados de los costos fueron reveladores, casi un tercio experimentaron sobrecostos de 150 a 200%. El promedio de todas las empresas es el 189% de la estimación de costos original. El sobregiro promedio es del 178% para las grandes empresas, del 182% para empresas medianas y 214% para pequeñas empresas. Más de un tercio también experimentó demoras del 200 al 300%. El rebasamiento promedio es el 222% del tiempo estimado original. Para las grandes empresas, el promedio es 230%; para empresas medianas, el promedio es 202%; y para pequeñas empresas, la media es de 239% (Group, 2019; Mulder, 2020).

Para estimar son utilizados métodos agrupados en tres categorías (Wright, 2021): Juicio de Expertos (Ortega-Hernandez, 2019; Robiolo, 2016), Modelos Algorítmicos (León & Hondares, 2019; Marquez, 2020; Villalobos, 2018), con el avance de las tecnologías la tendencia ha sido la utilización de métodos basados en la categoría de Aprendizaje Automático tales como: Lógica Difusa (Haloi et al., 2021; Vasykiv et al., 2020), Programación Genética (Mahadev & Gowrishankar, 2020; Sharma & Chaudhary, 2020), Razonamiento basado en Casos (Jung et al., 2020; Latif et al., 2021) y Redes Neuronales Artificiales (RNA) (Mohsin, 2021; D. Rankovic et al., 2021; N. Rankovic et al., 2021), esta última ha ganado popularidad en la industria del software por su capacidad para resolver problemas

complejos. El objetivo de la investigación es realizar un estudio comparativo de las RNA para la estimación del esfuerzo en los proyectos de desarrollo de software.

Materiales y métodos o Metodología computacional

La investigación realizada sigue la ruta cuantitativa con un enfoque descriptivo guiado por los aportes de (Hernández Sampieri & Mendoza Torres, 2018). Se realizó una revisión sistemática de la literatura siguiendo la metodología propuesta por (Fernández-Sánchez et al., 2020). En este sentido, se recopilaron artículos en inglés y español del periodo comprendido del 2019 al 2022, publicados en revistas indexadas en Springer, Scielo, IEEE y ELSEVIER. Durante la investigación se escogieron RNA que según (Kumar et al., 2020) son consideradas de orden superior: RNA Wavelet (Kumar et al., 2020); Functional Link Neural Network (FLANN) (Barik et al., 2022) y de aprendizaje profundo: , RNA Spiking (Tavanaei et al., 2019); Recurrent Neural Network (ELMAN) (Bilgaiyan et al., 2019), además se utilizan para la estimación del esfuerzo en el desarrollo de software. Los criterios de comparación fueron: Aplicabilidad en el proceso de desarrollo, Parámetros del desempeño, Procesamiento en la capa oculta y Capa de Salida.

Las redes neuronales artificiales son un sistema diseñado para modelar la técnica de procesamiento de información del cerebro humano. Estas han sido desarrolladas como generalizaciones de modelos matemáticos basándose en el procesamiento de información que se realiza en muchos elementos simples llamados neuronas. Las neuronas están interconectadas por enlaces que tienen un peso numérico, al ajustar estos pesos, se produce un aprendizaje en la red neuronal. A esta interconexión de neuronas se le denomina capa. Las RNA están compuestas por la capa de entrada, donde hay tantas neuronas como características de las muestras que deben enseñar a la red; la capa oculta, que está determinada de acuerdo a la solución del problema, no existe una regla específica para el número de capas ocultas y cambia de un problema a otro; y la capa de salida donde se realizan los cálculos que clasifican o etiquetan la información proveniente de la capa de entrada (AKGÜL & KAYA, 2022).

Las principales ventajas de las redes neuronales artificiales son: su capacidad de generalización, que permite dar una respuesta adecuada a entradas nunca vistas anteriormente; su naturaleza distribuida, que permite la construcción de sistemas eficientes; su capacidad de aproximar funciones no lineales, que resuelven problemas no complejos y su adaptabilidad frente a cambios en el entorno (Wright & Casañola, 2021b). Las redes neuronales artificiales tienen

una fuerza numérica que puede realizar más de un trabajo al mismo tiempo. Por su capacidad de análisis y eficiencia en la predicción son usadas en las diferentes esferas sociales y de producción tales como la medicina, la educación, la banca y la ingeniería. Resuelven problemas complejos relacionados a la ciberseguridad (Podder et al., 2021) y son utilizadas en aplicaciones para el reconocimiento de voz (Fandos Villanueva & Civera Sancho, 2020), el reconocimiento de patrones (Schilling & Stadelmann, 2020) y la estimación en los proyectos de desarrollo de software.

Resultados y discusión

Las RNA de orden superior utilizan combinaciones superiores de entrada y funciones de activación de orden superior. Proporcionan nodos en la capa de entrada con un conocimiento completo de sus patrones y sus relaciones, las entradas se convierten en una forma matemática bien entendida. Los resultados de estas son más precisos en comparación con las redes neuronales convencionales y requieren un período de entrenamiento más corto para adquirir la precisión deseada (Kumar et al., 2020). Por otro lado, las RNA de aprendizaje profundo, son redes de varias capas, con un cierto nivel de complejidad. Toma conjuntos de relaciones tanto lineales como no lineales como entradas y lo convierte en salida, mediante el uso de un conjunto de principios matemáticos. La probabilidad de la salida, se calcula en cada capa a medida que se mueve hacia adelante. Al principio, asigna pesos aleatorios a las conexiones entre las neuronas. Las unidades de entrada y los pesos se multiplican para obtener la salida. Si la salida deseada no es adquirida, entonces los pesos se ajustan usando algoritmos de aprendizaje. Permite el aprendizaje del modelo tanto en el procesamiento secuencial y paralelo de la información (Yu et al., 2022). La tabla 1 muestra los resultados de comparación entre los tipos de redes neuronales (Wavelet, Spiking, FLANN y ELMAN), basados en los estudios de (Benala & Dehuri, 2022; Kaushik & Singal, 2019; Song et al., 2021; Yamazaki et al., 2022)

Tabla 1. Comparación RNA de orden superior/ RNA de aprendizaje profundo

	Wavelet	FLANN	Spiking	ELMAN
Aplicabilidad en el proceso de desarrollo	Se utilizan para representar datos o	Aumentan las dimensiones del	Consisten en transformar la	Tienen neuronas tansig en su capa oculta

	<p>aproximar funciones. Se emplean como funciones de análisis y examinan la señal de interés para obtener sus características de espacio, tamaño y dirección.</p>	<p>espacio de entrada utilizando una combinación no lineal de de entrada, para manejar problemas linealmente no separables</p>	<p>información en el proceso de comunicación, permitiendo su uso en problemas no muy complejos de almacenamiento de información a largo plazo.</p>	<p>(recurrente) y neuronas purelin en su capa de salida. Con esta combinación son capaces de aproximar cualquier función (con un número finito de discontinuidades) con una precisión arbitraria.</p>
<p>Parámetros del desempeño</p>	<p>Es una red de 3 capas, la capa inferior representa la entrada, la capa intermedia representa la capa oculta y la capa superior representa la capa de salida. Requieren un menor número de coeficientes para concentrar la energía de una función y mejorar la aproximación.</p>	<p>Consta de dos capas con algoritmos de aprendizaje adecuados en cada estado. Aplica una función de expansión que aumenta la dimensionalidad del vector de entrada. Los hiperplanos generados proporcionan una mayor capacidad de discriminación en el espacio de patrones de entrada.</p>	<p>Se basan en el proceso de comunicación que utilizan las neuronas para la transformación de la información. En las neuronas biológicas, los picos se generan cuando el potencial de membrana alcanza un cierto umbral en un punto particular. Cuando la neurona se dispara, genera una señal (entrada sináptica) que se transmite a otras neuronas. La simulación de grandes redes suele</p>	<p>Es una red feedforward en la que la capa oculta, además de estar conectada a la capa de salida, se bifurca en otra capa idéntica, llamada capa de contexto, a la que está conectada con pesos iguales a uno. Esta conexión recurrente permite a la red detectar y generar patrones variables en el tiempo. Dado que la red puede almacenar información para futuras referencias, es capaz de aprender patrones temporales y</p>

			consumir demasiado tiempo.	espaciales por lo que puede entrenarse para responder y generar ambos tipos de patrones.
Procesamiento en la capa oculta	<p>Los pesos se modifican utilizando un algoritmo de aprendizaje y los coeficientes de wavelet se suman en la capa de salida para dar el resultado final.</p> <p>Las unidades en la capa oculta se conocen como wavelons y están definidos por los parámetros de dilatación, traslación y peso.</p>	<p>Figura sólo dos capas con mapeo no lineal (entrada-salida), siempre que sean alimentados con un conjunto adecuado de insumos funcionales.</p> <p>Este enfoque elimina la capa oculta para ayudar a reducir la complejidad de la arquitectura y proporciona una representación mejorada de los nodos de entrada para que la red pueda realizar tareas de clasificación no lineal separable.</p>	<p>Se componen de neuronas en espigas con sinapsis interconectadas y pesos escalares ajustables.</p> <p>Al recibir la señal sináptica, una neurona puede ser de naturaleza excitatoria (aumenta el potencial de membrana) o inhibidora (disminuye el potencial de membrana). Los pesos se ajustan y cambian como resultado del aprendizaje</p>	<p>El número de neuronas en la capa de contexto es igual al número de neuronas en la capa oculta. La capa de contexto se utiliza para recordar la salida de la capa oculta, por lo que es considerada como un operador de retardo. Esta salida se asocia con su entrada a través del retardo y el almacenamiento de la capa de contexto. Esta forma de asociación es sensible a los datos históricos, y la red de retroalimentación interna la cual, puede aumentar la capacidad de transmitir información dinámica lo que hace que el sistema se adapte a las</p>

				características variables en el tiempo.
Capa de salida	Es la combinación lineal de funciones wavelet de cada wavelons. Estas funciones pueden ser: Morlet, Gaussiana, Mexican Hat, transformadas, entre otras.	Se calcula con la suma ponderada de las entradas que pasan a través de una función de activación para producir la salida de la red deseada.	Las salidas se producen con el aumento o disminución de la señal, la cual determina el potencial de las respuestas en las neuronas.	La salida de la capa oculta se envía tanto a la capa de contexto como a la de salida para capitalizar esta información en las interacciones posteriores. De este modo, la red recuerda repetidamente la salida de la capa oculta de las iteraciones anteriores, lo que le permite conservar su memoria a corto plazo y mejorar el rendimiento de del sistema. Las salidas de la red son independientes entre sí y dependen únicamente de del paso de tiempo actual.

Fuente: Elaboración propia

Las investigaciones abordadas para la comparación de las RNA evidencian que este tipo de sistemas son eficientes para estimar el esfuerzo, como muestra el primer criterio. Las demás características destacan una ligera superioridad a las RNA de orden superior. Las de tipo Wavelet se caracterizan por necesitar menor cantidad de

neuronas para la aproximación de funciones facilitando un menor tiempo en su respuesta. Además, las funciones wavelet deben satisfacer ciertos requerimientos matemáticos, son usadas para representar datos y surgen como respuesta en la aproximación de señales discontinuas, mediante funciones más apropiadas que los senos y cosenos, dado que estas son no locales y se extienden hasta el infinito, teniendo como consecuencia un desempeño pobre en la detección de cambios repentinos. Por su parte, el rendimiento de los modelos FLANN es notable en términos de tiempo de procesamiento lo que demuestra su eficacia en tareas de clasificación. Se ha demostrado que esta red para la aproximación funcional y la clasificación de patrones obtiene una velocidad de convergencia más rápida y con menor carga computacional. Como no tiene capa oculta los cálculos y el algoritmo de aprendizaje utilizados en esta red son sencillos. Por su parte los sistemas de aprendizaje profundo de manera general, procesan los datos de forma jerárquica. Obtienen representaciones más significativas de los datos mediante el aprendizaje por capas, lo que suele consumir más tiempo. Sin embargo un estudio realizado por (Kumar et al., 2020), muestra que la comunidad científica apoya los sistemas de aprendizaje profundo y orden superior (en ese orden), por encima de otros métodos de RNA clásicos para el cálculo del esfuerzo. Es necesario aclarar que este tipo de sistemas no funcionan igual para diferentes conjuntos de datos de entrada. Sin embargo, pueden resolver problemas complejos de manera eficaz sobre todo si son combinadas con algoritmos de optimización, no solo en la industria del software sino en otros campos de la ciencia.

Conclusiones

Existen diversos artículos que abordan sobre el funcionamiento y la utilización de los diferentes tipos de redes neuronales artificiales utilizadas en la estimación distintos proyectos. El objetivo de esta investigación fue demostrar su eficiencia en la predicción del esfuerzo en el proceso de desarrollo del software, aportando una sistematización novedosa en este campo. La estimación precisa ayuda a una mejor planificación y ejecución de un proyecto de software, lo equivale a una reducción de costo, tiempo y es proporcional a aumentar la satisfacción del cliente. La utilización de sistemas como las RNA ha permitido resolver problemas complejos en diversos campos de la ciencia, demostrando así la importancia de su manejo en la industria de software. Todas las investigaciones mencionadas han obtenido resultados eficientes, sin embargo, el área de aprendizaje automático debería centrarse en estimaciones más precisas que no obstaculicen el proceso de desarrollo y evalúen factores que afecten la calidad del producto para obtener mejores resultados.

Referencias

- AKGÜL, İ., & KAYA, V. (2022). *A REVIEW ON ARTIFICIAL NEURAL NETWORKS*. Rumeli 1ST International Scientific Research Conference On Sustainable Engineering And Technology.
- Barik, S. K., Mohapatra, S., & Debdas, S. (2022). Multi-Neuron Functional Link Artificial Neural Network: A Novel Architecture and its Performance for Wind Energy Prediction. *2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*, 447-451. <https://doi.org/10.1109/ICSCDS53736.2022.9760812>
- Benala, T. R., & Dehuri, S. (2022). Tuning Functional Link Artificial Neural Network for Software Development Effort Estimation. En M. Panda, S. Dehuri, M. R. Patra, P. K. Behera, G. A. Tsihrintzis, S.-B. Cho, & C. A. Coello Coello (Eds.), *Innovations in Intelligent Computing and Communication* (pp. 62-81). Springer International Publishing. https://doi.org/10.1007/978-3-031-23233-6_5
- Bilgaiyan, S., Mishra, S., & Das, M. (2019). Effort estimation in agile software development using experimental validation of neural network models. *International Journal of Information Technology*, 11(3), 569-573. <https://doi.org/10.1007/s41870-018-0131-2>
- Fandos Villanueva, A., & Civera Sancho, J. (2020). *Reconocimiento de vocalistas mediante redes neuronales profundas* [Grado, Universidad de Zaragoza]. <https://zaguan.unizar.es/record/96551?ln=es>
- Fernández-Sánchez, H., King, K., & Enríquez-Hernández, C. B. (2020). Revisiones Sistemáticas Exploratorias como metodología para la síntesis del conocimiento científico. *Enfermería Universitaria*, 17(1). <https://doi.org/10.22201/eneo.23958421e.2020.1.697>
- Group, T. S. (2019). *The Standish Group Report Chaos*. 16.
- Haloj, N., Goyal, T., Zahoor, F., Jain, H., & Wali, R. S. (2021). *Estimation of cost overrun in construction projects using Fuzzy Logic*. 23(5).

- Hernández Sampieri, R., & Mendoza Torres, C. P. (2018). *Metodología de la investigación: Las rutas: cuantitativa, cualitativa y mixta*. Mc Graw Hill educación. <http://repositorio.uasb.edu.bo/handle/54000/1292>
- HEZAM, T. A. A. A. (2021). *Software project management*. <https://doi.org/10.6084/m9.figshare.14368646.v1>
- Huang, S.-J., & Chiu, N.-H. (2006). *Optimization of analogy weights by genetic algorithm for software effort estimation*. *48*, 1034-1045. <https://doi.org/10.1016/j.infsof.2005.12.020>
- Jung, S., Pyeon, J.-H., Lee, H.-S., Park, M., Yoon, I., & Rho, J. (2020). Construction Cost Estimation Using a Case-Based Reasoning Hybrid Genetic Algorithm Based on Local Search Method. *Sustainability*, *12*(19), 7920. <https://doi.org/10.3390/su12197920>
- Kaushik, A., & Singal, N. (2019). A hybrid model of wavelet neural network and metaheuristic algorithm for software development effort estimation. *International Journal of Information Technology*, *14*(3), 1689-1698. <https://doi.org/10.1007/s41870-019-00339-1>
- Kumar, P. S., Behera, H. S., K, A. K., Nayak, J., & Naik, B. (2020). Advancement from neural networks to deep learning in software effort estimation: Perspective of two decades. *Computer Science Review*, *38*, 100288. <https://doi.org/10.1016/j.cosrev.2020.100288>
- Latif, A., Fitriana, L. A., & Firdaus, M. R. (2021). COMPARATIVE ANALYSIS OF SOFTWARE EFFORT ESTIMATION USING DATA MINING TECHNIQUE AND FEATURE SELECTION. *JITK (Jurnal Ilmu Pengetahuan Dan Teknologi Komputer)*, *6*(2), Article 2. <https://doi.org/10.33480/jitk.v6i2.1968>
- León, D. P., & Hondares, Y. G. (2019). *Título: "Herramienta para la estimación del esfuerzo de desarrollo de proyectos de software a partir de modelos de procesos de negocio. SDEffort-BPMN"* [Grado, Universidad Central Marta Abreu de las Villas]. <https://dspace.uclv.edu.cu/bitstream/handle/123456789/12171/tesis%20Daydee.pdf?sequence=1&isAllowed=y>

- Mahadev, M., & Gowrishankar, G. (2020). Estimation of Effort in Software Projects using Genetic Programming. *International Journal of Engineering Research & Technology*, 9(7).
- Marquez, M. C. (2020). Cálculo del costo de proyectos de desarrollo informáticos en la Universidad de las Ciencias Informáticas. *Serie Científica de la Universidad de las Ciencias Informáticas*, 13(11), Article 11.
- Mohsin, Z. R. (2021). APPLICATION OF ARTIFICIAL NEURAL NETWORKS IN PREDICTION OF SOFTWARE DEVELOPMENT EFFORT. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12(14), Article 14.
- Mulder, H. (2020). *The Chaos Report*.
- Ortega-Hernandez, V. de J. (2019). *Diseño de una metodología para la estimación de esfuerzos en proyectos de consultoría en innovación*. [Maestría, Universidad Internacional de La Rioja]. <https://reunir.unir.net/handle/123456789/8280>
- Podder, P., Bharati, S., Mondal, M. R. H., Paul, P. K., & Kose, U. (2021). *Artificial Neural Network for Cybersecurity: A Comprehensive Review*. <https://doi.org/10.48550/arXiv.2107.01185>
- Rankovic, D., Rankovic, N., Ivanovic, M., & Lazic, L. (2021). Convergence rate of Artificial Neural Networks for estimation in software development projects. *Information and Software Technology*, 138, 106627. <https://doi.org/10.1016/j.infsof.2021.106627>
- Rankovic, N., Rankovic, D., Ivanovic, M., & Lazic, L. (2021). A New Approach to Software Effort Estimation Using Different Artificial Neural Network Architectures and Taguchi Orthogonal Arrays. *IEEE Access*, 9, 26926-26936. <https://doi.org/10.1109/ACCESS.2021.3057807>
- Robiolo, G. (2016). *Estimación de proyectos de software pequeños basada en el juicio de expertos: Un caso de estudio*. 12.
- Roger S. Pressman & Bruce R. Maxim. (2015). *Software Engineering. A PRACTITIONER'S APPROACH* (EIGHTH EDITION). McGraw-Hill Education,.

[https://raw.githubusercontent.com/bernardvery/Genap1718/master/Ebook/Software%20Engineering%20A%20Practitioner%E2%80%99s%20Approach%20eighth%20edition-\(www.downloadnema.com\).pdf](https://raw.githubusercontent.com/bernardvery/Genap1718/master/Ebook/Software%20Engineering%20A%20Practitioner%E2%80%99s%20Approach%20eighth%20edition-(www.downloadnema.com).pdf)

Schilling, F.-P., & Stadelmann, T. (Eds.). (2020). *Artificial Neural Networks in Pattern Recognition: 9th IAPR TC3 Workshop, ANNPR 2020, Winterthur, Switzerland, September 2–4, 2020, Proceedings* (1st ed. 2020 edition).

Springer. <https://www.amazon.com/Artificial-Neural-Networks-Pattern-Recognition/dp/3030583082>

Sharma, A., & Chaudhary, N. (2020). *Software Cost Estimation for Python Projects Using Genetic Algorithm* (pp. 137-148). https://doi.org/10.1007/978-981-15-3325-9_11

Solimán, E. R. M. (2018). *Estimación de esfuerzo en proyectos de desarrollo de software con metodologías ágiles* [Maestría, ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES]. <https://riunet.upv.es/handle/10251/109927>

Sommerville, I. (2016). *Software Engineering* (Tenth Edition). Pearson Education.

Song, K., Lv, S., Hu, D., & He, P. (2021). Software Defect Prediction Based on Elman Neural Network and Cuckoo Search Algorithm. *Mathematical Problems in Engineering*, 2021, e5954432. <https://doi.org/10.1155/2021/5954432>

Tavanaei, A., Ghodrati, M., Kheradpisheh, S. R., Masquelier, T., & Maida, A. (2019). Deep learning in spiking neural networks. *Neural Networks*, 111, 47-63. <https://doi.org/10.1016/j.neunet.2018.12.002>

Trendowicz, A., & Jeffery, R. (2014). *Software Project Effort Estimation: Foundations and Best Practice Guidelines for Success*. Springer International Publishing. <https://doi.org/10.1007/978-3-319-03629-8>

Vasylykiv, N., Dubchak, L., Sachenko, A., Lendyuk, T., & Sachenko, O. (2020). *Fuzzy Logic System for IT Project Management*.

Villalobos, L. C. S. (2018). *MEDICIÓN DEL TAMAÑO FUNCIONAL EN EL DESARROLLO DE SOFTWARE DIRIGIDO POR MODELOS* [Maestría]. Ciudad Universitaria Rodrigo Facio.

- Wright, W. C. (2021). *DISEÑO DE UNA RED NEURONAL ARTIFICIAL PARA EL MODELO DE ESTIMACIÓN DE PROYECTOS DESARROLLO DE SOFTWARE EN LA UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS* [Maestría]. Universidad de las Ciencias Informáticas.
- Wright, W. C., & Casañola, Y. T. (2021a). *Mejora del modelo de estimación del esfuerzo en proyectos de la Universidad de las Ciencias Informáticas*. 15.
- Wright, W. C., & Casañola, Y. T. (2021b). Redes Neuronales Artificiales en la estimación del esfuerzo. *Revista Cubana de Ciencias Informáticas*, 15(2). [https://rcci.uci.cu/?journal=rcci&page=article&op=view&path\[\]=2108](https://rcci.uci.cu/?journal=rcci&page=article&op=view&path[]=2108)
- Yamazaki, K., Vo-Ho, V.-K., Bulsara, D., & Le, N. (2022). Spiking Neural Networks and Their Applications: A Review. *Brain Sciences*, 12(7), Article 7. <https://doi.org/10.3390/brainsci12070863>
- Yu, H., Qi, Y., & Ding, Y. (2022). *Deep Learning in RNA Structure Studies*. 9. <https://doi.org/10.3389/fmolb.2022.869601>