

Tipo de artículo: Artículo original
Temática: Inteligencia Artificial
Recibido: 30/06/2021 | Aceptado: 01/10/2021

Mejora del modelo de estimación del esfuerzo en proyectos de la Universidad de las Ciencias Informáticas

Improvement of the effort estimation model in projects of the University of Informatics Sciences

Wisleidys Campos Wright^{1*} <https://orcid.org/0000-0003-0617-467X>

Yaimí Trujillo Casañola ² <https://orcid.org/0000-0002-3138-011x>

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

*Autor para la correspondencia. (wcamposw@uci.cu)

RESUMEN

La planificación en el mundo de 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. En junio del 2016, el proceso productivo de la Universidad de las Ciencias Informáticas estuvo siendo evaluado por el Software Industry Excellence Center de México en conjunto con Especialistas de la universidad. Se certificó el nivel 2 del

Capability Maturity Model Integration, detectándose brechas en el área de Planeación de Proyecto relacionadas a la estimación del esfuerzo. En ese contexto se propone la construcción de un modelo de estimación para el esfuerzo en el desarrollo de software adaptándolo a los distintos escenarios con que cuenta la universidad, que pretende mejorar la precisión utilizando una Red Neuronal Artificial de tipo Wavelet Radial Híbrida que combina los conceptos de un sistema heurístico de Búsqueda local y el algoritmo Colonia de abejas. Se utiliza el método Delphi y los criterios de expertos para validar la contribución de la propuesta. Se concluye que la técnica propuesta ayuda al proceso de estimación de los proyectos de desarrollo de la universidad.

Palabras clave: desarrollo de software; planificación; redes neuronales artificiales; estimación del esfuerzo.

ABSTRACT

Planning in the world of Software Industry has become one of the main challenges for project management, and a fundamental activity to develop high quality software. It requires estimates that guarantee the efficiency of the product. The application of good estimation methods is essential for planning because of the increasing influence they have on accurate and predictable control, production processes and software products. In June 2016, the production process of the University of Informatics Sciences was being evaluated by the Software Industry Excellence Center of Mexico in conjunction with specialists from the university. It was certified as Level 2 of the Capability Maturity Model Integration, detecting gaps in the Project Planning area related to effort estimation. In this context, the construction of an estimation model for the effort in software development is proposed, adapted to the different scenarios that the university has, which aims to improve accuracy using an Artificial Neural Network of Hybrid Radial Wavelet type, combining the concepts of a heuristic system of Local Search and the Colony of Bees algorithm. The Delphi method and expert criteria are used to validate the contribution of the proposal. It is concluded that the proposed methodology helps the estimation process of university software development projects.

Keywords: software development; project planning; artificial neural networks; effort estimation; learning algorithm.

Introducción

El éxito de un proyecto de desarrollo de software depende de que el producto obtenido cumpla con las especificaciones del usuario y se termine dentro del plazo y con el presupuesto establecido. En ocasiones pueden manifestarse determinados problemas en el proceso de estimación y planificación del proyecto (Huanca y Oré 2017). Pressman, define la estimación como la “necesidad real para 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” (Pressman y Maxim 2015). Por otra parte, el esfuerzo es una combinación de persona y tiempo. Se refiere a la cantidad de tiempo que una persona necesitaría para completar cierto trabajo productivo (Trendowicz y Jeffery 2014). Lograr una estimación en etapas temprana del proyecto nos permitirá tomar decisiones importantes aun cuando exista cierta incertidumbre, pero se requiere de experiencia y seguir un enfoque invariable. Pressman expresa que, “si se genera estimaciones usando datos históricos sólidos, si se establece un calendario realista y continuamente lo adapta conforme el proyecto avanza, se puede estar seguro de estar haciendo un mejor esfuerzo” (Pressman y Maxim 2015).

Con el avance de la tecnología, la tendencia para lograr una mayor exactitud ha sido combinar diferentes métodos de estimación, apelando a técnicas de inteligencia artificial tales como lógica difusa, sistemas basados en conocimiento, programación genética y redes neuronales artificia (Saini, Ahuja y Khatri 2018; Saruwatari et al. 2019; Sharma y Chaudhary 2020; Yousef, Alshaer y Alhammad 2017).

En Cuba se ejecutan estrategias para fortalecer la industria cubana del software, dando avances en sus procesos bajo los estándares de calidad. Una de las estrategias puestas en marcha fue la creación de la Universidad de las Ciencias Informáticas (UCI). En junio del 2016 la UCI fue certificada con el nivel 2 de CMMI (Capability Maturity Model Integration), no obstante, en su informe final la SIE Center (Software Industry Excellence Center) de México identificó como oportunidades de mejora en el área de Planeación de proyectos que el esfuerzo obtenido del modelo de estimación se distribuye en tareas en base a la experiencia del líder de proyecto y que el modelo de estimación no permite estimar el esfuerzo total de todos los proyectos. Un estudio realizado demostró que aún estos problemas persisten y continúan siendo un factor de

riesgo para la estimación correcta de los productos desarrollados en la universidad. La presente investigación propone una mejora al modelo de estimación de la UCI utilizando una Red Neuronal Artificial (RNA), que garantice la estimación del esfuerzo de los proyectos de desarrollo de software.

Métodos o Metodología Computacional

La investigación realizada sigue la ruta cuantitativa con un enfoque descriptivo según (Hernández-Sampieri y Torres 2018). El diseño de la misma está concebido como no experimental. Como el objetivo del estudio es proponer una mejora al método de estimación de la UCI, no se pretende manipular las variables, sino más bien se realiza el análisis de las mismas tal como se dan en su contexto natural. Para la obtención de información de la investigación se emplearon varios métodos científicos:

Métodos Teóricos

1. Analítico –Sintético: se empleó para extraer los elementos más relevantes que brinda la bibliografía, posibilitando el análisis del uso de las RNA en el ámbito de la estimación en proyectos de desarrollo de software.
2. Inductivo – Deductivo: se utiliza en el diagnóstico de la tendencia del uso de la RNA en la industria del software para identificar lo común en fenómenos individuales y a partir de ello realizar un razonamiento deductivo que contribuya a las ventajas de su uso en la planificación de proyectos.
3. La modelación: se utilizó en el diseño y modelación de la propuesta y sus componentes.

Métodos Empíricos

1. Observación participante: se utilizó para conocer la realidad que se estudia mediante la percepción directa, así como realizar la confrontación de los resultados obtenidos.
2. Encuesta: para obtener el diagnóstico de la situación actual de los proyectos de desarrollo de la universidad en relación a la manera en que ejecutan el modelo de estimación y para validar los resultados de la investigación.
3. Entrevistas: para obtener información que contribuya a argumentar la situación problemática.

Para conocer el estado del método de estimación actual que se utiliza en la UCI, se encuestaron y realizaron entrevistas de manera presencial a 15 especialistas de los centros de desarrollo que dentro de sus funciones se encontraba la realización del método de estimación en sus proyectos, además a 1 profesor de Ingeniería de Software (ISW) vinculado a la producción que aportó elementos importantes para profundizar en el funcionamiento del método en cuestión. Los años de experiencia oscilan entre los 2 y 5 años, y habían ejecutado el método de estimación de 2 a 4 años en distintos proyectos. Entre los roles que han desempeñado se encontraban: Líder de proyecto y Planificador

Los análisis de las encuestas arrojaron que el método de estimación utilizado, no es suficiente real en proyectos de gran complejidad y no cumple con las exigencias de los productos actuales, debido a que no es adaptable a proyectos que abarcan negocios muy amplios, donde las tecnologías y la arquitectura son complejas. Las métricas de tamaño no toman en cuenta las características propias de los proyectos, pues están basadas en el levantamiento de información para la creación de los paquetes funcionales a los cuales se le asigna una clasificación de pequeño, mediano o grande según su complejidad, esto trae como consecuencia que las entradas no sean ajustables a las particularidades de los centros de desarrollo. Las métricas del tiempo tienen un porcentaje de valor absoluto que no siempre permite la captura del esfuerzo real de cada actividad, esto deriva en que las horas hombres se colocan como un aproximado en dependencia de las actividades de los profesionales, si están o no doblemente vinculados a las actividades docentes- productivas y en algunos casos se introduzcan a partir de la experiencia del líder del proyecto y la cantidad de recursos asignados al mismo. A partir de los resultados de las encuestas y las entrevistas, se valoró la posibilidad de optimizar el método actual de tal manera que se ajuste los indicadores de estimación en un nivel más detallado. En el marco de la investigación, optimizar implica encontrar la mejor solución posible para un problema determinado. En este contexto y partiendo de la hipótesis: Si se propone una técnica basada en Redes Neuronales Artificiales, se obtendrá un instrumento válido para estimar el esfuerzo total de los proyectos de desarrollo de la UCI, se define como optimización del método: al proceso mediante el cual se logra obtener el esfuerzo total de los proyectos de desarrollo y que sea capaz de adaptarse a las características propias de los mismos. Se concluyó que contar con alternativas como las técnicas que usan

la categoría de aprendizaje automático permitirá obtener un nivel de certeza elevado. En ese sentido se decide que la utilización de métodos híbridos podría a corto plazo garantizar una estimación del esfuerzo eficaz y segura, permitiendo reducir la incertidumbre. Las RNA han ganado popularidad en la industria del software, por su capacidad de aumentar la precisión en la estimación del esfuerzo y tamaño de proyectos de software unidas a algoritmos de optimización.

Redes Neuronales Artificiales en la estimación de proyectos de software

Las RNA imitan la operación básica del cerebro. La información viaja entre las neuronas y, basada en la estructura y ganancia de los conectores neuronales, la red se comporta de manera diferente. En estas redes cada neurona está conectada con otra por medio de un peso o coeficiente de ajuste (Arnal 2018; Nielsen 2018). El procedimiento de entrenamiento de una RNA consiste en iterativamente suministrar a la red una secuencia de patrones de entrada, y ajustar los pesos de las conexiones en función de las salidas obtenidas. Se distinguen tres categorías de aprendizaje: Aprendizaje supervisado, por refuerzo y no supervisado, estos a su vez utilizan una variedad de algoritmos, dependiendo de la información que se posea sobre los patrones de entrada (Aggarwal 2019). Diversas predicciones relativas al desarrollo de software se encuentran ligadas a las redes neuronales artificiales para estimar tamaño, esfuerzo y costo. Para predecir el tamaño se puede encontrar trabajos como los de (Aggarwal y Singh 2005), donde se hacen uso de la regularización bayesiana de una RNA para estimar el tamaño del software mediante línea de códigos (SLOC) usando puntos de función y el de (Ajitha y Geetha 2010), quienes utilizan una RNA para estimar el tamaño usando puntos de caso de uso. Para proyectar el costo (DHANOPIYA, et al. 2017) utiliza una RNA multicapa y (Venkataiah, Mohanty y Nagaratna 2018) usa una RNA de tipo Spiking para operar en la misma dirección. Son abundantes los trabajos en el uso de las RNA para la estimación del esfuerzo, son los casos de los trabajos de (Azath, Mohanapriya y Rajalakshmi 2018; Kodmelwar, Joshi y Khanna 2018; Rao, Reddi y Rani 2017), los mismos combinan las RNA con diferentes algoritmos para optimizar y mejorar el aprendizaje. Todas estas investigaciones demostraron la efectividad y ventajas de la utilización de este tipo de sistemas. Sin embargo, aún estos enfoques no han alcanzado del todo la madurez que se necesita.

Proceso de desarrollo de software en la UCI

El proceso de desarrollo de software en la UCI está estructurado por un Sistema de Gestión de la Calidad y una metodología de desarrollo que integra prácticas de diferentes marcos de trabajo o metodologías, con el objetivo de insertarse de forma estándar en la producción de software en los distintos centros de la universidad. Esta metodología de desarrollo de software se denomina AUP-vUCI (Proceso Unificado Ágil-versión Universidad de las Ciencias Informáticas) (Paez 2017). Para el Modelado de negocio se proponen tres variantes a utilizar en proyectos (Casos de Uso del Negocio (CUN), Descripción de Procesos de Negocio (DPN) o Modelo Conceptual (MC)) y para encapsular los requisitos (Casos de Uso del Sistema (CUS), Historias de Usuario (HU) y Descripción de Requisitos por Procesos (DRP)). De ahí que, surgen los siguientes cuatro escenarios para modelar el sistema en los proyectos: (Escenario No. 1): $CUN + MC = CUS$, (Escenario No. 2): $MC = CUS$, (Escenario No. 3): $DPN + MC = DRP$, (Escenario No. 4): HU.

Resultados y discusión

El modelo de estimación de la UCI tiene el inconveniente de que no son adaptables a la variedad de proyectos y escenarios de la universidad, no permite la captura del esfuerzo total de las actividades que están dentro de las disciplinas por cada fase establecida en la metodología (Inicio, Ejecución y Cierre), lo que trae como consecuencia que en ocasiones la diferencia entre el plan y la ejecución real sea significativa. Según el trabajo de (Huang y Chiu 2006), uno de los pilares básicos para una correcta estimación de un proyecto software es la estimación del esfuerzo asociado a su desarrollo. En acuerdo con el planteamiento anterior, se considera que el esfuerzo permite predecir el tiempo de duración del software en meses/hombres. Como el tamaño es un derivado del esfuerzo teniendo en cuenta las actividades, los recursos, la programación y las limitaciones internas y externas (Pressman y Maxim 2015), esta investigación propone la utilización de una RNA Wavelet Radial híbrida (RWRH), que combina los conceptos de un sistema heurístico de búsqueda local y el algoritmo Colonia de abejas (ABC) que permitirá predecir el esfuerzo total tomando en cuenta los escenarios para modelar el sistema en los proyectos de desarrollo. Figura 1.

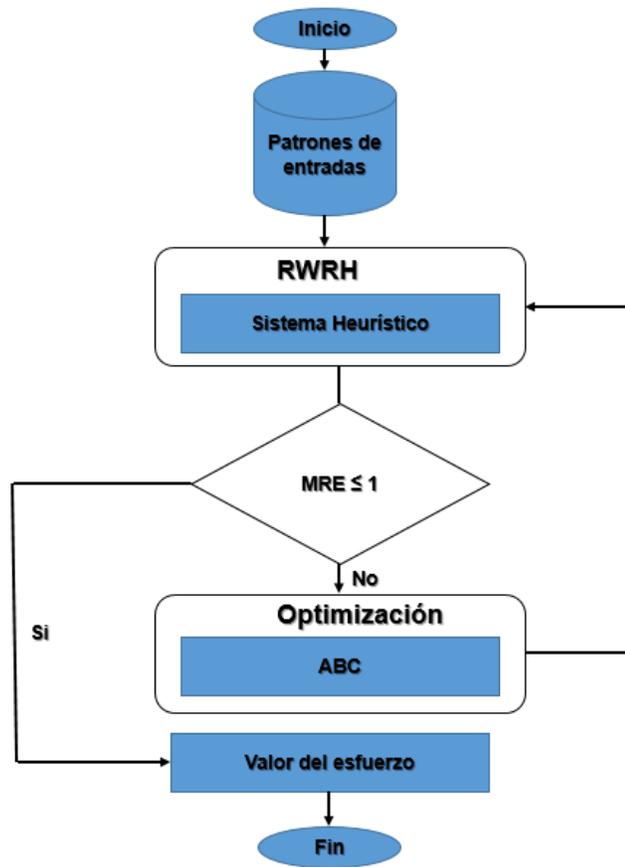


Fig. 1 – Estructura del método propuesto.

RNA Wavelet Radial

Las RNA de tipo wavelet (fig. 2), Combinan la teoría de procesamiento wavelet con las redes neuronales. En esta arquitectura, la capa oculta está formada por wavelons cuyos parámetros de entrada están definidos por peso, parámetro de traslación y dilatación. La salida de la red es una combinación lineal de las funciones wavelet de cada wavelon. Las wavelets son familia de funciones que se encuentran en el espacio y se emplean como funciones de análisis, examinan a la señal de interés para obtener sus características de espacio, tamaño y dirección. Las funciones de transferencias pueden ser Morlet, Gaussiana, Sombrero mexicano entre otras (Kumar et al. 2020). En este trabajo se hace uso de la función Gaussiana como función de activación, la cual se define de la siguiente manera en (1):

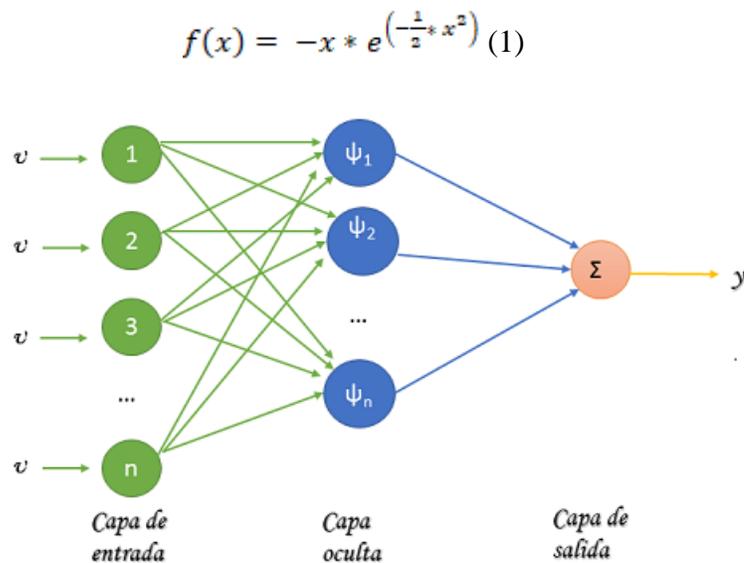


Fig. 2 – Estructura de la red Wavelet Radial.

La ecuación típica de una red wavelet (Firdous Ahmad Shah y Debnath 2017) está dada por (2):

$$\varphi^{\lambda, \tau}(x) = \frac{1}{\sqrt{\lambda}} \varphi\left(x - \frac{\tau}{\lambda}\right) \quad x \in \mathbb{R} \quad (2)$$

Donde $\varphi(x)$, es una función wavelet madre y $\varphi^{\lambda, \tau}(x)$ es el hijo formada por los parámetros de dilatación y traslación. Estos parámetros son inicializados en el rango de $[0 - 1]$ para los nodos en la capa oculta.

El número de entradas de la red viene determinado por el conjunto de atributos que definen el problema en cuestión, mientras que el número de salidas es el mismo que el número de valores a predecir. Para aproximar el número de las neuronas ocultas, se ha elegido la aproximación matemática de estimación descrita por (Tarassenko 1998) en (3)

$$j = \sqrt{ik} \quad (3)$$

Donde i es el número de neuronas de la capa de entrada, j el número de neuronas de la capa oculta y k el número de neuronas de la capa de salida.

Los atributos de entrada de la red se definen a partir de una serie de factores característicos que son comunes en los proyectos y que ayudan a predecir el esfuerzo. Se incluyen variables que aportan información del proyecto, la tabla 1 brinda información al respecto.

Tabla 1 – Atributos de entrada.

Atributos	Características
Recursos del proyecto	Recursos requeridos para acometer el esfuerzo como los componentes reutilizables, recursos materiales y las personas.
Tipo de arquitectura	Conjunto de patrones que proporciona un marco de trabajo para guiar la construcción del software.
Tipo de lenguaje	Nivel del lenguaje de programación utilizados por los especialistas para construir las aplicaciones.
Gestor de Base de Datos	Si utilizan un sistema de Gestor de Base de Datos.
Herramientas CASE	Si utilizan algunas herramientas que permite aumentar la productividad de su desarrollo.
Escenario (A, B, C, D)	los 4 tipos de escenario para modelar el sistema en los proyectos de la UCI

Fuente: Elaboración Propia

Sistema heurístico de búsqueda local

El Sistema heurístico propone buscar la solución óptima para cada escenario. A partir de los datos de entradas el sistema se encarga de evaluar el escenario asignándole un valor real pequeño cuanto mejor sea la

solución óptima para el mismo. Finalmente se elige la mejor solución como aquel que tenga la evaluación mínima y devuelve el mejor resultado para ese escenario. Figura 3.

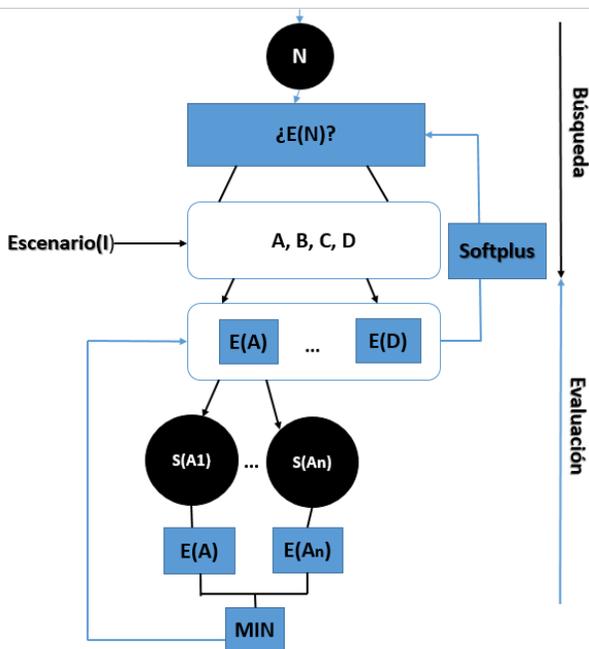


Fig. 3 – Sistema heurístico de búsqueda local

La función heurística recibe como entrada (N) escenario a evaluar para el cálculo del esfuerzo y el identificador del escenario I (A, B, C, D) de los 4 que pueden salir, se le asigna una evaluación (E) correspondiente a la mejor opción alcanzada para el esfuerzo, obtenida de forma habitual como el resultado mínimo de las evaluaciones de los escenarios mediante todas las posibles soluciones S11...SIn. La evaluación de las posibles soluciones se calcula mediante la función de activación Softplus usada en su capa de salida que devuelve un vector de probabilidades mediante (4):

$$[P_1, \dots, P_n] \text{ con } \sum_{i=1}^n P_i = 1 \quad (4)$$

En la que cada P_i , representa la probabilidad asignada por el Sistema heurístico de que el resultado del esfuerzo sea el óptimo.

Algoritmo de optimización Colonia Artificial de Abejas (ABC)

El algoritmo ABC simula el comportamiento de la búsqueda de alimento de las abejas. Consiste en crear un vecindario, donde las abejas comienzan a moverse buscando la mejor solución al patrón dado en donde se resuelve el problema de minimización. En una colonia hay 3 tipos de abejas: exploradoras, empleadas y observadoras. El trabajo que supone la búsqueda de alimento se divide en tres equipos que incluye la búsqueda del néctar, el intercambio de información sobre la fuente de alimento y la recogida del néctar. Una posición de fuente de alimento se refiere a una posible solución para el problema de optimización, mientras que la cantidad de néctar de una fuente de alimento se refiere a la calidad (Fitness) de la solución. El objetivo del algoritmo está configurado para determinar la mejor solución la Figura 4 muestra el pseudocódigo del mismo. (Azath, Mohanapriya y Rajalakshmi 2018; Rao, Reddi y Rani 2017).

```
Begin
Generar de forma aleatoria N fuentes de alimento en  $X_1, \dots, X_n$ 
Repeat
  For  $i=1$  a  $N$  do
    Enviar una abeja empleada a la fuente de alimento  $X_i$ 
    Modificar la fuente de alimento  $X_i$  y determinar la cantidad de néctar de la nueva solución  $X_{oi}$ 
    if la nueva solución es mejor else
       $X_{oi} \leftarrow X_i$ 
    END
  END
  For  $i=1$  a  $N$  do
    Calcular la probabilidad de cada fuente de alimento para ser elegida por una abeja observadora
  END
  For  $i=1$  a  $N$  do
    Elegir en probabilidad una fuente de alimento  $X_j$ 
    Enviar una abeja observadora a la fuente de alimento seleccionada
    Modificar la fuente de alimento  $X_j$  y determinar la cantidad de néctar de la nueva solución  $X_{oj}$ 
    if la nueva solución es mejor else
       $X_{oj} \leftarrow X_j$ 
    END
  END
END
```

Fig. 4 – Pseudocódigo algoritmo ABC.

Cuando una abeja abandona una fuente de alimento debido a su menor valor de néctar la abeja observadora genera una nueva fuente de alimento al azar de forma aleatoria y llena la posición abandonada. Esta abeja modifica la posición de la fuente de alimento almacenada en memoria y por tanto encuentra una nueva fuente de alimento y valida la cantidad de néctar (Fitness) de la misma. El resultado de la abeja observadora se aplica a la RWRH para una mejora en la optimización de los pesos.

Sistema Propuesto

El sistema propuesto funciona como muestra la Figura 1. El objetivo del método es mejorar la estimación del esfuerzo de cada uno de los proyectos sin descartar el esfuerzo real ya estimado por el mismo. Los valores de entrada se multiplican con los pesos asociados entre la entrada y la capa oculta. Se realiza la traslación y dilatación de los atributos de la red a algunos valores aleatorios, así se asegura que los valores se encuentren entre [0,1]. En la capa oculta la suma de la combinación entrada-peso se hacen pasar a través de la función wavelet madre para obtener la salida y k en cada neurona oculta k, la función de activación es la derivada de la gaussiana. La salida de los wavelones se multiplica por los pesos entre la capa oculta y la de salida. El resultado de cada wavelons se suma en la capa de salida de una sola neurona. Esta salida pasa a través del sistema de heurístico de búsqueda local que utiliza la función softplus y se multiplica por el tamaño del proyecto en punto de función ajustado para obtener el esfuerzo estimado. La fórmula del esfuerzo está dada por (5):

$$\text{Esfuerzo estimado} = \log \left(1 + e^{\left(\sum_{j=1}^p y_j \omega_j \right)} \right) * PFA \quad (5)$$

Donde p es el vector de probabilidades que devuelve la función softplus y PFA, la cantidad de unidades del sistema para el tamaño. Una vez obtenido el esfuerzo, se calcula la magnitud del error relativo (MRE). Se calcula para cada uno de los insumos del proyecto. Si la MRE obtenida es inferior o igual al valor aceptable (en este caso se toma como los valores aceptables todos los resultados $v \leq 1$), el proceso se termina (El cálculo del valor de MRE se explica en la sección siguiente), en caso contrario se actualizan los pesos mediante el algoritmo ABC y se repite todo el proceso.

Metodologías y métricas para probar y validar las RNA

La prueba y validación de las redes neuronales tras su entrenamiento ayudan a dar una indicación de la eficiencia del funcionamiento de la red, cuando se le pide que haga nuevas predicciones con datos que no ha visto antes (Aguilera 2018). Existen en la literatura trabajos que han utilizado varios métodos con este fin

los más conocidos son: la validación cruzada, K-iteraciones, validación cruzada triple y Validación cruzada con 10 iteraciones. Se recomienda la validación cruzada triple y la validación cruzada con 10 iteraciones basados en los aportes de (Kaushik y Singal 2019; Kodmelwar, Joshi y Khanna 2018) una vez implementada la RNA. En estos trabajos son utilizadas las métricas de:

Magnitud del error relativo (MRE): Se utiliza para encontrar el error relativo entre el esfuerzo real y el esfuerzo estimado para cada proyecto del conjunto de datos se calcula mediante la fórmula (6):

$$MRE = \frac{AE-EE}{AE} \quad (6)$$

Dónde AE es el esfuerzo actual estimado y el EE es el esfuerzo real estimado por el proyecto antes de la ejecución de la técnica.

MMRE: Es la mediana de los MRE de todos los proyectos de un conjunto de datos. Se calcula como (7):

$$MMRE = \frac{1}{N} \sum_{i=1}^N MRE_i \quad (7)$$

Donde N es el número de los proyectos del conjunto de datos.

Pred(i): Es la proporción de MRE menores o iguales a un valor determinado, i en todos los proyectos. Se calcula (8):

$$Pred(i) = \frac{K}{N} \quad (8)$$

Donde N es el número total de proyectos y K es el número de los puntos de datos cuyo MRE es menor o igual que los conjuntos de entrada a la red.

Validación de la propuesta

Para validar la contribución de la propuesta, se realizó una encuesta (constituida por 7 preguntas), para obtener el criterio de expertos. Los especialistas evaluaron la Relevancia: es significativo el aporte de la solución a las necesidades en los proyectos; Pertinencia: la estructura de la propuesta es congruente con los objetivos del mismo y es adaptable a los proyectos; aplicabilidad: la propuesta tiene una alta posibilidad de

aplicación y la precisión: obtener mejores resultados en la planificación. El panel de expertos, se seleccionó teniendo en cuenta haber desempeñado roles con más de 5 años en la producción de proyectos de software y más de 3 con experiencia en la estimación de software, tener experiencia en el trabajo con RNA para la estimación en proyectos de software y resultados científicos enfocados en el objetivo a evaluar. A partir de estos criterios, se realizó un cuestionario para el conocimiento curricular.

Procesamiento de la información

Luego de tabulados los resultados las categorías fueron evaluadas en su mayor por ciento como Altas y Muy altas como muestra la Figura 5, validando de esta manera la propuesta de mejora para el método de estimación de la UCI.



Fig. 5 – Análisis de los resultados de la validación por Criterio de Experto.

Se presentaron argumentos positivos relacionados con la complejidad del modelo, donde acotaron que permitirá un buen rendimiento en la capacidad de ajuste de las funciones y recomendaciones como que la aplicación de esta técnica ofrece perspectivas muy prometedoras en cuanto a fiabilidad y eficiencia, las cuales deben ser contrastadas con futuras experimentaciones.

Conclusiones

La estimación temprana de los proyectos de software en cuanto al esfuerzo tiene una gran importancia para la planificación. Disponer de esta información, facilitará al equipo del proyecto una gestión más eficiente, viabilizando la distribución de recursos y la disminución de riesgos. Con esta investigación se obtiene una técnica que ayudará al proceso de estimación de los proyectos de desarrollo de la UCI, debido a que la predicción del esfuerzo constituye la columna vertebral y aporta elementos importantes para manejar con precisión factores relacionados con el tamaño, costo y tiempo. La propuesta resuelve el problema del ajuste de los indicadores del esfuerzo a un nivel más detallado porque es adaptable a la variedad de proyectos de la universidad y toma en cuenta los 4 escenarios que modela el sistema en los proyectos. Su implementación mejorará la calidad del trabajo en los proyectos, reduciendo la probabilidad de retrasos en las entregas y la insatisfacción de los clientes.

Referencias

- AGGARWAL, C.C., 2019. Neural Networks and Deep Learning. S.l.: IBM T. J. Watson Research Center.
- AGGARWAL, K.K. y SINGH, Y., 2005. Bayesian Regularization in a Neural Network Model to Estimate Lines of Code Using Function Points. , vol. 1, no. 4. ISSN 1549-3636.
- AGUILERA, D.A., 2018. UN MODELO GLOBAL DE PREDICCIÓN DE QUIEBRA CON REDES NEURONALES. Doctoral. Málaga, España: Universidad de Málaga.
- AJITHA, S. y GEETHA, D.E., 2010. Neural Network Model For Software Size Estimation Using Use Case Point Approach. ,
- ARNAL, M.G., 2018. Estudio y aplicación de las redes neuronales convolucionales 3D. Grado. Barcelona: Universitat Politècnica de Catalunya.
- AZATH, H., MOHANAPRIYA, M. y RAJALAKSHMI, S., 2018. Software Effort Estimation Using Modified Fuzzy C Means Clustering and Hybrid ABC-MCS Optimization in Neural Network. , DOI <https://doi.org/10.1515/jisys-2017-0121>.

- BRITO, D.R., 2012. Método de Estimación para los proyectos de desarrollo de Software de la UCI. S.l.: Universidad de las Ciencias Informáticas.
- DHANOPIYA, K., GOYAL, A., MISHRA, N. y DAYMA, M., 2017. ESTIMATING SOFTWARE COST USING MULTILAYER NEURAL NETWORK. , vol. 5, no. 12, pp. 2320-2106.
- FIRDOUS AHMAD SHAH y DEBNATH, L., 2017. Wavelet Neural Model for Yield Spread Forecastin. , vol. 5, no. 72. DOI doi:10.3390/math5040072.
- HERNÁNDEZ-SAMPIERI, R. y TORRES, C.P.M., 2018. Metodología de la investigación. Las rutas cuantitativa, cualitativa y mixta. S.l.: McGraw-Hill Education.
- HUANCA, L.M. y ORÉ, S.B., 2017. Factores que Afectan la Precisión de la Estimación del Esfuerzo en Proyectos de Software Usando Puntos de Caso de Uso. , no. 21, pp. 18-32. ISSN 1696-9895.
- HUANG, S.-J. y CHIU, N.-H., 2006. Optimization of analogy weights by genetic algorithm for software effort estimation. , vol. 48, no. 11.
- KAUSHIK, A. y SINGAL, N., 2019. A hybrid model of wavelet neural network and metaheuristic algorithm for software development effort estimation. , DOI <https://doi.org/10.1007/s41870-019-00339-1>.
- KODMELWAR, M.K., JOSHI, S.D. y KHANNA, V., 2018. A Deep Learning Modified Neural Network Used for Efficient Effort Estimation. Journal of Computational and Theoretical Nanoscience. S.l.: s.n., pp. 3492-3500. DOI 10.1166/jctn.2018.7651.
- KUMAR, P.S., BEHERA, H.S., KUMARI, A., NAYAK, J. y NAIK, B., 2020. Advancement from neural networks to deep learning in software effort estimation:Perspective of two decades. ,
- NIELSEN, M., 2018. Neural Networks and Deep Learning. S.l.: s.n.
- PAEZ, A.H., 2017. Marco de trabajo ingenieril para el proceso de desarrollo de videojuegos. S.l.: Universidad de las Ciencias Informáticas.
- PRESSMAN, R.S. y MAXIM, B.R., 2015. Software Engineering. A PRACTITIONER'S APPROACH [en línea]. EIGHTH EDITION. S.l.: McGraw-Hill Education,. [Consulta: 16 diciembre 2020]. ISBN 978-0-07-802212-8. Disponible en: [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).

- RAO, S., REDDI, D.K.K. y RANI, D.U., 2017. Optimization of Neural Network for Software Effort Estimation. ,
- SAINI, A., AHUJA, L. y KHATRI, S.K., 2018. Effort Estimation of Agile Development using Fuzzy Logic. 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO). Noida, India: s.n., DOI DOI: 10.1109/ICRITO.2018.8748381.
- SARUWATARI, akuya, JIN, A., HAMURO, D. y HOSHINO, T., 2019. Estimation of Business Rules Using Associations Analysis. Knowledge-Based Software Engineering. S.l.: s.n., DOI https://doi.org/10.1007/978-3-319-97679-2_4.
- SHARMA, A. y CHAUDHARY, N., 2020. Software Cost Estimation for Python Projects Using Genetic Algorithm. , vol. 120. DOI https://doi.org/10.1007/978-981-15-3325-9_11.
- TARASSENKO, L., 1998. Guide to Neural Computing Applications. S.l.: Butterworth-Heinemann. ISBN 978-0-340-70589-6.
- TRENDOWICZ, A. y JEFFERY, R., 2014. Software Project Effort Estimation: Foundations and Best Practice Guidelines for Success [en línea]. S.l.: Springer International Publishing. [Consulta: 16 diciembre 2020]. ISBN 978-3-319-03628-1. Disponible en: <https://www.springer.com/gp/book/9783319036281>.
- VENKATAIAH, V., MOHANTY, R. y NAGARATNA, M., 2018. Prediction of Software Cost Estimation Using Spiking Neural Networks. Smart Intelligent Computing and Applications. S.l.: 101-112, DOI https://doi.org/10.1007/978-981-13-1927-3_11.
- YOUSEF, Q.M., ALSHAER, Y.A. y ALHAMMAD, N.K., 2017. Dragonfly Estimator: A Hybrid Software Projects' Efforts Estimation Model using Artificial Neural Network and Dragonfly Algorithm. , pp. 13.

Conflicto de interés

Los autores autorizan la distribución y uso de su artículo.

Contribuciones de los autores

1. Conceptualización: Wisleidys Campos Wright

2. Curación de datos: Yaimí Trujillo Casañola
3. Análisis formal: Wisleidys Campos Wright
4. Adquisición de fondos: -
5. Investigación: Wisleidys Campos Wright
6. Metodología: Wisleidys Campos Wright
7. Administración del proyecto: Yaimí Trujillo Casañola
8. Recursos: Wisleidys Campos Wright
9. Software: -
10. Supervisión: Yaimí Trujillo Casañola
11. Validación: Wisleidys Campos Wright
12. Visualización: Wisleidys Campos Wright
13. Redacción – Wisleidys Campos Wright
14. Redacción – Yaimí Trujillo Casañola