دار المنظومة
DAR ALMANDUMAH
الرواد في قواعد المعلومات العربية

العنوان:	تخمين الجهد اللازم لتطوير مشاريع هندسة البرمجيات بالاعتماد على تقنيات العنقدة مع المنطق المضبب
المصدر:	المجلة العراقية للعلوم الإحصائية
الناشر:	جامعة الموصل - كلية علوم الحاسوب والرياضيات
المؤلف الرئيسي:	عبدالمجيد، اشرف عبدالمنعم
مؤلفين آخرين:	سعيد، مصطفى غانم(م ، مشارك)
المجلد/العدد:	ع25
محكمة:	نعم
التاريخ الميلادي:	2013
الصفحات:	503 - 522
رقم MD:	636683
نوع المحتوى:	بحوث ومقالات
قواعد المعلومات:	EcoLink
مواضيع:	تكنولوجيا المعلومات ، هندسة البرمجيات ، التخمين الدقيق ، الانظمة التقنية ، أسلوب التعليمية ، علم الرياضيات ، مستخلصات الأبحاث
رابط:	http://search.mandumah.com/Record/636683

© 2021 دار المنظومة. جميع الحقوق محفوظة.
هذه المادة متاحة بناء على الإتفاق الموقع مع أصحاب حقوق النشر، علما أن جميع حقوق النشر محفوظة. يمكنك تحميل أو طباعة هذه المادة للاستخدام الشخصي فقط، ويمنع النسخ أو التحويل أو النشر عبر أي وسيلة (مثل مواقع الانترنت أو البريد الالكتروني) دون تصريح خطي من أصحاب حقوق النشر أو دار المنظومة.

المجلة العراقية للعلوم الإحصائية (25) 2013
عدد خاص بوقائع المؤتمر العلمي السادس لكلية علوم الحاسوب والرياضيات
ص ص[503-522]

تخمين الجهد اللازم لتطوير مشاريع هندسة البرمجيات بالاعتماد على تقنيات العنقدة مع المنطق المضبب

اشرف عبد المنعم عبد المجيد* مصطفى غانم سعيد**

الخلاصة

ان تقنيات التخمين في هندسة البرمجيات هي خطوة اساسية في عملية تطوير مشاريع هندسة البرمجيات اذ تتم عملية التخمين في المراحل البدائية لتطوير المشروع ضمن مرحلة التخطيط للمشروع المراد عمله حيث لا تتوفر معلومات دقيقة وكافية لأجراء التخمين بصورة دقيقة .

في هذا البحث تم دراسة استخدام تقنيات العنقدة مع المنطق المضبب لتكوين اداة هندسة البرمجيات لتخمين الجهد اللازم لتطوير المشاريع البرمجية (*Effort Estimating by FCM tool - EEFCM*).

لقد تم تدريب الاداة المقترحة على مجموعة بيانات *NASA* الفضائية التي تحتوي على خواص 60 مشروع برمجي بالإضافة الى الجهد الحقيقي لهذه المشاريع، ولقد بين تحليل نتائج تنفيذ الاداة المقترحة على ان عملية التخمين باستخدام عنقدة البيانات مع المنطق المضبب بصورة عامة قد حسنت من عملية التخمين بالمقارنة مع اساليب ذكائية وتقليدية في البحوث السابقة.

Abstract

The estimating techniques in software engineering is an essential step in the process of developing software engineering projects and the estimating process is done in the primitive stages of developing the project during the planning stage of the project to be done where there is no sufficient and accurate information is available to make an accurate estimate.

In this research we study the use of clustering techniques with the fuzzy logic to make a software engineering tool to estimate the required effort to develop the software projects (Effort Estimating by FCM tool (EEFCM)).

*مدرس مساعد / قسم هندسة البرمجيات/ كلية علوم الحاسوب والرياضيات/ جامعة الموصل

**باحث / قسم هندسة البرمجيات / كلية علوم الحاسوب والرياضيات/ جامعة الموصل

تاريخ استلام البحث 2013/10/9 ـــــــــــــــــــ **تأريخ القبول** 2013/11/21

The tool was trained on NASA space dataset which contains the properties of 60 software project in addition to the actual effort for these projects, and the analysis of the results of implementation the proposed tool shown that the process of estimating using data clustering with fuzzy logic in general have improved the estimating process compared with others intelligent and traditional approaches in previous researches.

1 المقدمة

التخمين الدقيق للكلفة البرمجية هي مهمة حرجة للمطورين والزبائن على حد سواء. الكلف " تحت التخمين" (*underestimating*) قد تنتج عن موافقة الإدارة على الأنظمة المقترحة التي ممكن أن تتجاوز ميزانياتها، مع استخدام دوال *functions* غير متطورة وجودة ضعيفة ورديئة والفشل في اكمال النظام في الوقت المحدد. و"فوق التخمين" (*Overestimating*) قد ينتج عن العدد الكبير في المصادر المخصصة للمشروع أو خلال عرض العقد قد يؤدي الى عدم الفوز به مما يؤدي الى فقدان الوظيفة. ولهذا فان تخمين الكلفة الدقيق مهم جداً[Parvinder,Porush,Amanpreet ,2008].

تقنية العنقدة هي تصنيف الكائنات المتشابهة الى مجاميع مختلفة أو بشكل اكثر دقة هي تقسيم مجموعة البيانات الى مجموعات جزئية (عناقيد) بحيث تمتلك البيانات في كل مجموعة جزئية بعض السمات المشتركة وغالباً مايتم تقريبها وفقاً لبعض مقاييس المسافة المعرفة. تعليم الآلة عادةً يَعتَبِر عنقدة البيانات كشكل من اشكال التعليم بدون مشرف [SERBAN,CAMELIA,POP,HORIA,2008].العنقدة هي عملية تقسيم البيانات إلى مجاميع إعتماداً على بعض المقاييس المتشابهة لهذه المجاميع وتعتبر عملية عنقدة البيانات عملية أساسية ومركزية في الذكاء الاصطناعي حيث يتم تعريف العنقود بواسطة مركز العنقود *cluster center*. التحليل العنقودي او عنقدة البيانات هي فرع تابع لتحليل البيانات ويتضمن مجموعة من الخوارزميات التابعة للتصنيف بدون مشرف [Vahid,Khatibi ,2011]. التحليل العنقودي هو تنـظيم مجموعة من النماذج الى عناقيـد بالاعتمـاد على التشابه [SERBAN,CAMELIA,POP,HORIA,2008, Mandal,2012]. التصنيف مهم للتنقيب عن قاعدة البيانات وبنائها [SERBAN,CAMELIA,POP,HORIA,2008].

خوارزميات العنقدة تستخدم على نطاق واسع ليس فقط لتنظيم وتصنيف البيانات وانما هي مفيده لضغط البيانات وبناء نموذج ترتيب البيانات. حيث أنه إذا كان بالإمكان ايجاد مجموعات من البيانات، فانه بالإمكان بناء نموذج للمشكلة على أساس تلك المجموعات [Mandal,2012].

ان اعتماد اسلوب الذكاء في حل مشكلة تخمين الجهد للمشروع البرمجي، حيث يحاول أن يعطي تخميناً أقرب ما يكون للكلفة الحقيقية للمشروع [Jagannath ,Bibhudatta ,2011]. ويتم الاعتماد في هذا البحث على مشاريع منجزة ومستكملة موجودة في قاعدة بيانات *NASA* لتدريب الاداة المقترحة،و أيضاً ايجاد وتحديد طريقة مناسبة لإعطاء أقرب تخمين ممكن إلى التخمين الحقيقي.

المنهجية المقترحة المعتمدة على تقنية تمييز الانماط باستخدام خوارزمية *FCM* المضببة لتخمين جهد مشاريع هندسة البرمجيات باعتماد مجموعة بيانات *N ASA* الفضائية التي تحتوي على خواص 60 مشروع برمجي بالإضافة الى الجهد الحقيقي لهذه المشاريع.

وقد تم استخدام طرق وتقنيات عنقدة البيانات مع المنطق المضبب وتم مقارنة نتائجها مع نتائج بحوث سابقة.

2 الدراسات السابقة

قدم الباحثون *Iris Fabiana de Barcelos Tronto* و *Jose Demisio Simoes da Silva* و *Nilson Sant'Anna* في العام 2006 بحثاً يركز على زيادة الدقة في عملية تخمين الجهد حيث تمت المقارنة بين طريقتين وهي طريقة شبكة الإدراك متعددة الطبقات (*multi layer perceptron*) وطريقة نماذج الانحدار الخطي (*linear regression models*) لمعرفة الطريقة الافضل في عملية تخمين الجهد وتم تطبيق هذه الطرق على قاعدة بيانات *NASA* والنتائج التي حصل عليها من طريقة الـ *ANN* هي أفضل من الطريقة الأُخرى ولوحظ أنه إذا تم استخدام بيانات متجانسة أكثر فإن أداء الطريقتين سوف يتحسن [Iris,Jose,Nilson ,2007]. وفي العام 2008 أيضاً قام الباحثون *Parvinder S. Sandhu* و *Porush Bassi* و *Amanpreet Singh Brar* باستخدام اسلوب الغموض العصبي *Neuro-Fuzzy* وتطبيقه على قاعدة بيانات *NASA* وعمل مقارنة بين أدائهو أداء النماذج *the Halstead, Walston-Felix, Bailey-Basili and Doty* و أظهرت النتائج أن الـ *Neuro-Fuzzy* كان لديه اقل *MMRE* و

RMSSE ويليها نموذج *Bailey-Basili* . وبهذا أثبت الباحث بأَن الـ *Neuro-Fuzzy* ممكن أن يستخدم في عملية تخمين الجهد ويناسب كل أنواع المشاريع [Parvinder,Porush,Amanpreet 2008,]. قام الباحثون *Rama Sree P* و *Sudha K. R* و *Prasad Reddy P.V.G.D* في العام 2011 باستخدام المنطق المضبب *Fuzzy Logic (FL)* وبينوا بأن استخدام دالة العضوية الثلاثية (*triangular membership function*) في المنطق المضبب لغرض عمل تخمين للجهد هو أفضل من استخدام دالة العضوية العامة بيل *Generalized Bell Membership Function (GBellMF)*) ، وفقط بتعديل القيم للمعاملات المستخدمة في أنظمة الاستدلال المضبب *FIS* ممكن ان نحصل على تخمين جيد للجهد [Prasad ,Sudha,Rama,2011]. وفي العام 2012 قام الباحثان *Roheet Bhatnagar* و *Mrinal Kanti Ghose* باستخدام نماذج مختلفة من الشبكات العصبية ونموذج مامداني *mamdani* من المنطق المضبب في عملية تخمين الجهد في مراحل المشروع الاولى وبالاعتماد على قاعدة بيانات طلاب جامعيين *student datast* وتم عمل مقارنة بين جميع الطرق باستخدام مقياس الـ *MMRE* لجميع الطرق وعلى مجموعة البيانات المذكورة ولوحظ بأن الشبكات العصبية ذات الانحدار الخطي*Linear Regression Neural network (LRNN)* لديها أقل قيمة بين نماذج الشبكات العصبية ولكن طريقة المنطق المضبب حصلت على قيمة أقل وهي تعتبر أفضل طريقة لعمل تخمين للجهد في مراحل مبكرة من عمر المشروع من بين الطرق المستخدمة في هذا البحث [Roheet,Mrinal,2012].

3 تخمين البرمجيات

التخمين الدقيق لحجم ، كلفة ، جهد والجدول الزمني للبرمجيات هو في الغالب التحدي الاكبر الذي يواجه مطوري البرمجيات هذه الأيام. حيث له تأثير أساسي على اوا ة تطوير البرمجيات وذلك بسبب التأثير المباشر لكل من "تحت التخمين" (*underestimates*) و"فوق التخمين" (*overestimates*). [Prasad ,Sudha,Rama,2011].

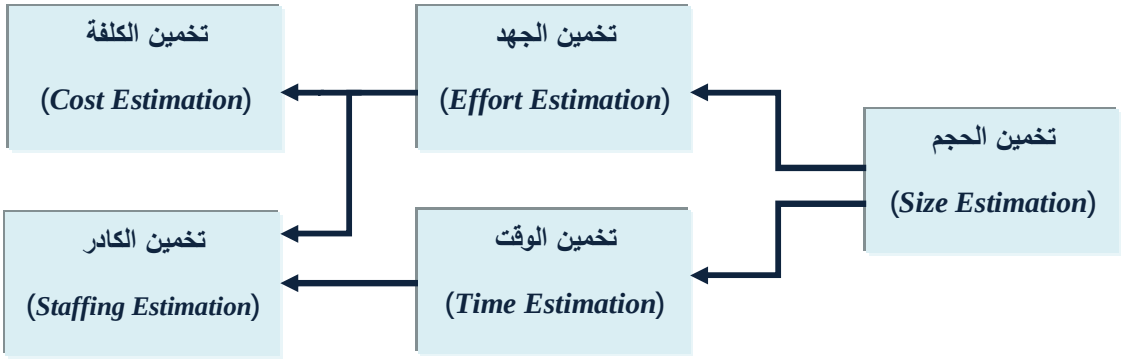
تخمين سلوك وأداء عملية المشاريع البرمجية مبكراً في دورة حياة البرمجيات هي تحدي كبير للغاية، وغالباً ما يصعب نمذجتها. والاستخدام التقليدي لتقنيات تخمين الكلفة البرمجية غير قادر على التعامل مع هذا الغموض والبيانات غير الدقيقة في نماذج تخمين الكلفة

الخاصة بها. [Zhiwei,Taghi,2003]. إنَ تخمين المشروع البرمجي لايمكن أن يكون علم ثابت ولكنه عبارة عن دمج بين بيانات تاريخية وتقنيات منظمة تستطيع برهنة دقة التخمين [Chetan,Anurag,2011].

يمكن تصنيف التخمين البرمجي إلى ثلاث مراحل[Esra'a,2012] :

- المرحلة الأولى: تشمل تخمين الحجم.
- المرحلة الثانية: تتمثل في تخمين الجهد وتخمين الوقت.
- المرحلة الثالثة: تتمثل في تخمين الكلفة وتخمين عدد الموظفين اللازمين.

و الشكل(1) يبين التداخل بين المراحل الثلاثة في عملية التخمين البرمجي النموذجية في دورة حياة تطوير البرمجيات [Parvinder,Porush,Amanpreet ,2008].



الشكل (1): يبين تسلسل عمليات التخمين في دورة حياة تطوير البرمجيات.

و عليه يمكن تحديد ثلاث خطوات رئيسة في عملية التخمين للبرنامج الذي يكون تحت التطوير [Esra'a,2012,Robert,Donald,Linda,2002] وهي:

- تخمين الحجم.
- تخمين الجهد.
- تخمين الجدولة.

الجهد *Effort* هو المصدر الاستهلاكي الرئيس في المشروع البرمجي. ففي مشروع تطوير البرمجيات كل تخمينات الجهد والجدولة تعتبر كشروط مسبقة لتخطيط المشروع. فتخمين الجهد هو أمر حيوي لمشاريع البرامج الناجحة [Pankaj,2008]. و غالباً ما يستخدم تخمين الجهد كمرادف لتخمين الكلفة. ففي تطوير البرمجيات، التكلفة الغالبة والمسيطرة

هي تكلفة اليد العاملة ولهذا من المهم جداً تخمين جهد التطوير البرمجي بأكثر دقة ممكنة. ونماذج الكلفة البرمجية تجهز بتخمين عن الجهد، الكلفة، والجدول الزمني [Karen,Michael,Jairus,John,Mori,Erik,2003]. في هذا البحث سوف نركز على عملية تخمين جهد المطور.

إن نماذج التخمين البرمجي يتم تصنيفها إلى نماذج حسابية *algorithmic models* ونماذج غير حسابية *non-algorithmic models* [Daniel,2006].

وهناك طرق ونماذج تخمين نموذجية، حيث هذه النماذج قد تم اشتقاقها من دراسة أعداد كبيرة من مشاريع البرمجيات المكتملة ولمؤسسات وتطبيقات متنوعة لاكتشاف كيفية تحويل حجم المشروع إلى جهد المشروع ولكنها بقيت لحد الآن غير قادرة على تخمين الجهد بشكل دقيق [Pankaj,2008,Nancy,Schett,2002]. الجدول (1) التالي يبين نقاط القوة والضعف بين طرق التخمين المختلفة .

الجدول (1) مقارنة بين نقاط القوة والضعف بين طرق التخمين المختلفة [Hareton,Zhang,1997, Sohaib,2009,Esra'a,2012].

النموذج	نقاط القوة	نقاط الضعف
النماذج الغير حسابية		
المقارنة	يعتمد على بيانات مشاريع حقيقية وخبرة سابقة	قد لاتتوفر المشاريع المشابهة. البيانات التا يخية قد لاتكون دقيقة.
حكم الخبراء	يمكن للخبراء أن يجهزوا بتخمين جيد مع توفر الخبرة المناسبة. يعطي تخمين سريع.	يعتمد على "الخبير". قد يكون به تحيز. يعاني من الاستدعاءات الغير كاملة.
باركنسون **التسعير من أجل كسب المشروع**	غالباً ما يفوز بالعقد (الصفقة).	ممارساته ضعيفة. قد يحدث به تجاوزات كثيرة.
من الأسفل إلى الأعلى	يعتمد على التحليل التفصيلي. يدعم تتبع المشروع بشكل أفضل من الطرق الأُخرى وتخميناته تعالج مهام المستويات المنخفضة.	قد يغفل عن عوامل كلفة مستوى النظام. يتطلب جهد تخمين أكبر مقارنةً بطريقة التخمين من الأعلى الى الأسفل. صعوبة عمل تخمين مبكر في دورة حياة المشروع.
من الأعلى الى الأسفل	التركيز على مستوى النظام. أسرع وأسهل من طريقة التخمين من الأسفل الى الأعلى. يتطلب الحد الأدنى من تفاصيل المشروع.	يقدم تفاصيل قليلة لتبرير التخمينات. أقل دقة من الطرق الأُخ ى.
النماذج الحسابية	موضوعية ونتائجها قابلة للتكرار. تحصل على فهم أفضل لطريقة التخمين.	مدخلاتها ذاتية (حساسة). حسابها للمشاريع السابقة والتي ربما لا تعكس البيئة الحالية. قد تكون الخوارزميات المستخدمة محددة من قبل

		شركة معينة ولا تكون مناسبة لتطوير البرمجيات بشكل عام.

من المقارنة ممكن أن استنتاج ان لا يوجد طريقة واحدة هي الأفضل لكل المشاريع و ان استخدام دمج بين التقنيات قد يعطي تخميناً أفضل. على سبيل المثال، الدمج بين طريقة التخمين من الأعلى إلى أسفل مع طريقة حكم الخبراء وطريقة المقارنة يمكن أن يعطي تخميناً أفضل.

4 العنقدة الضبابية

العنقدة الضبابية (*Fuzzy Clustering*) هي فئة من خوارزمية التحليل العنقودية التي تقسم البيانات إلى مجموعات. فهي عملية تقسيم عناصر البيانات إلى فئات أو مجموعات بحيث تكون العناصر في نفس الفئة متشابهة قدر الإمكان، والعناصر الموجودة في فئات مختلفة متباينة قدر الامكان. وبالاعتماد على طبيعة البيانات والغرض الذي من أجله يتم عنقدة البيانات، ويمكن استخدام مقاييس تشابه مختلفة لفرز العناصر إلى فئات، اذ تتحكم مقاييس التشابه بعملية تشكيل البيانات الى المجموعات. بعض الامثلة للمقاييس التي يمكن أن تستخدم في التجميع تشمل المسافة(*distance*)، والربط(*connectivity*)،و الكثافة(*intensity*) [SERBAN,CAMELIA,POP,HORIA,2008, Mandal,2012]. في التجميع الثابت(*Hard clustering*)، تنقسم البيانات إلى مجموعات متميزة، حيث ينتمي كل عنصر من عناصر البيانات لمجموعة واحدة فقط. بينما في التجميع الغامض، يمكن لعناصر البيانات ان تنتمي إلى اكثر من مجموعة واحدة، ويرتبط كل عنصر مع مجموعة من دوال العضوية في عدة مستويات. العنصر في البيانات يمكن أن ينتمي إلى أكثر من مجموعة واحدة، هذه هي الفكرة الرئيسية وراء تقنية نظام المجموعة الضبابي . لذلك، يتم تعريف مستوى العضوية لتحديد كيف يرتبط العنصر من المجموعة،. قيمة مستوى العضوية تتراوح بين 0 إلى 1 [Vahid,Khatibi ,2011]. وتشير الى قوة الارتباط بين احد عناصر البيانات ومجموعة معينة بالتجميع الضبابي و هو عملية تعيين مستويات العضوية، ومن ثم يتم استخدامها لتعيين عناصر البيانات إلى واحد أو أكثر من المجموعات[Vahid,Khatibi ,2011]. من اكثر خوارزميات العنقدة الضبابية استخداماً هي خوارزمية المجموعة الضبابية العظمى والصغرى (*Fuzzy max-min cluste*r). اسلوب عنقدة المجموعة (*C-Means*) هو احد اهم تقنيات المجموعات الضبابية الذي تم

اقتراحه في عام 1981 [Mandal,2012, Vahid,Khatibi ,2011]. في هذا الأسلوب كان الهدف النهائي هو التقليل من دالة الهدف كما المعادلة (1).

$$l_m = \sum_{i=1}^{N} \sum_{j=1}^{G} U_{ij}^{im} \left\| X_i - C_j \right\|^2 \quad 1 < m < \infty \qquad \ldots.(1)$$

حيث U_{ij} هو درجة عضوية X_i في المجموعة j وينتمي إلى [0، 1]. بينما Cj هو مركز المجموعة j والرمز | | * | | يشير الى مقياس التشابه بين X_i والمركز. يتم تقديم خوارزمية التجميع *KM* على النحو الاتي.

نظمت الخوارزمية *KM* في أربع خطوات [Vahid,Khatibi ,2011] :

الإدخال: *A* متجه X من نقاط البيانات N على النحو التالي:

$$X = \{x_1, x_2 \ldots\ldots, x_n\} \qquad \ldots.(2)$$

الإخراج: المصفوفة U تتكون من C × N، C هو عدد المجموعات.

الخطوة 1: تهيئة المصفوفة U=[uij] ، U=(0)، حيث U_{ij} هي قيمة دالة العضوية للنقطة i في المجموعة j .

الخطوة 2: في الخطوة K: حساب مراكز متجهات C(k)=[c]j مع U(k)،

$$cj = \frac{\sum_{L=1}^{N} U_{if}{}^{m} x_i}{\sum_{L=1}^{N} U_{if} m} \qquad \ldots.(3)$$

الخطوة 3: تحديث U(k) وU(k+1).

$$U_{if} = \frac{1}{\sum_{k=1}^{C} \left(\frac{|x_L - c_f|}{|x_L - c_k|} \right)} \qquad \ldots.(4)$$

الخطوة 4 : اذا كان ناتج (|| U(k+1) - U(k)||) اصغر من قيمة العتبة توقف والا العودة الى الخطوة 2.

5 مقاييس التقييم

تحليل الخطأ هو جزء مهم من التنبؤ . وتنبؤات السلاسل الزمنية لا يمكن ان تكون مثالية فمن المؤكد انها دائماً تمتلك بعض الاخطاء التنبئية. حساب الخطأ يساعدنا في تحليل

النتائج التي تم الحصول عليها من خلال الطريقة المطبقة. فمن المفيد تحليل وتلخيص دقة هذه التنبؤات[Esra'a,2012]

لإجراء مقارنة بين النماذج المستخدمة تم استخدام المقاييس التالية:

1- جذر متوسط مربع الخطأ Root Mean Square Error (RMSE): هذا المقياس يحسب الجذر التربيعي لمعدل مربع الخطأ المحسوب بين القيمة الحقيقة (الهدف *Target*) والقيمة الناتجة مقسوماً على N ومعادلته هي [Jaswinder, Satwinder, Karanjeet, Pourush, 2010]

$$RMSE= \sqrt{\frac{1}{N}\sum_{i=1}^{N}(E_i - \hat{E}_i)^2} \quad ....(5)$$

2- متوسط حجم الخطأ النسبي Mean Magnitude Relative Error(MMRE): هو النسبة المئوية للقيم المطلقة لمعدل الأخطاء النسبية (*mean relative error(MRE)*) مقسوماً على N ومعادلته هي [Alaa F.,Alaa,2010]:

$$MMRE = \frac{1}{N}\sum_{i=1}^{N} MRE_i \times 100 \quad ....(6)$$

حيث:

$$MRE = \left|\frac{E-\hat{E}}{E}\right| \quad ....(7)$$

3- خطأ التوازن النسبي Balanced Relative Error (BRE): هو القيمة المطلقة للخطأ مقسوماً على القيمة الأصغر بين الجهد المخمن $\hat{E}$ والجهد الحقيقي E [Parvinder,Porush,Amanpreet ,2008].

$$BRE = \frac{|E-\hat{E}|}{\min(E,\hat{E})} \quad ....(8)$$

4- نسبة التباين Variance-Accounted-For (VAF): ويستخدم لقياس مدى قرب الجهدالحقيقي من الجهد المخمن ومعادلته هي [Alaa F.,Alaa,2010]:

$$VAF = \left[1 \quad \frac{var(E-\hat{E})}{var(E)}\right] \times 100\% \quad ....(9)$$

حيث في جميع المقاييس الـ $(\hat{E})$ هي الجهد المخمن *Estimated Effort* والـ (E) هي الجهد الحقيقي*Actual Effort* و N هي عدد المشاريع الكلي. وفي جميع المقاييس كلما قلت

قيمة المقياس كلما كانت النتيجة أفضل عدا مقياس الـ *VAF* والذي كلما زادت قيمتة كان التخمين أفضل[Esra'a,2012] .

6 خطوات عمل تخمين الجهد البرمجي باستخدام العنقدة مع المنطق المضبب

عند استخدام العنقدة مع المنطق المضبب *FCM* كأداة في التخمين، يجب على المخمن *Estimator* أن يقوم بعدة خطوات أساسية لأي عملية عنقدة ، بدايةً بجمع البيانات لحين الوصول الى الحل المقبول ففي جمع البيانات يتم جمع البيانات من مشاريع مطورة سابقة وفي هذا البحث سيتم الاعتماد على قاعدة بيانات *NASA* والتي تتكون من *60* مشروع برمجي –كما ذكر سابقاً-. حيث إن هذه البيانات تحتوي على قيم خاصة بالخصائص المؤثرة في أي مشروع برمجي (*15* عامل كلفة *Cost Driver*) إضافة إلى حجم المشروع مقاساً بالـ *KLOC* وكذلك الجهد الحقيقي لهذا المشروع . وبعد اجراء محاولات عديدة لتحديد الخصائص التي سيتم اختيارها كمدخل للاداة المقترحة اثناء مرحلة التدريب حيث تجسدت هذه المحاولات في إدخال جميع البيانات (الـ*15* عامل كلفة اضافة الى الـ *KLOC*) أو اختيار العوامل الأكثر تأثيراً على المشروع البرمجي واستخدامها كمدخلات بالإضافة للـ *KLOC* – وهذا ماتقوم به غالبية البحوث- وغيرها من المحاولات العديدة لاختيار البيانات التي تعطينا أفضل تخمين برمجي. وبعد إجراء دراسة على هذه البيانات (بيانات *NASA*) ونظراً للتقارب الشديد بين قيم عوامل الكلفة تم في هذا البحث اقتراح استخدام قيمة واحدة بدل قيم الـ *15* عامل وهذه القيمة عبارة عن مضروب هذه العوامل وهي قيمة الـ *EAF* والتي استخدمت كمدخل لنظام العنقدة الضبابي إضافة للـ *KLOC* حيث بهذا سيتم أخذ جميع قيم العوامل بالحسبان ولايتم تجاهل أي عامل من العوامل المؤثرة بالمشروع البرمجي. والخطوة التالية لجمع البيانات هي تقسيم قاعدة البيانات حيث يتم تقسيم البيانات الى جزئين مجموعة بيانات التدريب *Training set* ومجموعة بيانات الاختبار *Testing set* حيث سيتم في بحثنا هذا إعطاء *50* مشروع كبيانات تدريب و *10* مشاريع كبيانات اختبار. كما في الجدولين (2) و(3) التاليين[Esra'a,2012] :

الجدول (2): مجموعة بيانات التدريب لنظام العنقدة الضبابي *Training set*

رقم المشروع	*EAF*	*KLOC*	*Actual effort (PM)*
1	0.4084	190	420
2	0.9694	48.5	239

60	19.7	0.8785	3
252	47.5	0.8577	4
48	10	0.7826	5
18	5.5	0.8785	6
750	101	1.1432	7
48	20	0.4714	8
107	21	1.8739	9
155	19.3	1.3900	10
815	161.1	0.9694	11
98.8	11.4	1.3900	12
117.6	25.9	0.8785	13
215	100	0.9128	14
31.2	7.7	0.8785	15
360	100	0.3837	16
3240	370	1.2034	17
42	8	1.1500	18
42	6.5	1.3225	19
90	15	0.9989	20
360	100	0.8021	21
170	32.6	0.9694	22
70	15.4	0.9694	23
400	79	0.6989	24
72	20	0.3837	25
192	35.5	0.9694	26
1368	282.1	0.6841	27
36	8.2	0.8785	28
117.6	24.6	0.8785	29
278	70	0.7949	30
60	32.5	0.9489	31
2300	423	0.4362	32
571.4	78	1.3900	33
324	150	0.5985	34
973	284.7	0.8044	35
36	11.3	0.6173	36
1181	227	0.9470	37
60	13	1.1500	38
114	16	1.2765	39
450	90	0.7780	40
352.8	66.6	0.8785	41
2120	219	1.2034	42
25.2	9.7	0.8785	43
50	10.4	0.8785	44
2400	302	0.9131	45
82	16.3	0.9694	46

480	115.8	0.8785	47
60	14	0.8785	48
24	6	0.4714	49
8.4	2.2	0.8785	50

الجدول (3): مجموعة بيانات الاختبار لنظام العنقدة الضبابي *Testing set*

Actual effort	*KLOC*	*EAF*	رقم المشروع
62	12.8	0.9694	51
300	66.6	0.8785	52
48	15	0.4714	53
10.8	3.5	0.8785	54
120	29.5	0.8785	55
370	50	1.2034	56
60	31.5	0.5231	57
210	38	1.0349	58
1248	177.9	1.2101	59
72	7.5	1.5867	60

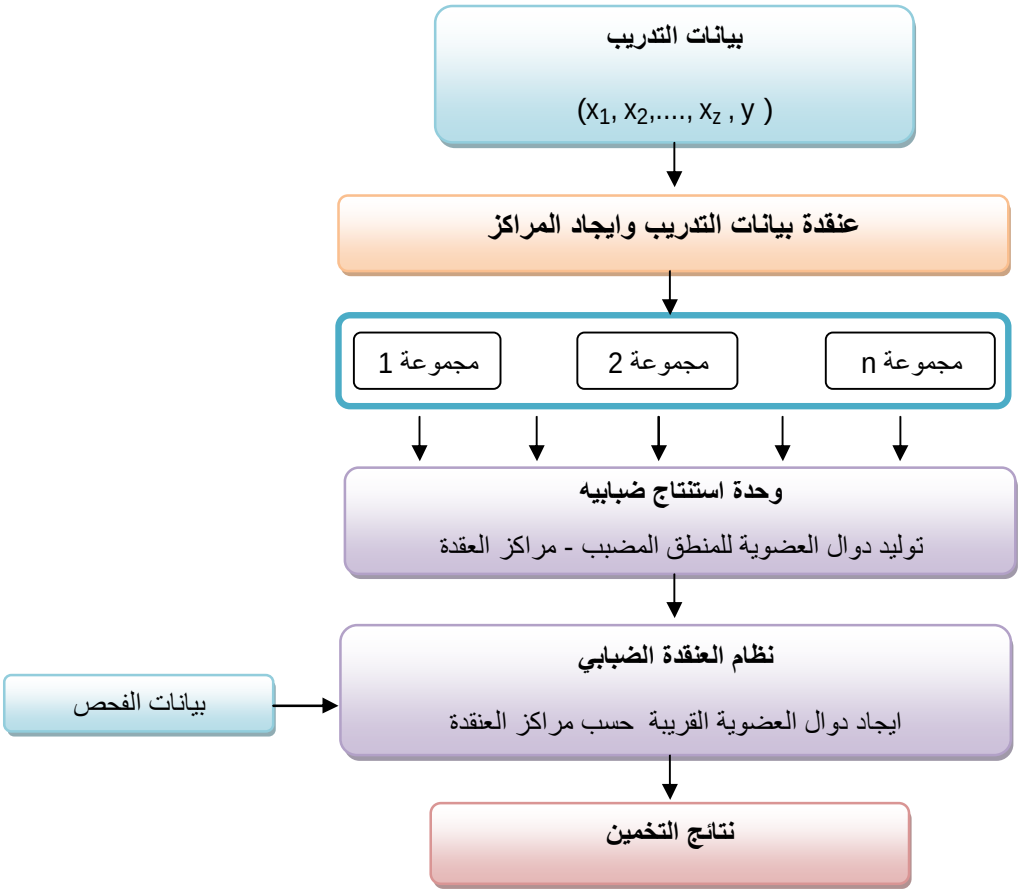
الهيكلية العامة لخوارزمية تخمين الجهد باستخدام نظام العنقدة الضبابي موضحة بالشكل (2) والذي تتوضح فيه كيفية استخدام العنقدة الضبابية لتكوين اداة هندسة البرمجيات لتخمين الجهد اللازم لتطوير المشاريع البرمجية (*Effort Estimating by FCM tool - EEFCM*) .

وبعد تهيئة البيانات لادخالها الى نظام العنقدة الضبابي تبدأ عملية التدريب و ايجاد مراكز العنقدة وتوليد دوال العضوية لكل مركز عنقدة وثم مرحلة فحص النظام وهذه الخطوات يمكن تلخيصها بالتالي:

اولا: تحضير ادخالات نظام العنقدة الضبابي في مرحلة تكوين دوال العضوية ومراكز البيانات وعدد اخراجات النظام

1. ادخالات العنقدة الضبابية هي متجه *EAF* ومتجه *KLOC* بشكل اعمدة، اذ كل صف يمثل مشروع واحد.
2. اخراجات النظام والمتمثلة بالجهد الحقيقي لهذه المشاريع.
3. عدد مراكز العنقدة التي من خلالها سوف يتحدد عدد الفئات في خوارزمية *KM* وايضاً يمثل عدد دوال العضوية في المنطق المضبب.اذ تم اختيار ثلاثة مراكز لعنقدة البيانات

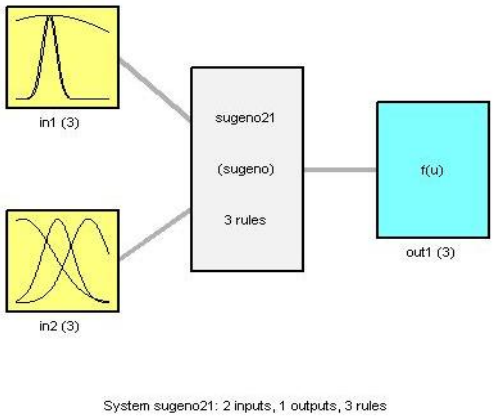
بعد ماتمت دراسة العدد الافضل لمراكز العنقدة الملائمة لحل مشكلة التخمين وكما موضح في الجدول (4).



الشكل (2) استخدام العنقدة الضبابية لتكوين اداة هندسة البرمجيات لتخمين الجهد اللازم لتطوير المشاريع البرمجية.

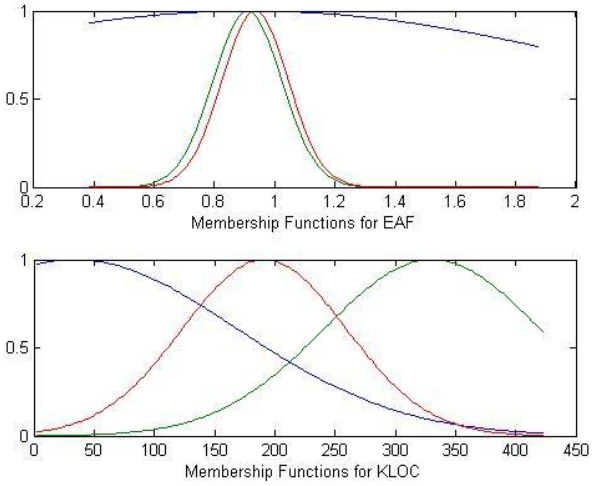
يحدد نوع الاستدلال في المنطق المضبب (*sugeno* و*mamdani*) حيث تم استخدام استدلال من نوع (*sugeno*) وذلك بسبب النتائج الجيدة في التخمين بالمقارنة مع نتائج الاستدلال باستخدام (*mamdani*) كما مبين في الجدول (4).

ثانيا: تكوين نظام العنقدة الضبابي وكما موضح بالشكل (3). اذ ان الجانب الايسر يمثل دوال العضوية الخاصة بالادخال والوكز يمثل وحدة استنتاج ضبابية (*fuzzy inferencing unit*).



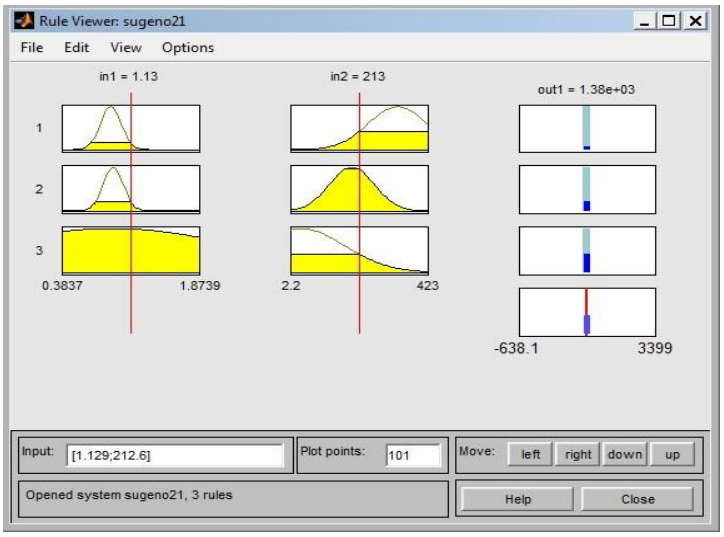
الشكل(3) نظام العنقدة الضبابي لتخمين جهد البرمجيات

اما الجانب الايمن فانه يمثل اخراج النظام والذي هو قيمة التخمين. ونلاحظ في الشكل (4) تداخل قيم دوال العضوية اذ يمثل الجزء الاعلى دوال العضوية الخاصة بمتجه *EAF*. والجزء الثاني يمثل دوال العضوية الخاصة بمتجه *KLOC* . اذ كل واحد من ادخالات نظام العنقدة الضبابي يمثل بثلاثة دوال عضوية وهذا يمثل عدد المراكز التي تم تحديدها في مرحلة تحضير البيانات.



الشكل(4) دوال العضوية لادخالات نظام العنقدة الضبابي

ثالثاً: وحدة استنتاج ضبابية تقوم بمعالجة القاعدة المعطاة من خلال صياغة مجموعة من القواعد المنطقية وتبين اخراجات النظام بشكل مباشر كما موضحة بالشكل (5).



الشكل(5) مجموعة من القواعد المنطقية وحدودها.

ولغرض عمل مقارنة بين نتائج هذه التخمينات لتحديد افضل نموذج استدلال وافضل عدد لمراكز العنقدة، فقد تم عمل جدول بقيم المقاييس الناتجة من جميع الطرق المستخدمة والجدول (4) يبين نسب هذه المقاييس:

الجدول (4): تحديد نوع الاستدلال

التسلسل			المقياس			
	عدد المراكز	نوع الاستدلال	RMSE	MMRE	BRE	VAF
1	3	*Sugeno*	96.7521	2.7117	0.2712	91.3616
		Mamdani	819.4595	5.47799	4.2318	32.6401
2	4	*Sugeno*	96.7521	2.7117	0.2712	91.3616
		Mamdani	277.6909	2.36515	2.3034	59.0261
3	5	*Sugeno*	96.7521	2.7117	0.2712	91.3616
		Mamdani	265.7599	2.08616	2.0446	63.7308
4	6	*Sugeno*	96.7521	2.7117	0.2712	91.3616
		Mamdani	161.3424	1.36535	1.6945	77.8513
5	7	*Sugeno*	96.7521	2.7117	0.2712	91.3616
		Mamdani	112.9566	1.50006	1.6087	57.6608
6	8	*Sugeno*	96.7521	2.7117	0.2712	91.3616
		Mamdani	156.7681	1.57459	1.5461	71.9101
7	9	*Sugeno*	96.7521	2.7117	0.2712	91.3616
		Mamdani	142.6493	1.27099	1.3431	88.1963

8	10	***Sugeno***	96.7521	2.7117	0.2712	91.3616
		Mamdani	144.1074	1.23661	1.3065	89.4935
9	11	***Sugeno***	96.7521	2.7117	0.2712	91.3616
		Mamdani	158.2909	1.59677	1.4136	74.3318

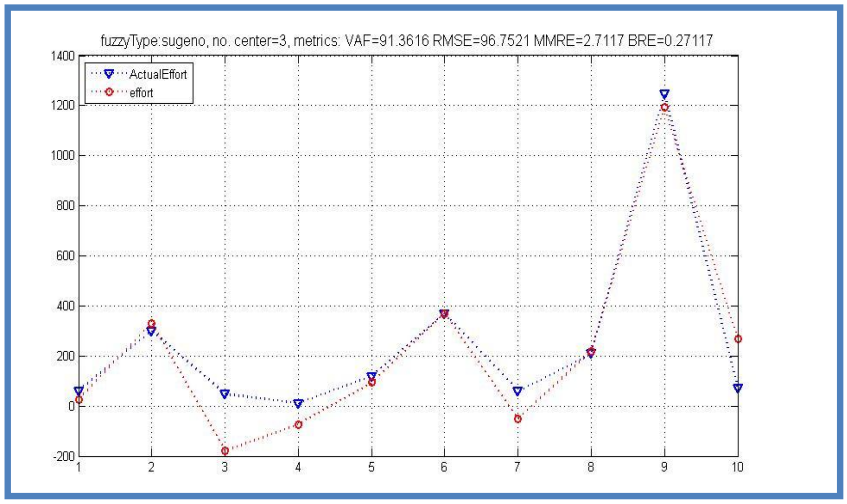
مما سبق في الجدول (4) يتبين لدينا أَن الاستدلال باستخدام *Mamdani* كان الأسوأ بين الطرق المستخدمة حتى مع زيادة عدد المراكز فانه يبقى اقل دقة من *Sugeno*، حيث حصل على أقل النسب بالمقاييس المستخدمة وان افضل عدد للمراكز هو 10. وبصورة عامة فإن نظام العنقدة الضبابي باستخدام *Sugeno* قد حسن من نتائج التخمين، اذ تم الحصول على نسب مقاييس أفضل من *Mamdani*. وان افضل عدد لمراكز العنقدة هو 3 مراكز والتي تم اعتمادها بنظام العنقدة الضبابي باستخدام *Sugeno* .

7 نتائج التخمين ومقارنة النتائج

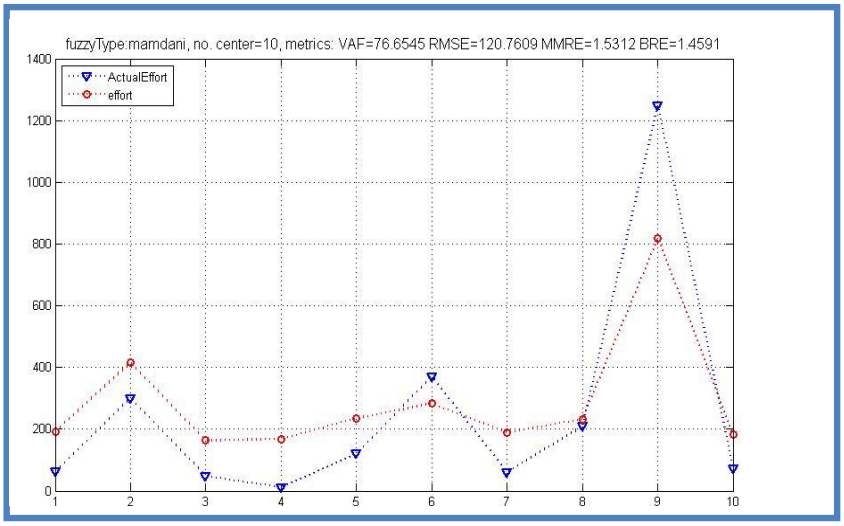
من خلال نتائج تطبيق اداة *EEFCM* على بيانات الفحص تم الاستنتاج بانه استخدام نظام العنقدة الضبابي في تخمين الجهد اللازم لإنجاز المشاريع قد حسن من نتائج التخمين و الشكل (6)-أ يوضح مخطط نتائج التخمين باستخدام العنقدة الضبابية (*FCM Sugeno*) بالمقارنة مع نتائج قيم كلفة الجهد الحقيقية و الشكل (6)- ب يوضح مخطط نتائج التخمين باستخدام العنقدة الضبابية (*FCM Mamdani*) بالمقارنة مع نتائج قيم كلفة الجهد الحقيقية.

وبالمقارنة مع نتائج بحوث عالمية كانت نتائج التخمين التي توصلنا إليها في بحثنا هذا أفضل من نتائج الباحثان *Jagannath Singh* و *Bibhudatta Sahoo* في العام *2011* [Parvinder,Porush,Amanpreet ,2008]، حيث قام الباحثان بتخمين الجهد باستخدام عدة شبكات عصبية تضمنت (*CNN, ENN, FFNN, Recurrent ANN*) وكانت شبكة الـ *CNN*هي الأفضل وشبكة *ENN* حصلت على أسوأ النتائج. حيث قام الباحثان باختيار اكثر عاملين مؤثرين في المشروع من بين عوامل الكلفة وهم (*CPLX,PCAP*) اضافة للـ*KLOC* كادخال للشبكة العصبية وتم اجراء مقارنة مع نتائج شبكة *CNN* لانها الافضل من حيث التخمين.

والجدول (5) يبين مقارنة بين نتائج الطرق المستخدمة في البحث مع نتائج البحث اعلاه وباستخدام المقاييس (*MMRE,RMSE,BRE*):



أ- نتائج التخمين باستخدام العنقدة الضبابية (*FCM Sugeno*).



ب- نتائج التخمين باستخدام العنقدة الضبابية (*FCM Mamdani*).

الشكل (6) يوضح مخطط نتائج التخمين باستخدم العنقدة الضبابية بالمقارنة مع نتائج قيم كلفة الجهد الحقيقي

الجدول (5): مقارنة نتائج البحث مع البحث العالمي

آلية التخمين		*المقياس*
CNN	*FCM*	
4.98	2.7117	*MMRE*
131.85	96.7521	*RMSE*
0.1	0.2712	*BRE*

حيث *FCM* تمثل النتائج التي توصلنا اليها و*CNN* تمثل نتائج الباحثان في [Parvinder,Porush,Amanpreet ,2008].

وكما هو واضح من الجدول (5) كانت نتائج الخوارزمية المقترحة أفضل .

8 الاستنتاجات

في هذه البحث، تم اعتماد اسلوب الذكاء في حل مشكلة تخمين الجهد للمشروع البرمجي وتم استخدام نظام العنقدة الضبابي في تخمين الجهد للمشروع. وقد تبين في الجدول (5)، ان الخوارزمية المقترحة تعطي نتيجة أفضل بالمقارنة مع الطرق الأخرى المطبقة في نفس المشكلة. فضلا عن ان استخدام نظام العنقدة الضبابي باستخدام الاستدلال من نوع *Sugeno* كان أفضل من الاستدلال من نوع *Mamdani* في تخمين الجهد للمشروع كما مبين في الجدول(4) .

9 المصادر

[1] Parvinder S. Sandhu, Porush Bassi, and Amanpreet Singh Brar, 2008, "Software Effort Estimation Using Soft Computing Techniques", World Academy of Science, Engineering and Technology.

[2] Jagannath Singh, Bibhudatta Sahoo, 2011, "Software Effort Estimation with Different Artificial Neural Network", IJCA, 2nd National Conference- Computing, Communication and Sensor Network, CCSN.

[3] SERBAN, CAMELIA AND POP, HORIA F. , 2008, "SOFTWARE QUALITY ASSESSMENT USING A FUZZY CLUSTERING APPROACH", STUDIA UNIV. BABES,BOLYAI, INFORMATICA, Volume LIII, Number 2.

[4] Vahid, Khatibi B. and others, 2011, "A New Fuzzy Clustering Based Method to Increase the Accuracy of Software Development Effort Estimation", World Applied Sciences Journal 14 (9): 1265-1275 ,ISSN 1818-4952.

[5] Iris Fabiana de Barcelos Tronto, Jose Demisio Simoes da Silva, Nilson Sant'Anna, 2007, "Comparison of Artificial Neural Network and Regression Models in Software Effort Estimation", IEEE.

[6] Prasad Reddy, Sudha K. R, Rama Sree p, 2011, "Application of Fuzzy Logic Approach to Software Effort Estimation", (IJACSA)

International Journal of Advanced Computer Science and Applications.

[7] Roheet Bhatnagar, Mrinal Kanti Ghose, 2012," early stage software development effort estimations-mamdani FIS VS neural network models", CS & IT , pp. 377–384.

[8] Zhiwei Xu, Taghi M. Khoshgoftaar, 2003, "Identification of fuzzy models of software cost estimation", Elsevier B.V,141-163.

[9] Chetan Nagar, Anurag Dixit, 2011, "Software Efforts and Cost Estimation with a Systematic Approach", ISSN, Journal of Emerging Trends in Computing and Information Sciences.

[10] Linda M. Laird, 2006, "The Limitations of Estimation" , IEEE and Wiley.

[11] Robert T. Futrell, Donald F. Shafer, Linda I. Safer, 2002, "Quality Software Project Management", Prentice Hall PTR.

[12] Pankaj Jalote , 2008, "A Concise Introduction to Software Engineering", Springer Science+Business Media.

[13] Daniel Meier, 2006, "E-Learning for Effort Estimation in Software Projects", Master's Thesis in Computer Science and Business Administration, Department of Informatics, University of Zurich, Switzerland.

[14] Karen Lum, Michael Bramble, Jairus Hihn, John Hackney, Mori Khorrami, Erik Monson, 2003, "Handbook for Software Cost Estimation", Jet Propulsion Laboratory, Pasadena, California.

[15] Nancy Merlo, Schett, 2002," COCOMO (Constructive Cost Model)", Requirements Engineering Research Group, Department of Computer Science, University of Zurich, Switzerland.

[16] Hareton Leung , Zhang Fan, 1997, "Software Cost Estimation", Department of Computing, The Hong Kong Polytechnic University.

[17] Sohaib Shahid Bajwa, 2009, "Investigating the Nature of Relationship between Software Size and Development Effort", Master Thesis, Computer Science, Department of Interaction and System Design,School of Engineering.

[18] Mandal1,S. Nath and others, 2012, "Prediction of Adsorption of Cadmium by Hematite Using Fuzzy C-Means Clustering Technique", I.J. Intelligent Systems and Applications, 12, 32,DOI:10.5815/ijisa.2012.12.05

[19] Jaswinder Kaur, Satwinder Singh, Dr. Karanjeet Singh Kahlon, Pourush Bassi, 2010,"Neural Network-A Novel Technique for Software Effort Estimation", International Journal of Computer Theory and Engineering.

[20] Alaa F. Sheta, Alaa Al-Afeef, 2010, "A GP Effort Estimation Model Utilizing Line of Code and Methodology for NASA Software Projects", IEEE, International Conference on Intelligent Systems Design and Applications.

[21] Esra'a Z.M.Quba, 2012, " Software Projects Estimation using Neural Networks "