

## استخدام خوارزمية أمثلية عناصر السرب لقياس جودة البرمجيات

صالحه رائد محمد  
(طالبة) قسم هندسة البرمجيات  
كلية علوم الحاسوب والرياضيات  
جامعة الموصل  
تاريخ قبول البحث : ٢٠١٨\٥\٢٠

ابراهيم احمد صالح  
قسم هندسة البرمجيات  
كلية علوم الحاسوب والرياضيات  
جامعة الموصل  
تاريخ استلام البحث : ٢٠١٨\١١\٢٩

### ABSTRACT

The process of improvement software quality began from early stages of software engineering development. It uses multiple quality metrics which are very important in software development. To calculate the standards quality of in software testing has been adopted. the software testing is focusing on the Software defect. In this paper is proposed new methods which combine the particle swarm optimization (PSO) to handle the best features Extraction with back-propagation networks to testing and evaluation of the data set . The paper depended database for NASA standards data. The result and experiment method improved quality performance for all classification methods used in the research"Combining Particle Swarm Optimization based Feature Selection and Bagging for Software Defect Prediction."

**Keywords:** *Software Defect, Particle Swarm Optimization, Neural Network Back Propagation, NASA.*

### الملخص

بدأت عملية تحسين الجودة للبرمجيات من مراحل مبكرة في تطوير هندسة البرمجيات؛ وذلك باستخدام مقاييس الجودة التي تعد مهمة جداً في عملية تنمية البرمجيات، وقد اعتمد حساب مقاييس الجودة في اختبار البرمجيات لتحسين عملية اختبار البرمجيات بالتركيز على عيوب البرمجيات (Software Defect). في هذا البحث اقترح نظام برمجي يجمع بين خوارزمية سرب الطيور (Particle Swarm Optimization) في استخلاص الخواص للتعامل مع أفضل المميزات واستخدام الشبكات العصبية ذات الانتشار العكسي (Neural Network Back Propagation) التي تقوم بعملية فحص مجموعة البيانات واختبارها وتقييمها من قاعدة بيانات مقاييس وكالة ناسا (NASA)، وقد بينت النتائج أن الطريقة المستخدمة قد حققت تحسناً ملحوظاً في عملية أداء الجودة بالنسبة لجميع طرائق التصنيف التي استخدمت في البحث "Combining Particle Swarm Optimization based Feature Selection and Bagging for Software Defect Prediction".

### 1. مقدمة

تعد عملية بناء برمجيات عالية الجودة للبرمجيات مهمة صعبة جداً من حيث عدد مكوناتها الأساسية ومرونتها والترابط فيما بينها، وأيضاً إدارة هذه البرمجيات وقياسها وضمان جودتها، ودور هندسة البرمجيات هو تقديم الطرائق والتقانات والنماذج والمنهجية التي تجعل من السهولة التعامل مع هذا التعقيد، وتعرّف هندسة البرمجيات وفق Institute of Electrical and Electronics Engineers: "هندسة البرمجيات بأنها تطبيق

المنهجية والنظامية والنهج القابل للقياس الكمي لتطوير البرمجيات وتنفيذها وصيانتها، أي بمعنى تطبيق الهندسة على البرامج" [١]. وعرفها Broy بأنها: "الهندسة بصورة عامة هي التطبيق المنظم للمبادئ والطرائق العلمية نحو بناء كفاءة وفعال للهياكل والمكائن، ومن ناحية هندسة نظم البرمجيات فإن الهدف الجوهرى هو تحسين القدرات لإدارة التطوير والتطور لنظم البرمجيات المكثفة، والأهداف الأساسية لهندسة نظم البرمجيات المكثفة هي الجودة المناسبة، وكلفة التطوير المنخفضة، والتسليم في الوقت المناسب" [٢].

ويمكن أن تُجزأ تقانة هندسة البرمجيات إلى أربع طبقات هي: الأدوات Tools، والطرائق Methods، والمعالجة Process، والجودة Quality كما هي موضحة في الشكل (١).



الشكل (١): المبادئ الأساسية لهندسة البرمجيات

إذ إن طبقة الجودة هي طبقة تحمل على عاتقها تحديد مجموع السمات والخصائص لمنتج أو لخدمة، و القدرة على تلبية احتياجات محددة أو ضمنية. وتعرف هندسة البرمجيات أيضاً بأنها انضباط هندسي يُعنى بجميع جوانب إنتاج البرمجيات من المراحل المبكرة التي تحدد خصائص النظام ومتطلباته إلى مرحلة صيانة النظام التي تأتي بعد فترة من استخدام النظام وهذا التعريف يشمل عبارتين مهمتين هما [٣][٤]:

#### ١. الانضباط الهندسي (Engineering Discipline):

يتضمن تطبيق النظريات والطرائق والأدوات حيثما تكون مناسبة وبطريقة انتقائية تتناسب مع حل المشكلة وإيجاد الحلول للمشاكل التي ليس لها نظريات أو طرائق مطبقة لها، كما يجب العمل ضمن قيود المؤسسة والقيود المالية وإيجاد الحلول ضمن تلك القيود.

#### ٢. جميع جوانب إنتاج البرمجيات (All aspects of software production):

هندسة البرمجيات ليست معنية بالعمليات التقانية لتطوير البرمجيات فحسب بل مسؤولة أيضاً عن فعاليات أخرى مثل إدارة المشاريع وتطوير الأدوات والنظريات والطرائق لدعم إنتاج البرمجيات. من أهم عيوب عملية اختبار جودة البرمجيات هو عندما تكون مجموعة البيانات تحتوي على خواص غير موثوقة [٥]. على أي حال، فإن هذا التشويش في مجموعة البيانات قد يؤدي إلى استعمالات خاطئة باستخدام الطرائق التقليدية؛ لذا خاصية استخلاص الخواص (feature selection) عادة ما تستخدم في عملية تقليل البيانات واختيار بيانات ذات سمات عالية وصاخبة، وتستخدم خوارزميه استخلاص الخواص لتحديد العناصر المعينة مابين الصحيح والأكثر صحة التي تساعد الباحثين من إيجاد أفضل الحلول التي تتراوح قيمها ما بين الأمثل وأقصى الأمثل والتي تساعد على تقليل مساحة البحث والتعقيد الحسابي، ونتيجة لذلك، شبه الأمثل للحلول المثلى يمكن تحقيقها باستخدام هذه الخوارزميات [٦].

إن عملية تحقيق الأكثر صحة هو لإيجاد أفضل حل مثالي في فضاء البحث واستخدام البحث الكامل، ويعمل على زيادة القدرة لإيجاد الحلول العالية النوعية على النحو ملحوظ ضمن مدة زمنية معقولة، وتتضمن عملية تحقيق الأمثلية باختيار الخواص التي تتضمن المستعمرات الطبيعية لمعظم الحيوانات البرية مثل خوارزمية سرب

الطيور (particle swarm optimization (PSO)) وخوارزمية النمل (ant colony optimization (ACO)) وغيرها [7].

في هذا البحث اعتمد موضوع جودة البرمجيات؛ لأنه موضوع مهم لهندسة البرمجيات التي يتخصص بها قسمنا، وقد استخدمت طريقة جديدة للجمع بين خوارزمية سرب الطيور المثالية (PSO) والشبكة العصبية ذات الانتشار العكسي (Neural Network Back Propagation) لفحص جودة البرمجيات ودقتها، إذ اعتمدت خوارزمية الأسراب لاستخلاص المميزات، وتستخدم الشبكة العصبية للتعامل مع مشكلة عدم التوازن الطبقة. تستخدم الشبكة بسبب فعالية في التعامل مع اختلال الطبقة وتقييم الطريقة المقترحة باستخدام أحدث مجموعات البيانات العامة من قاعدة بيانات وكالة ناسا (NASA). إن الطرائق الذكائية يمكن أن تجد أفضل الحلول في فضاء البحث واستخدام قدرة البحث العالمية (global search) وزيادة كبيرة في القدرة على إيجاد حلول ذات جودة عالية في غضون مدة معقولة من الزمن [7].

## 2. الاعمال السابقة

حاول الباحث غريغور بانفوسكي في عام ٢٠٠٨ [٨] إجراء دراسة لتقييم جودة منتج البرمجيات، فقد كانت دراسته تركز على تقييم جودة الخصائص الخارجية لمنتج البرمجيات التي تمثل تقييم السلوك لمنتج البرمجيات أثناء تنفيذه. وتم التركيز في هذه الدراسة على تطوير نموذج الجودة (ISO/IEC ٩١٢٦) على معايير البرمجيات، وأجريت هذه الدراسة على ٧ عينات لمنتج البرمجيات واستعمال معايير نموذج ISO/IEC ٩١٢٦-٢ لتقييمها. وتوصل الباحث الى أن خصائص جودة المنتج الخارجية يمكن اعتمادها كفتة أو مجال، وأن المعايير التي يوفرها ISO/IEC ٩١٢٦-٢ يمكن أن تكون نقطة انطلاق لتعريف المعايير، وأن تجهز على النحو مناسب، الأمر الذي يتطلب جعلها متكيفة مع خصائص منتج البرمجيات على النحو مناسب.

## 3. جودة البرمجيات Quality Software

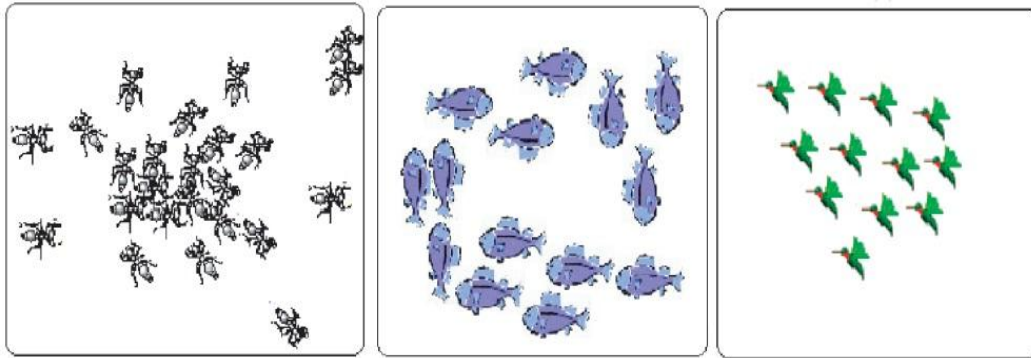
عرف رونان [٩] جودة البرمجيات بأنها امتداد للتعريف الصناعي الذي يضم مجموعة من الميزات المرغوب فيها التي ينبغي إدراجها في المنتج وذلك لتحسين أدائه مدى الحياة، وبعبارة أخرى هو التحسين المستمر للمنتج أثناء مدة استخدامه لكي يلبي احتياجات الزبون، وهذا التعريف يركز على وجود المنتج في المقام الأول الذي تكون جودته مرتبطة مع البعد الزمني. ويركز أيضاً على الميزات التي من شأنها تعزيز هذا المنتج، لهذا يعدّ التعريف هو الأشمل من ناحية تطور جودة المنتج أثناء مدة استخدامه.

اعتمد هذا البحث على تعريف رونان الذي يبحث عن جودة المنتج البرمجي في مراحل التطور ضمن البعد الزمني، وهذا يتماشى مع أساليب تطوير البرمجيات الحديثة التي تكون بحلقات متزايدة لمراحل التطوير، وفي كل حلقة نستطيع قياس الجودة على نحو جيد، وذلك لأن المنتج في أية مرحلة تطويرية يمكن أن يسلم كاملاً على المستوى الوظيفي كما في أسلوب تطوير البرمجة القصوى (Extreme Programming) [١٠]. بعد اجراء الاختبار والانتهاء من عملية التطوير يجب أن يعرف مدى جودة البرمجيات التي ستسلم. إذ يعتمد على الفرق بين البيانات الأساسية والبيانات بعد الانتهاء من عمليات الاختبار والتطوير.

## 4. الخوارزمية المقترحة في البحث:

## 1.4. خوارزمية سرب الطيور (PSO)

منذ أكثر من خمسة عقود أثبت العلماء وجود أنواع متعددة من التقانات الذكائية التي تستوحي من السلوكيات الاجتماعية الجماعية المتطورة لبعض الكائنات الحية مثل النمل والنحل والنديات ، وقد أثبتت هذه الخوارزميات فعاليتها - كما هي في الطبيعة - وكان لها دور كبير من التطبيقات الرائعة والفعالة التي شملت شتى المجالات. إذ يقوم بمحاكاة الكائنات الحية في عملية البحث عن غذائها في الطبيعة، وتكون مثالية لحل مسائل الأمثلية غير الخطية Nonlinear Optimization Problems مع بعض المحددات. والشكل رقم (٢) يمثل بعض المجتمعات الطبيعية التي يحاكيها ذكاء السرب [١١].



الشكل (٢) يمثل بعض السلوك الجماعي لبعض أنواع الأسراب

إن خوارزمية سرب الطيور (PSO) قد طورها Eberhart و Kennedy في عام ١٩٩٥ وهي تحاكي السلوك الاجتماعي لأسراب الطيور والأسماك التي تستخدمها هذه الأسراب لإيجاد المأوى والمسكن ومصادر الغذاء أو موطن آخر ملائم لها، وتعد هذه الخوارزمية من أفضل الخوارزميات المعتمدة في حل كثير من التقانات وأن المبادئ الأساسية للسلوك الجماعي للطيور تمثل البحث في فضاء البحث (D-dimention) [١٢]. وتمتاز خوارزميته بما يأتي:

- التجانس Homogeneity: الطيور كلها لها أنموذج السلوك نفسه ويتحرك السرب بدون قائد.
- المحلية Locality: حركة كل طير ستتأثر فقط بحركة الطيور المجاورة، والرؤية هي أهم حاسة لدى الطير لتنظيم السرب.
- تجنب التصادم Collision Avoidance: يقوم الطير بتجنب زملائه في السرب.
- التمرکز في السرب Flock Centering: يحاول الطير البقاء على مقربة من زملائه في السرب.
- مطابقة السرعة Velocity Matching: يقوم الطير بمحاولة مجاراة سرعة زملائه في السرب [١٣].

## 2.4 الطريقة المقترحة

في هذا البحث استخدمت خوارزمية أمثلية عناصر السرب في تحقيق جودة البرمجيات؛ إذ تقوم بأخذ عينة عشوائية لمجموعة التدريب لتكوين نماذج مستندة عليها لعملية قياس نوعية الأداء حيث البيانات التي أخذت من وكالة NASA هي (pc1) والتي تحتوي على 22 مقياساً و 1109 وحدة، وأن هذه المقاييس تستخدم لوصف جودة البرمجيات التي تكون في مرحلة التطوير واختبار البرمجيات ؛ إذ يستعمل 19 مقياساً منها خواص مدخلة ومقياس واحد يمثل الناتج الفعلي، وهذه المقاييس موضحة في الجدول (1).

الجدول (1) يمثل مجموعة المقاييس المستخدمة في البحث

رقم الوحدة	المقاييس	الوحدة ١	الوحدة ٥٠٠	الوحدة ١١٠٩
١.١	Lines of code	١	٤	٢٦
١.٤	Cyclomatic Complexity	١	١	١٨
١.٤	Essential Complexity	١	١	١٣
١.٤	Design Complexity	١	١	٦
١.٣	Program Length	٧	٧	٢٢٨
١.٣	Program Volume	١٩.٦٥	١٩.٦٥	١٣٣٥.٦٢
١.٣	Program Level	٠.٤	٠.٤	٠.٠٣
١.٣	Difficulty Level	٢.٥	٢.٥	٣٥.٨١
١.٣	Intelligent Content	٧.٨٦	٧.٨٦	٣٧.٢٩
١.٣	Programming Effort	٤٩.١٣	٤٩.١٣	٤٧٨٣٤.٢٦
١.٣	Program Vocabulary	٠.٠١	٠.٠١	٠.٤٥
١.٣	Programming Time	٢.٧٣	٢.٧٣	٢٦٥٧.٤٦
٢	LOCcode	٤	٤	٢٦
٢	LOCcodeAndComment	٠	٠	٠
١.٢	number of unique operators	٥	٥	٢٣
١.٢	number of unique operands	٢	٢	٣٥
١.٢	total occurrences of operators	٥	٥	١١٩
١.٢	total occurrences of operands	٢	٢	١٠٩
١.٤	BranchCount	١	١	٣٥
٠	Defects	٠	٠	٠

والخطوات المتبعة لتمثيل خوارزمية البحث تتلخص بالخطوات الآتية:

الخطوة الأولى : إدخال البيانات التي تقسم إلى بيانات تدريب وبيانات اختبار التي أخذت من وكالة NASA، إذ يمثل الجدول (1) البيانات كاملاً ؛ إذ تقسم إلى بيانات تدريب بنسبة 80% وبيانات اختبار 20%.

الخطوة الثانية : تحديد مصفوفة التي تستخدم لعمل استخلاص المواصفات [16\*10]

إذ يمثل الرقم 10 عدد الاختيارات التي سوف يختار قيمة واحدة منها، والرقم 16 يمثل عدد المواصفات التي سيختار من بين 19 صفة .

الخطوة الثالثة : البدء بتنفيذ خوارزمية أمثلية سرب الطيور على بيانات التدريب بالاعتماد على 16 صفة عند الاختيار 1= من أصل 10 اختيارات التي حددت في الخطوة السابقة.

الخطوة الرابعة : نهيء السرعة والمواقع بقيم ابتدائية حيث السرعة تكون قيمها صفر في البداية والمواقع تمثل 16 صفة التي حددت مسبقاً، إذ يوضح الجدول (2) القيم المختارة للمعاملات الأنموذج المتمثل بخوارزمية أمثلية سرب الطيور المدمجة بالشبكة العصبية ذات التغذية الجاهزة في ماتلاب، إذ تعتمد بصورة كبيرة على مجموعة المعلمات الخاصة بها فضلاً عن مدخلات هذا الأنموذج.

الجدول (2) مدخلات ومعلمات النموذج PSONN-SCQM

PSO Iteration No.	Data
PSO Iteration No.	100
Swarm Size	10
Inertia Weight(w)	0.9
$c_1$	0.2
$c_2$	0.1
Rand1	0.2
Rand2	0.3
No. of Hidden Neurons	{16,4}

الخطوة الخامسة: حساب دالة اللياقة للقيم لكل فرد (يمثل  $P_{best}$  أفضل موقع شخصي) بصورة ابتدائية التي تعتمد على دالة sphere، إذ تمثل الدالة المعادلة الآتية:

$$f(x) = \sum_{i=1}^n x_i^2 \quad \dots \dots \dots (1)$$

الخطوة السادسة: حساب قيمة  $g_{best}$  (أفضل موقع عام)، مقارنة قيمة دالة اللياقة مع أفضل موقع في السرب ككل، فإذا كانت القيمة الحالية أفضل من الـ  $g_{best}$  فتوضع قيمة الـ  $g_{best}$  مساوياً للموقع الحالي.

الخطوة السابعة : تحديث السرعة والموقع لكل فرد في السرب عند الدورة  $t=1$ ، وذلك بالاعتماد على المعادلتين الآتيتين:

$$v_i^{k+1} = w v_i^k + c_1 r_1^k (p_i^k - s_i^k) + c_2 r_2^k (p_g^k - s_i^k) \quad \dots \dots \dots (2)$$

$$s_i^{k+1} = s_i^k + v_i^{k+1} \quad \dots \dots \dots (3)$$

إذ إن  $w$  يمثل وزن القصور الذاتي،  $c_1$  و  $c_2$  هما ثوابت موجبة تسمى عوامل التعلم،  $r_1$  و  $r_2$  هما عدنان عشوائيان ضمن المدى  $[0,1]$ ،  $i=1,2,\dots,n$ ،  $k=1,2,\dots$  تمثل الدورة (Iteration) [14].

الخطوة الثامنة: حساب دالة اللياقة للقيم لكل فرد للقيم الجديدة.

الخطوة التاسعة: مقارنة قيمة دالة اللياقة لكل فرد بالـ  $P_{best}$  له فإذا كانت القيمة الجديدة أفضل من الـ  $P_{best}$  فتوضع قيمة  $P_{best}$  مساوية للقيمة الجديدة وموقع الـ  $P_{best}$  مساوياً للموقع الجديد في فضاء المسألة.

الخطوة التاسعة : تكرر الخطوات من الخطوة السابعة على وفق الدورات، بالنهاية نحصل على قيمة  $g_{best}$  وكذلك نقوم بخزن الأوزان الخاصة بالشبكة التي تكون خاصة بالموصفات المحددة.

الخطوة العاشرة : يكرر من الخطوة الثالثة على مواصفات محددة أخرى على وفق الاختيارات.

الخطوة الحادية عشر : في النهاية نختار المواصفات الأفضل بالاعتماد على أفضل قيمة  $g_{best}$ .

الخطوة الثانية عشر : تقييم الخوارزمية المقترحة بالاعتماد على المعايير الصحة.

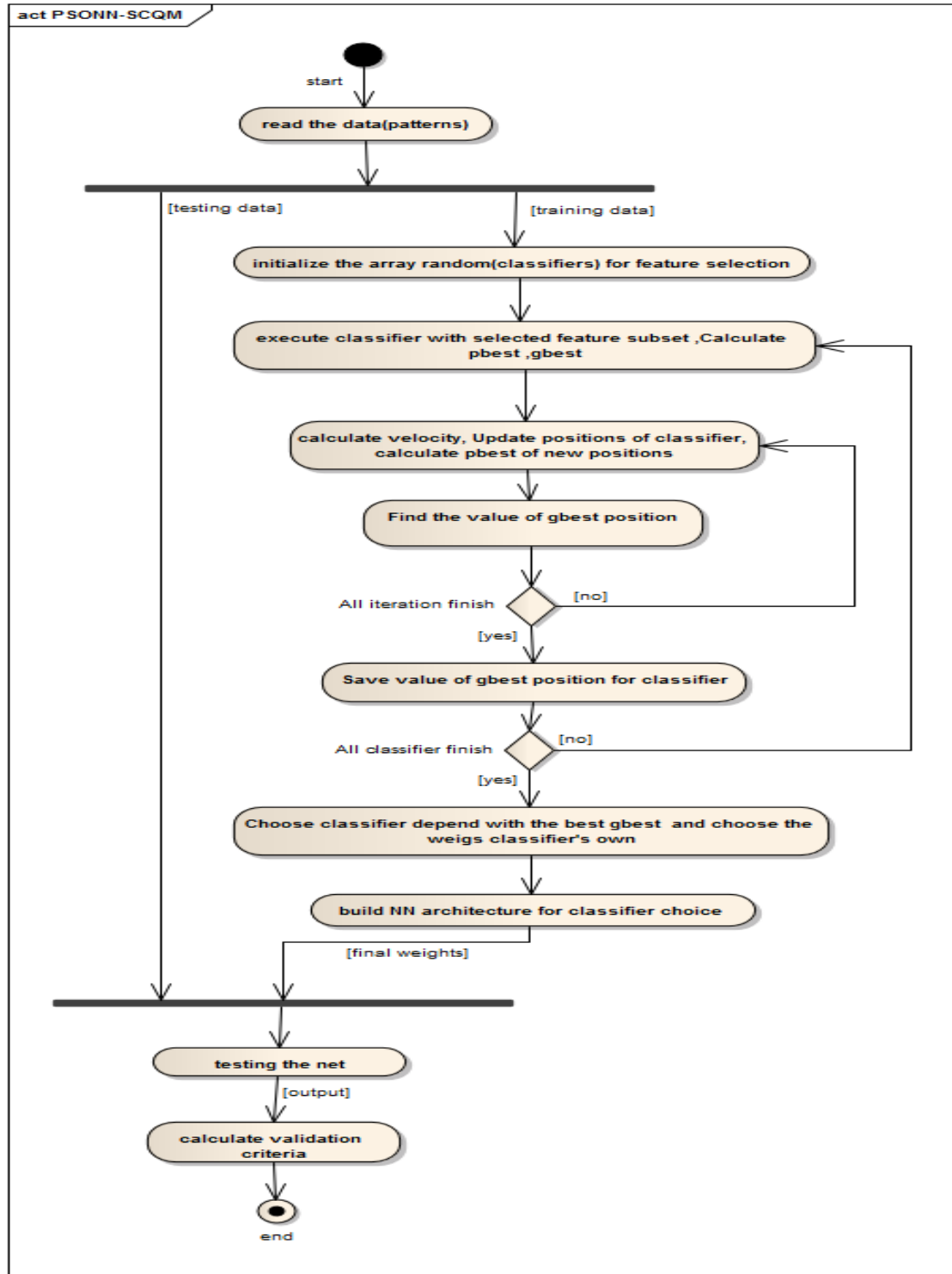
والشكل (3) يمثل مخطط الفاعلية للخوارزمية المقترحة لقياس جودة البرمجيات التي تستند على استخلاص الخواص لمجموعة من الجسيمات التي تولد مصفوفة ثنائية الأبعاد بصورة عشوائية، إذ إن موقع كل فرد (particle) من أفراد السرب يمثل مقياساً على النحو الاتي :

$$Particle(i) = [p_1 p_2 p_3 p_4 p_5 p_6 p_7 p_8 p_9 p_{10} p_{11} p_{12} p_{13} p_{14} p_{15} p_{16}] \dots \dots \dots (4)$$

SwarmMatrix

$$= \begin{bmatrix} Particle(1) \\ Particle(2) \\ \vdots \\ Particle(10) \end{bmatrix} \dots \dots \dots (5)$$

إن تغيير موقع كل فرد في السرب يعني تغير المقياس، وذلك لتقليل نسبة الخطأ في الدورة الواحدة، في كل دورة تقوم أفراد السرب بتغيير مواقعها بتحديث السرعة ومن ثم التحرك إلى الموقع الجديد الذي يمثل مقياساً جديداً الذي يعتمد على تقييم دالة اللياقة، تعاد هذه العملية لعدد من الدورات وقيمة  $P_{best}$  الأفضل تعد هي قيمة  $g_{best}$  والحصول على الأوزان المثالية للشبكة العصبية بالنسبة لـ 16 مقياس الأول ، وفي النهاية نحدد قيمة أفضل  $g_{best}$  من بين 10 قيم لـ  $g_{best}$  التي سوف تحدد المقاييس 16 الأفضل من بين 10 احتمالات المحدد مسبقاً و الأوزان المثالية لشبكة العصبية، وبعد أن يحدد الـ 16 مقياس الأفضل تحسب معايير الدقة لتصنيف جودة البرمجيات .



الشكل (٣) يمثل مخطط الفاعلية للخوارزمية المقترحة

## 5. النتائج والاستنتاجات

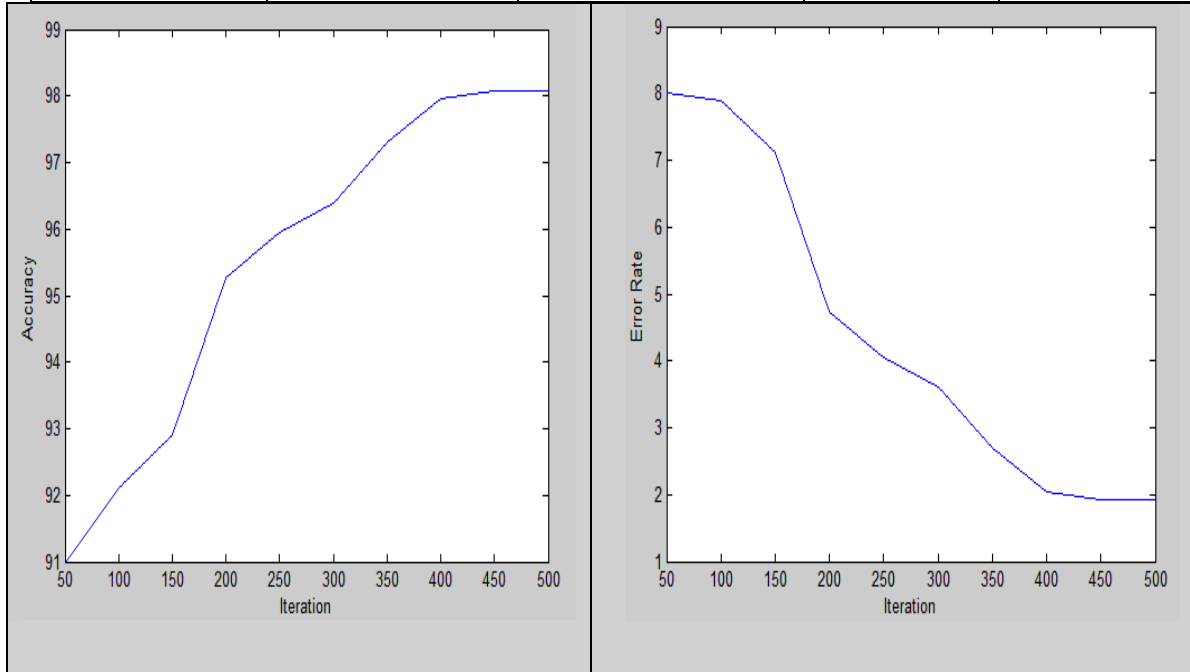
طبقت الخوارزمية المقترحة باستخدام (Matlab R2011a) بالاعتماد على معايير الصحة التي تشمل حساب دقة التصنيف (Classification Accuracy) وحساب نسبة الخطأ (Error Rate) ومتوسط مربع الخطأ (Mean Square Error) وجذر متوسط مربع الخطأ (Root Mean Square Error)، إذ كلما اقتربت نسبة

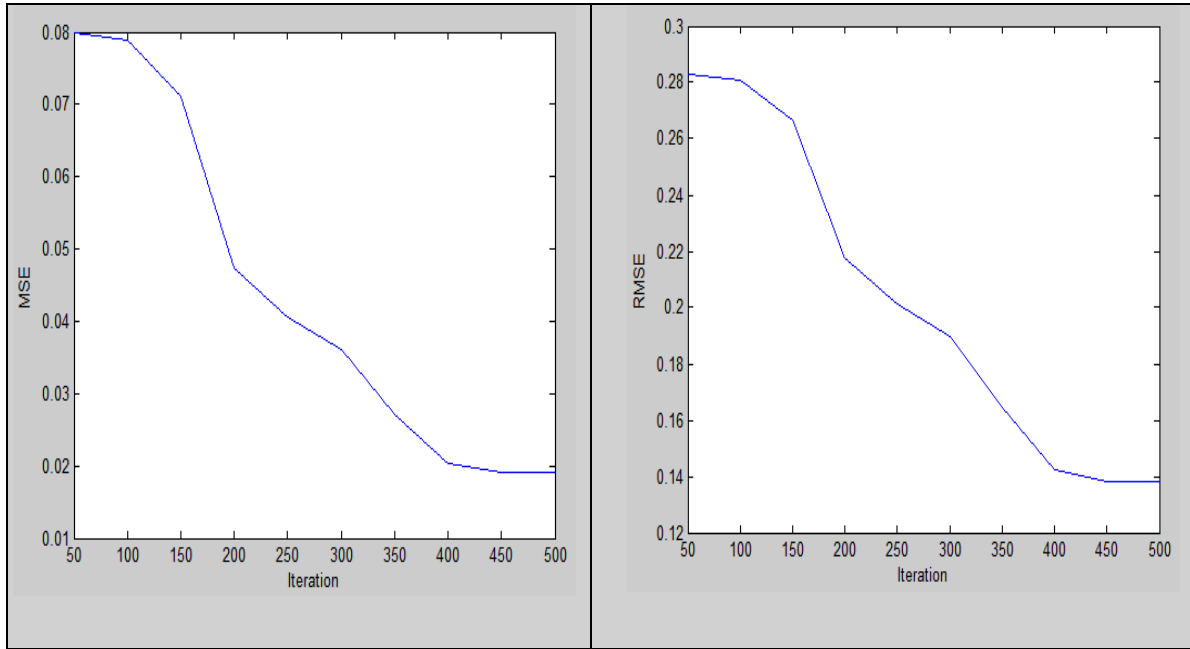


الدقة من المئة ازدادت فرصة الوصول إلى الحل الأمثل التي تمثل الفرق بين القيمة الفعلية والقيمة المتوقعة ،بينما نسبة الخطأ يحسب بالاعتماد على نسبة الدقة ، متوسط مربع الخطأ يحسب مجموع مربع الفرق بين القيمة الفعلية والقيمة الناتجة على عدد القيم المستخدمة ، جذر متوسط مربع الخطأ يحسب الجذر التربيعي لمتوسط مربع الخطأ [15]وكما موضح بالجدول (3) يمثل نتائج تقييم الخوارزمية المقترحة لقياس جودة البرمجيات ،أما الشكل (4) فيمثل نتائج تقييم الخوارزمية المقترحة لقياس جودة البرمجيات.

الجدول(3) يمثل نتائج تقييم الخوارزمية المقترحة لقياس جودة البرمجيات

Iteration	Accuracy	Error Rate	MSE	RMSE
50	91	8.0045	0.08	0.2829
100	92.1082	7.8918	0.0789	0.2809
150	92.8974	7.1086	0.0710	0.2665
200	95.2649	4.7351	0.0474	0.2176
250	95.9414	4.0586	0.0406	0.2015
300	96.3923	3.6077	0.0361	0.1899
350	97.2943	2.7057	0.0271	0.1645
400	97.9707	2.0293	0.0203	0.1425
450	98.0834	1.9166	0.0192	0.1384
500	98.0834	1.9166	0.0192	0.1384





الشكل (4) يمثل نتائج تقييم الخوارزمية المقترحة لقياس جودة البرمجيات

#### 6. مناقشة النتائج

يتبين من استخدام خوارزمية أمثلية عناصر السرب التي هجنت مع الشبكة العصبية ذات الانتشار العكسي لاستخلاص الصفات التي تكونت عشوائياً في مصفوفة؛ وذلك لاختيار الصفات ذات القيم الأفضل وتحديد الأوزان المثالية للشبكة وبعدها تحسب قيمة المعايير الخاصة بقياس جودة البرنامج الذي حدد، وتبين من قيم معايير الدقة الموضحة في الجدول (3) أن قيمة الدقة في تزايد والمعايير الأخرى في نقصان عند استخدام الخوارزمية المقترحة، وبمقارنة نسبة الدقة التي حصلنا عليها يتبين أن الخوارزمية أعطت فوائدها في زيادة الدقة لتصنيف جودة البرمجيات حيث توصلت إلى نسبة دقة 98% وهي نسبة لم تصلها الطرائق المستخدمة في البحث "Combining Particle Swarm Optimization based Feature Selection and Bagging for Software Defect Prediction" الوصل إليها.

المصادر

- [1] IEEE Std 610.12., 1990, IEEE Standard Glossary of Software Engineering Terminology.
- [2] Broy, M., (2011), " The Future of Software Engineering", S. Nanz (ed.), Springer-Verlag Berlin Heidelberg. ISBN 978-3-642-15186-6.
- [3] Pressman, R., 2010, "Software Engineering: A Practitioner's Approach", 7<sup>th</sup> Edition, McGraw-Hill, New York, USA, ISBN 978-0-07-337597-7
- [4] Sommerville, I. , 2011 , "Software Engineering" , 9th Edition , Addison Wesley, ISBN-13: 978-0-13-703515-1
- [5] S. Kim, H. Zhang, R. Wu and L. Gong, 2011 “Dealing with noise in defect prediction”, Proceeding of the 33rd International Conference on Software engineering - ICSE '11, (), pp. 481-490.
- [6] M. M. Kabir, M. Shahjahan and K. Murase, , (2012) “A new hybrid ant colony optimization algorithm for feature selection”, Expert Systems with Applications, vol. 39, no. 3, pp. 3747-3763
- [7] S. C. Yusta, (2009), “Different metaheuristic strategies to solve the feature selection problem”, Pattern Recognition Letters, vol. 30, no. 5, pp. 525-534.
- [8] Panovski , Gregor. ,(2008)," Product Software Quality ", MASTER'S THESIS, Department of Mathematics and Computing Science, TECHNISCHE UNIVERSITEIT EINDHOVEN.
- [9] Fitzpatrick, Ronan. ,(1996)." Software Quality: Definitions and Strategic Issues", MSc Computing Science Advanced Research Module, Staffordshire University.
- [10] Pressman, Roger S.,(2010)," Software Engineering: A Practitioner's Approach", 7th edition , McGraw-Hill, ISBN 978–0–07–337597–7, ISBN 0–07–337597–7.
- [11] Mohd. Ehmer Khan. 2011, “Different Approaches to White Box Testing Technique for Finding Errors", International Journal of Software Engineering and Its Applications.
- [12] Grosan, C., Abraham, A.,(2011), " Intelligence Systems", Springer, Verlag Berlin Heidelberg, ISBN 978-3-642-21003-7.
- [13] Chang R, Lai L. , Wang J. and Kouh J. , 2007, "Intrusion Detection by Backpropagation Neural Networks with Sample-Query and Attribute-Query", Department of Engineering Science and Ocean Engineering, National Taiwan University, International Journal of Computational Intelligence Research. Vol.3, No. 1 .
- [14] Gaitonde,V.,Karnik,S.,(2012),"Minimizing burr size in drilling using artificial neural network(ANN)-Particle swarm optimization(PSO) approach", Springer Science+ Business, J IntellManuf , doi: 10.1007/s10845-010-0481-5,Vol.23,pp.1783-1793.
- [15] Saleem Basha, Dhavachelvan P., 2010, " Analysis of Empirical Software Effort Estimation Models ", (IJCSIS) International Journal of Computer Science and Information Security, Vol. 7, No. 3.