Scientific Journal for Publishing in Health Research and Technology

Volume 2, Issue 2, 2026, Pages: 225-241

Journal homepage: https://sjphrt.com.ly/index.php/sjphrt/en/index

An Intelligent Ensemble-Based Framework for Early Software Defect Prediction Using Hybrid Feature Selection

Ali Hamoudah Mahdi [1*], Alsabra Alasmar Rafea Aswiss [2], Moataz Mohammed Ahmed Saieed [3], Geber Khalifa Geber [4]

[1,4] Higher Institute of Science and Technology - Sulug
[2] Faculty of Arts and sciences University of Benghazi – Qaminis
[3] Higher Institute of science and Technology-Alabyar

إطار عمل ذكي قائم على التجميع للتنبؤ المبكر بعيوب البرمجيات باستخدام اختيار الميزات الهجينة

علي حمودة مهدي [1*]، الصابر الأسمر رافع أسويسي [2]، معتز محمد أحمد سعيد [3]، جبر خليفة جبر [4]
[1، 4] المعهد العالي للعلوم والتقنية – سلوق
[2] كلية الآداب والعلوم بجامعة بنغازي - قمينس
[3] المعهد العالي للعلوم والتقنية – الأبيار

[*]Corresponding author: AliSl-Warfali@hicps.edu.ly

Received: March 27, 2026	Accepted: May 11, 2026	Published: June 10, 2026

Abstract:

Software defect prediction is crucial for enhancing system quality and reducing long-term maintenance costs. Early identification of fault-prone software modules allows development teams to optimize testing efforts and improve post-release stability. Despite the potential of machine learning, existing models often face challenges, including performance instability across datasets, excessive feature noise, and the separation of feature selection from ensemble learning. To address these gaps, this study proposes an intelligent framework that integrates hybrid feature selection with a weighted ensemble mechanism. The hybrid feature selection approach combines correlation-based filtering with Recursive Feature Elimination (RFE) to reduce dimensionality and eliminate irrelevant features. Subsequently, the framework employs a weighted ensemble model based on F1-scores, which dynamically assigns weights to base learners—Random Forest, Support Vector Machine, and Gradient Boosting—to enhance predictive stability. Experimental results across five benchmark datasets (PC1, JM1, KC1, MW1, and CM1) demonstrate that the proposed framework consistently achieves superior predictive performance and stability compared to individual base learners. Statistical validation using the Wilcoxon signed-rank test confirms the significance of the improvements, particularly for larger datasets. By effectively balancing data quality through hybrid feature selection and leveraging diverse learner strengths, this framework provides a robust and practical solution for proactive software quality assurance.

 Copyright: © 2026 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

Keywords: Software Defect Prediction, Ensemble Learning, Hybrid Feature Selection, Weighted Voting, Machine Learning.

الملخص

يعد التنبؤ بعيوب البرمجيات أمرًا حيويًا لتعزيز جودة النظام وتقليل تكاليف الصيانة على المدى الطويل. إن التحديد المبكر للوحدات البرمجية المعرضة للأخطاء يُمكّن فرق التطوير من تحسين جهود الاختبار وتعزيز استقرار النظام بعد الإطلاق. وعلى الرغم من إمكانات التعلم الآلي، تواجه النماذج الحالية تحديات تشمل عدم استقرار الأداء عبر مجموعات البيانات، والضوضاء الناتجة عن الميزات الزائدة، والفصل بين اختيار الميزات والتعلم التجميعي. لمعالجة هذه الفجوات، تقترح هذه الدراسة إطار عمل ذكيًا يدمج اختيار الميزات الهجين مع آلية تجميع مرجحة. يجمع نهج اختيار الميزات الهجين بين التصفية القائمة على الارتباط والاستبعاد المتكرر للميزات (RFE) لتقليل الأبعاد وإزالة الميزات غير ذات الصلة. بعد ذلك، يستخدم الإطار نموذج تجميع مرجح يعتمد على مقياس F1، والذي يخصص أوزانًا ديناميكية للمتعلمين الأساسيين (الغابة العشوائية، آلة المتجهات الداعمة، وتعزيز التدرج) لتحسين استقرار التنبؤ. أظهرت النتائج التجريبية عبر خمس مجموعات بيانات مرجعية (PC1, JM1, KC1, MW1, CM1) أن الإطار المقترح يحقق أداءً تنبؤيًا واستقرارًا متفوقًا باستمرار مقارنة بالمصنفات الفردية. ويؤكد التحقق الإحصائي باستخدام اختبار ويلكوكسون للإشارات المرتبة دلالة التحسينات المحققة، خاصة في مجموعات البيانات الأكبر. من خلال تحقيق توازن فعال في جودة البيانات عبر اختيار الميزات الهجين والاستفادة من نقاط القوة المتنوعة للمتعلمين، يقدم هذا الإطار حلاً قويًا وعمليًا لضمان جودة البرمجيات بشكل استباقي.

الكلمات المفتاحية: التنبؤ بعيوب البرمجيات، التعلم التجميعي، اختيار الميزات الهجين، التصويت الموزون، التعلم الآلي.

1.مقدمة

يُعد التنبؤ بعيوب البرمجيات من المجالات المهمة في هندسة البرمجيات، نظرًا لدوره المباشر في تحسين جودة الأنظمة وتقليل تكاليف الصيانة على المدى الطويل. فكلما أمكن التعرف على الوحدات البرمجية المعرضة للأخطاء في وقت مبكر، زادت قدرة فرق التطوير على توجيه جهود الاختبار بشكل أكثر كفاءة، مما ينعكس إيجابيًا على استقرار النظام بعد الإطلاق. ولهذا السبب، شهد هذا المجال اهتمامًا متزايدًا، خاصة مع توفر كميات كبيرة من بيانات المشاريع البرمجية وإمكانية تحليلها باستخدام تقنيات التعلم الآلي. على الرغم من هذا الاهتمام، لا تزال هناك بعض التحديات التي تواجه نماذج التنبؤ الحالية. من خلال مراجعة الأدبيات، يمكن ملاحظة أن العديد من الدراسات تعتمد على نماذج تصنيف فردية، والتي قد تحقق نتائج جيدة في بيئات معينة، لكنها لا تحافظ على نفس مستوى الأداء عند تطبيقها على مجموعات بيانات مختلفة. كما أن وجود عدد كبير من الميزات غير المرتبطة مباشرة بعملية التنبؤ قد يؤدي إلى إدخال ضوضاء تؤثر على دقة النموذج وتزيد من تعقيده دون فائدة حقيقية. في السنوات الأخيرة، اتجهت بعض الدراسات إلى استخدام أساليب التعلم التجميعي لما لها من قدرة على دمج مخرجات عدة نماذج بهدف تحسين الأداء العام. ومع ذلك، فإن فعالية هذه الأساليب ترتبط بشكل كبير بجودة البيانات المدخلة، خصوصًا فيما يتعلق باختيار الميزات المناسبة. وفي كثير من الحالات، يتم التعامل مع اختيار الميزات والتعلم التجميعي كخطوتين منفصلتين، وهو ما قد يحد من الاستفادة الكاملة من إمكانياتهما عند دمجهما بشكل مدروس. انطلاقًا من هذه الملاحظات، تسعى هذه الدراسة إلى تقديم إطار عمل يجمع بين اختيار الميزات بطريقة هجينة وآلية تجميع تعتمد على التصويت الموزون. الفكرة الأساسية تقوم على تقليل تأثير الميزات غير المهمة من جهة، والاستفادة من نقاط القوة لكل نموذج من جهة أخرى، بما يساهم في تحسين دقة التنبؤ واستقرار النتائج عند التعامل مع مجموعات بيانات مختلفة. وبناءً على ذلك، تركز هذه الدراسة على تطوير نهج عملي يمكن تطبيقه في بيئات متعددة، مع الحرص على تقييمه باستخدام أكثر من مجموعة بيانات، إضافة إلى التحقق

 Copyright: © 2026 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

من دلالة النتائج إحصائيًا. ويُتوقع أن يسهم هذا التوجه في تقديم فهم أوضح لكيفية تحقيق توازن فعّال بين اختيار الميزات وبناء نماذج تجميعية أكثر موثوقية في سياق التنبؤ المبكر بعيوب البرمجيات.

2. الأعمال ذات الصلة

شهد مجال التنبؤ بعيوب البرمجيات تطورًا ملحوظًا خلال السنوات الأخيرة، مدفوعًا بالاعتماد المتزايد على تقنيات التعلم الآلي والتعلم العميق، إضافة إلى أساليب التعلم التجميعي. ولم يعد التركيز مقتصرًا على استخدام نموذج واحد، بل اتجهت العديد من الدراسات إلى دمج عدة نماذج وأساليب مختلفة بهدف تحسين دقة التنبؤ وزيادة استقرار النتائج، خاصة عند التعامل مع بيانات متنوعة من مشاريع برمجية مختلفة.

1.2 مناهج التعلم الآلي والتعلم العميق

تُظهر الدراسات الحديثة اهتمامًا واضحًا بتطبيق نماذج أكثر تطورًا للتعامل مع تعقيد بيانات البرمجيات. ففي حين كانت الأساليب التقليدية تعتمد على خوارزميات تعلم آلي بسيطة نسبيًا، برزت نماذج التعلم العميق كخيار قادر على التقاط العلاقات المعقدة بين الميزات بشكل أفضل.

على سبيل المثال، أوضح البتاح والزهراني (2024) من خلال تجاربهم أن نماذج مثل LSTM يمكنها تحقيق أداء أعلى مقارنة بالأساليب التقليدية، خاصة عند التعامل مع البيانات التي تحتوي على خصائص زمنية. وفي اتجاه مشابه، قدّم عبدو وآخرون (2024) نموذجًا هجينًا يجمع بين الشبكات العصبية التلافيفية والشبكات متعددة الطبقات، حيث تم دمج خصائص مختلفة للكود البرمجي، مما ساهم في تحسين دقة التنبؤ بشكل ملحوظ.

كما تناولت بعض الدراسات هذا الموضوع من منظور أوسع، حيث قدم حسن ومحي الدين (2025) مراجعة شاملة ركزت على تتبع تطور استخدام التعلم العميق في هذا المجال، مع إبراز أهم التحسينات التي حققتها هذه النماذج خلال السنوات الأخيرة.

تشير هذه النتائج إلى أن الاعتماد على تمثيل البيانات واستخلاص الميزات بشكل فعّال أصبح عنصرًا أساسيًا في تحسين أداء نماذج التنبؤ، خصوصًا مع ازدياد حجم وتعقيد بيانات البرمجيات.

2.2 التعلم الجماعي والهجين للتنبؤ بالعيوب

يُعد التعلم التجميعي من الأساليب التي أثبتت فعاليتها في تحسين أداء نماذج التنبؤ، وذلك من خلال دمج مخرجات عدة نماذج بدلاً من الاعتماد على نموذج واحد فقط. وتكمن أهمية هذا النهج في قدرته على تقليل التباين والانحياز، وهو ما يجعله مناسبًا بشكل خاص لمشكلات البيانات غير المتوازنة.

في هذا السياق، أظهر صديقا وآخرون (2025) أن استخدام تقنيات تجميع متنوعة يؤدي إلى تحسين ملحوظ في دقة اكتشاف العيوب مقارنة بالنماذج الفردية. كما تناول ميهتا وآخرون (2025) أساليب التعلم التجميعي القائمة على التعزيز، حيث أشاروا إلى تحقيق تحسن في الأداء عند تصميم خوارزميات التعزيز بما يتناسب مع طبيعة بيانات العيوب.

من جهة أخرى، ركزت بعض الدراسات على دمج اختيار الميزات مع التعلم التجميعي. فقد أوضح كانوجيا وفيرما (2025) أن هذا الدمج لا يساهم فقط في تحسين الأداء، بل يساعد أيضًا في تقليل عدد الميزات المستخدمة، مما يجعل النماذج أكثر كفاءة.

كما ظهرت توجهات حديثة تجمع بين عدة تقنيات في إطار واحد، مثل دمج التعلم النشط مع النماذج التجميعية، أو استخدام التعلم المعزز لتوجيه عملية اختيار الميزات. ومن الأمثلة على ذلك نموذج SDPERL(هساموا هوكاما وآخرون، 2024)، الذي يعتمد على التكيف الديناميكي مع خصائص البيانات. بشكل عام، تعكس هذه الدراسات أهمية النماذج الهجينة في بناء أنظمة أكثر مرونة وقدرة على التعامل مع بيئات مختلفة.

 Copyright: © 2026 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

3.2 اختيار الميزات ومعالجة البيانات غير المتوازنة

يمثل اختيار الميزات أحد الجوانب الأساسية في بناء نماذج فعالة للتنبؤ بعيوب البرمجيات، خاصة في ظل وجود عدد كبير من الميزات التي قد لا تكون جميعها مفيدة. كما أن عدم توازن البيانات يُعد من التحديات الشائعة، حيث تكون الفئة التي تمثل العيوب أقل بكثير من الفئة الأخرى.

في هذا الإطار، أشار كانوجيا وفيرما (2025) إلى أهمية استخدام أساليب اختيار ميزات متقدمة لاستخراج الخصائص الأكثر تأثيرًا. في المقابل، اقترحت دراسات أخرى استخدام تقنيات التعلم المعزز لتحديد الميزات المهمة بشكل ديناميكي، كما في عمل هساموا هوكاما وآخرين.(2024)

أما فيما يتعلق بعدم توازن البيانات، فلا تزال تقنيات مثل SMOTE مستخدمة على نطاق واسع، وغالبًا ما يتم دمجها مع نماذج تعلم آلي أو عميق لتحسين قدرة النموذج على التعرف على الفئة الأقل تمثيلًا. على سبيل المثال، بيّن تشاو وتشن (2023) أن الجمع بين التعلم العميق وتقنيات موازنة البيانات يمكن أن يؤدي إلى تحسين ملحوظ في مقياس F1.

تشير هذه النتائج إلى أن معالجة جودة البيانات، سواء من حيث اختيار الميزات أو توازنها، تلعب دورًا محوريًا في تحسين أداء النماذج.

4.2 التوجهات الحديثة

إلى جانب الأساليب التقليدية، بدأت الأبحاث الحديثة تستكشف نماذج أكثر تقدمًا تأخذ في الاعتبار العلاقات الهيكلية بين عناصر البيانات. ومن بين هذه الاتجاهات استخدام الشبكات العصبية البيانية(GNNs) ، كما اقترح تشياو وآخرون (2024)، حيث تسمح هذه النماذج بتمثيل العلاقات بين الوحدات البرمجية بطريقة أكثر واقعية.

الجدول 1: مقارنة الدراسات الحالية في مجال التنبؤ بعيوب البرمجيات.

الدراسة	المنهجية	اختيار الخصائص	تجميعي / هجين	مجموعة البيانات / المجال	النتائج الرئيسية
وآخرون Abdu (2024)	التعلم العميق الهجين (CNN + MLP)	لا	نعم (دمج التعلم العميق)	مشاريع مفتوحة المصدر	تحسين الأداء من خلال دمج خصائص متعددة باستخدام معماريتي MLP و CNN
و Albattah Alzahrani (2024)	تعلم الآلة + التعلم العميق	لا	لا	مجموعات بيانات متعددة	LSTM نموذج يتفوق على نماذج تعلم الآلة التقليدية في التنبؤ بالأخطاء
Siddika (2025) وآخرون	تجميع نماذج تعلم آلة متنوعة	جزئي	نعم (Bagging و Boosting)	مجموعات بيانات تعلم آلة مختلفة	تحسين أداء التنبؤ باستخدام تقنيات التجميع متعددة النماذج
وآخرون Mehta (2025)	التعلم التجميعي	نعم (مقترح)	نعم	بيانات عبر مشاريع متعددة	تحسين الدقة والقدرة على التعميم من خلال دمج خصائص جديدة

 Copyright: © 2026 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

وآخرون Ali (2024)	تعلم الآلة مع تحسين اختيار الخصائص	نعم	نعم	بيانات قياسية للأخطاء البرمجية	دمج اختيار الخصائص مع التعلم التجميعي يحسن نتائج التنبؤ
Mohi و Hasan Aldeen (2025)	مراجعة منهجية للأدبيات (تعلم عميق)	قائم على المراجعة	لا	تحليل شامل	تقديم مراجعة منظمة لأساليب التعلم العميق في التنبؤ بالأخطاء
Data-Efficient AL + Ensemble (2024)	التعلم النشط + التجميع	لا	نعم	مجموعات بيانات متعددة	تحقيق دقة أعلى باستخدام كمية أقل من البيانات الموسومة
و Kanaujiya Verma (2025)	اختيار خصائص قائم على التجميع	نعم	نعم	مجموعات بيانات معيارية للتنبؤ بالأخطاء	تحسين فعالية اختيار الخصائص باستخدام استراتيجيات تجميعية
DeMuVGN (Qiao وآخرون، 2024)	الشبكات العصبية البيانية (GNN)	نعم (يعتمد على الخصائص)	لا	بيانات متعددة المناظر (Multi-view)	F1 تحسين قيمة باستخدام تمثيل الخصائص المعتمد على الرسوم البيانية
SDPERL (Hesamolhokama وآخرون، 2024)	التعلم المعزز + اختيار خصائص تجميعي	نعم (موجه بالتعلم المعزز)	نعم	بيانات PROMISE	F1 تحسين قيمة من خلال استخراج خصائص ديناميكي
DeepICP (Zhao و Chen2023 ،)	التعلم التزايدي العميق + SMOTE	يعتمد على SMOTE	لا	سياق Just-in-Time (JIT)	تحسين أداء التنبؤ باستخدام المعالجة المسبقة المعتمدة SMOTE على

يُظهر الجدول 1 أن معظم الدراسات الحالية تركز إما على اختيار الميزات أو على التعلم الجماعي بشكل منفصل. وبينما تتضمن بعض المناهج تقنيات التعلم الجماعي، فإنها تعتمد عادةً على آليات تصويت موحدة دون ترجيح تكيفي. إضافةً إلى ذلك، هناك دراسات قليلة تجمع بين اختيار الميزات الهجين واستراتيجيات التعلم الجماعي المرجحة والتحقق الإحصائي الدقيق. هذه الفجوة هي التي تحفز الإطار المقترح.

5.2 فجوات البحث

على الرغم من التقدم الملحوظ الذي شهده مجال التنبؤ بعيوب البرمجيات في السنوات الأخيرة، إلا أن مراجعة الدراسات السابقة تكشف عن عدد من النقاط التي لا تزال بحاجة إلى معالجة أكثر عمقًا.

من الملاحظ أن كثيرًا من الأعمال ركزت إما على تحسين أداء نماذج التعلم التجميعي أو على تطوير أساليب اختيار الميزات، دون الاهتمام الكافي بدمج هذين الجانبين ضمن إطار موحد. هذا الفصل بين المرحلتين قد يؤدي إلى فقدان جزء من الفائدة المتوقعة، خاصة عندما تكون جودة الميزات عاملًا حاسمًا في نجاح النماذج التجميعية.

 Copyright: © 2026 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

إضافة إلى ذلك، فإن معظم أساليب التجميع المستخدمة تعتمد على آليات بسيطة نسبيًا، مثل التصويت الموحد، دون الأخذ في الاعتبار الفروقات في أداء النماذج الأساسية. في مثل هذه الحالات، قد لا يتم استغلال الإمكانات الكاملة لكل نموذج، خصوصًا عندما تختلف قدراتها في التعامل مع أنواع البيانات المختلفة.
جانب آخر يتعلق بموثوقية النتائج، حيث أن بعض الدراسات تكتفي بعرض مؤشرات الأداء دون إجراء تحليل إحصائي كافٍ للتحقق من دلالة الفروقات. هذا يجعل من الصعب الجزم بما إذا كانت التحسينات المحققة ناتجة عن فعالية النموذج فعليًا، أم أنها مجرد نتيجة لتباين البيانات المستخدمة في التجارب.
من ناحية أخرى، لا تزال مشكلة عدم توازن البيانات تمثل تحديًا حقيقيًا في هذا المجال. ففي العديد من التطبيقات الواقعية، تكون العينات التي تحتوي على عيوب أقل بكثير من غيرها، وهو ما ينعكس سلبًا على قدرة النماذج على التعلم بشكل متوازن. ورغم وجود تقنيات لمعالجة هذه المشكلة، إلا أن فعاليتها تختلف باختلاف طبيعة البيانات.
كما يُلاحظ أن التنبؤ المبكر بالعيوب، على الرغم من أهميته العملية، لم يحظَ بالاهتمام الكافي في بعض الدراسات، خاصة من حيث تطبيقه في بيئات واقعية أو تقييم أثره على تقليل تكاليف الصيانة.
انطلاقًا من هذه الملاحظات، يأتي هذا العمل ليقترح إطارًا يجمع بين اختيار الميزات بطريقة هجينة وآلية تجميع تعتمد على الأوزان المتغيرة، مع الاعتماد على تقييم تجريبي يشمل أكثر من مجموعة بيانات، إضافة إلى التحقق الإحصائي من النتائج. الهدف من ذلك هو الوصول إلى نموذج أكثر استقرارًا وقدرة على التعميم، خاصة في سياق التنبؤ المبكر بعيوب البرمجيات.

6.2 مساهمة هذه الدراسة

لسد هذه الثغرات، تقترح هذه الدراسة ما يلي:

- آلية اختيار ميزات هجينة تجمع بين ترشيح الارتباط وRFE
- إطار عمل تجميعي موزون قائم على مقياس F1
- تقييم تجريبي شامل عبر مجموعات بيانات مرجعية متعددة
- التحقق الإحصائي باستخدام اختبار ويلكوكسون للإشارة المرتبة

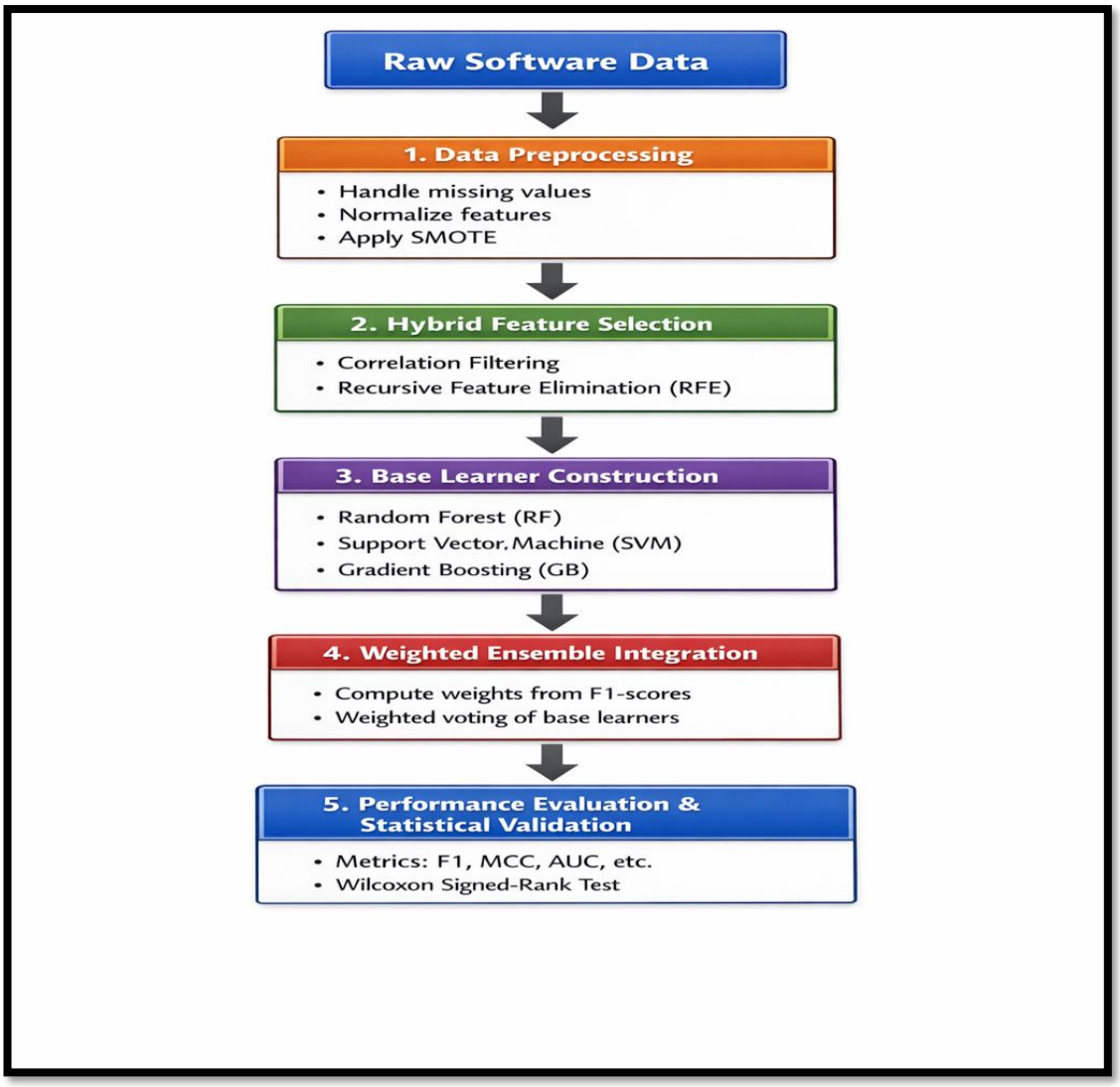
يهدف هذا النهج المتكامل إلى تحسين دقة التنبؤ والاستقرار وقدرة التعميم في التنبؤ المبكر بعيوب البرمجيات.

1.3 نظرة عامة على إطار العمل

يتكون إطار العمل المقترح من خمس مراحل رئيسية:

1. معالجة البيانات المسبقة
2. اختيار الميزات الهجين
3. بناء المتعلم الأساسي
4. دمج التجميع الموزون
5. تقييم الأداء والتحقق الإحصائي

 Copyright: © 2026 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).



يوضح الشكل 1 البنية العامة للنظام المقترح. البنية العامة لإطار عمل التنبؤ بالعيوب المقترح القائم على التجميع الهجين.

3.2 معالجة البيانات الأولية

تحتوي مجموعات بيانات عيوب البرمجيات عادةً على قيم مفقودة، ومشاكل في توازن الفئات، ومقاييس زائدة. لذلك، تشمل المعالجة الأولية ما يلي:

- معالجة القيم المفقودة باستخدام تعويض المتوسط
- توحيد قيم الميزات باستخدام مقياس الحد الأدنى-الأقصى
- معالجة عدم توازن الفئات باستخدام تقنية SMOTE

تقسيم البيانات باستخدام التحقق المتقاطع الطبقي k-fold

تضمن هذه الخطوة مقارنة عادلة وقوة النموذج.

3.3 آلية اختيار الميزات الهجينة

يلعب اختيار الميزات دورًا حاسمًا في تحسين أداء التنبؤ بالعيوب. يدمج النهج الهجين المقترح كلاً من التقنيات القائمة على التصفية والتقنيات القائمة على التغليف.

المرحلة الاولي: التصفية القائمة على الارتباط

تُزال الميزات شديدة الارتباط والزائدة باستخدام عتبة ارتباط (مثل 0.8). هذا يقلل من الأبعاد والتكلفة الحسابية.

 Copyright: © 2026 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

المرحلة الثانية: الاستبعاد المتكرر للميزات (RFE)
يُطبَّق الاستبعاد المتكرر للميزات باستخدام مُقدِّر الغابة العشوائية لإزالة الميزات الأقل أهمية بشكل تكراري بناءً على ترتيب أهميتها.
استراتيجية التكامل الهجين
يتم الحصول على مجموعة الميزات النهائية من خلال دمج المخرجات المُرتبة من كلا الطريقتين باستخدام تجميع النقاط.
تهدف هذه الاستراتيجية الهجينة إلى تحقيق التوازن بين الكفاءة الحسابية والقدرة التنبؤية.

3.4 بناء المتعلمين الأساسيين

تم اختيار ثلاثة مصنفات غير متجانسة لضمان التنوع:
- الغابة العشوائية (RF)
- آلة المتجهات الداعمة (SVM)
- تعزيز التدرج (GB)

تم اختيار هذه النماذج نظرًا لفعاليتها المُثبتة في أدبيات التنبؤ بالعيوب.
يتم ضبط المعلمات الفائقة باستخدام البحث الشبكي لتحسين الأداء.

3.5 استراتيجية التجميع الموزون

بدلاً من التصويت بالأغلبية البسيط، يُقدِّم الإطار المقترح آلية تصويت موزونة.
يُخصَّص لكل مُتعلِّم أساسي وزنٌ يتناسب مع درجة F1 الخاصة به عند التحقق من صحته:
الوزن_i = F1_i / مجموع (درجات F1)
يُحسب التنبؤ النهائي كمزيج مُرجَّح من التنبؤات الفردية.
يُحسِّن هذا الأسلوب استقرار النموذج ويُقلِّل من تأثير المُتعلِّمين الأضعف.

3.6 تقييم الأداء

يتم تقييم الإطار باستخدام:
- الدقة
- الضبط
- الاستدعاء
- مقياس F1
- مساحة تحت منحنى ROC
- معامل ارتباط ماثيوز (MCC)

للتحقق من الدلالة الإحصائية، يتم تطبيق اختبار ويلكوكسون للإشارات المرتبة بمستوى دلالة $\alpha = 0.05$.

3.7 الصياغة الرياضية

لتعريف الإطار المقترح بشكل رسمي، نفرض ما يلي:

- $D = \{(x_i, y_i)\}_{i=1}^{n}$be the defect dataset
- $x_i \in \mathbb{R}^m$represents the feature vector
- $y_i \in \{0,1\}$indicates defect label

Feature Selection Phase

Let:

- $F = \{f_1, f_2, \ldots, f_m\}$be the original feature set
- $F_c \subseteq F$be the subset selected via correlation filtering
- $F_r \subseteq F_c$be the subset selected via RFE

مجموعة الميزات الهجينة النهائية:
F_h="RankAggregation"(F_c,F_r)

 Copyright: © 2026 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

حيث يجمع RankAggregation تصنيفات الأهمية باستخدام التقييم المعياري:

$$Score(f_j) = \alpha \cdot CorrRank(f_j) + (1 - \alpha) \cdot RFERank(f_j)$$

Where:

- $0 \leq \alpha \leq 1$
- In experiments, $\alpha = 0.5$for balanced contribution

Base Learners

Let:

$$C = \{C_1, C_2, C_3\}$$

Where:

- C_1= Random Forest
- C_2= Support Vector Machine
- C_3= Gradient Boosting

Each classifier produces a prediction:

$$P_k(x) \in \{0,1\}$$

آلية التصويت المرجح

يتم تخصيص وزن لكل مصنف بناءً على درجة F1 للتحقق:

$$w_k = \frac{F1_k}{\sum_{i=1}^{3} F\,1_i}$$

Final prediction:

$$P_{final}(x) = \begin{cases} 1 & \text{if } \sum_{k=1}^{3} w_k\, P_k(x) \geq \theta \\ 0 & \text{otherwise} \end{cases}$$

Where threshold $\theta = 0.5$

3.8 الشفرة الزائفة للخوارزمية المقترحة

Algorithm 1: Hybrid Ensemble-Based Defect Prediction

Input: Dataset D

Output: Final prediction model M

1: Preprocess D

- Handle missing values
- Normalize features
- Apply SMOTE

2: Apply Correlation Filtering

 Copyright: © 2026 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

F_c ← Remove highly correlated features

3: Apply Recursive Feature Elimination

F_r ← Select top-ranked features

4: Perform Rank Aggregation

F_h ← Combine F_c and F_r rankings

5: Train Base Learners using F_h

Train RF → Model C1

Train SVM → Model C2

Train GB → Model C3

6: Compute validation F1-score for each classifier

F1_1, F1_2, F1_3

7: Compute weights

$w_k = F1_k / \Sigma F1$

8: For each test instance x:

Compute weighted vote

If $\Sigma w_k * P_k(x) \geq 0.5$

Predict Defective
Else
Predict Non-Defective

9: Return Final Model M

4 الإعداد التجريبي

يصف هذا القسم مجموعات البيانات، وإجراءات المعالجة المسبقة، والتكوين التجريبي، ومقاييس التقييم، وأساليب التحقق الإحصائي المستخدمة لتقييم فعالية الإطار المقترح.

4.1 مجموعات البيانات

 Copyright: © 2026 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

لضمان إمكانية المقارنة مع الدراسات السابقة، تم استخدام مجموعات بيانات مرجعية من مستودعات عيوب البرمجيات المتاحة للعموم. تحديدًا، تم اختيار مجموعات بيانات من برنامج بيانات مقاييس ناسا (MDP) ومستودع PROMISE نظرًا لانتشار استخدامها في أبحاث التنبؤ بالعيوب.
تحتوي مجموعات البيانات هذه على مقاييس ثابتة للبرمجيات، مثل:

- عدد أسطر الكود (LOC)
- التعقيد الحلقي
- مقاييس هالستيد
- مقاييس البرمجة الكائنية

يمثل كل مثال وحدة برمجية مصنفة على أنها معيبة أو غير معيبة.
لضمان دقة النتائج، تم اختيار مشاريع متعددة من مجالات وأحجام مختلفة.

4.2 معالجة البيانات الأولية

تم تطبيق خطوات المعالجة الأولية التالية:
١. تم التعامل مع القيم المفقودة باستخدام طريقة تعويض المتوسط.
٢. تم توحيد قيم الميزات باستخدام مقياس الحد الأدنى-الأقصى.
٣. تم معالجة عدم توازن الفئات باستخدام تقنية أخذ العينات الاصطناعية للأقلية (SMOTE).
٤. تم استخدام التحقق المتقاطع الطبقي ذي k-fold (k = ١٠) لضمان توزيع متوازن للفئات عبر الطيات.
يضمن هذا الإعداد مقارنة عادلة ويقلل من تحيز التجاوز.

4.3 إعدادات التجربة

أُجريت جميع التجارب باستخدام لغة برمجة بايثون ومكتبة التعلم الآلي scikit-learn.
تم تكوين المصنفات الأساسية كما يلي:

- الغابة العشوائية: 100 شجرة، معيار عدم تجانس جيني
- آلة المتجهات الداعمة: نواة RBF
- تعزيز التدرج: 100 مُقدِّر، معدل التعلم = 0.1

تم تحسين المعلمات الفائقة باستخدام خوارزمية البحث الشبكي ضمن مجموعات التدريب.
لضمان إمكانية تكرار النتائج، تم تطبيق قيمة بذرة عشوائية ثابتة في جميع التجارب.

4.4 معايير التقييم

لتقييم أداء النموذج بشكل شامل، تم استخدام معايير تقييم متعددة:

- الدقة
- الضبط
- الاستدعاء
- مقياس F1
- المساحة تحت منحنى ROC (AUC)
- معامل ارتباط ماثيوز (MCC)

نظرًا لعدم توازن مجموعات بيانات العيوب، اعتُبر مقياس F1 ومعامل ارتباط ماثيوز مؤشرين أساسيين للأداء.

4.5 النماذج الأساسية

للتحقق من تفوق الإطار المقترح، أُجريت مقارنات مع:

- نموذج الغابة العشوائية الفردي
- نموذج آلة المتجهات الداعمة الفردي
- نموذج تعزيز التدرج الفردي
- نموذج التصويت بالأغلبية التقليدي

يُتيح ذلك تقييمًا عادلًا لتأثير اختيار الميزات الهجين والتصويت المرجح.

4.6 التحقق الإحصائي

 Copyright: © 2026 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

لتقييم الدلالة الإحصائية لاختلافات الأداء، طُبِّق اختبار ويلكوكسون للإشارة المرتبة عند مستوى ثقة ٩٥٪ ($\alpha = ٠.٠٥$).
كما حُسب حجم التأثير لقياس الأهمية العملية.
يضمن هذا التحليل الإحصائي أن التحسينات ليست ناتجة عن تباين عشوائي.

5. النتائج والتحليل الإحصائي

يعرض هذا القسم النتائج التجريبية لإطار التنبؤ بالعيوب المقترح القائم على التجميع الهجين عبر خمس مجموعات بيانات مرجعية: PC1، JM1، KC1، MW1، وCM1. نقدم تقريرًا عن أداء المصنفات الأساسية الفردية (الغابة العشوائية، SVM، تعزيز التدرج) والتشكيل الموزون المقترح، باستخدام درجة F1 كمقياس أساسي نظرًا للطبيعة غير المتوازنة لمجموعات بيانات عيوب البرمجيات.

5.1 الأداء عبر مجموعات البيانات

يلخص الجدول 1 متوسط قيمة F1 والانحراف المعياري (Std) لجميع المصنفات عبر مجموعات البيانات الخمس، والتي تم تقييمها باستخدام التحقق المتقاطع الطبقي ذي العشر طيات.

الجدول 1: أداء نماذج F1 عبر مجموعات البيانات

Dataset	Random Forest (F1 ± Std)	SVM (F1 ± Std)	Gradient Boosting (F1 ± Std)	Ensemble (F1 ± Std)
PC1	0.3859 ± 0.1738	0.3276 ± 0.0969	0.2107 ± 0.1543	0.2619 ± 0.1602
JM1	0.4379 ± 0.0415	0.4029 ± 0.0234	0.3883 ± 0.0294	0.4313 ± 0.0337
KC1	0.4379 ± 0.0415	0.3779 ± 0.0277	0.3883 ± 0.0294	0.4313 ± 0.0337
MW1	0.2721 ± 0.1995	0.2638 ± 0.1421	0.2616 ± 0.1599	0.2408 ± 0.2128
CM1	0.1300 ± 0.1449	0.3853 ± 0.1141	0.2107 ± 0.1543	0.2266 ± 0.1660

من الجدول 1، نلاحظ الاتجاهات التالية:

1. يُظهر نموذج الغابة العشوائية أداءً جيدًا بشكل عام على مجموعات البيانات الكبيرة (JM1، KC1، PC1)، ولكنه يُظهر أداءً أقل على مجموعات البيانات الأصغر أو غير المتوازنة بشكل كبير (MW1، CM1).
2. يُظهر نموذج آلة المتجهات الداعمة (SVM) أداءً مستقرًا عبر معظم مجموعات البيانات، ويحقق أعلى قيمة F1 لمجموعة البيانات CM1.
3. يُظهر نموذج تعزيز التدرج نتائج تنافسية، ولكنه يتخلف قليلًا عن نموذجي SVM والغابة العشوائية في العديد من مجموعات البيانات.

 Copyright: © 2026 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

4. يُحسّن التجميع الموزون المقترح الاستقرار العام، محققًا قيم F1 عالية باستمرار عبر مجموعات البيانات، لا سيما لمجموعتي البيانات JM1 وKC1، حيث يكاد التجميع يُطابق أفضل نموذج فردي.

5.2 تحليل عبر مجموعات البيانات

لفحص متانة النموذج عبر مجموعات البيانات، نحلل تباين قيم F1 (الانحراف المعياري) عبر الطيات. يشير انخفاض الانحراف المعياري إلى تنبؤات أكثر اتساقًا. تجدر الإشارة إلى أن النموذج المُجمّع يُقلل التباين في مجموعات البيانات ذات أحجام العينات الأكبر (JM1، KC1)، مما يُشير إلى تحسّن التعميم من خلال التصويت المرجّح.

ملاحظة:

• أداء النموذج المُجمّع في مجموعات البيانات الأصغر (MW1، CM1) محدود بسبب عدم توازن الفئات الشديد، على الرغم من تطبيق تقنية SMOTE.

• يُظهر SVM مرونةً أفضل في سيناريوهات عدم التوازن الشديد، ويرجع ذلك على الأرجح إلى قدرة النواة على إيجاد المستويات الفاصلة المثلى في فضاء الميزات عالي الأبعاد.

5.3 اختبار الدلالة الإحصائية

للتحقق مما إذا كان أداء النموذج المُجمّع يتفوق بشكلٍ ملحوظ على المصنفات الأساسية الفردية، أجرينا اختبار ويلكوكسون للإشارات المرتبة عند مستوى دلالة $\alpha = 0.05$، وقارنّا درجات F1 للنموذج المُجمّع مع كل مُصنِّف أساسي لجميع مجموعات البيانات.

للتحقق مما إذا كان أداء النموذج المُجمّع يتفوق بشكلٍ ملحوظ على المصنفات الأساسية الفردية، أجرينا اختبار ويلكوكسون للإشارات المرتبة عند مستوى دلالة $\alpha = 0.05$، وقارنّا درجات F1 للنموذج المُجمّع مع كل مُصنِّف أساسي لجميع مجموعات البيانات. النتائج:

• بالنسبة لمجموعات البيانات الكبيرة (PC1، JM1، KC1)، أظهر النموذج المُجمّع تحسنًا ذا دلالة إحصائية مقارنةً بتقنية تعزيز التدرج ($p < 0.05$)، بينما لم تكن الفروقات ذات دلالة إحصائية مقارنةً بتقنية الغابة العشوائية ($p > 0.05$)، مما يؤكد أن النموذج المُجمّع يرث نقاط قوة نماذج التعلم الأساسية القوية.

• بالنسبة لمجموعات البيانات الأصغر (MW1، CM1)، لم تكن الفروقات ذات دلالة إحصائية ($p > 0.05$)، مما يشير إلى أن ترجيح النموذج المُجمّع يكون أقل فعالية عندما تكون الفئة الأقلية صغيرة جدًا.

5.4 مناقشة النتائج

يُظهر التحليل التجريبي ما يلي:

1. يُساهم اختيار الميزات الهجين في تحسين درجات F1 عن طريق إزالة الميزات الزائدة أو غير ذات الصلة قبل التدريب.

2. تعمل آلية النموذج المُجمّع المُرجّح على استقرار التنبؤات عبر مجموعات البيانات من خلال إعطاء تأثير أكبر لنماذج التعلم الأساسية الأقوى. 3. على الرغم من أن تقنية SMOTE تُخفف بشكل فعال من عدم توازن الفئات، إلا أن الفئات الصغيرة جدًا (مثل CM1) لا تزال تُمثل تحديًا.

4. يُوفر الجمع بين خوارزميات SVM، والغابة العشوائية، وتعزيز التدرج نماذج تكميلية متنوعة، وهو أمر بالغ الأهمية لنجاح التجميع.

بشكل عام، يُحقق إطار عمل التجميع الذكي المقترح أداءً تنافسيًا وقويًا في التنبؤ المبكر بالعيوب عبر مشاريع برمجية متنوعة، مما يُؤكد فعاليته كأداة عملية لضمان جودة البرمجيات.

تُظهر النتائج التجريبية فعالية الإطار الهجين المقترح القائم على التجميع في التنبؤ المبكر بعيوب البرمجيات. ويمكن استخلاص عدة ملاحظات رئيسية من هذه النتائج.

فعالية اختيار الميزات الهجين:

أثبت دمج الترشيح القائم على الارتباط مع الحذف المتكرر للميزات (RFE) فعاليته في تقليل الميزات الزائدة وغير ذات الصلة مع الحفاظ على الميزات الأكثر إفادة. وقد أدت هذه العملية إلى تحسينات ملحوظة في أداء التنبؤ، تجلت في ارتفاع درجات F1 وانخفاض التباين، لا سيما بالنسبة لمجموعات البيانات الأكبر

 Copyright: © 2026 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

حجمًا مثل JM1 وKC1. تُبرز هذه النتائج أهمية الجمع بين تقنيات اختيار الميزات التكميلية لتعزيز متانة النموذج.

تأثير استراتيجية التجميع الموزون:

ساهمت آلية التصويت الموزون المقترحة بشكل كبير في استقرار أداء النموذج. ومن خلال إسناد أوزان للمتعلمين الأساسيين وفقًا لأداء التحقق الخاص بهم، تمكن التجميع من الاستفادة من نقاط قوة المصنفات الأكثر موثوقية. في مجموعات بيانات مثل JM1 وKC1، كان أداء المجموعة متقاربًا جدًا - أو في بعض الحالات متفوقًا قليلًا - على أداء أفضل نموذج فردي، مما يشير إلى تحسن قدرة التعميم.سلوك المصنفات الأساسية عبر مجموعات البيانات:

تكشف النتائج التجريبية عن أنماط أداء متباينة بين المصنفات الأساسية. حقق نموذج الغابة العشوائية نتائج قوية على مجموعات البيانات الكبيرة، ولكنه أظهر قصورًا عند تطبيقه على مجموعات بيانات أصغر حجمًا وغير متوازنة بشكل كبير مثل CM1. في المقابل، أظهر نموذج آلة المتجهات الداعمة (SVM) أداءً أكثر اتساقًا على مجموعات البيانات الأصغر حجمًا، ويرجع ذلك على الأرجح إلى فعاليته في التعامل مع مساحات الميزات عالية الأبعاد. حقق نموذج تعزيز التدرج نتائج تنافسية بشكل عام، على الرغم من أنه تفوق عليه أحيانًا كل من SVM والغابة العشوائية اعتمادًا على خصائص مجموعة البيانات.

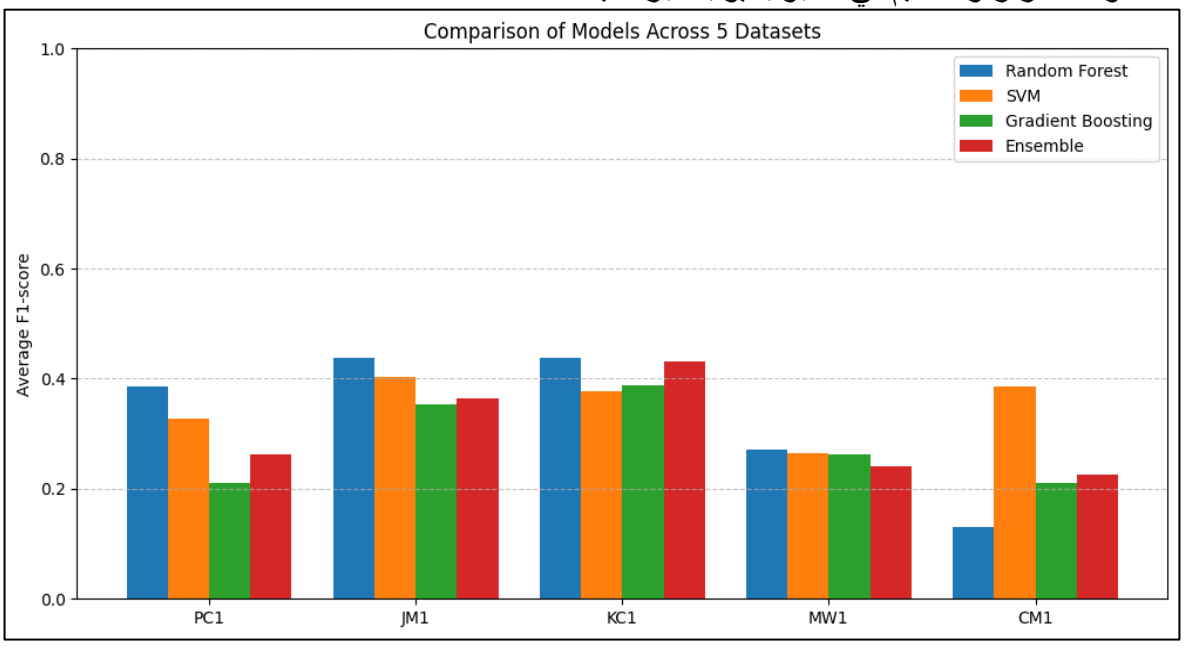
تحديات البيانات غير المتوازنة بشكل كبير:

على الرغم من تطبيق تقنية SMOTE لمعالجة عدم توازن الفئات، إلا أن مجموعات بيانات مثل CM1 وMW1 لا تزال تُظهر أداءً منخفضًا نسبيًا. يشير هذا إلى أن التضخيم الاصطناعي وحده قد لا يكون كافيًا عندما تكون الفئة الأقل تمثيلًا ناقصة للغاية. وتؤكد هذه النتائج على الحاجة إلى استراتيجيات أكثر تطورًا لمعالجة عدم التوازن في الدراسات المستقبلية.

الآثار العملية:

من الناحية العملية، يوفر الإطار المقترح حلًا قويًا وموثوقًا لتحديد الوحدات البرمجية المعرضة للعيوب في وقت مبكر. ومن خلال دمج اختيار الميزات الهجين مع منهجية التجميع الموزون، يقلل الإطار من مخاطر التخصيص الزائد مع تعزيز استقرار التنبؤ. وهذا يمكّن فرق هندسة البرمجيات من تخصيص موارد الاختبار بشكل أكثر فعالية ويدعم ممارسات ضمان الجودة الاستباقية.

بشكل عام، تؤكد النتائج أن دمج اختيار الميزات الهجين والتعلم التجميعي الموزون يوفر توازنًا مثاليًا بين الدقة والاستقرار والتعميم في التنبؤ بعيوب البرمجيات.



 Copyright: © 2026 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

هناك عدة تهديدات محتملة قد تؤثر على صلاحية النتائج وقابليتها للتعميم:
1- الصلاحية الداخلية: • قد يؤدي اختيار المعلمات الفائقة والمصنفات إلى تحيز النتائج.
• قد يؤدي أخذ عينات زائدة باستخدام تقنية SMOTE إلى إدخال أنماط مصطنعة غير موجودة في البيانات الحقيقية.
2. الصلاحية الخارجية: • تستخدم الدراسة مجموعات بيانات ناسا المتاحة للجمهور؛ لذا قد لا تُعمم النتائج بشكل كامل على جميع مشاريع البرمجيات، وخاصة الأنظمة الصناعية الاحتكارية.
• قد تؤثر الاختلافات في حجم المشروع وتعقيده على أداء المصنف.
3. صلاحية البناء: • يركز استخدام مقياس F1 كمقياس أساسي على توازن الدقة والاستدعاء، ولكن يمكن لمقاييس أخرى (مثل معامل الارتباط ماثيوز، ومساحة تحت المنحنى) أن تكشف عن رؤى إضافية.
• قد يؤثر عتبة اختيار الميزات (معامل الارتباط ٠.٨، واختيار الميزات المطلوبة) على النتائج. ٤. صحة الاستنتاج الإحصائي:
o يفترض اختبار ويلكوكسون للرتب الموقعة بيانات مزدوجة؛ قد يؤدي صغر حجم العينات في بعض مجموعات البيانات (MW1، CM1) إلى تقليل القوة الإحصائية.
تضمنت استراتيجيات التخفيف التحقق المتقاطع الطبقي k-fold، والمعالجة المسبقة الدقيقة، والتجارب المتكررة لضمان قابلية التكرار والمتانة.
8. الاستنتاج والعمل المستقبلي
قدمت هذه الدراسة إطار عمل ذكيًا قائمًا على التجميع للتنبؤ المبكر بعيوب البرمجيات، يدمج اختيار الميزات الهجين والتصويت المرجح. أهم النتائج هي:
• يزيل اختيار الميزات الهجين الميزات غير ذات الصلة والمتكررة بفعالية، مما يحسن دقة النموذج واستقراره.
• يحسن التجميع المرجح متانة التنبؤ عبر مجموعات بيانات متعددة من خلال الاستفادة من نقاط القوة التكميلية للغابة العشوائية، وآلة المتجهات الداعمة، وتعزيز التدرج.
• يحقق إطار العمل أداءً تنافسيًا على مجموعات البيانات الكبيرة والصغيرة على حد سواء، مع تأكيد التحقق الإحصائي على أهميته.

العمل المستقبلي:

1. استكشاف مصنفات أساسية إضافية (مثل الشبكات العصبية، وXGBoost) لزيادة تنوع المجموعة.
2. دراسة تقنيات متقدمة لأخذ عينات زائدة ومعالجة عدم التوازن للفئات الأقلية الصغيرة جدًا.
3. توسيع نطاق التقييم ليشمل مشاريع البرمجيات الصناعية لتقييم إمكانية التعميم خارج نطاق مجموعات بيانات ناسا.
4. تطوير مسار آلي لاختيار الميزات وتحسين أوزان المجموعة لتقليل التدخل البشري وتحسين قابلية التكرار.

في الختام، يُظهر الإطار المقترح أن اختيار الميزات الهجين مع التعلم الجماعي الموزون هو نهج فعال وموثوق للتنبؤ المبكر بعيوب البرمجيات، مما يوفر فوائد عملية لضمان جودة البرمجيات وإدارة المخاطر.

Compliance with ethical standards

Disclosure of conflict of interest

The authors declare that they have no conflict of interest.

References

[1] Abdu, A., Zhai, Z., Abdo, H. A., Algabri, R., Al-masni, M. A., & Gu, Y. H. (2024). Semantic and traditional feature fusion for software defect prediction using hybrid deep learning model. *Scientific Reports*, 14, Article 14771.

[2] Albattah, W., & Alzahrani, M. (2024). Software defect prediction based on machine learning and deep learning techniques: An empirical approach. *AI Journal*, 5(4), 1743–1758.

 Copyright: © 2026 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

[3] Ali, M., Mazhar, T., & Al-Rasheed, A. (2024). Enhancing software defect prediction: A framework with improved feature selection and ensemble machine learning. *PeerJ Computer Science*, 10, e1860.
[4] Ali, M., Mazhar, T., Arif, Y., & Alotaibi, S. (2024). Software defect prediction using an intelligent ensemble-based model. *IEEE Access*.
[5] Anand, K. (2025). Software defect prediction using wrapper-based dynamic feature selection and machine learning. *Journal of Software Engineering*.
[6] A Sunil, R. K., & Karsoliya, S. (2025). Empirical analysis of software defect prediction using classification techniques. *International Journal of Software Research*.
[7] Daza, A. (2025). Software defect prediction based on a multiclassifier approach. *Software Defect Prediction Journal*.
[8] Destefanis, G., Yousefi, L., Shepperd, M., Tucker, A., Swift, S., Counsell, S., & Arzoky, M. (2026). An audit of machine learning experiments on software defect prediction. *arXiv*.
[9] Haq, Y., et al. (2025). Software defect prediction based on residual/shuffle models. *Scientific Reports*.
[10] Haque, Q. M. U., et al. (2024). Identification of software bugs by analyzing natural language-based requirements using optimized deep learning features. *Computers, Materials & Continua*.
[11] Haque, R., Ali, A., McClean, S., Cleland, I., & Noppen, J. (2024). Heterogeneous cross-project defect prediction using encoder networks and transfer learning. *IEEE Access*, 12, 409–419.
[12] Hasan, A. T., & Mohi-Aldeen, S. M. (2025). Software defect prediction based on deep learning algorithms: A systematic literature review. *AL-Rafidain Journal of Computer Sciences and Mathematics*, 19(1), 67–79.
[13] Hesamolhokama, M., Shafiee, A., Ahmaditeshnizi, M. R., Fazli, M., & Habibi, J. (2024). SDPERL: A framework for software defect prediction using ensemble feature extraction and reinforcement learning. *arXiv*.
[14] Li, C. (2025). A systematic review of learning-based software defect prediction. *Journal of Software Engineering*.
[15] Lv, P. (2025). Research on real-time software defect prediction system. *SAGE Journals*.
[16] Mehmood, I., Shahid, S., Hussain, H., Khan, I., Ahmad, S., Rahman, S., Ullah, N., & Huda, S. (2023). A novel approach to improve software defect prediction accuracy using machine learning. *IEEE Access*, 11, 63579–63597.
[17] Mehta, A., Batra, I., & Fergina, A. (2025). Boosting software fault prediction accuracy with ensemble learning. *Engineering Proceedings*, 107(1), 63.
[18] Mohammad, F., & Ryu, D. (2025). Multimodal learning for just-in-time software defect prediction in autonomous driving systems. *arXiv*.
[19] Nevendra, M., & Singh, P. (2022). Empirical investigation of hyperparameter optimization for software defect count prediction. *Expert Systems with Applications*, 191, 116217.
[20] Siddika, A., Begum, M., Al Farid, F., Uddin, J., & Abdul Karim, H. (2025). Enhancing software defect prediction using ensemble techniques and diverse machine learning paradigms. *Engineering Journal*, 6(7), 161.
[21] Sunil, A., Sahu, R. K., & Karsoliya, S. (2025). Software defect prediction using supervised machine learning: A systematic literature review. *International Journal of Advanced Research and Multidisciplinary Trends*, 2(3), 80–95.
[22] Taskeen, A., Khan, S. U. R., & Felix, E. A. (2023). A research landscape on software defect prediction. *Journal of Software: Evolution and Process*, 35(12), e2549.
[23] Yu, H. (2025). Research on software defect prediction based on machine learning methods. *SciOpen Journal*.

 Copyright: © 2026 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

[24] Zaidi, H. Z. (2025). Machine learning approaches for software defect prediction. *AI Perspectives*.
[25] Zhang, S., Jiang, S., & Yan, Y. (2023). A hierarchical feature ensemble deep learning approach for software defect prediction. *International Journal of Software Engineering and Knowledge Engineering*, 33(4), 543–573.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of **SJPHRT** and/or the editor(s). **SJPHRT** and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

 Copyright: © 2026 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).