**Apéndice B.** Descripción de Algoritmos propuestos y no propuestos

| Estudio | Algoritmo propuesto | Descripción | Algoritmo no propuesto | Descripción |
| --- | --- | --- | --- | --- |
| [R01] | 2M-GWO (SVM, RF, GB, AB, KNN) | Two-Phase Modified Grey Wolf Optimizer combinado con clasificadores **SVM**: Support Vector Machine; **RF:** Random Forest; **GB**: Gradient Boosting; **AB**: AdaBoost; **KNN:** K-Nearest Neighbors, para optimización y clasificación | HHO, SSO, WO, JO, SCO | **HHO**: Harris Hawks Optimization, metaheurística inspirada en el comportamiento cooperativo de los halcones para resolver problemas de optimización; **SSO**: Social Spider Optimization, algoritmo de optimización basado en la comunicación y cooperación de las arañas sociales; **WO**: Whale Optimization, algoritmo bioinspirado en la estrategia de caza de las ballenas jorobadas; **JO**: Jellyfish Optimization, técnica de optimización basada en los patrones de movimiento de las medusas; **SCO**: Sand Cat Optimization, algoritmo inspirado en la estrategia de caza de los gatos del desierto para la búsqueda de soluciones óptimas. |
| [R02] | ANN, SVM | **ANN**: Artificial Neural Network, red neuronal básica utilizada para clasificación o regresión; **SVM**: Support Vector Machine, clasificador supervisado robusto para problemas de clasificación binaria | n/a | n/a |
| [R03] | LineFlowDP (Doc2Vec+R-GCN+GNNExplainer) | Enfoque de predicción de defectos basado en representación semántica de código y grafos neuronales | CNN, DBN, BoW, Bi-LSTM, CodeT5, DeepBugs, IVDetect, LineVD, DeepLineDP, N-gram | **CNN**: Convolutional Neural Network, red neuronal profunda utilizada para extracción automática de características en datos estructurados o no estructurados; **DBN**: Deep Belief Network, red neuronal basada en capas de autoencoders para aprender representaciones jerárquicas de datos; **BoW**: Bag of Words, modelo de representación de texto o código basado en la frecuencia de aparición de palabras sin considerar el orden; **Bi-LSTM**: Bidirectional Long Short-Term Memory, red neuronal recurrente bidireccional utilizada para capturar información contextual en secuencias; **CodeT5**: Transformer Model, modelo basado en transformers pre entrenado para tareas de análisis y generación de código fuente; **DeepBugs**: DeepBugs Defect Detection, sistema de aprendizaje profundo diseñado para detectar errores en código fuente.; **IVDetect**: Invariant Violation Detection, técnica que busca detectar violaciones de invariantes lógicos en programas de software; **LineVD**: Line-level Vulnerability Detector, sistema automatizado que identifica vulnerabilidades en líneas específicas de código; **DeepLineDP**: Deep Line-based Defect Prediction, modelo basado en aprendizaje profundo para predecir defectos a nivel de línea de código; **N-gram**: N-gram Language Model, modelo estadístico para procesar secuencias basado en la frecuencia de ocurrencia de subsecuencias adyacentes |
| [R13] | CNN | Convolutional Neural Network, red neuronal utilizada para extracción automática de características | n/a | n/a |
| [R22] | SDP-CMPOA (CMPOA+Bi-LSTM+Deep Maxout) | Software Defect Prediction usando CMPOA optimizado con Bi-LSTM y activación Deep Maxout | CNN, DBN, RNN, SVM, RF, GH+LSTM, FA, POA, PRO, AOA, COOT, BES | **RNN**: Recurrent Neural Network, red neuronal diseñada para procesar datos secuenciales mediante conexiones recurrentes; **SVM**: Support Vector Machine, clasificador supervisado robusto para problemas de clasificación binaria y multiclase; **RF**: Random Forest, ensamble de árboles de decisión utilizado para clasificación y regresión, robusto a sobreajuste; **GH+LSTM**: Genetic Hybrid + Long Short-Term Memory, combinación de optimización genética con red neuronal LSTM para mejorar el aprendizaje; **FA**: Firefly Algorithm, algoritmo de optimización inspirado en el comportamiento luminoso de las luciérnagas para resolver problemas complejos; **POA**: Pelican Optimization Algorithm, técnica de optimización basada en el comportamiento colectivo de los pelícanos; **PRO**: Progressive Optimization, enfoque de optimización que ajusta iterativamente los parámetros para mejorar resultados; **AOA**: Arithmetic Optimization Algorithm, metaheurística basada en operaciones aritméticas para explorar y explotar el espacio de búsqueda; **COOT**: Coot Bird Optimization, algoritmo de optimización inspirado en los movimientos de aves acuáticas tipo fúlica; **BES**: Bacterial Foraging Optimization, metaheurística inspirada en la estrategia de búsqueda de alimento de las bacterias |
| [R24] | DT, NB, RF, LSVM | DT: Decision Tree, clasificador basado en árboles de decisión, NB: Naïve Bayes, clasificador probabilístico basado en la teoría de Bayes, RF: Random Forest, conjunto de árboles de decisión para clasificación y regresión, LSVM: Linear Support Vector Machine, versión lineal de SVM | n/a | n/a |
| [R10] | PoPL(Hybrid) | Paired Learner Approach, técnica híbrida para manejar Concept Drift en predicción de defectos | n/a | n/a |
| [R11] | bGWO (ANN, DT, KNN, NB, SVM) | Binary Grey Wolf Optimizer combinado con múltiples clasificadores | ACO | Ant Colony Optimization, técnica metaheurística basada en el comportamiento colectivo de las hormigas para resolver problemas de optimización de rutas o combinatorios |
| [R12] | FMR, FMRT | Fuzzy Min-Max Regression y su variante para predicción | NB, RF, ACN, ACF | **NB**: Naïve Bayes, clasificador probabilístico sencillo basado en la aplicación del teorema de Bayes con independencia entre atributos; **ACN**: Artificial Cognitive Network, modelo de red artificial inspirado en sistemas cognitivos para clasificación o análisis de patrones; **ACF**: Artificial Cooperative Framework, marco cooperativo artificial diseñado para mejorar la precisión en tareas de predicción o clasificación |
| [R15] | LM, BP, BR, BR+NN | LM: Linear Model, modelo de regresión lineal, BP: Backpropagation, algoritmo de entrenamiento para redes neuronales, BR: Bayesian Regularization, técnica para evitar sobreajuste en redes neuronales, BR+NN: Bayesian Regularized Neural Network, red neuronal regularizada bayesiana | SVM, DT, KNN, NN | **DT**: Decision Tree, modelo de clasificación o regresión basado en una estructura de árbol de decisiones; **KNN**: K-Nearest Neighbors, clasificador basado en la similitud entre instancias en el espacio de características; **NN**: Neural Network, red neuronal artificial utilizada para aprendizaje supervisado o no supervisado en distintas tareas |
| [R16] | DEPT-C, DEPT-M1, DEPT-M2, DEPT-D1, DEPT-D2 | Variantes de un enfoque específico DEPT para priorización o predicción en pruebas de software | DE, GS, RS | **DE**: Differential Evolution, algoritmo de optimización evolutiva utilizado para resolver problemas continuos y no lineales; **GS**: Grid Search, método sistemático de búsqueda para optimización de hiperparámetros en modelos de machine learning; **RS**: Random Search, técnica de optimización de hiperparámetros basada en la selección aleatoria de combinaciones |
| [R42] | MLP | Multilayer Perceptron, red neuronal con múltiples capas ocultas | n/a |  |
| [R18] | C4.5 +ADB | Algoritmo de árbol de decisión C4.5 combinado con AdaBoost para mejorar precisión | ERUS, NB, NB+Log, RF, DNC, SMT+NB, RUS+NB, SMTBoost, RUSBoost | **ERUS**: Ensemble Random Under Sampling, método de balanceo de clases basado en submuestreo aleatorio combinado en ensamble; **NB+Log**: Naïve Bayes + Logistic Regression, enfoque híbrido que combina probabilidades de Naïve Bayes con un clasificador logístico; **DNC**: Dynamic Nearest Centroid, clasificador basado en centroides dinámicos para mejorar la precisión; **SMT+NB**: Synthetic Minority Technique + Naïve Bayes, combinación de balanceo de clases con clasificación bayesiana; **RUS+NB**: Random Under Sampling + Naïve Bayes, técnica de reducción de clases mayoritarias combinada con Naïve Bayes; **SMTBoost**: Synthetic Minority Oversampling Technique Boosting, método de balanceo combinado con boosting para mejorar clasificación; **RUSBoost**: Random Under Sampling Boosting, método de ensamble basado en submuestreo y boosting para mejorar predicción. |
| [R28] | KPCA+ELM | Kernel Principal Component Analysis combinado con Extreme Learning Machine | SVM, NB, LR, MLP, PCA+ELM | **LR**: Logistic Regression, modelo estadístico utilizado para clasificación binaria mediante la función sigmoide; **MLP**: Multilayer Perceptron, red neuronal artificial con una o más capas ocultas para clasificación o regresión; **PCA+ELM**: Principal Component Analysis + Extreme Learning Machine, enfoque híbrido que reduce dimensionalidad y aplica ELM para clasificación |
| [R47] | rejoELM, IrejoELM | Variantes mejoradas del Extreme Learning Machine aplicando técnicas propias. | rejoNB, rejoRBF | **rejoNB**: Re-joined Naïve Bayes, variante mejorada de Naïve Bayes para clasificación; **rejoRBF**: Re-joined Radial Basis Function, variante basada en RBF para tareas de clasificación o regresión |
| [R29] | WPA-PSO+DNN, WPA-PSO+self-encoding | Whale + Particle Swarm Optimization combinados con Deep Neural Networks o Autoencoders. | Grid, Random, PSO, WPA | **Grid**: Grid Search, técnica de búsqueda exhaustiva para optimización de hiperparámetros; **Random**: Random Search, estrategia de optimización aleatoria de parámetros; **PSO**: Particle Swarm Optimization, algoritmo de optimización inspirado en el comportamiento de enjambres de partículas; **WPA**: Whale Particle Algorithm, metaheurística que combina estrategias de optimización de ballena y partículas |
| [R30] | ACO | Ant Colony Optimization, técnica inspirada en comportamiento de hormigas para optimización | NB, J48, RF | **J48**: J48 Decision Tree, implementación del algoritmo C4.5 en el software WEKA para clasificación |
| [R41] | DP+GCNN | Defect Prediction usando Graph Convolutional Neural Network | LRC, RFC, DBN, CNN, SEML, MPT, DP-T, CSEM | **LRC**: Logistic Regression Classifier, variante de regresión logística aplicada a tareas de clasificación; **RFC**: Random Forest Classifier, ensamble de árboles de decisión para clasificación robusta; **SEML**: Software Engineering Machine Learning, enfoque que aplica técnicas de machine learning en ingeniería de software; **MPT**: Modified Particle Tree, algoritmo basado en árboles y partículas para optimización; **DP-T**: Defect Prediction - Tree, enfoque basado en árboles para predicción de defectos; **CSEM**: Code Structural Embedding Model, modelo que utiliza embeddings estructurales de código para predicción o clasificación |
| [R44] | RNNBDL | Recurrent Neural Network con Bayesian Deep Learning | LSTM, BiLSTM, CNN, SVM, NB, KNN, KStar, Random Tree | **LSTM**: Long Short-Term Memory, red neuronal recurrente especializada en aprender dependencias a largo plazo en secuencias; **BiLSTM**: Bidirectional Long Short-Term Memory, versión bidireccional de LSTM que captura contexto pasado y futuro en secuencias; **KStar**: KStar Instance-Based Classifier, clasificador basado en vecinos más cercanos con función de distancia basada en transformaciones; **Random Tree**: Random Tree Classifier, clasificador basado en árboles de decisión generados aleatoriamente |
| [R50] | Naïve Bayes (GaussianNB) | Variante de Naïve Bayes usando distribución Gaussiana | n/a | n/a |
| [R51] | Stacking+MLP (J48,RF,SMO,IBK,BN)+BF,GS,GA,PSO,RS,LFS | Ensamble Stacking de múltiples clasificadores y meta-heurísticas | n/a | n/a |
| [R53] | TS-ELA (ELA+IG+SMOTE+INFFC)+(BaG, RaF, AdB, LtB, MtB, RaB, StK, StC, VoT, DaG, DeC, GrD, RoF) | Técnica híbrida que combina múltiples técnicas de balanceo, selección e inducción | DTa, DSt | **DTa**: Decision Tree (Adaptive), variante de árbol de decisión adaptativo para clasificación; **DSt**: Decision Stump, árbol de decisión de una sola división, utilizado en métodos de ensamble. |
| [R55] | CBA2 | Classification Based on Associations versión 2 | C4.5, CART, ADT, RIPPER, DT | **C4.5**: C4.5 Decision Tree, algoritmo clásico de árboles de decisión utilizado en clasificación; **CART**: Classification and Regression Tree, técnica de árboles para tareas de clasificación o regresión; **ADT**: Alternating Decision Tree, algoritmo basado en árboles con nodos de predicción y de decisión alternados; **RIPPER**: Repeated Incremental Pruning to Produce Error Reduction, algoritmo de reglas para clasificación |
| [R57] | HyGRAR (MLP, RBFN, GRANUM) | Híbrido de MLP, redes de base radial y algoritmo GRAR para clasificación | SOM, KMeans-QT, XMeans, EM, GP, MLR, BLR, LR, ANN, SVM, CCN, GMDH, GEP, SCART, FDT-O, FDT-E, DT-Weka, BayesNet, MLP, RBFN, ADTree, DTbl, CODEP-Log, CODEP-Bayes | **SOM**: Self-Organizing Map, red neuronal no supervisada utilizada para clustering y visualización de datos; **KMeans-QT**: K-Means Quality Threshold, variante del algoritmo K-Means con umbrales de calidad para los clústeres; **XMeans**: Extended K-Means, versión extendida de K-Means que optimiza automáticamente el número de clústeres; **EM**: Expectation Maximization, técnica estadística iterativa para estimación de parámetros en modelos de mezcla; **GP**: Genetic Programming, técnica de programación evolutiva para resolver problemas de optimización o aprendizaje; **MLR**: Multiple Linear Regression, modelo estadístico para predecir una variable continua usando múltiples predictores; **BLR**: Bayesian Linear Regression, regresión lineal bajo un enfoque bayesiano para incorporar incertidumbre; **ANN**: Artificial Neural Network, red neuronal artificial utilizada en tareas de clasificación, regresión o predicción; **CCN**: Convolutional Capsule Network, red de cápsulas convolucionales para reconocimiento de patrones; **GMDH**: Group Method of Data Handling, técnica basada en redes de polinomios para modelado predictivo; **GEP**: Gene Expression Programming, técnica evolutiva basada en programación genética para modelado simbólico; **SCART**: Soft Classification and Regression Tree, variante de árbol de decisión que permite clasificación difusa o suave; **FDT-O**: Fuzzy Decision Tree - Option, variante de árbol de decisión con incorporación de lógica difusa; **FDT-E**: Fuzzy Decision Tree - Enhanced, versión mejorada de árboles de decisión difusos; **DT-Weka**: Decision Tree Weka, implementación de árboles de decisión dentro de la plataforma WEKA; **BayesNet**: Bayesian Network, clasificador probabilístico basado en redes bayesianas; **RBFN**: Radial Basis Function Network, red neuronal basada en funciones de base radial para clasificación o regresión; **ADTree**: Alternating Decision Tree, técnica basada en árboles alternados de decisión y predicción; **DTbl**: Decision Table, clasificador simple basado en tablas de decisión; **CODEP-Log**: Code Execution Prediction - Logistic Regression, enfoque de predicción de defectos utilizando regresión logística; **CODEP-Bayes**: Code Execution Prediction - Naïve Bayes, enfoque de predicción basado en Naïve Bayes |
| [R65] | ME-SFP+[DT], ME-SFP+[MLP] | Multiple Ensemble with Selective Feature Pruning con clasificadores base. | Bagging+DT, Bagging+MLP, Boosting+DT, Boosting+MLP, Stacking+DT, Stacking+MLP, Indi+DT, Indi+MLP, Classic+ME | **Bagging+DT**: Bootstrap Aggregating + Decision Tree, método de ensamble que utiliza árboles de decisión para mejorar precisión; **Bagging+MLP**: Bagging + Multilayer Perceptron, método de ensamble que aplica redes MLP; **Boosting+DT**: Boosting + Decision Tree, método de ensamble donde los clasificadores débiles son árboles de decisión; **Boosting+MLP**: Boosting + MLP, combinación de boosting con redes neuronales MLP; **Stacking+DT**: Stacking + Decision Tree, ensamble apilado que usa árboles de decisión; **Stacking+MLP**: Stacking + MLP, ensamble apilado con redes MLP; **Indi+DT**: Individual Decision Tree, enfoque basado en árboles de decisión individuales dentro de un esquema de comparación o ensamble; **Indi+MLP**: Individual MLP, red neuronal MLP utilizada de forma independiente en experimentos o ensambles; **Classic+ME**: Classic Multiple Ensemble, configuración clásica de métodos de ensamble |
| [R66] | AST n-gram+J48, AST n-gram+Logistic, AST n-gram+Naive Bayes | Enfoque basado en extracción de características AST n-gram combinado con diferentes clasificadores | n/a | n/a |
| [R07] | IECGA (RF+SVM+NB+GA) | Improved Evolutionary Cooperative Genetic Algorithm con múltiples clasificadores | RF, SVM, NB | **NB**: Naïve Bayes, clasificador probabilístico sencillo basado en la teoría de Bayes |
| [R09] | VESDP (RF+SVM+NB+ANN) | Variant Ensemble Software Defect Prediction | RF, SVM, NB, ANN | **ANN**: Artificial Neural Network, red neuronal artificial utilizada en tareas de clasificación o regresión |
| [R17] | MLP, BN, Lazy IBK, Rule ZeroR, J48, LR, RF, DStump, SVM | BN: Bayesian Network, clasificador basado en redes bayesianas, **Lazy IBK**: Instance-Based K Nearest Neighbors, **Rule ZeroR**: Clasificador trivial sin variables predictoras, **J48**: Implementación de C4.5 en WEKA, **LR**: Logistic Regression, regresión logística, **DStump**: Decision Stump, árbol de decisión de profundidad 1 | n/a | n/a |
| [R19] | CONVSDP (CNN), DNNSDP (DNN) | Convolutional Neural Network aplicado a defect prediction., Deep Neural Network aplicado a defect prediction | RF, DT, NB, SVM | **RF**: Random Forest, ensamble de árboles de decisión que mejora precisión y control de sobreajuste |
| [R21] | ISDPS (NB+SVM+DT) | Intelligent Software Defect Prediction System combinando clasificadores | NB, SVM, DT, Bagging, Vouting, Stacking | **Bagging**: Bootstrap Aggregating, técnica de ensamble que mejora la estabilidad de los clasificadores; **Vouting**: Voting Ensemble, método de ensamble que combina las predicciones de múltiples clasificadores mediante votación; **Stacking**: Stacked Generalization, técnica de ensamble que combina diversos modelos mediante un meta-clasificador |
| [R33] | 2SSEBA (2SSSA, ELM, Bagging Ensemble) | Two-Stage Salp Swarm Algorithm + ELM con ensamble | ELM , SSA+ELM, 2SSSA+ELM, KPWE, SEBA | **ELM**: Extreme Learning Machine, red neuronal de una sola capa de aprendizaje rápido.  **SSA+ELM**: Salp Swarm Algorithm + ELM, combinación de algoritmo bioinspirado SSA con ELM; **2SSSA+ELM**: Two-Stage Salp Swarm Algorithm + ELM, versión mejorada del enfoque SSA combinado con ELM; **KPWE**: Kernel Principal Wavelet Ensemble, método que combina transformadas wavelet con técnicas kernel para clasificación; **SEBA**: Swarm Enhanced Bagging Algorithm, técnica de ensamble mejorada mediante algoritmos de enjambre |
| [R38] | MODL-SBP (CNN-BiLSTM+CQGOA) | Modelo híbrido que combina CNN, BiLSTM y optimización CQGOA | SVM-RBF, KNN+EM, NB, DT, LDA, AdaBoost, | **SVM-RBF**: Support Vector Machine with Radial Basis Function, SVM utilizando kernel RBF para separación no lineal; **KNN+EM**: K-Nearest Neighbors + Expectation Maximization, combinación de clasificación KNN con algoritmo EM para clustering o imputación; **LDA**: Linear Discriminant Analysis, técnica estadística para reducción de dimensionalidad y clasificación; **AdaBoost**: Adaptive Boosting, técnica de ensamble que combina clasificadores débiles para mejorar precisión |
| [R46] | MVFS (MVFS+NB, MVFS+J48, MVFS+IBK) | Multiple View Feature Selection aplicado a distintos clasificadores | IG, CO, RF, SY | **IG**: Information Gain, medida estadística utilizada para seleccionar atributos en modelos de decisión; **CO**: Cut-off Optimization, técnica que ajusta puntos de corte en modelos de clasificación; **SY**: Symbolic Learning, enfoque basado en aprendizaje simbólico para tareas de clasificación o descubrimiento de patrones |
| [R06] | HFEDL(CNN, BiLSTM+Attention) | Hierarchical Feature Ensemble Deep Learning | n/a | n/a |
| [R40] | KELM+WSO | Kernel Extreme Learning Machine combinado con Weight Swarm Optimization | SNB, FLDA, GA+DT, CGenProg | **SNB**: Selective Naïve Bayes, versión mejorada de Naïve Bayes basada en selección de atributos relevantes; **FLDA**: Fisher Linear Discriminant Analysis, técnica de reducción de dimensionalidad optimizada para separación de clases; **GA+DT**: Genetic Algorithm + Decision Tree, combinación de algoritmo genético con árboles de decisión para optimización de selección o parámetros; **CGenProg**: Code Genetic Programming, aplicación de programación genética para mejora o reparación automática de código |
| [R49] | CCFT+CNN | Combination of Code Feature Transformation + CNN | RF, DBN, CNN, RNN, CBIL, SMO | **CBIL**: Classifier Based Incremental Learning, enfoque incremental de aprendizaje supervisado basado en clasificadores; **SMO**: Sequential Minimal Optimization, algoritmo eficiente para entrenar SVMs |
| [R58] | KTC (IDR+NB, IDR+SVM, IDR+KNN, IDR+J48) | Keyword Token Clustering combinado con distintos clasificadores | NB, KNN, SVM, J48 | Conjunto de clasificadores estándar (Naïve Bayes, K-Nearest Neighbors, Support Vector Machine, J48 Decision Tree) aplicados en diversas tareas de clasificación |
| [R45] | Flakify (CodeBERT) | Modelo basado en CodeBERT para detección de tests inestables | FlakeFlagger | **FlakeFlagger**: Flaky Test Flagging Model, modelo diseñado para identificar tests inestables o flakiness en software testing |
| [R34] | SVM+MLP+RF | SVM: Support Vector Machine + MLP: Multilayer Perceptron + RF: Random Forest, ensamble híbrido que combina SVM, redes neuronales MLP y Random Forest para mejorar precisión. | SVM, ANN, RF | **SVM**: Support Vector Machine, clasificador robusto ampliamente utilizado para problemas de clasificación supervisada; **ANN**: Artificial Neural Network, red neuronal artificial para tareas de clasificación, regresión o predicción; **RF**: Random Forest, técnica de ensamble basada en múltiples árboles de decisión para mejorar precisión y robustez |
| [R56] | FRBS | Fuzzy Rule-Based System, sistema basado en reglas difusas utilizado para clasificación o toma de decisiones | C4.5, RF, NB | **C4.5:** Decision Tree, algoritmo clásico de árboles de decisión usado para clasificación; **NB**: Naïve Bayes, clasificador probabilístico sencillo basado en la aplicación del teorema de Bayes |
| [R04] | XCSF-ER | Extended Classifier System with Function approximation - Enhanced Rule, sistema basado en reglas extendido con capacidades de aproximación y mejora | ANN, RS, XCSF | **RS**: Random Search, técnica de optimización de hiperparámetros basada en selección aleatoria; **XCSF:** Extended Classifier System with Function approximation, sistema de aprendizaje evolutivo basado en reglas |
| [R60] | KNN | K-Nearest Neighbors, clasificador basado en la similitud entre instancias cercanas en el espacio de características | LR, LDA, CART, NB, SVM | **LR**: Logistic Regression, modelo estadístico para clasificación binaria o multiclase; **LDA**: Linear Discriminant Analysis, método para reducción de dimensionalidad y clasificación supervisada; **CART**: Classification and Regression Trees, técnica de árboles usada en clasificación y regresión |
| [R64] | AFSA | Artificial Fish Swarm Algorithm, metaheurística bioinspirada en el comportamiento de cardúmenes de peces para optimización | GA, K-means Clustering, NSGA-II, IA | **GA**: Genetic Algorithm, algoritmo evolutivo basado en selección natural para resolver problemas complejos; **K-means Clustering:** K-means Clustering Algorithm, técnica no supervisada para agrupar datos en clústeres basados en distancia; **NSGA-II:** Non-dominated Sorting Genetic Algorithm II, algoritmo evolutivo multiobjetivo ampliamente utilizado; **IA:** Intelligent Agent, sistema computacional que percibe su entorno y toma decisiones autónomas |
| [R35] | T5 (YOLOv5) | Text-to-Text Transfer Transformer + You Only Look Once v5, combinación de procesamiento de lenguaje con detección de objetos en imágenes | n/a |  |
| [R39] | EfficientDet, DETR, T5, GPT-2 | **EfficientDet**: EfficientDet Object Detector, modelo de aprendizaje profundo optimizado para detección de objetos en imágenes; **DETR**: Detection Transformer, modelo basado en transformers para detección de objetos en visión por computadora; **T5**: Text-to-Text Transfer Transformer, modelo de aprendizaje profundo para traducción, resumen y otras tareas de NLP; **GPT-2**: Generative Pre-trained Transformer 2, modelo de lenguaje autoregresivo basado en transformers | n/a |  |
| [R14] | MFO | Moth Flame Optimization, algoritmo de optimización bioinspirado en el comportamiento de polillas alrededor de llamas | FA, ACO | **FA**: Firefly Algorithm, metaheurística inspirada en el comportamiento lumínico de luciérnagas; **ACO**: Ant Colony Optimization, metaheurística bioinspirada en la búsqueda cooperativa de caminos por hormigas |
| [R48] | IFROWANN av-w₁ | Improved Fuzzy Rough Weighted Artificial Neural Network, red neuronal con ponderación difusa y aproximación | EUSBoost, SMOTE+C4.5, CS+SVM, CS+C4.5 | **EUSBoost:** Evolutionary Undersampling Boosting, técnica de ensamble que balancea clases usando submuestreo evolutivo; **SMOTE+C4.5:** Synthetic Minority Oversampling + C4.5, técnica híbrida para balanceo de clases y clasificación; **CS+SVM:** Cost-Sensitive SVM, versión sensible al costo de los errores del clasificador SVM; **CS+C4.5:** Cost-Sensitive C4.5, versión sensible al costo aplicada a árboles C4.5 |
| [R32] | NN (LSTM+MLP) | Neural Network (LSTM + Multilayer Perceptron), red neuronal híbrida que combina redes LSTM y MLP | Hierarchical Clustering | Hierarchical Clustering Algorithm, técnica no supervisada que agrupa datos en forma jerárquica |
| [R43] | EfficientNet-B1 | EfficientNet-B1, red neuronal convolucional optimizada para clasificación de imágenes con alta eficiencia | CNN, VGG-16, ResNet-50, MobileNet-V3 | **CNN**: Convolutional Neural Network, red neuronal profunda utilizada para extracción automática de características en imágenes, texto o datos estructurados; **VGG-16**: Visual Geometry Group 16-layer CNN, arquitectura de red convolucional profunda con 16 capas diseñada para tareas de clasificación de imágenes; **ResNet-50**: Residual Neural Network 50 layers, red neuronal convolucional con conexiones residuales que facilitan el entrenamiento de redes profundas; **MobileNet-V3**: MobileNet Version 3, arquitectura ligera de red convolucional optimizada para dispositivos móviles y tareas de visión computacional con baja demanda de recursos |
| [R62] | NMT | Neural Machine Translation, sistema basado en redes neuronales para traducción automática de idiomas | n/a |  |
| [R23] | GPT-4o | Generative Pre-trained Transformer 4 Omni, modelo de lenguaje multimodal avanzado para procesamiento de texto, imágenes y audio | GitHub Copilot, GPT-4 Turbo | **GitHub Copilot:** GitHub Copilot, sistema de inteligencia artificial asistido por OpenAI para autocompletado y generación de código en entornos de desarrollo; **GPT-4 Turbo:** Generative Pre-trained Transformer 4 Turbo, modelo de lenguaje avanzado optimizado para generación de texto, asistencia en programación y tareas de NLP. |
| [R36] | ACO+NSA | Ant Colony Optimization + Negative Selection Algorithm, combinación de optimización basada en hormigas con detección negativa inspirada en sistemas inmunológicos | Random Testing, ACO, NSA | **Random Testing:** Random Testing, técnica de prueba de software que genera entradas de manera aleatoria para descubrir errores; **NSA:** Negative Selection Algorithm, algoritmo bioinspirado en el sistema inmunológico utilizado para detección de anomalías o intrusiones |
| [R05] | SFLA | Shuffled Frog-Leaping Algorithm, algoritmo metaheurístico basado en el comportamiento social de ranas para resolver problemas complejos | GA, PSO, ACO, ABC, SA | **GA**: Genetic Algorithm, algoritmo evolutivo basado en principios de selección natural para resolver problemas de optimización compleja; **PSO**: Particle Swarm Optimization, algoritmo de optimización inspirado en el comportamiento de enjambres para encontrar soluciones óptimas; **ABC**: Artificial Bee Colony, algoritmo de optimización bioinspirado en el comportamiento de las abejas para búsqueda de soluciones; **SA**: Simulated Annealing, técnica probabilística de optimización basada en el proceso de recocido físico de materiales |
| [R26] | ERINet | Enhanced Residual Inception Network, arquitectura neuronal mejorada para reconocimiento de patrones complejos | SIFT, SURF, ORB | **SIFT**: Scale-Invariant Feature Transform, algoritmo de visión computacional para detección y descripción de puntos clave en imágenes; **SURF**: Speeded-Up Robust Features, algoritmo rápido y robusto para detección de características locales en imágenes; **ORB**: Oriented FAST and Rotated BRIEF, método eficiente para detección de características visuales y emparejamiento de imágenes |
| [R63] | ER  -Fuzz (Word2Vec+LSTM) | Error-Revealing Fuzzing con Word2Vec y LSTM, enfoque híbrido para generación y análisis de entradas que causan fallos | AFL, AFLFast, DT, LSTM | **AFL**: American Fuzzy Lop, herramienta de fuzz testing utilizada para descubrir vulnerabilidades mediante generación automática de entradas maliciosas; **AFLFast:** American Fuzzy Lop Fast, versión optimizada de AFL que mejora la velocidad y eficiencia en la detección de fallos mediante fuzzing; **DT:** Decision Tree, clasificador basado en estructura jerárquica de decisiones para tareas de clasificación o regresión; **LSTM**: Long Short-Term Memory, red neuronal recurrente diseñada para aprender dependencias a largo plazo en secuencias |
| [R27] | HashC-NC | Hash Coverage - Neuron Coverage, enfoque de cobertura de pruebas basado en activación de neuronas en redes profundas | *\*(Criterios de evaluación)* NC, 2-way, 3-way, INC, SC, KMNC, HashC-KMNC, TKNC | (Criterios de evaluación) NC, 2-way, 3-way, INC, SC, KMNC, HashC-KMNC, TKNC: Conjunto de métricas o técnicas de evaluación de cobertura y diversidad en pruebas de software basadas en activación de neuronas, combinatorias y cobertura estructural |
| [R20] | NSGA-II, MOPSO | **NSGA-II**: Non-dominated Sorting Genetic Algorithm II, algoritmo evolutivo multiobjetivo ampliamente utilizado en optimización; **MOPSO**: Multi-Objective Particle Swarm Optimization, versión multiobjetivo de la optimización por enjambre de partículas | Single-objective GA, PSO | **Single-objective GA:** Single-Objective Genetic Algorithm, algoritmo genético clásico enfocado en optimizar un solo objetivo específico |
| [R37] | CVDF DYNAMIC (Bi-LSTM+GA) | Cross-Validation Dynamic Feature Selection usando Bi-LSTM y Algoritmo Genético para selección adaptativa de características | NeuFuzz , VDiscover , AFLFast | **NeuFuzz:** Neural Fuzzing System, sistema basado en técnicas de aprendizaje profundo para generación de datos de prueba automatizada; **VDiscover:** Vulnerability Discoverer, herramienta de detección de vulnerabilidades automatizada mediante análisis dinámico o estático; **AFLFast**: American Fuzzy Lop Fast, (repetido) sistema optimizado para fuzz testing eficiente |
| [R52] | ARTDL | Adaptive Random Testing Deep Learning, enfoque de pruebas de software que combina técnicas de muestreo adaptativo con modelos de deep learning | RT | **RT**: Random Testing, estrategia básica de generación aleatoria de datos para pruebas de software |
| [R25] | MTUL (Autoencoder) | Multi-Task Unsupervised Learning basado en autoencoder, utilizado para aprendizaje no supervisado y detección de anomalías | n/a |  |
| [R61] | RL | Reinforcement Learning, técnica de aprendizaje automático basada en recompensas para toma de decisiones secuenciales | GA, ACO, RS | **GA**: Genetic Algorithm, **ACO**: Ant Colony Optimization y **RS**: Random Search, metaheurísticas o estrategias de búsqueda combinadas o aplicadas individualmente para optimización o clasificación |
| [R08] | FrMi | Fractional Minkowski Distance, métrica de distancia mejorada para clasificadores basados en distancia | SVM, RF, DT, LR, NB, CNN | Conjunto de clasificadores tradicionales **SVM:** Support Vector Machine, **RF:** Random Forest, **DT:** Decision Tree, **LR:** Logistic Regression, **NB:** Naïve Bayes, **CNN:** Convolutional Neural Network, aplicados a diferentes tareas de predicción o clasificación. |
| [R31] | MLP | Multilayer Perceptron, red neuronal con múltiples capas ocultas ampliamente usada en clasificación. | Random Strategy, Total Strategy, Additional Strategy | Estrategias de selección o priorización de casos de prueba basadas en criterios aleatorios, exhaustivos o incrementales. |
| [R54] | LSTM | Long Short-Term Memory, red neuronal recurrente especializada en el aprendizaje de dependencias temporales a largo plazo | n/a |  |
| [R59] | MiTS | Minimal Test Suite, enfoque para generación automática de un conjunto mínimo de casos de prueba | n/a |  |

**Apéndice C.** Variables utilizadas en estudios de IA para las PS

| Categoría | Variable | Descripción | Estudio |
| --- | --- | --- | --- |
| Estructurales de Código Fuente | LOC | Líneas totales de código fuente | [R11], [R12], [R15], [R22], [R16], [R18], [R28], [R47], [R44], [R51], [R55], [R65], [R07], [R09], [R17], [R46], [R40], [R66], [R34], [R56], [R64], [R42], [R13], [R10], [R19], [R06] |
| Estructurales de Código Fuente | v(g) | Complejidad ciclomática del grafo de control | [R11], [R12], [R15], [R18], [R28], [R29], [R30], [R44], [R51], [R55], [R46], [R40], [R56], [R36], [R05], [R42], [R10], [R06] |
| Estructurales de Código Fuente | eV(g) | Complejidad esencial (EVG) | [R11], [R12], [R15], [R18], [R28], [R29], [R44], [R46], [R40], [R56] |
| Estructurales de Código Fuente | iv(g) | Complejidad del flujo de información (IVG) | [R11], [R15], [R18], [R28], [R29], [R30], [R44], [R40], [R56] |
| Estructurales de Código Fuente | npm | Número de métodos públicos | [R01], [R16], [R28], [R65], [R49], [R34] |
| Estructurales de Código Fuente | NOM | Número total de métodos | [R47], [R46], [R06] |
| Estructurales de Código Fuente | NOPM | Número de métodos públicos | [R47], [R46] |
| Estructurales de Código Fuente | NOPRM | Número de métodos protegidos | [R47], [R46] |
| Estructurales de Código Fuente | NOMI | Número de métodos internos o privados | [R01], [R47], [R46] |
| Estructurales de Código Fuente | Loc\_com | Líneas de código que contienen comentarios | [R01], [R15], [R11], [R28], [R29], [R44], [R50], [R51], [R21], [R46], [R66], [R56] |
| Estructurales de Código Fuente | Loc\_blank | Líneas en blanco en el archivo fuente | [R01], [R11], [R15], [R28], [R29], [R30], [R50], [R51], [R21], [R46], [R34], [R56] |
| Estructurales de Código Fuente | Loc\_executable | Líneas que contienen código ejecutable | [R01], [R28], [R51], [R07], [R34], [R56] |
| Estructurales de Código Fuente | LOCphy | Total de líneas físicas de código fuente | [R29], [R41] |
| Estructurales de Código Fuente | CountLineCodeDecl | Líneas dedicadas a declaraciones | [R01] |
| Estructurales de Código Fuente | CountLineCode | Total de líneas de código sin comentarios | [R01], [R28], [R44], [R46], [R49], [R45] |
| Estructurales de Código Fuente | Locomment | Número de líneas que contienen solo comentarios | [R15], [R22], [R28], [R29], [R44], [R50], [R51], [R09], [R46], [R66], [R34] |
| Estructurales de Código Fuente | Branchcount | Número total de ramas condicionales (if, switch, etc.) | [R15], [R30], [R50], [R51], [R07], [R46], [R34], [R56],. , [R19] |
| Estructurales de Código Fuente | Avg\_CC | Promedio de la complejidad ciclomatica de los métodos | [R28], [R65], [R34] |
| Estructurales de Código Fuente | max\_cc | Complejidad ciclomatica máxima de todos los métodos | [R16], [R28], [R30], [R07], [R34] |
| Estructurales de Código Fuente | NOA | Número total de atributos en una clase | [R47], [R46] |
| Estructurales de Código Fuente | NOPA | Número de atributos públicos | [R47], [R46] |
| Estructurales de Código Fuente | NOPRA | Número de atributos protegidos | [R47], [R46] |
| Estructurales de Código Fuente | NOAI | Número de atributos internos/privados | [R47], [R46] |
| Estructurales de Código Fuente | NLoops | Número de bucles totales (for, while) | [R29] |
| Estructurales de Código Fuente | NLoopsD | Número de bucles anidados | [R29] |
| Estructurales de Código Fuente | max\_cc | Complejidad ciclomatica máxima observada entre métodos | [R50], [R51], [R65], [R17] |
| Estructurales de Código Fuente | CALL\_PAIRS | Número de pares de llamadas entre funciones | [R51], [R09], [R56] |
| Estructurales de Código Fuente | CONDITION\_COUNT | Número de condiciones booleanas (if, while, etc.) | [R51], [R56] |
| Estructurales de Código Fuente | CYCLOMATI C\_DENSITY (vd(G)) | Densidad de la complejidad ciclomatica respecto al tamaño del código | [R51], [R21], [R56] |
| Estructurales de Código Fuente | DECISION\_count | Número de puntos de decisión | [R51], [R56] |
| Estructurales de Código Fuente | DECISION\_density (dd(G)) | Proporción de decisiones respecto al total del código | [R51], [R56] |
| Estructurales de Código Fuente | EDGE\_COUNT | Número de aristas en el grafo de flujo de control | [R51], [R56] |
| Estructurales de Código Fuente | ESSENTIAL\_COMPLEXITY (ev(G)) | Parte no estructurable del flujo de control (estructuración mínima) | [R51], [R40], [R34], [R56] |
| Estructurales de Código Fuente | ESSENTIAL\_DENSITY (ed(G)) | Densidad de la complejidad esencial | [R51], [R56] |
| Estructurales de Código Fuente | PARAMETER\_COUNT | Número de parámetros utilizados en funciones o métodos | [R51], [R21], [R56], [R02] |
| Estructurales de Código Fuente | MODIFIED\_CONDITION\_COUNT | Conteo de condiciones modificadas (e.g., if, while) | [R51], [R56] |
| Estructurales de Código Fuente | MULTIPLE\_CONDITION\_COUNT | Conteo de decisiones compuestas (e.g., if (a && b)) | [R51], [R56] |
| Estructurales de Código Fuente | NODE\_COUNT | Número total de nodos en el grafo de control | [R51], [R56] |
| Estructurales de Código Fuente | NORMALIZED\_CYLOMATIC\_COMP (Normv(G)) | Complejidad ciclomatica dividida entre líneas de código | [R51], [R56] |
| Estructurales de Código Fuente | NUMBER\_OF\_LINES | Número total de líneas en el archivo fuente | [R51], [R56] |
| Estructurales de Código Fuente | PERCENT\_COMMENTS | Porcentaje de líneas que son comentarios | [R51], [R17], [R21], [R56] |
| Métricas Halstead | n1, n2 / N1, N2 | Número de operadores (n1) y operandos únicos (n2) | [R24], [R50], [R56] |
| Métricas Halstead | V | Volumen del programa | [R11], [R24], [R15], [R29], [R50], [R55], [R46], [R66], [R56] |
| Métricas Halstead | L | Longitud esperada del programa | [R11], [R24], [R15], [R44], [R51], [R53], [R55], [R46], [R66], [R56] |
| Métricas Halstead | D | Dificultad del código | [R11], [R24], [R15], [R29], [R46], [R66], [R56] |
| Métricas Halstead | E | Esfuerzo de implementación | [R11], [R24], [R15], [R46], [R66], [R56] |
| Métricas Halstead | N | Longitud total: suma de operadores y operandos | [R15], [R29], [R50], [R46], [R66], [R53], [R57], [R11], [R12], [R18], [R66], [R34] |
| Métricas Halstead | B | Número estimado de errores | [R15], [R46], [R66], [R56] |
| Métricas Halstead | I | Nivel de inteligencia requerido | [R11], [R15], [R29], [R46], [R56], [R56] |
| Métricas Halstead | T | Tiempo estimado para programar el software | [R11], [R15], [R29], [R46], [R56] |
| Métricas Halstead | uniq\_Op | Número de operadores únicos | [R11], [R12], [R15], [R28], [R29], [R51], [R53], [R57], [R46], [R34], [R19] |
| Métricas Halstead | uniq\_Opnd | Número de operandos únicos | [R11], [R12], [R15], [R28], [R29], [R51], [R53], [R57], [R46], [R34], [R19] |
| Métricas Halstead | total\_Op | Total de operadores usados | [R11], [R15], [R28], [R29], [R30], [R51], [R53], [R55], [R21], [R46] |
| Métricas Halstead | total opnd | Total de operandos usados | [R15], [R28], [R29], [R51], [R53], [R55], [R46], [R66] |
| Métricas Halstead | hc | Halstead Complexity (puede ser variante específica) | [R28] |
| Métricas Halstead | hd | Halstead Difficulty | [R28] |
| Métricas Halstead | he | Halstead Effort | [R28], [R30], [R51], [R07], [R34] |
| Métricas Halstead | hee | Halstead Estimated Errors | [R28], [R51], [R53], [R34] |
| Métricas Halstead | hl | Halstead Length | [R28], [R51], [R34] |
| Métricas Halstead | hlen | Longitud Halstead estimada | [R28], [R09] |
| Métricas Halstead | hpt | Halstead Programming Time | [R28], [R51] |
| Métricas Halstead | hv | Halstead Volume | [R28], [R51], [R34] |
| Métrica Halstead (o alternativa) | Lv | Nivel lógico de complejidad del programa | [R29], [R34] |
| Métricas Halstead | HALSTEAD\_CONTENT | Contenido calculado de acuerdo al modelo Halstead | [R51], [R21], [R34] |
| Métricas Halstead | HALSTEAD\_DIFFICULTY | Dificultad estimada de comprensión del código | [R51], [R34] |
| Métricas OO | amc | Complejidad media de métodos (Average Method Complexity) | [R16], [R28], [R65], [R33], [R38], [R34] |
| Métricas OO | ca | Afferent coupling: número de clases que dependen de esta | [R16], [R28], [R65], [R49] |
| Métricas OO | cam | Cohesión entre métodos de clase | [R16], [R28], [R65], [R17] |
| Métricas OO | cbm | Acoplamiento entre métodos de clases | [R16], [R28], [R65], [R49], [R34] |
| Métricas OO | cbo | Coupling Between Object classes | [R16], [R28], [R47], [R57], [R65], [R46], [R49], [R34] |
| Métricas OO | dam | Data Access Metric | [R16], [R28], [R65], [R49], [R34] |
| Métricas OO | dit | Depth of Inheritance Tree | [R16], [R28], [R47], [R65], [R46], [R49], [R34] |
| Métricas OO | ic | Índice de reutilización por herencia (Inheritance Coupling) | [R16], [R28], [R65], [R49], [R34] |
| Métricas OO | lcom | Lack of Cohesion of Methods (cohesión débil entre métodos) | [R16], [R28], [R47], [R65], [R17], [R46], [R49], [R34] |
| Métricas OO | lcom3 | Variante mejorada de LCOM para detectar cohesión | [R16], [R28], [R65], [R34] |
| Métricas OO | mfa | Measure of Functional Abstraction | [R16], [R28], [R65], [R34] |
| Métricas OO | moa | Measure of Aggregation | [R16], [R28], [R65], [R34] |
| Métricas OO | noc | Number of Children: número de clases derivadas | [R16], [R28], [R47], [R17], [R46], [R34] |
| Métricas OO | wmc | Weighted Methods per Class | [R16], [R28], [R47], [R57], [R65], [R46], [R34] |
| Métricas OO | FanIn | Número de funciones o clases que llaman a una función dada | [R47], [R29], [R44], [R46] |
| Métricas OO | FanOut | Número de funciones llamadas por una función dada | [R47], [R29], [R44], [R46] |
| Métricas de Calidad del Software | rfc | Fan-in OO: clases que llaman a esta clase | [R01], [R16], [R28], [R47], [R57], [R46], [R66], [R34] |
| Métricas de Calidad del Software | ce | Fan-out OO: clases que esta clase utiliza | [R01], [R16], [R28], [R65], [R49], [R34] |
| Métricas de Calidad del Software | DESIGN\_COMPLEXITY (iv(G)) | Medida compuesta de la complejidad del diseño | [R51], [R09], [R40], [R34], [R56] |
| Métricas de Calidad del Software | DESIGN\_DENSITY (id(G)) | Densidad de elementos de diseño por unidad de código | [R51], [R56] |
| Métricas de Calidad del Software | GLOBAL\_DATA\_COMPLEXITY (gdv) | Complejidad derivada del uso de datos globales | [R51], [R56] |
| Métricas de Calidad del Software | GLOBAL\_DATA\_DENSITY (gd(G)) | Densidad del acceso a datos globales respecto al total | [R51], [R56] |
| Métricas de Calidad del Software | MAINTENANCE\_SEVERITY | Severidad en el mantenimiento del software | [R51], [R56] |
| Métricas de Calidad del Software | HCM | Medida compuesta de complejidad para mantenimiento | [R46] |
| Métricas de Calidad del Software | WHCM | Weighted HCM | [R46] |
| Métricas de Calidad del Software | LDHCM | Layered Depth of HCM | [R46] |
| Métricas de Calidad del Software | LGDHCM | Generalized Depth of HCM | [R46] |
| Métricas de Calidad del Software | EDHCM | Extended Depth of HCM | [R46] |
| Historial de Cambios | NR | Número de revisiones | [R46] |
| Historial de Cambios | NFIX | Número de correcciones realizadas | [R46] |
| Historial de Cambios | NREF | Número de referencias a errores anteriores | [R46] |
| Historial de Cambios | NAUTH | Número de autores que modificaron el archivo | [R46] |
| Historial de Cambios | LOC\_ADDED | Líneas de código añadidas en una revisión | [R46] |
| Historial de Cambios | maxLOC\_ADDED | Máximo de líneas añadidas en una única revisión | [R46] |
| Historial de Cambios | avgLOC\_ADDED | Promedio de líneas añadidas por revisión | [R46] |
| Historial de Cambios | LOC\_REMOVED | Total de líneas eliminadas | [R46] |
| Historial de Cambios | max LOC\_REMOVED | Máximo de líneas eliminadas en una revisión | [R46] |
| Historial de Cambios | avg LOC\_REMOVED | Promedio de líneas eliminadas por revisión | [R46] |
| Historial de Cambios | AGE | Antigüedad del archivo desde su creación | [R46] |
| Historial de Cambios | WAGE | Antigüedad ponderada por el tamaño de las modificaciones | [R46] |
| Historial de Cambios | CVSEntropy | Entropía del historial de cambios en el repositorio | [R01], [R44] |
| Historial de Cambios | numberOfNontrivialBugsFoundUntil | Número acumulado de bugs significativos hallados | [R01] |
| Historial de Cambios | Entropía mejorada | Variante refinada de la entropía de modificaciones | [R22] |
| Historial de Cambios | fault | Conteo total de fallos registrados | [R16], [R44] |
| Historial de Cambios | Defects | Número total de defectos registrados | [R15], [R46], [R10] |
| Historial de Defectos | Bugs | Conteo de bugs encontrados o relacionados al archivo | [R46] |
| Métrica de Cambios | codeCHU | Medida de cambio en unidades de código (Code Change History Unit) | [R46] |
| Métrica de Cambios | maxCodeCHU | Máximo valor de codeCHU en una revisión | [R46] |
| Métrica de Cambios | avgCodeCHU | Promedio de codeCHU a lo largo del tiempo | [R46] |
| Estadística descriptiva | mea | Valor promedio (media aritmética) | [R22] |
| Estadística descriptiva | median | Valor central de la distribución de datos | [R22] |
| Estadística descriptiva | SD | Desviación estándar: dispersión de los datos | [R22] |
| Estadística descriptiva | Curtosis | Medida de la concentración de valores en la media | [R22] |
| Estadística descriptiva | moments | Momentos estadísticos de una distribución | [R22] |
| Estadística descriptiva | skewness | Asimetría de la distribución | [R22] |
| Comunicación MPI | send\_num | Número de envíos MPI (bloqueantes) | [R24] |
| Comunicación MPI | recv\_num | Número de recepciones MPI | [R24] |
| Comunicación MPI | Isend\_num | Envíos MPI no bloqueantes | [R24] |
| Comunicación MPI | Irecv\_num | Recepciones MPI no bloqueantes | [R24] |
| Comunicación MPI | recv\_precedes\_send | Recepción ocurre antes del envío | [R24] |
| Comunicación MPI | mismatching\_type, size | Tipos o tamaños incompatibles en la comunicación | [R24] |
| Comunicación MPI | any\_source, any\_tag | Uso de comodines en comunicación MPI (MPI\_ANY\_SOURCE, etc.) | [R24] |
| Comunicación MPI | recv\_without\_wait | Recepción sin espera activa (non-blocking) | [R24] |
| Comunicación MPI | send\_without\_wait | Envío sin espera activa | [R24] |
| Comunicación MPI | request\_overwrite | Posible sobrescritura de peticiones MPI | [R24] |
| Comunicación MPI | collective\_order\_issue | Problemas de orden en operaciones colectivas | [R24] |
| Comunicación MPI | collective\_missing | Falta de llamadas colectivas requeridas | [R24] |
| Métricas Sintácticas | LCSAt | Tamaño total del Árbol de Sintaxis Abstracta (AST) | [R29] |
| Métricas Sintácticas | LCSAr | Profundidad del AST | [R29] |
| Métricas Sintácticas | LCSAu | Número de nodos únicos en el AST | [R29] |
| Métricas Sintácticas | LCSAm | Promedio de nodos por rama del AST | [R29] |
| Métricas Sintácticas | N\_AST | Número total de nodos en el árbol de sintaxis abstracta (AST) | [R41] |
| Semántica textual | Línea + flujo de datos/control | Representación lógica del flujo de control/datos | [R03] |
| Semántica textual | Doc2Vec vector (100 dimensiones) | Embedding textual vectorizado del código fuente | [R03] |
| Semántica textual | Token Vector | Representación tokenizada del código | [R24], [R63] |
| Semántica textual | Bag of Words | Representación basada en frecuencia de palabras | [R24] |
| Semántica textual | Padded Vector | Vector normalizado con padding para redes neuronales | [R24] |
| Métricas de Red | degree\_norm, Katz\_norm | Métricas de centralidad en grafos de dependencias | [R03] |
| Métricas de Red | closeness\_norm | Métrica de cercanía normalizada en grafo de dependencias | [R03] |
| Métrica de Concurrencia | reading\_writing\_same\_buffer | Acceso concurrente a un mismo buffer | [R24] |
| Métricas estáticas de código | 60 métricas estáticas (calculadas con OSA), originalmente 22 en algunos datasets. | Variables del código fuente como líneas de código, complejidad ciclomática, y métricas orientadas a objetos, usadas para predecir defectos. | [R42], [R06] |
| Dinámicas de Ejecución | Tiempo de ejecución relativo | relación entre duración del test y la suma total | [R04], [R02] |
| Dinámicas de Ejecución | Historial de ejecución | vector binario con resultados previos: 0 = falló, 1 = pasó | [R04] |
| Dinámicas de Ejecución | Última ejecución | proximidad temporal normalizada | [R04] |
| Elementos de Interfaces | EIem\_Inter (\*) | Elementos de interfaz extraídas | [R60], [R35], [R39] |
| Programas | Programas (Source code, test case sets, injected fault points, and running scripts.) | Contenido de programas | [R64] |
| Modelos gráficos / diagramas de estado | Diagramas de Transición de estado | Sistemas OO: traductor Braille, microondas y cajero automático | [R14] |
| Semántica textual | BoW | Representa el texto por la frecuencia de palabras. | [R48] |
| Semántica textual | TF-IDF | Resalta palabras frecuentes en un texto pero raras en el corpus. | [R48] |
| Trazas y llamadas | Function names | Nombres de las funciones llamadas en la traza | [R32] |
| Trazas y llamadas | Return values | Valores de retorno de las funciones | [R32] |
| Trazas y llamadas | Arguments | Argumentos de entrada usados en cada llamada | [R32] |
| Visuales / imagenes | UI\_imagenes | Capturas pantallas (UI) representadas por imágenes. | [R43] |
| Trazas y llamadas | Nombre de clase | Extraído y separado de clases JUnit en Java | [R62] |
| Trazas y llamadas | Nombre de método | Generado desde test methods (@Test) | [R62] |
| Trazas y llamadas | Cuerpo del método | Código fuente tokenizado | [R62] |
| BDD Scenario / Texto | BDD Scenario (texto Given-When) | CSV generado a partir de user stories | [R23], [R02] |
| Visuales GUI / Procesamiento de interfaces | Imagenes GUI, | Visuales (imagen) + estructurales derivadas (máscaras) | [R26] |
| Semántica textual | Condiciones if + tokens | Fragmentos condicionales y estructuras tokenizadas para clasificación de manejo de errores. | [R63] |
| Representación embebida | Word2Vec embedding | Representación vectorial de código fuente para entrada al clasificador. | [R63] |
| Etiquetado supervisado | Etiqueta error-handling | Variable binaria para entrenar el clasificador (manejo de error / normal) | [R63] |
| Representación embebida | Activaciones neuronales | Salidas internas de las neuronas en diferentes capas del modelo bajo entradas de prueba | [R27] |
| Representación embebida | Combinaciones activas | Conjuntos de neuronas activadas simultáneamente durante la ejecución | [R27] |
| Representación embebida | Hash de combinaciones | Representación hash de combinaciones activas para acelerar la evaluación de cobertura (HashC-NC | [R27 |
| Interacción GUI | Eventos (secuencias de interacción) | Clicks, teclas presionadas, secuencia de acciones | [R20] |
| Set de pruebas | Test Paths | Conjuntos de eventos ejecutados por un test case | [R20] |
| Semántica textual | Cadena de entrada (input sequence) | Secuencia de caracteres (fuzz inputs) procesada por Bi-LSTM | [R37] |
| Fuzzing | Caminos únicos ejecutados | Medida de efectividad estructural del test en cobertura | [R37] |
| Fuzzing (basado en búsqueda) | Fitness de entrada | Evaluación probabilística del valor del input dentro de GA | [R37] |
| Visuales / imagenes | Activaciones de capas conv3\_2 y conv4\_2 | Representaciones vectoriales de imágenes extraídas desde capas de VGGNet para medir diversidad en fuzzing. | [R52] |
| Latent representations (autoencoding) | Salidas del autoencoder, entradas mutadas, distancias latentes | Representaciones del autoencoder mutadas y evaluadas por su efecto en el clustering. | [R25] |
| Estructura de integración / Dependencias OO | Dependencias entre clases, número de stubs generados, tamaño del grafo | Relaciones entre clases y número de stubs necesarios para ejecutar el orden de integración propuesto. | [R61] |
| Métricas de ejecución de mutantes | Número de test cases que matan al mutante, severidad de killability, código mutado, clase de operador | Atributos estadísticos y estructurales de los mutantes usados como features para clasificar su capacidad de revelar fallas reales. | [R08] |
| Multifuente (histórico + código) | 04 features (52 métricas de código, 8 métricas de clones, 42 violaciones de reglas de codificación, 2 métricas de Git) | Atributos del código fuente y de su historial de cambios usados para estimar la propensión a fallas mediante MLP. | [R31] |
| Secuencia temporal (interacción) | Secuencia de estados del jugador (acciones, objetos, puntuación, tiempo, eventos) | Variables temporales de interacción en el juego utilizadas como entrada de una red LSTM para generar eventos de prueba y evaluar la jugabilidad. | [R54] |
| Estructural combinatoria | Tamaño del array, niveles por factor, cobertura ttt, cardinalidades mixtas | Parámetros de diseño combinatorio (valores por factor y fuerza de interacción) usados para construir arrays de prueba óptimos mediante búsqueda tabú. | [R59] |

(\*)EIem\_Inter: Text Box, E-mail, Text Box, Password, Button Back, Button, Login, Link Forgot Password,Button Create an account, Text Box First Name, Text Box, Last Name, Text Box Password, Text Box Date of Birth, Text Box, e-mail address, Text Box Country, Button Sign Up, Button Back, Item Product, Dropdown colour, Dropdown size, Dropdown, quantity, Button Buy Now.

**Apéndice D.** Métricas utilizadas en estudios de IA para las PS

| Disciplina | Descripción | Metrics / Formula | Estudios |
| --- | --- | --- | --- |
| Rendimiento clásico | Proporción de predicciones correctas sobre el total de casos evaluados. |  | [R22], [R24], [R11], [R15], [R44], [R51], [R53], [R55], [R57], [R07], [R09], [R17], [R21], [R38], [R40], [R49], [R34], [R43], [R63], [R37], [R08], [R42], [R02], [R10], [R19], [R06] |
| Rendimiento clásico | Mide la proporción de verdaderos positivos entre todas las predicciones positivas realizadas. |  | [R22], [R24], [R11], [R15], [R16], [R42], [R28], [R29], [R55], [R57], [R65], [R07], [R09], [R21], [R49], [R66], [R60], [R32], [R63], [R08], [R02], [R13], [R10], [R19], [R06] |
| Rendimiento clásico | Evalúa la capacidad del modelo para identificar correctamente todos los casos positivos. |  | [R22], [R24], [R11], [R15], [R42], [R18], [R29], [R50], [R55], [R57], [R65], [R07], [R09], [R21], [R40], [R49], [R66], [R60], [R32], [R63], [R08], [R02], [R10], [R19], [R06] |
| Rendimiento clásico | Balance armónico entre precisión y recall, útil en escenarios con clases desbalanceadas. |  | [R22], [R11], [R15], [R16], [R42], [R28], [R47], [R29], [R41], [R44], [R51], [R53], [R55], [R65], [R07], [R40], [R49], [R66], [R60], [R63], [R08], [R02], [R10], [R19], [R06] |
| Clasificación avanzada | Evalúa la calidad de las predicciones considerando verdaderos y falsos positivos y negativos. |  | [R03], [R22], [R28], [R51], [R53], [R65], [R33], [R66] |
| Clasificación avanzada | Resume la capacidad del modelo para discriminar entre clases positivas y negativas a diferentes umbrales |  | [R01], [R03], [R16], [R42], [R18], [R28], [R29], [R30], [R41], [R44], [R51], [R55], [R57], [R65], [R07], [R38], [R40], [R48], [R08], [R19], [R06] |
| Clasificación avanzada | Promedia la sensibilidad y especificidad, útil cuando las clases están desbalanceadas. |  | [R03] |
| Clasificación avanzada | Geométrica entre sensibilidad y especificidad, mide el equilibrio en clasificación binaria. |  | [R03], [R16], [R18], [R55], [R65], [R33], [R46] |
| Alarmas y riesgo | Mide la proporción de verdaderos negativos detectados entre todos los casos negativos reales. |  | [R22], [R15], [R55], [R57], [R09], [R21], [R40] |

| Disciplina | Descripción | Metrics / Formula | PDS |
| --- | --- | --- | --- |
| Alarmas y riesgo | Proporción de verdaderos negativos entre todas las predicciones negativas. |  | [R22], [R09], [R21] |
| Alarmas y riesgo | Proporción de falsos positivos entre todas las predicciones positivas. |  | [R22] |
| Alarmas y riesgo | Proporción de positivos no detectados entre todos los verdaderos positivos. |  | [R22], [R12], [R57], [R09], [R21], [R33] |
| Alarmas y riesgo | Proporción de negativos incorrectamente clasificados como positivos. |  | [R22], [R12], [R18], [R50], [R57], [R65], [R09], [R21], [R33]. [R18], [R37] |
| Alarmas y riesgo | Capacidad del modelo para identificar correctamente los positivos reales. |  | [R37] |
| Alarmas y riesgo | Capacidad del modelo para identificar correctamente los negativos reales. |  | [R32] |
| Alarmas y riesgo | Proporción de verdaderos positivos correctamente identificados por el modelo. |  | [R18] |
| Alarmas y riesgo | Proporción de negativos incorrectamente identificados como positivos. |  | [R18] |
| Métricas específicas en PS | Mide el esfuerzo requerido (en porcentaje de LOC o archivos) para alcanzar el 20% del recall. |  | [R03] |
| Métricas específicas en PS | Porcentaje de defectos encontrados dentro del 20% de las líneas de código más sospechosas. |  | [R03] |
| Métricas específicas en PS | Número de falsos positivos antes de encontrar el primer verdadero positivo. | IFA = Number of non-defective instances before the first defect found | [R03], [R06] |
| Métricas específicas en PS | Precisión entre los k elementos mejor rankeados por el modelo. |  | [R03] |
| Métricas específicas en PS | Métrica de esfuerzo que combina precisión y recall con ponderación del código inspeccionado. |  | [R44] |
| Métricas específicas en PS | Se utiliza para comparar qué tan efectivamente un modelo detecta fallos tempranamente en relación con un modelo base. |  | [R04] |
| Métricas específicas en PS | Número esperado de casos de prueba generados hasta detectar el primer fallo. |  | [R52] |
| Métricas específicas en PS | Número de filas necesarias para cubrir todas las combinaciones t |  | [R59] |
| Métricas específicas en PS | Tiempo requerido por MiTS para construir el array |  | [R59] |
| Métricas específicas en PS | Mejora respecto a los mejores valores conocidos anteriormente |  | [R59] |
| Costo/Error y probabilísticas | Mide el error cuadrático medio entre probabilidades predichas y resultados reales (menor es mejor). |  | [R16] |
| Costo/Error y probabilísticas | Distancia del modelo a un clasificador ideal con 100% TPR y 0% FPR. |  | [R16] |
| Costo/Error y probabilísticas | Raíz del error cuadrático medio entre valores predichos y reales; útil para modelos regresivos. |  | [R53] |
| Costo/Error y probabilísticas | Tiempo esperado que toma al modelo detectar una instancia positiva (defecto) correctamente. |  | [R53] |
| Costo/Error y probabilísticas | Ratio entre el esfuerzo real necesario para alcanzar cierto recall y el esfuerzo óptimo posible. |  | [R57] |
| Costo/Error y probabilísticas | Proporción de instancias clasificadas incorrectamente respecto al total. |  | [R09], [R21], [R56] |
| Cobertura, ejecución, GUI y aprendizaje profundo | Evalúa la rapidez de cobertura de puntos de prueba. Cuanto más cercano a 1, mejor. |  | [R64] |
| Cobertura, ejecución, GUI y aprendizaje profundo | Evalúa el tiempo total de ejecución hasta lograr cobertura completa. Cuanto más bajo, mejor. |  | [R64] |
| Cobertura, ejecución, GUI y aprendizaje profundo | Evalúa la similitud entre un texto generado (ej. test case) y uno de referencia, mediante coincidencias de n-gramas y penalización por brevedad. |  | [R35], [R39], [R62] |
| Cobertura, ejecución, GUI y aprendizaje profundo | Mide la precisión promedio del modelo en detección de objetos a distintos umbrales de coincidencia (IoU). |  | [R39] |
| Cobertura, ejecución, GUI y aprendizaje profundo | Mide el tiempo total que tarda un algoritmo en generar todos los caminos de prueba. |  | [R14], [R20], [R25], [R27], [R37], [R61] |
| Cobertura, ejecución, GUI y aprendizaje profundo | Indica la proporción de caminos de prueba repetidos o innecesarios generados por el algoritmo. |  | [R14] |
| Cobertura, ejecución, GUI y aprendizaje profundo | Fracción de métodos de paso generados que tienen implementación |  | [R23] |
| Cobertura, ejecución, GUI y aprendizaje profundo | Fracción de métodos de paso generados sin implementación |  | [R23] |
| Cobertura, ejecución, GUI y aprendizaje profundo | Fracción de métodos POM generados con implementación funcional |  | [R23] |
| Cobertura, ejecución, GUI y aprendizaje profundo | Promedio de caminos cubiertos por el algoritmo |  | [R36], [R05] |
| Cobertura, ejecución, GUI y aprendizaje profundo | Promedio de generaciones necesarias para cubrir todos los caminos |  | [R36], [R05] |
| Cobertura, ejecución, GUI y aprendizaje profundo | Porcentaje de ejecuciones que cubren todos los caminos |  | [R36], [R05] |
| Cobertura, ejecución, GUI y aprendizaje profundo | Tiempo promedio de ejecución del algoritmo |  | [R36], [R05] |
| Cobertura, ejecución, GUI y aprendizaje profundo | Es equivalente a una métrica de accuracy, aplicada a una tarea de correspondencia visual. |  | [R26] |
| Cobertura, ejecución, GUI y aprendizaje profundo | Mide cuántas combinaciones neuronales únicas han sido cubiertas |  | [R27] |
| Cobertura, ejecución, GUI y aprendizaje profundo | Mide si una neurona fue activada al menos una vez |  | [R27] |
| Cobertura, ejecución, GUI y aprendizaje profundo | Cobertura de combinaciones de 2 neuronas activadas juntas |  | [R27] |
| Cobertura, ejecución, GUI y aprendizaje profundo | Cobertura de combinaciones de 3 neuronas activadas juntas |  | [R27] |
| Cobertura, ejecución, GUI y aprendizaje profundo | Porcentaje de test paths cubiertos por los test cases generados |  | [R20] |
| Cobertura, ejecución, GUI y aprendizaje profundo | % de eventos únicos cubiertos (equivalente a coverage por GUI widgets) |  | [R20] |
| Cobertura, ejecución, GUI y aprendizaje profundo | Porcentaje del código ejecutado durante las pruebas. |  | [R37] |
| Cobertura, ejecución, GUI y aprendizaje profundo | Medida ponderada de la diversidad de cobertura entre los casos generados. |  | [R37] |
| Cobertura, ejecución, GUI y aprendizaje profundo | Proporción de mutantes detectados por cambio en el resultado del sistema |  | [R25] |
| Cobertura, ejecución, GUI y aprendizaje profundo | Distancia euclidiana en el espacio latente entre input original y mutado |  | [R25] |
| Cobertura, ejecución, GUI y aprendizaje profundo | Número total de stubs necesarios para cada orden |  | [R61] |
| Cobertura, ejecución, GUI y aprendizaje profundo | Reducción del número de stubs en comparación con baseline |  | [R61] |
| Cobertura, ejecución, GUI y aprendizaje profundo | Evalúa la efectividad de priorización de test cases |  | [R31] |
| Cobertura, ejecución, GUI y aprendizaje profundo | Porcentaje de predicciones del LSTM que coinciden con la jugabilidad esperada |  | [R54] |
| Cobertura, ejecución, GUI y aprendizaje profundo | Medida de equilibrio entre las acciones y respuestas del juego |  | [R54] |