Jurnal Ilmiah MEDIA SISFO
Vol. 20, No. 1, April 2026
ISSN: 1978-8126
e-ISSN: 2527-7340

Optimasi Software Effort Estimation Menggunakan Random Forest

Maria Rosario Borroek [1], Jasmir[2], Fachruddin[3], Marrylinteri [4], Yosefina Venus[5]

Sistem Informasi[1,2,3,5], Teknik Informatika[4], Universitas Dinamika Bangsa
Jl. Jendral Sudirman, Thehok, Jambi, Indonesia
diamar_ros@yahoo.com[1], Jasmir@unama.ac.id.ac.id[2], fachruddin.stikom@gmail.com[3],*
Marrylinteri@unama.ac.id[4], ydifera@gmail.com[5]

Submitted : **24/02/2026**; Reviewed :**11/03/2026**; Accepted :**29/04/2026**; Published : **30/04/2026**

Abstract
Software development effort estimation is crucial as it is one of the key factors for successful software development. This research employs Random Forest to estimate software development effort. To achieve better results, the study combines the Random Forest method with Genetic Algorithm. The results show that the China dataset provides more accurate estimation compared to the Desharnais dataset, because the China dataset uses relevant feature selection for estimation.

Keywords: effort; software effort estimation, random forest, china dataset, desharnais dataset.

Abstrak

Estimasi usaha pengembangan perangkat lunak adalah sesuatu yang penting dilakukan karena merupakan salah satu kunci keberhasilan pengembangan perangkat lunak. Penelitian menggunakan Random Forest dalam mengestimasi usaha pengembangan perangkat lunak. Agar hasilnya lebih baik, penelitian mengkombinasikan metoda Random Forest dengan Genetic Algorthim. Hasilnya Dataset Cina memberikan hasil estimasi yang lebih akurat dibandingkan dengan dataset desharnais, dikarenakan pada dataset Cina menggunakan pemilihan fitur yang relevan untuk mengestimasinya.

Kata kunci : effort; estimasi perangkat lunak; random forest; dataset cina, dataset derhanais

1. Pendahuluan

Estimasi usaha pengembangan perangkat lunak (SEE) menjadi sangat penting untuk mencegah atau mengurangi kegagalan proyek, estimasi pada tahap awal siklus hidup perangkat lunak menjadi sangat diperlukan [1], [2], [3]. Semakin awal estimasi dilakukan, semakin baik pengelolaan proyek yang dapat dilakukan [4]. Pentingnya estimasi awal terungkap ketika dibutuhkan untuk mengajukan penawaran pada proyek atau membuat komitmen kontrak antara pelanggan dan pengembang [5], [6]. Hal tersebut penting dilakukan agar efisiensi dalam manajemen pengembangan perangkat lunak [7], [8].

Effort adalah segala sumber daya dikeluarkan dalam mengembangkan perangkat lunak. *Effort* biasanya dievaluasi berdasarkan jumlah jam kerja seseorang atau jumlah uang yang dibutuhkan untuk mengimbangi pekerjaan tersebut [9], [10], [11]. Dalam memprediksi ada banyak cara baik mengenai program perangkat lunak yang membantu menghitung perkiraan atau teknik spesifik yang memandu cara membuat prediksi ini. Pada metoda pengembangan ada beberapa metoda yang digunakan yakni: Model Agile yang fokusnya pada bagaimana menghasilkan perangkat lunak yang cepat, COCOMO yang focus kepada estimasi biaya proyek perangkat lunak berdasarkan ukuran dan kompleksitasnya, SEER-SEM yang fokusnya memperkirakan biaya perangkat lunak dengan mempertimbangkan berbagai faktor, termasuk ukuran proyek dan pengalaman tim, PUTNAM yang menggunakan data historis untuk memprediksi berapa lama proyek akan berlangsung berdasarkan ukuran dan kompleksitas perangkat lunak dan PRICE-S dan SLIM yang membantu dalam memperkirakan biaya perangkat lunak dengan menganalisis atribut dan karakteristik proyek yang berbeda [5], [6], [12].

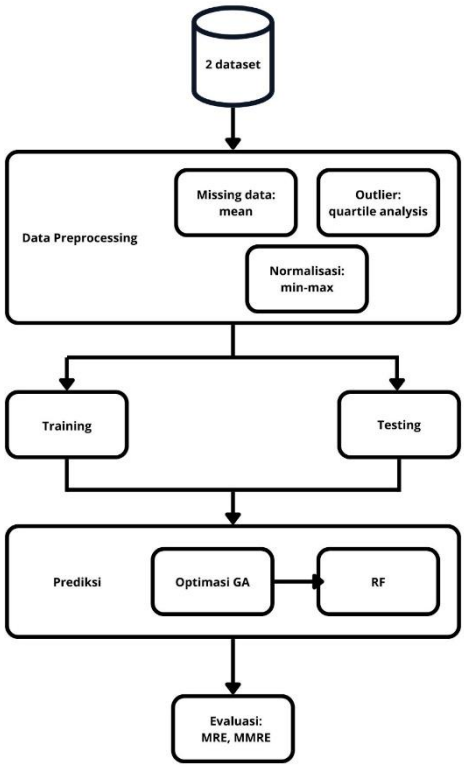
Banyak peneliti telah mencoba membuat model yang dapat memperkirakan upaya ini secara akurat [13]. Zhang dan Nashaat menggunakan Case Based Reasoning (CBR), Decission Tree (DT), Random Forest (RF) dan Support Vector Machine (SVM) [3][14]. Begitu pula Crosa dan Mahes menggunakan RF, SVM, NN dan DT, sedangkan [15], [16]. Sedangkan Carvalho menggunakan Linear Regression (LR) [17].

ISSN: 1978-8126
e-ISSN: 2527-7340

Namun, pengembangan perangkat lunak selalu berubah. Bahasa dan metode pemrograman baru sedang dikembangkan, yang berarti bahwa teknik estimasi lama mungkin tidak berfungsi dengan baik lagi.
Metoda ML metoda yang digunakan untuk untuk memprediksi usaha SEE diantaranya adalah CBR, Ensemble model, FAHP (Fuzzy analytic hierarchy process), Regression, RBFNN (feed forward neural network), fuzzy logic , ANN, Bayesian, DT, Least Square Support Vector Machine (LSSVM), RF, SVM, LR [4], [9], [18]

Agar hasil Estimasi menjadi lebih akurat beberapa penelitian melakukan optimasi parameter [19]. Optimasi dilakukan untuk membantu memilih fitur yang lebih berpengaruh dan meningkatkan akurasi estimasi. Beberapa metoda yang digunakan antara lain: Genetic Algorithm (GA), Particle Swam Optimization (PSO), Genetic programming (GP), Artificial Bee Colony (ABC)

2. Metodologi



Gambar1. *Model Yang Diusulkan*

Penelitian ini bertujuan untuk mengestimasi SEE dengan menggunakan RF yang dioptimasi dengan dengan GA. Adapun skema penelitiannya adalah sebagai berikut:

1. *Dataset*
   Pada penelitian dataset yang digunakan ada 2 yakni Cina dan Desharnais dataset dalam SEE
2. *Preprocessing Data*
   Proses *preprocessing data* pada 2 dataset SEE sangat penting untuk memastikan kualitas data sebelum diterapkan pada algoritma prediksi pada ML. Langkah-langkah yang dilakukan mencakup penanganan nilai hilang, konversi variabel kategorikal ke bentuk numerik, serta normalisasi fitur**.**
   Dalam menangani *missing value*, digunakan metode rata-rata (*mean imputation***)** untuk menggantikan nilai yang hilang agar tetap mencerminkan distribusi data asli. Untuk normalisasi, diterapkan teknik *Min-Max Scaling*, yang mengubah nilai fitur ke dalam rentang tertentu guna memastikan model tidak bias terhadap skala variabel. Sementara itu, deteksi dan penanganan *outlier* dilakukan menggunakan *quartile analysis (IQR method)* agar data tetap representatif tanpa dipengaruhi nilai ekstrem.

ISSN: 1978-8126
e-ISSN: 2527-7340

3. Pembagian Data *(Data Splitting)*
   Dalam SEE, pembagian data menjadi *training set* dan ***testing set*** merupakan langkah penting untuk mengevaluasi kinerja algoritma ML secara efektif. *Training set* digunakan untuk melatih model agar mampu mengenali pola dalam data, sedangkan *testing set* berfungsi untuk mengukur akurasi model terhadap data yang belum pernah dilihat sebelumnya. Pembagian ini membantu mencegah *overfitting*, yaitu kondisi di mana model terlalu menyesuaikan diri dengan data latih sehingga kurang mampu melakukan generalisasi pada data baru.
   Rasio 70:30 menyediakan lebih banyak data untuk pengujian, sehingga lebih cocok untuk analisis performa model yang lebih mendalam. Pemilihan rasio ini bergantung pada ukuran dataset serta kebutuhan analisis yang dilakukan.
4. *Estimasi Perangkat Lunak*
   Selanjutnya data yang sudah terstandarisai dilakukan estimasi dengan menggunakan RF yang selanjutnya akan dilakukan optimasi.
5. *Evaluasi Kinerja Model*
   Evaluasi dilakukan dengan menggunakan metrik MRE dan MMRE untuk menguji model yang diusulkan.

3. Hasil dan Pembahasan

3.1 Analisa Dataset

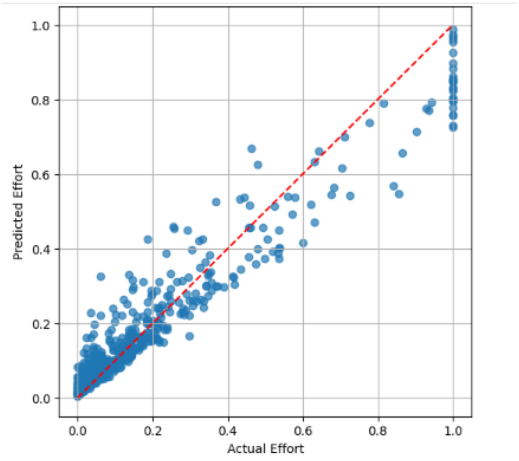
Penelitian ini menggunakan 2 dataset yakni Cina dan Desharnais. Untuk dataset Cina terdiri 19 fitur yang akhirnya hanya 11 fitur yang berperan dalam SEE, sedangkan untuk dataset Desharnais menggunakan semua fitur (11 fitur), dengan tipe untuk semua fitur numerik, kecuali bahasa pemograman yang digunakan (dataset Desharnais). Adapun fiturnya adalah sebagai berikut:

Tabel 1. *Fitur Pada Dataset Cina dan Desharnais*

NO	Fitur Dataset Cina	Fitur Dataset Desharnais
1	Input	TeamExp
2	Output	ManagerExp
3	Enquiry	YearEnd
4	File	Length
5	Interface	Effort
6	Add	Transaction
7	Change	Entities
8	Deleted	PointAdjust
9	Resource	Envergure
10	Duration	PontsNonAjust
11	Effort	Language

3.2 Evaluasi Model Yang Diusulkan

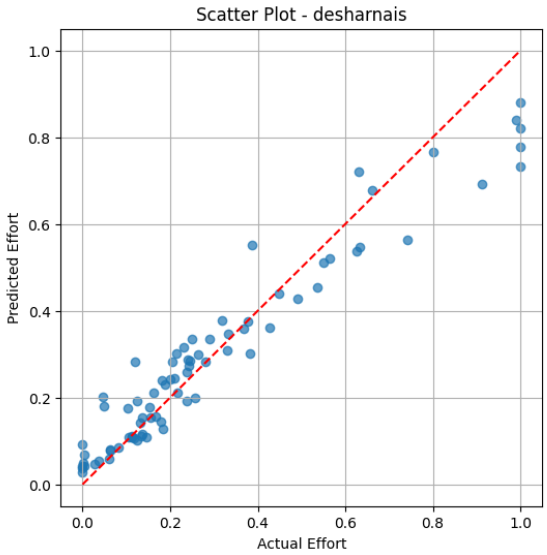
Dalam menganalisis aktual dan hasil prediksi, penulis menggunakan *scatter plot*.



Gambar 1. *Scatter Plot Dataset Cina*

ISSN: 1978-8126
e-ISSN: 2527-7340

Berdasarkan *scatter plot* yang ditampilkan, dapat dilihat hubungan antara usaha pengembangan aktual dengan usaha pengembangan yang merupakan hasil dari prediksi pada dataset Desharnais. Gambar 2 menunjukkan adanya hubungan positif antara kedua variabel tersebut, dimana penyebaran titik-titik biru cenderung mengikuti garis diagonal merah putus-putus yang merupakan garis prediksi ideal. Hal yang menarik untuk diperhatikan adalah sebagian besar data terkonsentrasi pada rentang SEE rendah, yang menunjukkan bahwa mayoritas proyek dalam dataset ini termasuk proyek berskala kecil hingga menengah. Pada rentang nilai tersebut, titik-titik data tersebar cukup rapat di sekitar garis referensi, sehingga dapat disimpulkan bahwa model prediksi memiliki tingkat akurasi yang cukup baik untuk proyek-proyek berukuran kecil. Namun demikian, seiring dengan meningkatnya nilai actual effort, pola penyebaran data mengalami perubahan yang cukup signifikan dimana pada rentang SEE tinggi, jumlah titik data menjadi lebih sedikit dan tingkat variasinya semakin besar, dengan beberapa titik menyebar cukup jauh dari garis ideal baik di atas maupun di bawah garis tersebut. Kondisi ini mengindikasikan bahwa model mengalami kesulitan yang lebih besar dalam melakukan prediksi terhadap proyek-proyek berskala besar atau kompleks, dimana ketidakseimbangan jumlah data antara proyek kecil dan besar menyebabkan model kurang terlatih dengan baik untuk kasus-kasus proyek berskala besar. Adanya beberapa outlier, khususnya titik-titik yang berada jauh di atas garis referensi pada area SEE tinggi, menunjukkan terjadinya prediksi yang terlalu tinggi yang perlu dilakukan analisis lebih lanjut guna meningkatkan kualitas dan keandalan model dalam mengestimasi proyek-proyek dengan skala yang lebih besar



Gambar 2. *Scatter Plot Dataset Desharnais*

Pada dataset Desharnais untuk estimasi proyek perangkat lunak. Terlihat adanya korelasi positif yang kuat antara kedua variabel, di mana sebaran titik-titik biru cenderung mengikuti garis diagonal merah putus-putus yang merepresentasikan prediksi yang cukup akurat. Hal ini mengindikasikan bahwa model prediksi yang digunakan memiliki kemampuan yang cukup baik dalam mengestimasi usaha SEE. Sebagian besar titik data berada relatif dekat dengan garis referensi, menunjukkan bahwa prediksi model tidak terlalu jauh menyimpang dari nilai aktual.

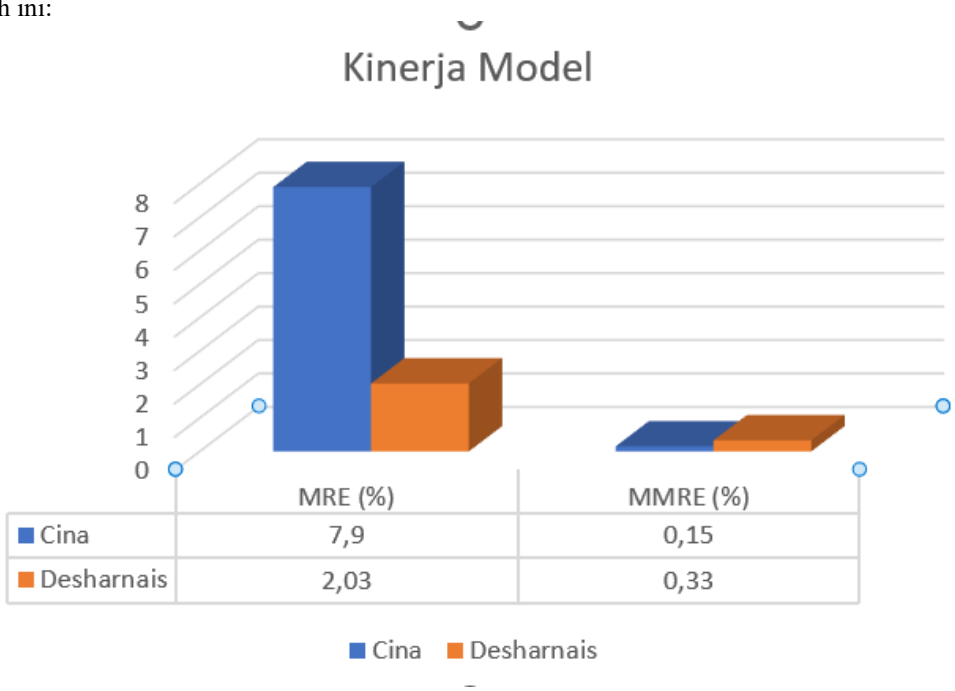
Namun demikian, terdapat beberapa hal yang perlu diperhatikan dari pola sebaran data. Pada range usaha SEE rendah hingga menengah, prediksi cenderung lebih akurat dengan titik-titik yang lebih rapat ke garis ideal. Sebaliknya, pada range usaha SEE tinggi, terlihat variabilitas yang lebih besar dengan beberapa titik yang menyebar cukup jauh dari garis referensi. Gejala ini mengindikasikan bahwa model mengalami kesulitan dalam memprediksi proyek-proyek besar atau kompleks dengan akurasi yang sama seperti proyek kecil. Adanya beberapa titik yang berada jauh di atas atau di bawah garis diagonal juga menunjukkan keberadaan outlier atau kasus-kasus khusus yang mungkin memerlukan investigasi lebih lanjut untuk meningkatkan performa model secara keseluruhan.

Kedua model juga menunjukkan bahwa hubungan variasi antar data sangat baik dengan nilai R^2 sebesar 91%.

ISSN: 1978-8126
e-ISSN: 2527-7340

3.3 Kinerja Model

Penelitian ini menggunakan 2 dataset yakni Cina Desharnais, dimana hasilnya dapat dilihat pada gambar 4 dibawah ini:



Gambar 3. *Kinerja Model (MRE dan MMRE)*

Pada gambar 4, dapat diketahui bahwa Hasil pengujian menunjukkan bahwa model RF-GA mencapai kinerja yang lebih baik pada dataset China (MMRE 7,90%) dibandingkan dengan dataset Desharnais (MMRE 22,60%). Perbedaan kinerja ini dapat dijelaskan melalui proses pemilihan fitur yang diterapkan pada dataset China. Pada dataset China, fitur-fitur yang tidak relevan seperti PDR-AFT dan fitur lainnya telah dieliminasi melalui proses pemilihan fitur, sehingga model hanya menggunakan fitur-fitur yang berkontribusi signifikan terhadap prediksi. Sebaliknya, dataset Desharnais menggunakan seluruh fitur tanpa proses seleksi, yang berpotensi mengandung *noise* dan fitur-fitur yang tidak relevan, sehingga menurunkan akurasi model. Hal ini mengonfirmasi pentingnya tahap preprocessing, khususnya pemilihan fitur, dalam meningkatkan kinerja model estimasi usaha perangkat lunak

3.4 Perbandingan Kinerja Dengan Model

Peneliti juga membandingan kinerja model dengan peneliti terdahulu, dimana hasilnya terlihat pada tabel 4.

Tabel 1. *Perbandingan Kinerja Model*

Ref	Dataset	Metoda	MMRE %	MRE %
[2]	NASA, NASA 92 dan NASA 93	Fuzzy Logic	1,01	6,55
[20]	COCOMO	ANN	22	-
[21]	QUES and UIMS.	NN	6	-
[22]	ISBSG	RF	1,29	-
[23]	ISBSG	ANN	-	0,03
[24]	Privat	Extension of Optimizing Correction Factors	-	6
Model yang Diusulkan	Cina	RF-GA	7,90	0,15
Model yang Diusulkan	Desharnais	RF-GA	2,03	0,33

ISSN: 1978-8126
e-ISSN: 2527-7340

Berdasarkan Tabel 4, model yang diusulkan untuk dataset yang memiliki fitur kecil memiliki tingkat eror yang rendah, sehingga bisa disimpulkan model ini handal untuk dataset yang memiliki dimensi rendah. Untuk dataset yang dimensi tinggi ANN ternyata lebih baik.

4. Kesimpulan

Penelitian ini menggunakan RF sebagai metoda prediksi dalam SEE, dimanauntuk mendapatkan hasil yang optimal maka digunakan GA untuk *hyper* parameternya. Untuk menguji model, maka digunakan 2 dataset yang berbeda yakni Cina dan Desharnais, dengan fitur dan tipe data yang berbeda, akan tetapi jumlah fitur kedua dataset ini sama. Hasil yang didapat adalah dataset Cina yang dilakukan seleksi fitur memiliki hasil yang lebih baik dibandingkan dataset Desharnais

Selanjutnya agar model ini lebih teruji, maka diperlukan datset yang berbeda yang memiliki banyak fitur dan data. Perlu juga menguji model dengan karkateristik yang memiliki missing value yang cukup banyak.

Daftar Pustaka

[1] N. Rankovic, D. Rankovic, M. Ivanovic, and L. Lazic, "A New Approach to Software Effort Estimation Using Different Artificial Neural Network Architectures and Taguchi Orthogonal Arrays," *IEEE Access*, vol. 9, pp. 26926–26936, 2021, doi: 10.1109/ACCESS.2021.3057807.

[2] D. Vanathi, K. Anusha, A. Ahilan, and A. Salinda Eveline Suniram, "Software cost and effort estimation using dragonfly whale optimized multilayer perceptron neural network," *Alexandria Engineering Journal*, vol. 103, pp. 30–37, Sep. 2024, doi: 10.1016/j.aej.2024.04.043.

[3] M. Nashaat and J. Miller, "Refining software defect prediction through attentive neural models for code understanding," *Journal of Systems and Software*, vol. 220, p. 112266, Feb. 2025, doi: 10.1016/j.jss.2024.112266.

[4] B. K. Park and R. Y. C. Kim, "Effort Estimation Approach through Extracting Use Cases via Informal Requirement Specifications," *Applied Sciences*, vol. 10, no. 9, p. 3044, Apr. 2020, doi: 10.3390/app10093044.

[5] A. B. Nassif, D. Ho, and L. F. Capretz, "Towards an early software estimation using log-linear regression and a multilayer perceptron model," *Journal of Systems and Software*, vol. 86, no. 1, pp. 144–160, Jan. 2013, doi: 10.1016/j.jss.2012.07.050.

[6] Y. Mahmood, N. Kama, A. Azmi, A. S. Khan, and M. Ali, "Software effort estimation accuracy prediction of machine learning techniques: A systematic performance evaluation," *Softw Pract Exp*, vol. 52, no. 1, pp. 39–65, Jan. 2022, doi: 10.1002/spe.3009.

[7] A. Sharma and N. Chaudhary, "Prediction of Software Effort by Using Non-Linear Power Regression for Heterogeneous Projects Based on Use case Points and Lines of code," *Procedia Computer Science*, vol. 218, pp. 1601–1611, 2023, doi: 10.1016/j.procs.2023.01.138.

[8] A. Jadhav, M. Kaur, and F. Akter, "Evolution of Software Development Effort and Cost Estimation Techniques: Five Decades Study Using Automated Text Mining Approach," *Mathematical Problems in Engineering*, vol. 2022, pp. 1–17, May 2022, doi: 10.1155/2022/5782587.

[9] Z. Sakhrawi, A. Sellami, and N. Bouassida, "Software enhancement effort estimation using correlation-based feature selection and stacking ensemble method," *Cluster Comput*, vol. 25, no. 4, pp. 2779–2792, Aug. 2022, doi: 10.1007/s10586-021-03447-5.

[10] A. Sharma and D. S. Kushwaha, "Estimation of Software Development Effort from Requirements Based Complexity," *Procedia Technology*, vol. 4, pp. 716–722, 2012, doi: 10.1016/j.protcy.2012.05.116.

[11] A. J. J. Tan, C. Y. Chong, and A. Aleti, "REARRANGE: Effort estimation approach for software clustering-based remodularisation," *Information and Software Technology*, vol. 176, p. 107567, Dec. 2024, doi: 10.1016/j.infsof.2024.107567.

[12] S. Denard, A. Ertas, S. Mengel, and S. Ekwaro-Osire, "Development Cycle Modeling: Resource Estimation," *Applied Sciences*, vol. 10, no. 14, p. 5013, Jul. 2020, doi: 10.3390/app10145013.

[13] P. Suresh Kumar, H. S. Behera, A. K. K, J. Nayak, and B. Naik, "Advancement from neural networks to deep learning in software effort estimation: Perspective of two decades," *Computer Science Review*, vol. 38, p. 100288, Nov. 2020, doi: 10.1016/j.cosrev.2020.100288.

[14] Y. C. Zhang and L. Sakhanenko, "The naive Bayes classifier for functional data," *Statistics and Probability Letters*, vol. 152, pp. 137–146, 2019, doi: 10.1016/j.spl.2019.04.017.

[15] J. Crossa *et al.*, "Expanding genomic prediction in plant breeding: harnessing big data, machine learning, and advanced software," *Trends in Plant Science*, p. S1360138524003455, Jan. 2025, doi: 10.1016/j.tplants.2024.12.009.
[16] B. Mahesh, "Machine Learning Algorithms - A Review," *International Journal of Science and Research (IJSR)*, vol. 9, no. 1, pp. 381–386, 2020, doi: 10.21275/art20203995.
[17] H. D. P. De Carvalho, R. Fagundes, and W. Santos, "Extreme Learning Machine Applied to Software Development Effort Estimation," *IEEE Access*, vol. 9, pp. 92676–92687, 2021, doi: 10.1109/ACCESS.2021.3091313.
[18] A. B. Nassif, M. Azzeh, A. Idri, and A. Abran, "Software Development Effort Estimation Using Regression Fuzzy Models," *Computational Intelligence and Neuroscience*, vol. 2019, pp. 1–17, Feb. 2019, doi: 10.1155/2019/8367214.
[19] M. S. Khan, F. Jabeen, S. Ghouzali, Z. Rehman, S. Naz, and W. Abdul, "Metaheuristic Algorithms in Optimizing Deep Neural Network Model for Software Effort Estimation," *IEEE Access*, vol. 9, pp. 60309–60327, 2021, doi: 10.1109/ACCESS.2021.3072380.
[20] C. H. Rashid, I. Shafi, B. H. A. Khattak, M. Safran, S. Alfarhood, and I. Ashraf, "ANN-based software cost estimation with input from COCOMO: CANN model," *Alexandria Engineering Journal*, vol. 113, pp. 681–694, Feb. 2025, doi: 10.1016/j.aej.2024.11.042.
[21] A. Vescan and D. Barac-Antonescu, "Software maintainability prediction based on change metric using neural network models," *Engineering Applications of Artificial Intelligence*, vol. 144, p. 110032, Mar. 2025, doi: 10.1016/j.engappai.2025.110032.
[22] Z. Abdelali, H. Mustapha, and N. Abdelwahed, "Investigating the use of random forest in software effort estimation," *Procedia Computer Science*, vol. 148, pp. 343–352, 2019, doi: 10.1016/j.procs.2019.01.042.
[23] S. S. Ali, J. Ren, K. Zhang, J. Wu, and C. Liu, "Heterogeneous Ensemble Model to Optimize Software Effort Estimation Accuracy," *IEEE Access*, vol. 11, pp. 27759–27792, 2023, doi: 10.1109/ACCESS.2023.3256533.
[24] H. L. T. K. Nhung, V. Van Hai, R. Silhavy, Z. Prokopova, and P. Silhavy, "Parametric Software Effort Estimation Based on Optimizing Correction Factors and Multiple Linear Regression," *IEEE Access*, vol. 10, pp. 2963–2986, 2022, doi: 10.1109/ACCESS.2021.3139183.