Jurnal TEKNOINFO, Vol. 14, No. 2, 2020, 138-142, ISSN: 2615-224X
DOI : 10.33365/jti.v14i2.647

KOMBINASI MEDIAN WEIGHTED INFORMATION GAIN DENGAN K-NEAREST NEIGHBOR PADA DATASET LABEL *MONTHS SOFTWARE EFFORT ESTIMATION*

Indra Kurniawan[1)]

[1]Prodi Rekayasa Perangkat Lunak, Politeknik Balekambang Jepara
[1] Jl. Ponpes Balekambang Rt. 02/07 Jepara Jawa Tengah 59466
Email: [1] indrakurniawan.politbang@gmail.com

Abstract

Software Effort Estimation is one of the oldest and most important problems faced in software project management. The k-Nearnest Neighbor (k-NN) algorithm shows accurate results for small data sets and is able to handle outlier data. However, the k-NN algorithm has the disadvantage that it cannot tolerate irrelevant features. The public software effort dataset with labels has several main types of problems that are most commonly encountered in datasets with monts labels, namely data outliers, missing values and irrelevant features. This study proposes a combination method of selection of Weighted Information Gain (WIG) features using median threshold criteria with the k-NN algorithm to solve irrelevant feature problems. The comparison results show that the smallest RMSE value in the Chinese dataset is 1936,004 using the Median-WIG k-NN method, the Desharnais dataset is 2690,469 using the Median-WIG k-NN method, the Kitchenham dataset is 1382,579 using the Median-WIG k-NN method. The results of the comparison of all tests revealed that the proposed method of Median Weigthed Information Gain with k-NN was proven to be able to improve the accuracy of the estimation as indicated by a significant decrease in RMSE values in the entire dataset. By using the proposed method by eliminating irrelevant features, it can increase the accuracy of the estimation as seen from the significant decrease in RMSE value.

Keywords: *Effort Estimation Software, irrelevant features, Median Weighted Information Gain, k-Nearnest Neighbor*

Abstrak

Software Effort Estimation adalah salah satu masalah tertua dan paling penting yang dihadapi dalam manajemen proyek perangkat lunak. Algoritma k-Nearnest Neighbor (k-NN) menunjukkan hasil yang akurat untuk kumpulan data kecil dan mampu mengatasi data yang outlier. Namun algoritma k-NN memiliki kelemahan yaitu tidak dapat toleran terhadap fitur yang tidak relevan. Dataset public software effort dengan label memiliki beberapa jenis masalah utama yang paling sering ditemui dalam dataset dengan label monts yaitu data outlier, nilai yang hilang dan fitur yang tidak relavan (*irrelevant featur)*. Penelitian ini mengusulkan metode kombinasi seleksi fitur Weighted Information Gain (WIG) dengan menggunakan kriteria (threshold) median dengan algoritma k-NN untuk menyelesaikan masalah fitur yang tidak relevan. Hasil komparasi memperlihatkan bahawa nilai RMSE terkecil pada dataset China sebesar 1936.004 menggunakan metode Median-WIG k-NN, pada dataset Desharnais sebesar 2690.469 menggunakan metode Median-WIG k-NN, pada dataset Kitchenham sebesar 1382.579 menggunakan metode Median-WIG k-NN. Hasil komparasi seluruh pengujian tersebut diketahui bahwa metode usulan Median *Weigthed Information Gain* dengan k-NN terbukti mampu meningkatkan akurasi estimasi yang ditunjukkan dengan penurunan nilai RMSE yang signifikan pada seluruh dataset. Dengan penggunaan metode yang diusulkan dengan menghilangkan fitur yang tidak relevan mampu meningkatkan akurasi estimasi yang dilihat dari penurunan nilai RMSE secara signifikan menjadi lebih kecil.

Kata Kunci: *Software Effort Estimation, irrelevant feature, Median Weighted Information Gain, k-Nearnest Neighbor*

1. Pendahuluan

Salah satu kegiatan manajemen proyek perangkat lunak adalah kegiatan *Software Effort Estimation*. *Software Effort Estimation* adalah salah satu masalah tertua dan paling penting yang dihadapi dalam manajemen proyek perangkat lunak. Bisa merencanakan dengan benar adalah dasar untuk semua kegiatan manajemen proyek. Seseorang tidak dapat mengelola proyek tanpa mengetahui sumber daya apa yang dibutuhkan untuk mencapai tujuan proyek[1].

Menurut Steve McConnell estimasi yang bagus adalah estimasi yang dapat memberikan gambaran yang cukup jelas tentang keadaan proyek sehingga pimpinan proyek dapat mengambil keputusan yang baik tentang bagaimana mengendalikan sebuah proyek agar mencapai target yang ditentukan. Metode untuk *Software Effort Estimation* terbagi menjadi dua kelompok yaitu menjadi metode Non

Machine Learning (non-ML) dan Machine Learning (ML) [2].

Beberapa metode non machine learning yang telah diusulkan dan digunakan untuk melakukan *Software Effort Estimation* diantaranya expert judgement [3], COCOMO [4], function point (FP) [5]. Namun menurut hasil penelitian yang dilakukan oleh Nunes et.al metode-metode non machine learning merupakan metode konvensional yang memiliki tingkat keakuratan yang relatif rendah dalam melakukan estimasi usaha pengembangan perangkat lunak [6].

Sedangkan metode machine learning yang sudah digunakan untuk estimasi usaha pengembangan perangkat lunak diantaranya k-Nearnest Neighbor (k-NN) [7], [8]. Artificial Neural Networks (ANN) atau Neural Network (NN) [7], [9], [10], Support Vector Machines [7], [11], Naive Bayes (NB) [12], Decision Trees (DT) [13], Linear Regression (LR) [14].

Hasil tinjauan literatur review tentang metode machine learning yang dilakukan Wen et all [15] memperlihatkan hasil bahwa pendekatan CBR dengan menggunakan learning k-NN adalah teknik yang paling banyak digunakan untuk estimasi usaha pengembangan perangkat lunak yaitu sebanyak 37%. Hasil perbandingan menunjukkan bahwa pendekatan CBR menggunakan learning k-NN lebih akurat daripada model regresi. Penggunaan algoritma k-NN mempunyai hasil yang lebih akurat dalam estimasi usaha perangkat lunak dibandingkan metode ML lainnya.

Algoritma k-NN adalah metode yang mudah dimengerti dikarenakan menggunakan penalaran yang mirip dengan pemecahan masalah manusia, sehingga pengguna lebih mudah memahami. Algoritma k-NN juga menunjukkan hasil yang lebih akurat untuk kumpulan data kecil dan mampu mengatasi data yang outlier. Namun algoritma k-NN memiliki kelebahan yaitu tidak dapat toleran terhadap fitur yang tidak relevan dan hal tersebut sangat mempengaruhi akurasi k-NN [15]. Oleh sebab itu model integrasi atau kombinasi dapat digunakan untuk meningkatkan hasil akurasi estimasi.

Dataset public software effort dengan label months yang sering digunakan untuk estimasi usaha perangkat lunak diantaranya China, Desharnais, Kitchenham dan Cocomo. Beberapa studi yang sudah teridentifikasi, menunjukkan bahwa terdapat beberapa jenis masalah utama yang paling sering ditemui dalam dataset dengan label monts yaitu data outlier, nilai yang hilang dan kategori fitur [15]. Masalah kategori fitur dalam dataset seperti fitur yang tidak relavan, bobot fitur yang belum optimal, fitur yang terlalu banyak dan fitur yang sama.

Hasil literatur review yang dilakukan Idri at all menunjukkan bahwa *Software Effort Estimation* masih memiliki beberapa tantangan serius seperti pada karakteristik kumpulan data yaitu estimasi yang dilakukan sensitif terhadap fitur yang tidak relevan dan tingkat pengaruh dari setiap fitur pada data estimasi usaha perangkat lunak [16].

Beberapa penelitian telah menunjukkan bahwa tingkat akurasi dalam *Software Effort Estimation* sangat bergantung pada fitur yang digunakan. Selain itu, pemilihan fitur yang digunakan telah menunjukkan pengaruh yang penting terhadap ketepatan estimasi [11]. Beberapa teknik dikembangkan untuk mengatasi masalah pengurangan fitur yang tidak relevan dan redundant fitur (fitur yang sama). Seleksi Fitur (*variable elimination*) membantu dalam memahami data, mengurangi kebutuhan komputasi, mengurangi efek dimensi dan meningkatkan kinerja prediksi [17].

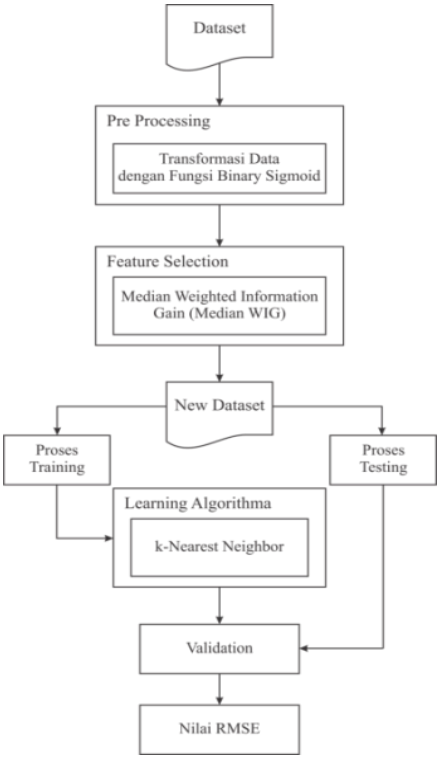
Tujuan utama dari seleksi fitur adalah untuk menyederhanakan dan meningkatkan kualitas dataset dengan memilih fitur-fitur yang relevan. Seleksi fitur dilakukan dengan menghapus fitur yang kurang relevan dari kumpulan data asli tanpa mengorbankan kinerjanya [18].

Hasil penelitian Khoshgoftaar dan Gao diketahui bahwa *Weigthed Information Gain* (WIG) memperlihatkan hasil yang baik dalam bobot fitur untuk pemilihan fitur yang relevan dengan menggunakan kriteria median fitur [19]. Tujuan pembobotan ini untuk merangking fitur yang memenuhi kriteria (threshold) yang ditentukan dipertahankan untuk menjadi kumpulan data baru untuk digunakan oleh algoritma.

Maka dalam penelitian ini peneliti akan mengusulkan metode kombinasi seleksi fitur dengan algoritma k-NN pada dataset label monts *Software Effort Estimation.* Kombinasi yang di usulkan yaitu metode seleksi fitur Weighted Information Gain (WIG) dengan menggunakan kriteria (threshold) median dengan algoritma k-NN untuk menyelesaikan masalah fitur yang tidak relevan sehingga di dapatkan model kombinasi yang lebih akurat untuk *Software Effort Estimation*.

2. Metode

Tahapan penelitian metode yang diusulkan dalam penelitian ini penulis gambarkan pada Gambar 1 sebagai berikut:



Gambar 1. Tahapan Penelitian

Berdasarkan alur penelitian pada gambar 1 tahapan-tahapan metode yang diusulkan dalam penelitian ini adalah sebagai berikut:

1. Lakukan pengolahan data awal dengan transformasi setiap data dengan fungsi Binary

Sigmoid. Fungsi binary sigmoid yaitu mentransformasikan data dalam range 0 sampai 1. Namun, menurut Yu dan Xu hasil yang lebih baik diperoleh dengan menggunakan range kisaran 0.1 sampai 0.9. Karena itu, peneliti mengadopsi range tersebut [20]. Adapun rumus pada persamaan (1) untuk fungsi binary sigmoid (Logsig) yaitu:

$$y' = \frac{x - x\,min}{x\,max - x\,min} \times (0.9 - 0.1) + 0.1 \qquad (1)$$

y' = Hasil transformasi data
x = Nilai asli
$x\,min$ = Nilai minimal
$x\,max$ = Nilai maksimal

2. Lakukan seleksi fitur dengan Median Weighted Information Gain (Median WIG)

Melakukan seleksi fitur yang relevan dan membentuk dataset baru hasil seleksi fitur dengan cara fitur dengan kriteria (threshold) yaitu nilai fitur yang diatas median (fitur relevan) akan dipertahankan untuk digunakan dan fitur dibawah median (fitur tidak relevan) akan dihilangkan. Adapun seleksi fitur dengan cara yaitu: 1) Hitung nilai entropy setiap data yang sudah di transformasi, 2) Hitung nilai gain setiap data, 3) Hitung median kumpulan data dengan median untuk kumpulan data genap yaitu diambil di titik tengah antara angka dengan kedua nilai di jumlah dan di bagi 2 dan median untuk kumpulan data ganjil diambil dari nilai yang berada di titik tengah 4) Bentuk dataset baru hasil transformasi dan seleksi fitur Median WIG dengan mengeliminasi fitur yang kurang dari nilai median.

3. Melakukan training dan testing dengan dataset baru *Software Effort Estimation* hasil transformasi dan seleksi fitur menggunakan algoritma k-Nearest Neighbor dengan k = 1 serta menggunakan metrik jarak Euclidean distance k-NN (k-NN Euc)

4. Validasi metode dengan leave-one-out cross validation (LOOCV)

Selanjutnya tahapan eksperimen dan pengujian metode yang diusulkan dengan metode lain dalam penelitian ini adalah sebagai berikut:

1. Melakukan pengujian metode yang diusulkan dengan menggunakan dataset public *Software Effort Estimation* dengan label months yaitu: China, Desharnais dan Kitchenham.

Tahapan-tahapan pengujian metode yang diusulkan dalam penelitian ini adalah sebagai berikut:

a. Lakukan pengolahan data awal dengan transformasi setiap data pada dataset asli dengan fungsi Binary Sigmoid (Logsig).
b. Lakukan seleksi fitur dengan Median Weighted Information Gain (Median WIG) dan bentuk dataset baru dengan cara menentukan fitur yang relevan dengan kriteria (threshold) yaitu nilai fitur yang diatas median akan dipertahankan untuk digunakan dan fitur dibawah median akan dihilangkan.
c. Melakukan pengujian menggunakan algoritma k-Nearest Neighbor dengan k = 1 serta menggunakan metrik jarak Euclidean distance pada data baru hasil seleksi fitur Median WIG kemudian dari hasil pengujian yang dilakukan dicatat hasil yang didapat.

2. Melakukan pengujian metode individual k-Nearest Neighbor .
3. Melakukan pengujian metode kombinasi seleksi fitur *Forward Selection* (FS) dengan k-Nearest Neighbor.
4. Melakukan pembahasan komparasi hasil estimasi menggunakan nilai RMSE antara metode individual k-Nearest Neighbor, metode kombinasi seleksi fitur *Forward Selection* (FS) dengan k-Nearest Neighbor dan metode usulan seleksi fitur Median Weighted Information Gain (Median WIG) dengan k-Nearest Neighbor.

3. Hasil dan Pembahasan

Pengujian Metode Seleksi Fitur Median WIG k-Nearest Neighbor

Pengujian Metode Median WIG k-Nearest Neighbor pada Dataset China diperolah nilai median sebesar 0.018 Sehingga terdapat 2 fitur yang akan dihilangkan dikarenakan nilai dibawah median (fitur tidak relevan) yaitu: Resource, Dev.Type. Serta terdapat 14 fitur diatas median (fitur relevan) yaitu AFP, Input, Output, Enquiry,

File, Interface, Added, Changed, Deleted, PDR_AFP, PDR_UFP, NPDR_AFP, NPDU_UFP, Duration serta 1 label Effort yang akan dipertahankan untuk di jadikan dataset baru China dalam pengujian menggunakan Algoritma k-Nearest Neighbor di *rapidminer*. Pengujian dataset baru *Software Effort Estimation* dengan algoritma k-Nearest Neighbor menggunakan *rapidminer*. Hasil perhitungan dengan *Rapidminer* metode usulan Median-WIG k-NN dengan parameter k-NN menggunakan k = 1 dan perhitungan jarak Euclidean distance pada Dataset China diperoleh hasil perhitungan nilai RMSE sebesar 1936.004.

Pengujian Metode Median WIG k-Nearest Neighbor pada Dataset Desharnais diperolah nilai median sebesar 0.075 Sehingga terdapat 1 fitur yang akan dihilangkan dikarenakan nilai dibawah median (fitur tidak relevan) yaitu: Language. Serta terdapat 9 fitur diatas median (fitur relevan) yaitu TeamExp, ManagerExp, YearEnd, Length, Transactions, Entities, PointsNonAdjust, Adjustment, PointsAjust serta 1 label Effort yang akan dipertahankan untuk di jadikan dataset baru Desharnais dalam pengujian menggunakan Algoritma k-Nearest Neighbor di *rapidminer*. Pengujian dataset baru *Software Effort Estimation* dengan algoritma k-Nearest Neighbor menggunakan *rapidminer*. Hasil perhitungan dengan *Rapidminer* metode usulan Median-WIG k-NN dengan parameter k-NN menggunakan k = 1 dan perhitungan jarak Euclidean distance pada Dataset Desharnais diperoleh hasil perhitungan nilai RMSE sebesar 2690.469.

Pengujian Metode Median WIG k-Nearest Neighbor pada Dataset Kitchenham diperolah nilai median sebesar 0.156 Sehingga terdapat 1 fitur yang akan dihilangkan dikarenakan nilai dibawah median (fitur tidak relevan). Serta terdapat 2 fitur diatas median (fitur relevan) yang akan dipertahankan untuk di jadikan dataset baru Kitchenham dalam pengujian menggunakan Algoritma k-Nearest Neighbor di *rapidminer*. Pengujian dataset baru *Software Effort Estimation* dengan algoritma k-Nearest Neighbor menggunakan *rapidminer*. Hasil perhitungan dengan *Rapidminer* metode usulan Median-WIG k-NN dengan parameter k-NN menggunakan k = 1 dan perhitungan jarak Euclidean distance pada Dataset Kitchenham diperoleh hasil perhitungan nilai RMSE sebesar 1382.579.

Rekap hasil pengujian Metode Seleksi Fitur Median WIG k-Nearest Neighbor dengan perhitungan nilai RMSE pada dataset *Software Effort Estimation* dengan label months sebagai berikut:

Tabel 1. Hasil Nilai RMSE Metode Usulan Kombinasi Median-WIG k-Nearest Neighbor

Dataset	Median-WIG k-NN
China	1936.004
Desharnais	2690.469
Kitchenham	1382.579

Pengujian Metode Individual k-Nearest Neighbor

Hasil perhitungan dengan *Rapidminer* metode individual k-NN menggunakan parameter k = 1 dan perhitungan jarak Euclidean Distance diperoleh hasil perhitungan nilai RMSE pada dataset *Software Effort Estimation* dengan label months sebagai berikut:

Tabel 2. Hasil Nilai RMSE Metode Individual k-Nearest Neighbor

Dataset	k-NN
China	2854.956
Desharnais	2894.914
Kitchenham	1400.648

Dapat diketahui berdasarkan tabel diatas hasil perhitungan nilai RMSE yang diperoleh pada dataset China sebesar 2854.956, dataset Desharnais sebesar 2894.914 dan dataset Kitchenham sebesar 1400.648.

Pengujian Metode Seleksi Fitur Forward Selection (FS) dengan Algoritma k-Nearest Neighbor

Hasil perhitungan dengan *Rapidminer* metode Forward Selection (FS) k-NN dengan parameter k-NN menggunakan k = 1 dan perhitungan jarak Euclidean distance diperoleh hasil perhitungan nilai RMSE pada dataset *Software Effort Estimation* dengan label months sebagai berikut:

Tabel 3. Hasil Nilai RMSE Metode Kombinasi FS k-NN

Dataset	FS k-NN
China	3240.611
Desharnais	2839.210
Kitchenham	1391.497

Dapat diketahui bahwa hasil perhitungan metode FS k-NN pada tabel diatas menunjukkan nilai RMSE yang diperoleh untuk dataset China sebesar 3240.611, dataset Desharnais sebesar 2839.210 dan dataset Kitchenham sebesar 1391.497.

Pembahasan Komparasi Hasil Nilai RMSE

Untuk melihat kinerja metode terbaik dilakukan perbandingan nilai RMSE dari seluruh metode dengan parameter k-NN menggunakan k = 1 dan perhitungan jarak Euclidean Distance. Berdasarkan hasil pengujian metode individual k-Nearest Neighbor, seleksi fitur Forward Selection (FS) dengan k-Nearest Neighbor, metode usulan seleksi fitur Median Weighted Information Gain (Median WIG) dengan kNearest Neighbor yang keseluruhan hasil nilai RMSE dapat dilihat pada tabel dibawah ini:

Tabel 4. Komparasi Hasil Pengujian Keseluruhan Metode

Dataset	k-NN	FS k-NN	Median-WIG k-NN
China	2854.956	3240.611	1936.004

Desharnais	2894.914	2839.210	2690.469
Kitchenham	1400.648	1391.497	1382.579

Hasil komparasi pada tabel 4 memperlihatkan bahawa nilai RMSE terkecil pada dataset China sebesar 1936.004 menggunakan metode Median-WIG k-NN, pada dataset Desharnais sebesar 2690.469 menggunakan metode Median-WIG k-NN, pada dataset Kitchenham sebesar 1382.579 menggunakan metode Median-WIG k-NN.
Dari hasil komparasi seluruh pengujian tersebut diketahui bahwa metode usulan Median-WIG dengan k-NN terbukti mampu meningkatkan akurasi estimasi yang ditunjukkan dengan penurunan nilai RMSE yang signifikan pada seluruh dataset. Sehingga terjadi peningkatan kinerja setelah menggunakan metode seleksi fitur. Hal ini membuktikan fitur yang tidak relevan sangat mempengaruhi akurasi dalam melakukan estimasi. Dengan penggunaan metode yang diusulkan dengan menghilangkan fitur yang tidak relevan mampu meningkatkan akurasi estimasi yang dilihat dari penurunan nilai RMSE secara signifikan menjadi lebih kecil

4. Kesimpulan dan Saran

Berdasarkan hasil pengujian yang telah dilakukan maka kesimpulan dalam penelitian ini yaitu:

Fitur yang tidak relevan pada dataset *Software Effort Estimation* dengan label months sangat mempengaruhi akurasi dalam melakukan estimasi. Penggunaan metode seleksi fitur mampu mengatasi masalah fitur yang tidak relevan dan dapat meningkatkan akurasi estimasi dilihat dari penurunan nilai RMSE secara signifikan menjadi lebih kecil.

Adapun saran yang dapat diberikan untuk penelitian selanjutnya, yaitu:

Dapat dilakukan penelitian yang lebih jauh lagi mengenai fungsi transformasi data lain selain fungsi Binary Sigmoid (Logsig) digabungkan dengan seleksi fitur Median-WIG dan dapat dibandingkan hasil yang diperoleh.

Daftar Pustaka

[1] I. Sommerville, *Software Engineering Ninth Edition*, 9th ed. Boston: PEARSON, 2011.

[2] M. Shepperd and S. Macdonell, "Evaluating prediction systems in software project estimation," *Inf. Softw. Technol.*, vol. 54, no. 8, pp. 820–827, 2012, doi: 10.1016/j.infsof.2011.12.008.

[3] S. Grimstad and M. Jørgensen, "Inconsistency of expert judgment-based estimates of software development effort," *J. Syst. Softw.*, vol. 80, pp. 1770–1777, 2007, doi: 10.1016/j.jss.2007.03.001.

[4] X. Huang, D. Ho, J. Ren, and L. F. Capretz, "Improving the COCOMO model using a neuro-fuzzy approach," *Appl. Soft Comput.*, vol. 7, pp. 29–40, 2007, doi: 10.1016/j.asoc.2005.06.007.

[5] G. R. Finnie and G. E. Wittig, "A Comparison of Software Effort Estimation Techniques : Using Function Points with Neural Networks , Case-Based Reasoning and Regression Models," *J. Systens Softw.*, vol. 1212, no. 97, pp. 281–289, 1997.

[6] E. U. Points, N. J. Nunes, and L. Constantine, "i UCP : Software Project Size with," *IEEE Softw.*, pp. 64–73, 2011.

[7] E. K. Adhitya, R. Satria, and H. Subagyo, "Komparasi Metode Machine Learning dan Non Machine Learning untuk Estimasi Usaha Perangkat Lunak," *J. Softw. Eng.*, vol. 1, no. 2, pp. 109–113, 2015.

[8] Q. Liu, J. Xiao, and H. Zhu, "Feature selection for software effort estimation with localized neighborhood mutual information," *Cluster Comput.*, no. 1, 2018, doi: 10.1007/s10586-018-1884-x.

[9] V. S. Dave, "Comparison of Regression model , Feed-forward Neural Network and Radial Basis Neural Network for Software Development Effort Estimation," *ACM SIGSOFT Softw. Eng. Notes*, vol. 36, no. 5, pp. 1–5, 2011, doi: 10.1145/2020976.2020982.

[10] C. López-martín, "Predictive accuracy comparison between neural networks and statistical regression for development effort of software projects," *Appl. Soft Comput. J.*, pp. 1–16, 2014, doi: 10.1016/j.asoc.2014.10.033.

[11] A. L. I. Oliveira, P. L. Braga, R. M. F. Lima, and M. L. Cornélio, "GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation," *Inf. Softw. Technol.*, vol. 52, no. 11, pp. 1155–1166, 2010, doi: 10.1016/j.infsof.2010.05.009.

[12] J. Shivhare and S. K. Rath, "Software Effort Estimation using Machine Learning Techniques," *ISEC*, 2014.

[13] A. Bakır, B. Turhan, and A. Bener, "A comparative study for estimating software development effort intervals," *Softw. Qual J*, vol. 19, pp. 537–552, 2011, doi: 10.1007/s11219-010-9112-9.

[14] R. Malhotra, A. Kur, and Y. Sigh, "Application of Machine Learning Methods for Software Effort Prediction," *ACM SIGSOFT Softw. Eng. Notes*, vol. 35, no. 3, pp. 1–6, 2010, doi: 10.1145/1764810.1764825.

[15] J. Wen, S. Li, Z. Lin, Y. Hu, and C. Huang, "Systematic literature review of machine learning based software development effort estimation models," *Inf. Softw. Technol.*, vol. 54, no. 1, pp.

41–59, 2012, doi: 10.1016/j.infsof.2011.09.002.

[16] A. Idri and A. Abran, "Analogy-based software development effort estimation : A systematic mapping and review," *Inf. Softw. Technol.*, 2014, doi: 10.1016/j.infsof.2014.07.013.

[17] G. Chandrashekar and F. Sahin, "A survey on feature selection methods q," *Comput. Electr. Eng.*, vol. 40, no. 1, pp. 16–28, 2014, doi: 10.1016/j.compeleceng.2013.11.024.

[18] M. Kabir and K. Murase, "A new hybrid ant colony optimization algorithm for feature selection," *Expert Syst. Appl.*, vol. 39, no. 3, pp. 3747–3763, 2012, doi: 10.1016/j.eswa.2011.09.073.

[19] T. M. Khoshgoftaar and K. Gao, "Feature Selection with Imbalanced Data for Software Defect Prediction," in *International Conference on Machine Learning and Applications*, 2009, pp. 235–240, doi: 10.1109/ICMLA.2009.18.

[20] F. Yu and X. Xu, "A short-term load forecasting model of natural gas based on optimized genetic algorithm and improved BP neural network," *Appl. Energy*, vol. 134, pp. 102–113, 2014, doi: 10.1016/j.apenergy.2014.07.104.