

Analisis Tren Penelitian *Hyperparameter Tuning* dalam *Software Engineering* melalui *Systematic Literature Review* dan *Bibliometric Analysis*

Rizqi Fadilatul Fajriyah^{*1}, Yusuf Sulistyo Nugroho²

^{1,2}Magister Teknik Informatika, Fakultas Komunikasi dan Informatika, Universitas Muhammadiyah Surakarta, Indonesia

Email: ¹l208230005@student.ums.ac.id, ²yusuf.nugroho@ums.ac.id

Abstrak

Hyperparameter tuning merupakan salah satu aspek penting dalam meningkatkan performa model *machine learning* di bidang *software engineering*. Meskipun memiliki dampak signifikan, kajian terkait tren dan perkembangan penelitian *hyperparameter tuning* di bidang ini masih terbatas dan belum banyak dieksplorasi secara sistematis. Penelitian ini bertujuan untuk menganalisis tren penelitian *hyperparameter tuning* dalam *software engineering* melalui pendekatan *Systematic Literature Review (SLR)* dan *Bibliometric Analysis*. Metode *bibliometrix* menggunakan *Bibliometrix R-Package*, sebanyak 503 artikel diperoleh dari *database Scopus* dianalisis untuk mengidentifikasi metode *tuning* yang dominan, tantangan yang dihadapi, serta peluang penelitian masa depan. Hasil kajian menunjukkan adanya fluktuasi jumlah publikasi dari tahun 2020 hingga 2025, dengan peningkatan signifikan pada tahun 2024 (132 artikel), namun diiringi penurunan rata-rata sitasi pada tahun 2025 (10 artikel) dapat dikaitkan dengan waktu yang lebih singkat bagi publikasi baru untuk mendapatkan kutipan. Wang Y menjadi penulis terbanyak dengan 11 artikel dan sebagai penulis yang paling berpengaruh dengan skor fraksionalisasi 1,75. Sumber paling relevan dan populer berdasarkan jumlah publikasi kategori Jurnal ilmiah adalah *IEEE Transactions on Software Engineering* (18 artikel), untuk kategori konferensi yaitu *ACM International Conference Proceeding Series* (15 artikel). Tren topik berdasarkan kata kunci yang sering muncul adalah *deep learning* dengan 89 kemunculan. Tren penelitian juga menunjukkan peningkatan signifikan dalam eksplorasi teknik *tuning* otomatis guna mengatasi kompleksitas model dan biaya komputasi yang tinggi. Kajian ini memberikan wawasan mengenai perkembangan terkini dan tantangan, seperti kurangnya generalisasi hasil *tuning* dan pendekatan akan pendekatan yang lebih adaptif, serta membuka peluang penelitian untuk inovasi di bidang *hyperparameter tuning* dalam *software engineering*.

Kata kunci: *Bibliometrix analysis, Hyperparameter tuning, Software engineering, Systematic Literature Review.*

Research Trend Analysis of Hyperparameter Tuning in Software Engineering: A Systematic Review

Abstract

Hyperparameter tuning is a crucial aspect in enhancing the performance of machine learning models within the field of software engineering. Although it has a significant impact, research trends and developments in hyperparameter tuning in this area remain limited and have not been extensively explored systematically. This study aims to analyze research trends in hyperparameter tuning within software engineering through a Systematic Literature Review (SLR) and Bibliometric Analysis. Using the Bibliometrix R-Package, 503 articles sourced from the Scopus database were analyzed to identify dominant tuning methods, challenges faced, and future research opportunities. The findings show fluctuations in publication volume from 2020 to 2025, with a notable increase in 2024 (132 articles), but a decline in average citations in 2025 (to 10 citations), which can be attributed to the shorter time for recent publications to accrue citations. Wang Y is the most prolific author with 11 articles and the most influential author based on a fractionalization score of 1.75. The most relevant and popular sources by publication volume are IEEE Transactions on Software Engineering (18 articles) for journals, and ACM International Conference Proceeding Series (15 articles) for conferences. The most frequent keyword associated with the research is deep learning, with 89 appearances. The study also highlights a significant increase in exploration of automatic tuning techniques to address model complexity and high computational costs. This review provides insights into current developments and ongoing challenges, such as the limited generalization of tuning results and the need for more adaptive approaches, while opening opportunities for innovation in hyperparameter tuning within the field of software engineering.

Keywords: *Bibliometrix analysis, Hyperparameter tuning, Software engineering, Systematic Literature Review.*

1. PENDAHULUAN

Teknologi kecerdasan buatan (*Artificial Intelligence/AI*) dan pembelajaran mesin (*Machine Learning/ML*) telah mengalami kemajuan dalam beberapa dekade terakhir [1][2]. *ML* memungkinkan sistem untuk belajar dari data dan membuat prediksi atau keputusan tanpa perlu pemrograman yang rumit [3]. Berbagai algoritma *ML*, seperti *Supervised Learning* [4][5], *Unsupervised Learning* [6][7], dan *Reinforcement Learning* [8][9], telah diterapkan dalam berbagai bidang, termasuk pengenalan pola, analisis data besar, serta pengambilan keputusan berbasis data [10][11][12]. Seiring dengan meningkatnya kompleksitas permasalahan yang dihadapi, optimasi model *ML* menjadi sangat penting untuk meningkatkan akurasi dan efisiensi sistem [13][14][15].

Penerapan *ML* dalam bidang rekayasa perangkat lunak (*Software Engineering/SE*), telah membawa perubahan signifikan dalam berbagai aspek, termasuk pemeliharaan perangkat lunak [16][17][18], pengujian otomatis [19][20][21][22], deteksi kerentanan keamanan [23][24][25], dan estimasi biaya proyek [26][27][28]. Salah satu tantangan utama dalam penerapan *ML* dalam *SE* adalah bagaimana mengoptimalkan performa model agar sesuai dengan kebutuhan spesifik dalam bidang ini [29][13]. Proses penyesuaian parameter eksternal yang dikenal sebagai *Hyperparameter tuning* memainkan peran penting dalam meningkatkan efisiensi dan akurasi model [30][31]. Pemilihan teknik tuning yang tepat dapat meningkatkan kualitas perangkat lunak secara keseluruhan, mulai dari akurasi prediksi, efisiensi komputasi, hingga adaptabilitas model terhadap berbagai skenario penggunaan [15][32][33][34].

Penelitian sebelumnya telah melakukan studi empiris untuk menganalisis dampak *hyperparameter tuning* dan optimasi model terhadap performa *Deep Neural Network*, menunjukkan bahwa penggunaan teknik *hyperparameter tuning*, seperti *Bayesian optimization*, dan optimasi model melalui *pruning* serta *quantization* dapat meningkatkan akurasi tanpa mengurangi performa secara signifikan [35]. Penelitian lain menegaskan bahwa *iterative tuning* dapat mencapai akurasi maksimum hingga 95,1% dalam mengidentifikasi *software defect*, serta kerangka kerja tersebut menunjukkan peningkatan efisiensi, yang secara signifikan mengurangi waktu eksekusi selama fase pelatihan dan pengujian masing-masing sebesar 51,52% dan 52,31%. Sehingga lebih ekonomis secara komputasi [36]. Selain itu, berbagai teknik-teknik seperti *grid search* [37], *Bayesian* [38], *random search* [39], *Genetic Algorithm* [40], dan *Optuna* [41] serta metode *metaheuristik* [42] terbukti tidak hanya meningkatkan akurasi model, tetapi juga mengurangi waktu pelatihan dan penggunaan memori secara signifikan. Hal ini menunjukkan pentingnya efisiensi komputasi dalam proses *tuning hyperparameter*, terutama di lingkungan dengan keterbatasan sumber daya [43][44][45]. *SmartTuning* adalah alat otomatis yang dapat mengoptimalkan konfigurasi aplikasi *cloud* secara dinamis dan otomatis, meningkatkan efisiensi hingga 58% dan mengurangi biaya 27% dengan penyesuaian otomatis berdasarkan *Bayesian Optimization* saat aplikasi berjalan [46]. Namun, tantangan utama dalam *hyperparameter tuning* adalah memastikan hasil yang dapat digeneralisasi hasil di berbagai skenario pengembangan perangkat lunak.

Meningkatnya kompleksitas sistem perangkat lunak telah mendorong perhatian terhadap penelitian tentang *hyperparameter tuning* dalam bidang ini [47]. Namun, dari penelitian tersebut belum ada yang membahas secara komprehensif mengenai penerapan *hyperparameter tuning* di bidang *SE* sehingga dilakukan analisis *Systematic Literature Review* (SLR) dan *bibliometrix analysis* untuk mengetahui gaps penelitian dan tren penelitian [48]. Kajian ini akan mengidentifikasi metode yang paling banyak digunakan, tantangan utama dalam implementasi, serta arah penelitian di masa depan. Hasil kajian diharapkan dapat menjadi panduan bagi peneliti, pengembang perangkat lunak, serta industri dalam mengoptimalkan teknik *hyperparameter tuning* yang lebih efektif dan efisien. Pemahaman mengenai potensi kolaborasi antara *machine learning* dan *Software Engineering* juga dapat membuka peluang inovasi dalam menciptakan sistem yang lebih adaptif dan cerdas di masa depan.

Untuk mempermudah analisis terhadap pokok pembahasan, beberapa pertanyaan penelitian yang relevan dirumuskan sebagai berikut:

- a. RQ1: Bagaimana tren penelitian terkait *hyperparameter tuning* dari waktu ke waktu?
Motivasi: Motivasi RQ1 untuk mengidentifikasi perkembangan dalam fokus penelitian, metodologi dan tren terkait *hyperparameter tuning*, sehingga dapat memberikan pemahaman tentang bagaimana topik ini berkembang dari waktu ke waktu dan mengarahkan penelitian di masa mendatang.
- b. RQ2: Bagaimana kontribusi penelitian terkait *hyperparameter tuning* terhadap komunitas akademik?
Motivasi: Motivasi RQ2 fokus pada mengevaluasi dampak praktis dan teoritis dari *hyperparameter tuning* terhadap berbagai penelitian yang lain, serta efektivitasnya.
- c. RQ3: Apa saja sumber utama yang berkontribusi secara signifikan terhadap penelitian tentang *hyperparameter tuning*?
Motivasi: Motivasi RQ3 fokus pada mengidentifikasi publikasi pada konferensi dan jurnal terkemuka yang memiliki kontribusi signifikan dalam memajukan pengetahuan terkait *hyperparameter tuning*, sehingga dapat digunakan sebagai referensi utama oleh peneliti lain.
- d. RQ4: Apa tema dan topik utama yang sering dibahas pada penelitian tentang *hyperparameter tuning*?

Motivasi: Motivasi RQ4 fokus pada menganalisis tema dominan dan topik utama yang muncul dalam penelitian terkait, sehingga memberikan gambaran menyeluruh tentang area yang telah dieksplorasi dan potensi pengembangan tema baru.

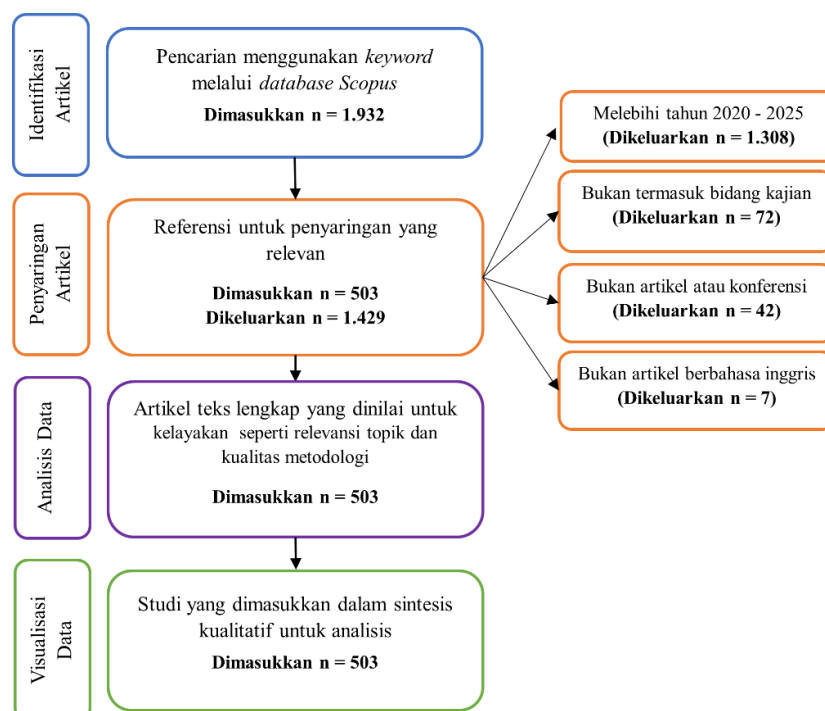
- e. RQ5: Apa *research gaps* dan arah penelitian di masa mendatang terkait *hyperparameter tuning*?

Motivasi: Motivasi RQ5 fokus pada mengidentifikasi kekurangan dalam penelitian saat ini dan peluang untuk penelitian di masa depan, sehingga dapat memberikan arah yang jelas bagi peneliti dalam mengeksplorasi topik ini lebih mendalam.

Hasil penelitian ini diharapkan dapat memberikan kontribusi yang signifikan dalam bidang *SE*, khususnya dalam optimasi *hyperparameter tuning*. Kontribusi utama penelitian ini meliputi: (1) Pemetaan topik penelitian yang menggambarkan tema-tema utama dan tren terkait *hyperparameter tuning*. (2) Identifikasi celah penelitian (*research gaps*) yang menunjukkan kurangnya eksplorasi pada metode *tuning* adaptif, kombinasi algoritma berbasis kecerdasan buatan, dan penerapan *tuning* dalam sistem *real-time*, serta (3) Rekomendasi arah penelitian masa depan yang mencakup pengembangan metode *tuning* otomatis berbasis *AI*, peningkatan efisiensi komputasi untuk model kompleks, dan integrasi *hyperparameter tuning* dalam pengembangan perangkat lunak berbasis *cloud* dan *edge computing*.

2. METODE PENELITIAN

Penelitian ini menggunakan dua pendekatan utama yaitu *Systematic Literature Review* dan *Bibliometric analysis*, mengikuti penelitian sebelumnya [49][50]. *Bibliometric analysis* dilakukan menggunakan *Bibliometric R-package* untuk memvisualisasikan peta hubungan antar topik dan mengidentifikasi topik penelitian terkini. *Bibliometric* merupakan sebuah *R package* (perpustakaan/ perangkat lunak tambahan dalam bahasa pemrograman R) yang dirancang khusus untuk melakukan *bibliometric analysis* secara komprehensif [51]. Pada *package* ini, fungsi *biblioshiny* memungkinkan pengguna melakukan *bibliometric analysis* dan dimanfaatkan untuk menghasilkan visualisasi data [52]. Pengumpulan data untuk analisis *bibliometric* dilakukan pada tanggal 23 Februari 2025 melalui *database scopus*, selanjutnya Artikel yang diseleksi dievaluasi berdasarkan jumlah kutipan dan publikasi jurnal bereputasi tinggi untuk memastikan relevansi dan kualitas hasil penelitian.



Gambar 1. Diagram Alir PRISMA dalam Penelitian *Bibliometric*

Gambar 1 menjelaskan tahapan penelitian yang terdiri dari empat tahap utama yaitu Identifikasi Artikel, Penyaringan Artikel, Analisis Data, dan Visualisasi Data. Berikut adalah penjelasan dari masing-masing tahapan tersebut:

- a. Identifikasi Artikel: Tahap ini dilakukan proses identifikasi sumber artikel yang akan digunakan dari *database Scopus*. Sumber ini dipilih karena merupakan salah satu *database* ilmiah yang populer dan memiliki tingkat

- kredibilitas tinggi. Pengumpulan data artikel dilakukan pada tanggal 23 Februari 2025. Proses identifikasi artikel relevan dilakukan dengan menerapkan kata kunci (*keyword*) “*hyperparameter OR tuning*” AND “*Software AND Engineering*”. Berdasarkan kata kunci tersebut, diperoleh sebanyak 1.932 artikel.
- b. Penyaringan Artikel: Tahap ini bertujuan untuk menyaring artikel sesuai dengan kriteria inklusi dan eksklusi yang dijelaskan pada Tabel 1, meliputi rentang tahun, bidang kajian, jenis publikasi, dan bahasa. Proses penyaringan dilakukan sebagai berikut:
- 1) Penyaringan berdasarkan rentang tahun: Artikel disaring berdasarkan rentang tahun 2020 – 2025. Sehingga dari 1.932 artikel tersisa 624 artikel, sedangkan 1.308 artikel yang berada di luar rentang tahun tersebut di eliminasi.
 - 2) Penyaringan berdasarkan bidang kajian: Penyaringan dilakukan untuk memastikan artikel dalam bidang kajian *computer science* dan *Engineering*. Sehingga dari 624 artikel, tersisa 552 artikel. Terdapat 72 artikel yang di eliminasi karena bukan termasuk bidang kajian tersebut.
 - 3) Penyaringan berdasarkan jenis penelitian: Penyaringan hanya artikel yang berasal dari jurnal dan konferensi yang disertakan. Sehingga dari 552 artikel tersisa sebanyak 510 artikel, setelah 42 artikel di eliminasi karena bukan termasuk jurnal atau konferensi.
 - 4) Penyaringan berdasarkan bahasa: Penyaringan hanya artikel berbahasa Inggris yang dipertahankan. Sehingga dari 510 artikel tersisa 503 artikel, setelah 7 artikel yang tidak berbahasa Inggris di eliminasi.

Tabel 1. Kriteria inklusi dan eksklusi untuk menyaring publikasi yang relevan

Kriteria	Inklusi	Eksklusi
Tahun Publikasi	2020 – 2025	-
Bidang Kajian	<i>Computer Science, Engineering</i>	Bukan <i>Computer Science</i> , Bukan <i>Engineering</i>
Jenis Publikasi	<i>Article, Conference</i>	<i>Conference review, Book Chapter, Review, Book</i>
Bahasa	Inggris	Bukan Inggris

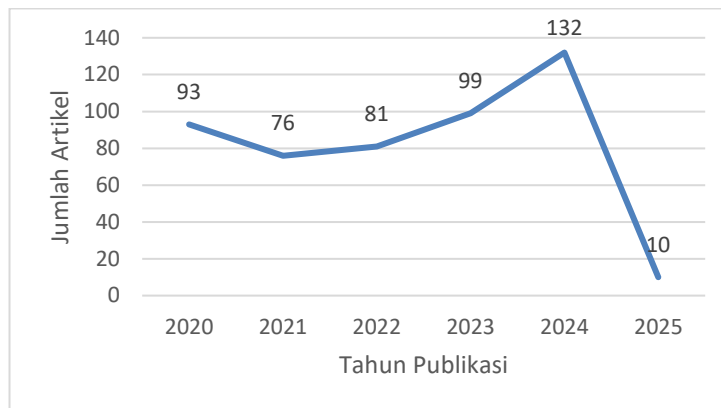
- c. Analisis Data: Tahap ini artikel yang telah lolos proses penyaringan dievaluasi lebih lanjut berdasarkan kriteria inklusi, seperti relevansi topik dan kualitas metodologi. Tujuannya adalah memastikan bahwa hanya artikel yang benar-benar relevan dengan tujuan penelitian yang dipertahankan. Setelah proses ini, sebanyak 503 artikel dinyatakan layak sebagai referensi.
- d. Visualisasi Data: Tahap terakhir yaitu memilih artikel yang benar-benar memenuhi seluruh kriteria inklusi yang akan digunakan dalam penelitian. Setelah melalui proses *screening* dan *eligibility*, hanya data yang dianggap paling relevan dan berkualitas yang tersisa untuk mendukung tujuan penelitian. Artikel yang relevan dan memiliki kualitas terbaik akan dijadikan referensi utama dalam penelitian. Hasil akhir dari proses seleksi ini adalah 503 artikel yang akan digunakan sebagai dasar teori dan data pendukung penelitian.

3. HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil analisis secara komprehensif tentang studi terkait *hyperparameter tuning*. Analisis ini memberikan wawasan tentang tren penelitian terkait *hyperparameter tuning*.

3.1. RQ1: Bagaimana tren penelitian terkait *hyperparameter tuning* dari waktu ke waktu?

Penelitian ini menganalisis perkembangan tren dan topik terkait *hyperparameter tuning* pada rentang tahun 2020 hingga 2025. Berdasarkan Gambar 2, jumlah publikasi ilmiah per tahun sejak tahun 2020 hingga 2025 mengalami fluktuasi. Pada tahun 2020 terdapat 93 artikel yang diterbitkan, sedangkan jumlah ini sedikit menurun pada tahun 2021 menjadi 76 artikel. Meskipun tidak signifikan, jumlah artikel mengalami kenaikan di tahun 2022 yaitu sebanyak 81 artikel. Tren ini terus meningkat pada tahun 2023 dengan 99 publikasi, dan mencapai puncaknya pada tahun 2024 dengan 132 artikel. Adapun jumlah publikasi pada tahun 2025 baru mencapai 10 artikel pada saat pengumpulan data, sehingga jumlah ini belum bisa mencerminkan produktivitas ilmiah sepanjang tahun 2025.



Gambar 2. Tren publikasi ilmiah terkait *hyperparameter tuning* dalam bidang *software engineering* sejak 2020

Tren penelitian terkait *hyperparameter tuning* pada rentang tahun 2020 hingga 2025 juga dapat dijelaskan berdasarkan nilai rata-rata kutipan artikel, seperti yang ditunjukkan dalam Tabel 2. Secara umum, rerata kutipan tiap artikel mengalami penurunan selama 6 tahun terakhir. Pada tahun 2020, rata-rata kutipan per artikel mencapai 16,22 dengan rata-rata kutipan per tahun sebesar 2,70. Namun, angka ini mengalami penurunan secara bertahap pada tahun-tahun berikutnya, dengan penurunan yang cukup signifikan setelah tahun 2021. Pada tahun 2022 dan 2023, rata-rata kutipan per artikel masing-masing adalah 9,27 dan 6,28, dengan rata-rata kutipan per tahun yang relatif stabil di sekitar 2,09–2,32. Tren ini semakin drastis pada tahun 2024 dan 2025, dengan rata-rata kutipan per artikel turun drastis menjadi 0,88 dan 0,10, serta rata-rata kutipan per tahun yang semakin kecil, mencapai 0,44 dan 0,10. Penurunan pada tahun 2025 ini dapat dikaitkan dengan waktu yang diperlukan bagi artikel baru untuk mendapatkan kutipan, dimana publikasi yang lebih baru memiliki waktu yang lebih singkat untuk dikutip dibandingkan publikasi yang lebih lama.

Tabel 2. Perkembangan kutipan artikel dalam 5 tahun terakhir

Tahun	Rata-rata Kutipan Per Artikel	Rata-rata kutipan per Tahun
2020	16,22	2,70
2021	11,25	2,25
2022	9,27	2,32
2023	6,28	2,09
2024	0,88	0,44
2025	0,10	0,10

Hasil ini dapat disimpulkan bahwa meskipun jumlah publikasi mengalami peningkatan dalam beberapa tahun terakhir, tren kutipan menunjukkan penurunan yang cukup signifikan. Hal ini menunjukkan bahwa meskipun semakin banyak penelitian yang dilakukan dalam bidang *hyperparameter tuning* untuk *software engineering*, dampak atau relevansi artikel-artikel tersebut dalam komunitas akademik cenderung menurun. Salah satu kemungkinan penyebabnya adalah semakin banyaknya artikel yang diterbitkan dalam topik ini, sehingga distribusi kutipan menjadi lebih tersebar. Selain itu, penurunan tajam dalam rata-rata kutipan pada tahun 2024 dan 2025 bisa disebabkan oleh waktu yang masih relatif singkat sejak publikasi, sehingga artikel-artikel terbaru belum sempat mendapatkan banyak kutipan. Secara keseluruhan, tren ini menunjukkan bahwa meskipun minat penelitian dalam topik ini masih tinggi, daya tarik atau pengaruhnya dalam komunitas ilmiah mungkin mulai berkurang, atau ada pergeseran ke arah pendekatan baru dalam optimasi model dalam *software engineering*.

3.2. RQ2: Bagaimana kontribusi penelitian terkait *hyperparameter tuning* terhadap komunitas akademik?

Untuk mengidentifikasi kontribusi penelitian tentang *hyperparameter tuning* dapat dilihat dari artikel yang paling banyak dikutip dan penulis yang paling relevan terhadap topik tersebut.

Tabel 3. Daftar 10 artikel yang paling banyak dikutip antara tahun 2020 dan 2025

Penulis dan Sumber Artikel	DOI	Tahun	Jumlah Kutipan
Lutellier T, 2020, <i>ISSTA - Proc ACM Sigsoft Int Symp Softw Test Anal</i> [53]	10.1145/3395363.3397369	2020	231
Choe J, 2020, <i>Proc IEEE Comput Soc Conf Comput Vision Pattern Recognit</i> [54]	10.1109/CVPR42600.2020.00320	2020	142
Luo JH, 2020, <i>Pattern Recogn</i> [55]	10.1016/j.patcog.2020.107461	2020	126
Bui NDQ, 2021, <i>Proc Int Conf on Software Eng</i> [56]	10.1109/ICSE43902.2021.00109	2021	88
Rahman MM, 2021, <i>Symmetry</i> [57]	10.3390/sym13020247	2021	87
Liu Y, 2020, <i>Inf Sci</i> [58]	10.1016/j.ins.2018.12.078	2020	87
Zhang CL, 2020, <i>Proc IEEE Comput Soc Conf Comput Vision Pattern Recognit</i> [59]	10.1109/CVPR42600.2020.01347	2020	86
Yang Z, 2022, <i>Proc Int Conf on Software Eng</i> [60]	10.1145/3510003.3510146	2022	83
Lawrence NP, 2022, <i>Control Eng Pract</i> [61]	10.1016/j.conengprac.2021.105046	2022	71
Zou Z, 2020, <i>Proc IEEE Comput Soc Conf Comput Vision Pattern Recognit</i> [62]	10.1109/CVPR42600.2020.01282	2020	60

Tabel 3 menjelaskan 10 artikel yang paling banyak dikutip terkait penelitian *hyperparameter tuning* pada rentang waktu 2020–2025. Artikel yang memiliki jumlah kutipan tertinggi adalah Lutellier T, 2020 [53] (*ISSTA - Proceedings of the ACM SIGSOFT International Symposium on Software testing and Analysis*) dengan 231 kutipan yang menunjukkan pengaruhnya yang kuat dalam bidang pengujian perangkat lunak dan analisis perangkat lunak. Choe J, 2020 [54] (*Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern recognition*) menempati posisi kedua terbanyak dengan 142 kutipan, yang mencerminkan tingginya minat pada pendekatan *computer vision* dalam pengembangan perangkat lunak. Posisi ketiga adalah artikel oleh Luo JH, 2020 [55] dengan 126 kutipan, menunjukkan adanya relevansi *pattern recognition* dalam otomatisasi perangkat lunak dan pengenalan pola untuk *tuning* model. Selain itu, artikel-artikel lain seperti Bui NDQ, 2021 [56] (*Proceedings of the International Conference on Software engineering*) dengan 88 kutipan, Rahman MM, 2021 [57] (*Symmetry*) dengan 87 kutipan, Liu Y, 2020 [58] (*Information Sciences*) dengan 87 kutipan, menunjukkan bahwa penelitian pada tahun 2020 dan 2021 menjadi titik penting dalam pengembangan keilmuan terkait topik ini, terutama yang berfokus pada *software testing*, *symmetry-based modeling*, dan *information sciences*.

Kontribusi terbaru juga terlihat dari artikel tahun 2022 seperti Zhang CL (2022) [59] sebanyak 86 kutipan, Yang Z (2022) [60] dengan 83 kutipan, dan Lawrence NP (2022) [61] sebanyak 71 kutipan. Hal ini menandakan adanya dorongan riset yang berkelanjutan, terutama yang berkaitan dengan konferensi dan jurnal internasional bergengsi. Dengan demikian, secara umum artikel-artikel tersebut berpengaruh sebagai landasan teoritis dan praktis dalam mengembangkan metodologi *hyperparameter tuning* dalam *software engineering*, dan menunjukkan tren perkembangan riset terkini.

Berdasarkan pola ini dapat disimpulkan bahwa meskipun jumlah publikasi dalam topik *hyperparameter tuning* meningkat dalam beberapa tahun terakhir, tidak semua artikel memiliki tingkat kutipan yang tinggi. Faktor seperti relevansi topik dengan tren penelitian terkini, cakupan aplikasi dalam berbagai domain, serta kualitas dan kredibilitas *venue* publikasi juga berperan dalam menentukan tingkat kutipan. Selain itu, penelitian yang menghubungkan *hyperparameter tuning* dengan bidang yang lebih luas, seperti *computer vision* dan *machine learning*, mungkin lebih menarik perhatian komunitas ilmiah dibandingkan yang berfokus secara eksklusif pada *software engineering*. Hal ini menunjukkan bahwa pengaruh *hyperparameter tuning* dalam *software engineering* dapat diperluas dengan menjangkau disiplin ilmu lain yang memiliki kebutuhan serupa dalam optimasi model.

Tabel 4 menyajikan daftar 10 penulis dengan jumlah artikel terbanyak serta skor fraksionalisasi penulis artikel terkait *hyperparameter tuning* pada periode 2020–2025. Konsep fraksionalisasi digunakan untuk mengukur kontribusi penulis secara proporsional [51][52]. Jika sebuah artikel memiliki banyak penulis, maka bobot kontribusi akan dibagi rata [64]. Misalnya, untuk artikel dengan 4 penulis, masing-masing akan mendapat skor fraksional sebesar 0,25 [63]. Dengan demikian, penulis yang sering berkolaborasi akan memiliki skor fraksionalisasi lebih kecil dibanding mereka yang lebih banyak menulis secara individu atau sebagai kontributor utama.

Tabel 4. Daftar 10 penulis yang paling relevan pada tahun 2020 - 2025

Penulis	Jumlah Artikel	Skor Fraksionalisasi
Wang Y.	11	1,75
Li X.	9	2,57
Chen T.	8	4,20
Li M.	8	2,20
Zhang J.	8	1,58
Li Y.	7	1,45
Wang J.	7	1,11
Liu Y.	6	1,45
Wang Z.	6	1,03
Wang S.	6	0,85

Berdasarkan Tabel 4 jumlah artikel yang tinggi tidak selalu berbanding lurus dengan kontribusi individu. Sebagai contoh, Wang Y menjadi penulis dengan jumlah artikel terbanyak yakni 11 artikel dengan skor fraksionalisasi sebesar 1,75 yang diikuti oleh Li X dengan 9 artikel dan skor fraksionalisasi sebesar 2,57. Hal ini berbanding terbalik dengan Chen T yang memiliki 8 artikel namun dengan skor fraksionalisasi tertinggi sebesar 4,20 yang menunjukkan signifikansi kontribusi yang dilakukan dalam penelitian. Selain itu, Li M dan Zhang J juga aktif berkontribusi dengan 8 artikel serta skor fraksionalisasi masing-masing sebesar 2,20 dan 1,58. Hal ini menegaskan bahwa jumlah artikel bukan satu-satunya indikator utama dalam menilai dampak akademik seorang penulis, melainkan juga seberapa besar kontribusi langsung mereka dalam setiap publikasi. Oleh karena itu, dalam mengidentifikasi pakar di bidang *hyperparameter tuning* dalam *software engineering*, pendekatan fraksionalisasi memberikan wawasan yang lebih proporsional terhadap kontribusi individu dalam penelitian yang bersifat kolaboratif.

3.3. RQ3: Apa saja sumber utama yang berkontribusi secara signifikan terhadap penelitian tentang *hyperparameter tuning*?

Distribusi artikel di berbagai sumber menandakan upaya kolaboratif antara jurnal dan konferensi dalam menyebarluaskan pengetahuan serta memfasilitasi pemahaman yang komprehensif tentang *hyperparameter tuning* di komunitas akademik. Identifikasi sumber utama yang berkontribusi terhadap penelitian ini dapat digunakan untuk mengetahui tren perkembangan keilmuan, akses referensi yang relevan, serta menentukan arah penelitian yang masih memiliki peluang eksplorasi lebih lanjut. Selain itu, analisis terhadap sumber-sumber ini juga dapat membantu dalam menilai kredibilitas dan dampak dari penelitian yang telah dilakukan.

Tabel 5. Sumber paling relevan untuk jurnal dan konferensi pada tahun 2020 - 2025

Jenis	Sumber	Jumlah Artikel
Jurnal	1. <i>IEEE Transactions On Software engineering</i>	18
	2. <i>IEEE Access</i>	10
	3. <i>Empirical Software Engineering</i>	6
	4. <i>Applied Sciences (Switzerland)</i>	5
	5. <i>ACM Transactions on Software Engineering and Methodology</i>	4
Konferensi	1. <i>ACM International Conference Proceeding Series</i>	15
	2. <i>Ceur Workshop Proceedings</i>	12
	3. <i>Lecture Notes in Computer Science (Including Subseries Lecture Notes In Artificial intelligence and Lecture Notes in Bioinformatics)</i>	12
	4. <i>Proceedings - International Conference on Software Engineering</i>	11
	5. <i>SAE Technical Papers</i>	10

Tabel 5 menunjukkan sumber paling relevan berdasarkan jumlah artikel yang dipublikasikan pada tahun 2020-2025. Jurnal ilmiah *IEEE Transactions on Software Engineering* menjadi jurnal dengan kontribusi terbesar sebanyak 18 artikel terkait *hyperparameter tuning* yang diterapkan dalam bidang *software engineering*, yang diikuti oleh Jurnal *IEEE Access* dengan 10 artikel relevan. Jurnal *Empirical Software Engineering* yang berfokus pada studi

empiris dan analisis data dalam rekayasa perangkat lunak berkontribusi sebanyak 6 artikel, yang diikuti oleh jurnal *Applied Sciences (Switzerland)* dengan 5 artikel. Adapun Jurnal *ACM Transactions on Software Engineering and Methodology* berkontribusi sebanyak 4 artikel selama 6 tahun terakhir yang merefleksikan kontribusi komunitas ACM dalam metodologi pengembangan perangkat lunak.

Sementara itu, pada kategori konferensi, *ACM International Conference Proceeding Series* dengan 15 artikel relevan menjadi konferensi terkemuka untuk diskusi terkait penerapan *hyperparameter tuning* dalam bidang pengembangan perangkat lunak. *Ceur Workshop Proceedings* (12 artikel relevan) memfasilitasi publikasi hasil penelitian terkini yang sering kali melibatkan kajian eksperimental. *Lecture Notes in Computer Science* dengan 11 artikel relevan menjadi sumber penting untuk riset komputasi, termasuk sub-seri AI dan bioinformatika. Adapun *Proceedings - International Conference on Software Engineering* yang telah menerbitkan 11 artikel relevan pada tahun 2020-2025 dapat mencerminkan pentingnya forum internasional dalam memajukan teknik pengembangan perangkat lunak. *SAE Technical Papers* juga telah menerbitkan sebanyak 10 artikel relevan untuk riset interdisipliner yang menghubungkan rekayasa perangkat lunak dengan sistem teknis lainnya.

Berdasarkan hal tersebut maka dapat disimpulkan bahwa jurnal dan prosiding dari IEEE dan ACM mendominasi sebagai sumber yang paling banyak digunakan dalam penelitian terkait *hyperparameter tuning*, yang menunjukkan bahwa komunitas akademik sangat mengandalkan publikasi dari organisasi tersebut dalam bidang rekayasa perangkat lunak.

Tabel 6. Sumber utama publikasi ilmiah terkait *hyperparameter tuning* pada tahun 2020 sampai 2025

Tahun	2020	2021	2022	2023	2024	2025	Total Publikasi
<i>IEEE Transactions on Software Engineering</i>	1	1	6	5	4	1	18
<i>ACM International Conference Proceeding Series</i>	4	2	0	4	5	0	15
<i>Ceur Workshop Proceedings</i>	6	1	0	2	3	0	12
<i>Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial intelligence and Lecture Notes in Bioinformatics)</i>	3	0	2	2	4	1	12
<i>Proceedings - International Conference on Software Engineering</i>	1	1	1	1	7	0	11

Analisis lebih lanjut juga dilakukan untuk mengetahui perkembangan jumlah publikasi dari lima sumber utama dalam kurun waktu 2020 hingga 2025. Berdasarkan Tabel 6, dapat dilihat bahwa publikasi ilmiah terkait *hyperparameter tuning* selama periode 2020–2025 didominasi oleh beberapa sumber utama. *IEEE Transactions on Software Engineering* secara konsisten menjadi salah satu sumber utama dengan jumlah publikasi yang relatif stabil setiap tahun, menunjukkan peran penting jurnal ini dalam mendukung penelitian. Sementara itu *ACM International Conference Proceeding Series* menunjukkan tren peningkatan dari 2020 ke 2025, mencerminkan semakin besarnya perhatian komunitas akademik terhadap topik ini di berbagai konferensi internasional.

Selain itu *Ceur Workshop Proceedings* memiliki total 12 publikasi, meskipun jumlahnya berfluktuasi, tertinggi pada tahun 2020 (6 publikasi) dan menurun tajam di tahun-tahun berikutnya juga turut yang mengindikasikan adanya minat khusus terhadap diskusi teknis dan praktik terbaik terkait *hyperparameter tuning*. Sumber lainnya seperti *Lecture Notes in Computer Science* (12 publikasi) juga turut berkontribusi, khususnya dalam mengaitkan *hyperparameter tuning* dengan kecerdasan buatan dan bioinformatika. Menariknya *Proceedings - International Conference On Software Engineering* memiliki jumlah publikasi yang meningkat tajam pada tahun 2024 (7 publikasi), menunjukkan bahwa topik ini semakin menjadi perhatian utama dalam konferensi perangkat lunak global.

Secara keseluruhan, data ini mencerminkan tren peningkatan publikasi dari berbagai sumber kredibel, menunjukkan bahwa *hyperparameter tuning* dalam *software engineering* terus berkembang dan mendapatkan perhatian besar dari komunitas ilmiah dan industri.

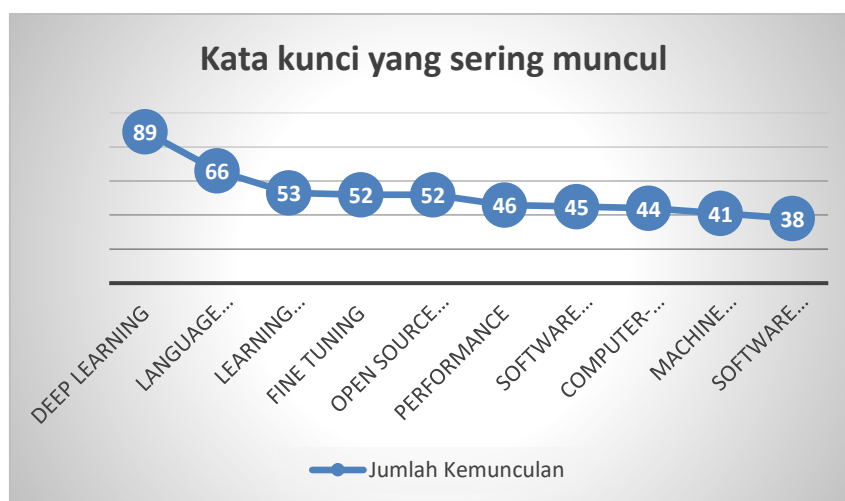
3.4. RQ4: Apa tema dan topik utama yang sering dibahas pada penelitian tentang *hyperparameter tuning*?

Analisis ini menjelaskan tentang tema dan topik utama yang sering dibahas dalam penelitian terkait *hyperparameter tuning*. Hal ini dapat dimanfaatkan untuk mengetahui topik-topik yang populer selama 6 tahun terakhir, sehingga dapat diidentifikasi celah penelitian yang dapat diatasi dalam penelitian berikutnya.

Tabel 7 menunjukkan 10 kata kunci yang paling sering muncul dalam penelitian terkait *hyperparameter tuning*. Kata kunci *deep learning* menempati posisi teratas dengan 89 kemunculan, menunjukkan bahwa pendekatan ini memiliki peran penting dalam penelitian terkait. *language model* muncul sebanyak 66 kali, mengindikasikan perhatian besar terhadap penggunaan model bahasa. Kata *learning systems* (53 kali) dan *fine tuning* (52 kali) merefleksikan perhatian peneliti pada proses penyempurnaan model melalui penyesuaian *hyperparameter*. Selain itu, *open source software* (52 kali) menunjukkan adanya tren pemanfaatan perangkat lunak terbuka, yang memungkinkan kolaborasi dan replikasi eksperimen. Kata *performance* (46 kali) dan *software testing* (45 kali) menggaris bawahi pentingnya evaluasi kinerja serta pengujian perangkat lunak dalam konteks *hyperparameter tuning*. Terakhir, *computer-aided software engineering* (44 kali) dan *machine learning* (41 kali) menegaskan keterkaitan kuat antara rekayasa perangkat lunak berbantuan komputer dan metode pembelajaran mesin sebagai bagian integral dalam mengembangkan model yang optimal.

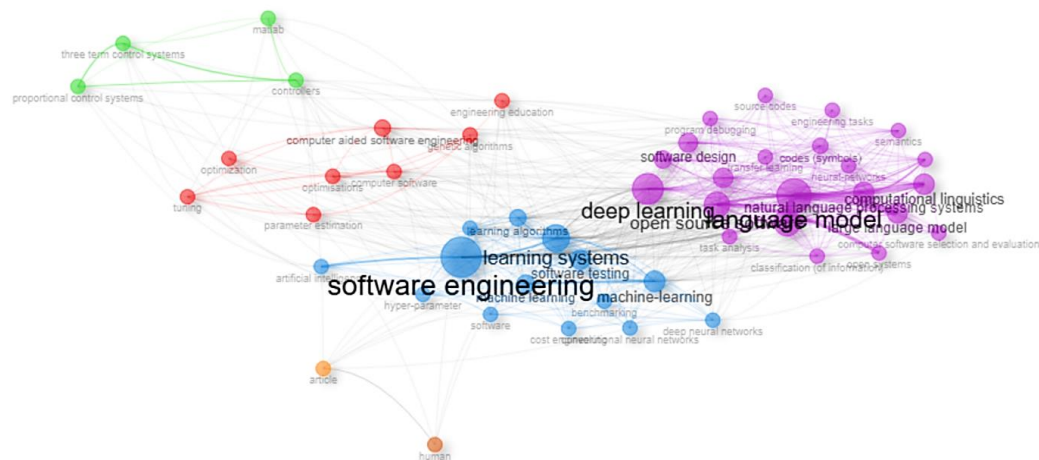
Tabel 7. Daftar 10 kata kunci yang sering muncul dalam artikel relevan

Kata Kunci	Jumlah kemunculan
<i>deep learning</i>	89
<i>language model</i>	66
<i>learning systems</i>	53
<i>fine tuning</i>	52
<i>open source software</i>	52
<i>performance</i>	46
<i>software testing</i>	45
<i>computer-aided software engineering</i>	44
<i>machine learning</i>	41
<i>software design</i>	38



Gambar 3. Kata Kunci yang sering muncul dalam arikel relevan

Dari hasil ini, dapat disimpulkan bahwa penelitian *hyperparameter tuning* erat kaitannya dengan pengembangan dan optimalisasi model pembelajaran mesin, terutama dalam *deep learning* dan *language models*. Selain itu, aspek kinerja dan pengujian perangkat lunak juga menjadi perhatian utama dalam studi ini. Dominasi kata kunci yang berkaitan dengan sistem pembelajaran dan rekayasa perangkat lunak menunjukkan bahwa penelitian ini tidak hanya berfokus pada aspek teoritis, tetapi juga pada implementasi praktis untuk meningkatkan kualitas dan efisiensi pengembangan perangkat lunak.



Gambar 3. Co-occurrence Network

Beberapa kata kunci yang sering muncul dalam Tabel 7 juga tampak menonjol dalam jaringan yang ditunjukkan pada Gambar 3, menjelaskan tentang peta visualisasi *co-occurrence* kata kunci terkait dengan tren penelitian tentang *hyperparameter tuning*. Ukuran label mencerminkan frekuensi kemunculan kata kunci, di mana *software engineering* dan *language models* tampak menjadi topik dominan. Garis penghubung antar kata kunci merepresentasikan hubungan ko-eksistensi, menunjukkan bagaimana istilah-istilah ini muncul bersamaan dalam literatur dan keterkaitan antar konsep dalam kerangka penelitian *hyperparameter tuning*. Hubungan antar kelompok menunjukkan irisan berbagai bidang penelitian, misalnya *machine learning* dan *software engineering* saling berhubungan dalam konteks pengembangan dan pengujian perangkat lunak.

Selain itu, peta jaringan ini juga memperlihatkan adanya 5 klaster utama yang ditandai dengan warna berbeda, dengan masing-masing klaster merepresentasikan topik spesifik yang saling berkaitan:

- Klaster biru berpusat pada *software engineering* sebagai kata kunci dominan, dengan koneksi kuat ke konsep seperti *machine learning*, *deep learning*, *software testing*, dan *learning systems*. Hal ini mencerminkan integrasi metode pembelajaran mesin dalam proses pengembangan perangkat lunak serta penerapan model-model cerdas untuk meningkatkan kualitas perangkat lunak.
- Klaster ungu berfokus pada *language models* dan *natural language processing systems (NLP)*. Istilah-istilah seperti *computational linguistics*, *software design*, dan program *debugging* menunjukkan bahwa penelitian di area ini mengkaji bagaimana model bahasa digunakan untuk menganalisis, memahami, dan mengotomatisasi tugas-tugas pemrograman dan rekayasa perangkat lunak.
- Klaster merah mencakup konsep seperti *Optimization*, *Parameter estimation*, dan *computer-aided software engineering*. Hal ini menunjukkan bahwa *hyperparameter tuning* erat kaitannya dengan proses optimasi model, baik dari segi akurasi maupun efisiensi komputasi, sehingga relevan dalam konteks rekayasa perangkat lunak berbasis AI.
- Klaster hijau terdiri dari konsep *control systems*, *proportional control systems*, dan *matlab*. Meski lebih teknis, klaster ini berkaitan dengan strategi pengendalian dan pengoptimalan model yang mendukung proses *tuning*, terutama untuk memastikan stabilitas dan kinerja algoritma yang digunakan.
- Klaster oranye memuat kata kunci *human* dan *article* yang kemungkinan mencerminkan aspek sosio-teknis, termasuk peran manusia dalam proses *tuning*, pengambilan keputusan, serta evaluasi model.

Secara keseluruhan, visualisasi ini mengungkapkan bahwa tren penelitian terkini tidak hanya berfokus pada peningkatan performa model melalui *hyperparameter tuning*, tetapi juga pada integrasi pembelajaran mesin dalam *software engineering*, eksplorasi model bahasa dalam NLP, serta pengoptimalan model secara otomatis. Temuan ini memberikan wawasan mendalam mengenai bagaimana berbagai pendekatan dan teknologi saling berinteraksi dalam konteks rekayasa perangkat lunak berbasis AI.

Tabel 8 menyajikan tren topik penelitian berdasarkan kata kunci yang sering muncul terkait *hyperparameter tuning*. Terlihat bahwa *Deep Learning* merupakan topik yang paling sering muncul dengan frekuensi 89 kali. Topik ini mulai marak diteliti sejak tahun 2022 (Q1), mencapai median pada tahun 2023, dan tetap relevan hingga Q3 di tahun yang sama.

Tabel 8. Tren topik penelitian berdasarkan kata yang sering muncul

Topik	Frekuensi	Tahun (Q1)	Tahun (Median)	Tahun (Q3)
<i>Deep Learning</i>	89	2022	2023	2023
<i>Language Model</i>	66	2023	2024	2024
<i>Learning Systems</i>	53	2021	2023	2023
<i>Fine Tuning</i>	52	2023	2024	2024
<i>Open Source Software</i>	52	2022	2024	2024

Language Model juga menunjukkan tren yang kuat, dengan frekuensi kemunculan sebanyak 66 kali. Topik ini mulai muncul pada tahun 2023 (Q1), mencapai puncak median pada tahun 2024, dan tetap signifikan hingga Q3 di tahun 2024. Ini mencerminkan meningkatnya perhatian terhadap pengembangan model bahasa, sejalan dengan kemajuan dalam pemrosesan bahasa alami (*Natural Language Processing*).

Learning Systems memiliki frekuensi kemunculan 53 kali, dengan awal tren pada tahun 2021 (Q1), median di tahun 2023, dan Q3 pada tahun 2024. Ini menunjukkan stabilitas topik tersebut dalam rentang waktu tertentu.

Sementara itu, *Fine Tuning* dan *Open Source Software* memiliki frekuensi yang sama, yakni 52 kali. *Fine Tuning* mulai diteliti pada tahun 2023 (Q1), dengan median tahun 2024 dan tetap relevan hingga Q3 di tahun yang sama. Sedangkan *Open Source Software* menunjukkan pola yang sedikit berbeda, dimulai lebih awal pada tahun 2022 (Q1), dengan median pada tahun 2024, serta Q3 di tahun 2024, menandakan bahwa pengembangan perangkat lunak sumber terbuka terus menjadi perhatian peneliti.

Hasil ini mengindikasikan bahwa topik terkait *Deep Learning* dan *Language Model* mendominasi topik penelitian terkini, sementara *Learning Systems*, *Fine Tuning*, dan *Open Source Software* menunjukkan relevansi yang kuat dan konsisten sepanjang periode yang diamati. Secara keseluruhan, tren ini menjelaskan gambaran yang dinamis dari bidang rekayasa perangkat lunak yang terus beradaptasi dengan kemajuan teknologi baru. Integrasi pembelajaran mesin dan kecerdasan buatan menjanjikan untuk merevolusi cara kita mengembangkan, memvalidasi, dan memelihara perangkat lunak. Penelitian di masa depan kemungkinan akan mengeksplorasi aplikasi baru dari teknologi ini, yang mengarah pada peningkatan efisiensi, keandalan, dan kualitas perangkat lunak. Memahami tren ini sangat penting bagi para peneliti dan praktisi untuk tetap berada di garis depan inovasi dan memastikan bahwa mereka siap untuk tantangan dan peluang di masa depan.

Selain itu, distribusi tema penelitian dianalisis berdasarkan dua parameter utama, yaitu *Relevance degree (Centrality)* dan *Development degree (Density)*, seperti yang ditunjukkan dalam Gambar 4. *Relevance degree (Centrality)* menggambarkan pentingnya suatu tema dalam bidang penelitian secara keseluruhan sedangkan *Development degree (Density)* mengindikasikan tingkat pengembangan internal atau kedalaman eksplorasi suatu tema. Berdasarkan peta tematik ini, terdapat empat kuadran utama dengan karakteristik berikut:

a. *Niche Themes* (Kuadran kiri atas)

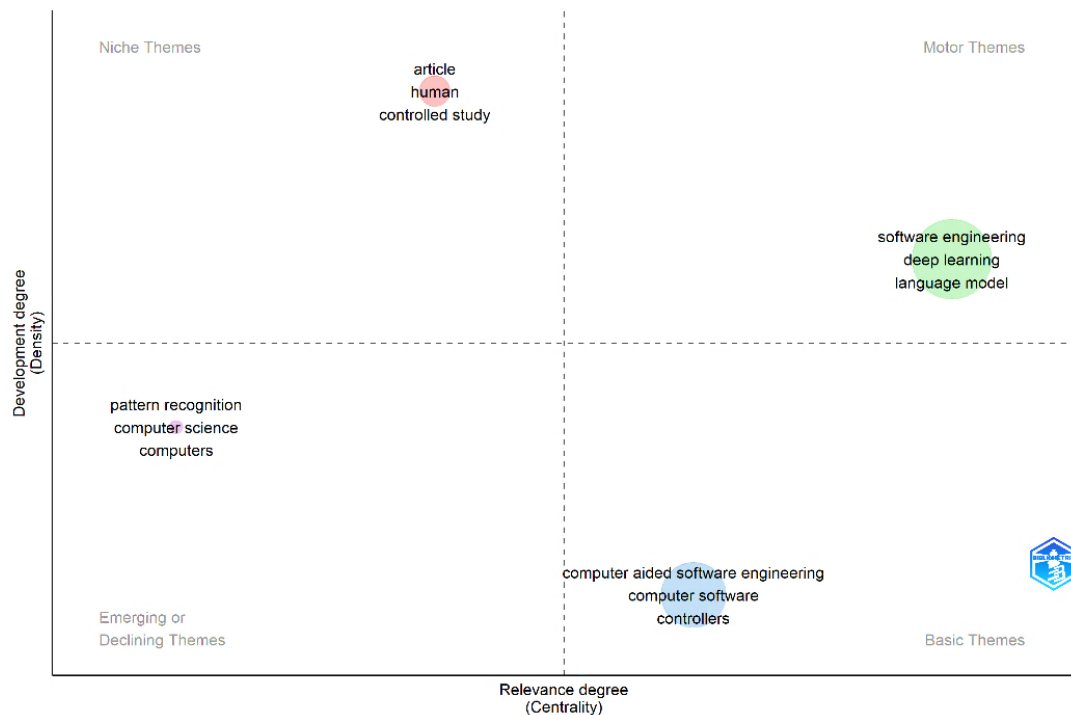
Tema di kuadran ini memiliki pengembangan tinggi namun relevansinya terbatas, menandakan bahwa topik tersebut berkembang secara spesifik tetapi belum memiliki pengaruh besar dalam keseluruhan bidang. *Article*, *human*, dan *controlled study* termasuk dalam kelompok ini, mencerminkan studi-studi yang lebih terfokus pada aspek sosio-teknis, seperti bagaimana manusia berinteraksi dengan teknologi atau eksperimen terkontrol terkait *tuning hyperparameter*. Meskipun tidak dominan, tema ini penting karena membantu memahami dampak sosial dan teknis dari penerapan *AI* dalam *software engineering*.

b. *Emerging or Declining Themes* (Kuadran kiri bawah)

Tema di sini memiliki relevansi dan pengembangan rendah, menunjukkan topik yang sedang meredup atau baru muncul. *Pattern recognition*, *computer science*, dan *computers* masuk dalam kelompok ini. Hal ini menunjukkan bahwa meskipun pengenalan pola dan prinsip dasar ilmu komputer menjadi fondasi penelitian, topik tersebut kini mulai beralih ke aplikasi yang lebih maju seperti pembelajaran mesin dan model bahasa.

c. *Motor Themes* (Kuadran Atas Kanan)

Tema di kuadran ini memiliki tingkat relevansi dan pengembangan yang tinggi, menunjukkan bahwa topik ini menjadi pendorong utama dalam bidang penelitian. *Software engineering*, *deep learning*, dan *language model* muncul sebagai tema utama, menegaskan bahwa riset terkini banyak berfokus pada penerapan model bahasa dan pembelajaran mendalam untuk meningkatkan performa perangkat lunak. Keterkaitan antar konsep ini mencerminkan tren penggunaan *AI*, khususnya *deep learning* dan *language models*, dalam proses *tuning hyperparameter* untuk mengoptimalkan model perangkat lunak.

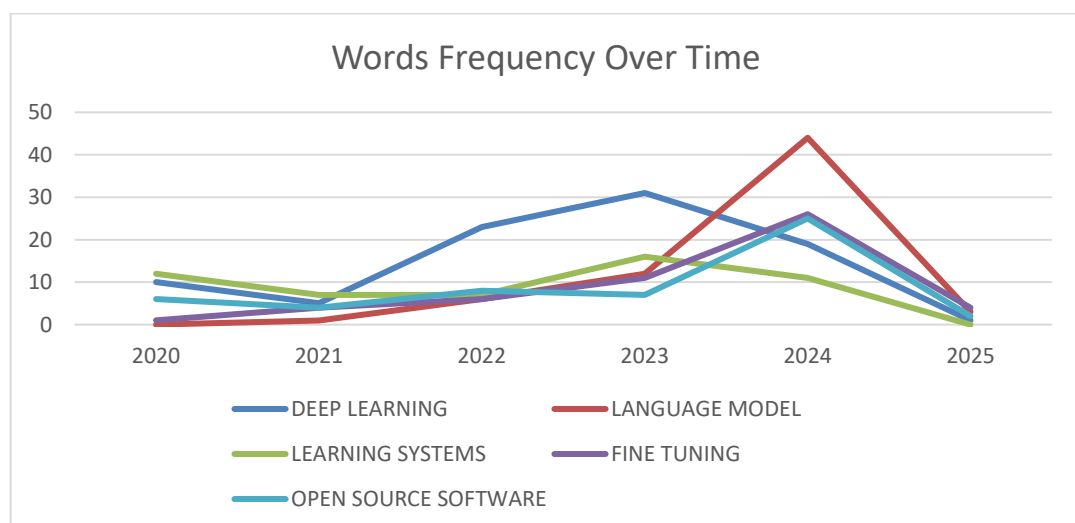


Gambar 4. Distribusi tema penelitian berdasarkan perkembangan dan relevansinya

d. *Basic Themes* (Kuadran Bawah Kanan)

Tema dalam kategori ini memiliki relevansi tinggi tetapi tingkat pengembangannya masih rendah, menunjukkan bahwa topik ini fundamental namun belum dieksplorasi secara mendalam. *Computer-aided software engineering*, *computer software*, dan *controllers* termasuk dalam kategori ini. Tema-tema ini mendasari proses *tuning hyperparameter*, terutama terkait otomatisasi rekayasa perangkat lunak dan kontrol algoritma, meskipun belum mendapatkan eksplorasi yang mendalam.

Peta tematik ini menunjukkan bahwa penelitian terkait *hyperparameter tuning* saat ini berpusat pada penerapan *deep learning* dan *language models*, dengan *software engineering* sebagai inti utama. Sementara itu, *computer-aided software engineering* menjadi fondasi yang mendukung penelitian ini, serta terdapat potensi eksplorasi lebih dalam pada aspek sosio-teknis seperti studi manusia dan eksperimen terkontrol.



Gambar 5. Perkembangan jumlah kata kunci relevan dari tahun 2020 sampai 2025

Hasil tersebut juga diperkuat dengan tren penelitian *hyperparameter tuning* pada penerapan teknologi seperti *deep learning* dan *language models*, seperti yang dijelaskan dalam Gambar 5. Secara keseluruhan, data ini

menunjukkan bahwa *Language Model* dan *Deep Learning* menjadi dua topik paling dominan dalam rentang waktu tersebut, sementara *Learning Systems*, *Fine Tuning*, dan *Open Source Software* tetap memiliki kontribusi penting dalam diskusi ilmiah.

Temuan ini dapat memberikan wawasan mengenai perkembangan berbagai topik yang sedang menjadi perhatian dalam bidang penelitian terkait. Berikut ini merupakan analisis tren utama:

- a. Pertumbuhan Signifikan “*Deep learning*” dan “*Language model*”
Deep Learning menunjukkan pertumbuhan signifikan, dengan frekuensi meningkat tajam mulai tahun 2021, mencapai puncaknya pada tahun 2023, sejalan dengan meningkatnya penggunaan model pembelajaran mendalam dalam optimasi algoritma, namun mengalami penurunan setelahnya.
Language Model memperlihatkan pola pertumbuhan yang lebih dramatis, dengan lonjakan besar antara tahun 2023 dan 2024, mencapai frekuensi tertinggi di antara semua topik, sebelum turun drastis pada tahun 2025. Hal ini mengindikasikan peningkatan perhatian terhadap model bahasa dalam kurun waktu tertentu, yang kemungkinan besar dipengaruhi oleh kemajuan teknologi NLP (*Natural Language Processing*).
- b. Stabilitas “*Learning Systems*” dan “*Fine Tuning*”
Sementara itu, *Learning Systems* memiliki tren yang lebih stabil, meskipun terjadi sedikit peningkatan antara tahun 2022 dan 2023, lalu mengalami penurunan bertahap setelah tahun 2024.
Topik *Fine Tuning* menunjukkan pola pertumbuhan yang relatif konsisten, dengan puncak pada tahun 2024, mencerminkan pentingnya proses penyempurnaan model dalam penelitian terkini.
- c. Kenaikan “*Open Source Software*”
memperlihatkan tren pertumbuhan yang moderat hingga tahun 2024, sebelum akhirnya menurun pada tahun 2025. Hal ini menandakan bahwa meskipun topik ini tetap relevan, fokus penelitian tampaknya mulai bergeser ke area lain.

3.5. RQ5: Apa research gaps dan arah penelitian di masa mendatang terkait *hyperparameter tuning*?

Berdasarkan analisis yang telah dilakukan, terdapat beberapa celah dalam penelitian terkait *hyperparameter tuning* yang memerlukan perhatian lebih lanjut. Salah satu kesenjangan utama adalah keterbatasan dalam generalisasi hasil *tuning* [2]. Sebagian besar studi masih berfokus pada penggunaan *dataset* dan skenario tertentu, sehingga hasil *tuning* belum sepenuhnya dapat diterapkan pada kasus nyata yang lebih kompleks dan dinamis. Selain itu meskipun berbagai metode seperti *grid search*, *random search*, dan *Bayesian Optimization* telah diterapkan, eksplorasi terhadap pendekatan *hyperparameter tuning* berbasis kecerdasan buatan seperti *reinforcement learning*, *meta-learning*, dan algoritma evolusi masih terbatas [65][66][11][67]. Hal ini menunjukkan adanya peluang besar untuk mengembangkan metode otomatisasi *tuning* yang lebih cerdas dan adaptif.

Tantangan lain yang ditemukan adalah kurangnya penelitian terkait optimasi multi-objektif, di mana sebagian besar studi hanya berfokus pada peningkatan akurasi model tanpa mempertimbangkan efisiensi waktu, penggunaan sumber daya komputasi, dan kompleksitas model [68][32]. Padahal dalam konteks rekayasa perangkat lunak, keseimbangan antara akurasi dan efisiensi menjadi aspek krusial.

Arah penelitian di masa depan dapat difokuskan pada beberapa hal berikut:

- a. Mengembangkan metode *hyperparameter tuning* adaptif yang mampu secara dinamis menyesuaikan parameter sesuai perubahan data dan model sehingga meningkatkan efisiensi dan akurasi hasil.
- b. Mengintegrasikan *hyperparameter tuning* dengan pendekatan *meta-learning* dan *reinforcement learning* untuk meningkatkan efisiensi dan akurasi model.
- c. Mengeksplorasi strategi optimasi multi-objektif yang tidak hanya memperhatikan akurasi, tetapi juga efisiensi komputasi dan penggunaan sumber daya dalam satu proses *tuning*.
- d. Membangun kerangka kerja standar serta *benchmark dataset* untuk mengevaluasi metode *tuning* secara konsisten dan memungkinkan perbandingan metode *hyperparameter tuning* secara objektif.
- e. Mengkaji penerapan *hyperparameter tuning* pada sistem perangkat lunak berbasis *cloud* dan *edge computing*, sehingga model dapat dioptimalkan dalam lingkungan yang lebih variatif dan dinamis.

Memahami celah penelitian ini dan mengarahkan fokus ke area penelitian yang relevan, diharapkan studi mendatang mampu berkontribusi dalam pengembangan strategi *hyperparameter tuning* yang lebih efektif, efisien, dan adaptif, sehingga memperkuat integrasi teknologi kecerdasan buatan dalam rekayasa perangkat lunak.

4. KESIMPULAN

Penelitian ini bertujuan untuk menganalisis tren penelitian terkait *hyperparameter tuning* dalam bidang *software engineering* melalui pendekatan *Systematic Literature Review (SLR)* dan *Bibliometric Analysis*. Sebanyak 503 artikel yang diperoleh dari *database Scopus* dan dipublikasikan antara tahun 2020 hingga 2025. Perolehan artikel tersebut dianalisis untuk mengidentifikasi peluang dan arah penelitian di masa depan. Hasil kajian menunjukkan bahwa tren topik berdasarkan kata kunci yang paling sering muncul dalam penelitian terkait

hyperparameter tuning adalah *deep learning*, dengan 89 kemunculan, diikuti oleh tema-tema utama seperti *language model* dan *learning systems*. Tren penelitian juga mengindikasikan peningkatan signifikan dalam eksplorasi teknik tuning otomatis untuk mengatasi kompleksitas model dan tingginya biaya komputasi.

Penelitian terkini berfokus pada penggunaan *hyperparameter tuning* untuk meningkatkan akurasi model dalam mendeteksi *bug* dan kesalahan perangkat lunak, pengembangan metode *hyperparameter tuning* untuk *deep learning*, penerapan *language model* yang canggih, serta sistem pembelajaran otomatis yang adaptif. Hal ini sejalan dengan upaya mengintegrasikan *hyperparameter tuning* dengan teknologi terkini guna meningkatkan performa model *machine learning* dan mengurangi biaya yang terkait. Kombinasi *hyperparameter tuning* dengan *deep learning* digunakan untuk memprediksi kualitas perangkat lunak serta untuk optimalisasi algoritma pencarian *hyperparameter* yang dapat menekan biaya komputasi. Tantangan utama yang dihadapi meliputi kompleksitas model, biaya komputasi yang tinggi, dan keterbatasan generalisasi hasil tuning untuk berbagai skenario aplikasi.

Arah penelitian di masa depan dapat difokuskan pada pengembangan metode *hyperparameter tuning* adaptif, integrasi dengan pendekatan *meta-learning* dan *reinforcement learning* untuk meningkatkan efisiensi dan akurasi. Eksplorasi strategi optimasi multi-objektif juga sangat penting untuk memenuhi berbagai kebutuhan dalam pengembangan model. Pembangunan kerangka kerja standar dan *benchmark dataset* yang komprehensif menjadi langkah penting untuk evaluasi metode *tuning* dapat dilakukan secara konsisten dan objektif. Selain itu, penerapan *hyperparameter tuning* pada sistem perangkat lunak berbasis *cloud* dan *edge computing* menawarkan peluang besar untuk optimasi dalam lingkungan yang lebih variatif dan dinamis.

Kajian ini dapat menjadi referensi yang berguna bagi peneliti dan praktisi dalam mengembangkan strategi tuning yang lebih efektif dan efisien dalam konteks *software engineering*, sehingga dapat mendukung percepatan inovasi dan peningkatan kualitas perangkat lunak.

DAFTAR PUSTAKA

- [1] M. A. Khan *et al.*, "Machine Learning-based Test Case Prioritization using Hyperparameter Optimization," *Proc. - 2024 IEEE/ACM Int. Conf. Autom. Softw. Test, AST 2024*, pp. 125–135, 2024, doi: 10.1145/3644032.3644467.
- [2] J. Wu, S. P. Chen, and X. Y. Liu, "Efficient hyperparameter optimization through model-based reinforcement learning," *Neurocomputing*, vol. 409, pp. 381–393, 2020, doi: 10.1016/j.neucom.2020.06.064.
- [3] N. Arkabaev, E. Rahimov, A. Abdullaev, H. Padmanaban, and V. Salmanov, "Modelling and Analysis of Optimization Algorithms," *J. Ilm. Ilmu Terap. Univ. Jambi*, vol. 9, no. 1, pp. 161–177, Feb. 2025, doi: 10.22437/JIITUI.V9I1.38410.
- [4] N. Agarwal, N. Gupta, and V. Sharma, "Enhancing Fault Detection Accuracy and Reliability in Software Engineering Through Supervised Machine Learning Algorithm," *Int. J. Glob. Res. Innov. Technol.*, vol. 02, no. 03, pp. 108–112, Sep. 2024, doi: 10.62823/IJGRIT/02.03.6892.
- [5] V. Gupta, V. K. Mishra, P. Singhal, and A. Kumar, "An Overview of Supervised Machine Learning Algorithm," *Proc. 2022 11th Int. Conf. Syst. Model. Adv. Res. Trends, SMART 2022*, pp. 87–92, 2022, doi: 10.1109/SMART55829.2022.10047618.
- [6] L. Zaadnoordijk, T. R. Besold, and R. Cusack, "Lessons from infant learning for unsupervised machine learning," *Nat. Mach. Intell.* 2022 46, vol. 4, no. 6, pp. 510–520, Jun. 2022, doi: 10.1038/s42256-022-00488-2.
- [7] C. S. Wickramasinghe, K. Amarasinghe, D. L. Marino, C. Rieger, and M. Manic, "Explainable Unsupervised Machine Learning for Cyber-Physical Systems," *IEEE Access*, vol. 9, pp. 131824–131843, 2021, doi: 10.1109/ACCESS.2021.3112397.
- [8] A. Kanso and K. Patra, "Engineering a Platform for Reinforcement Learning Workloads," *Proc. - 1st Int. Conf. AI Eng. - Softw. Eng. AI, CAIN 2022*, pp. 88–89, 2022, doi: 10.1145/3522664.3528609.
- [9] R. T. Icarte, T. Q. Klassen, R. Valenzano, and S. A. McIlraith, "Reward Machines: Exploiting Reward Function Structure in Reinforcement Learning," *J. Artif. Intell. Res.*, vol. 73, pp. 173–208, Jan. 2022, doi: 10.1613/JAIR.1.12440.
- [10] J. G. Brandão *et al.*, "Optimization of machine learning models for sentiment analysis in social media," *Inf. Sci. (Ny)*, vol. 694, p. 121704, Mar. 2025, doi: 10.1016/J.INS.2024.121704.
- [11] H. Zhang, J. Sun, Y. Wang, J. Shi, and Z. Xu, "Variational Reinforcement Learning for Hyper-Parameter Tuning of Adaptive Evolutionary Algorithm," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 7, no. 5, pp. 1511–1526, 2023, doi: 10.1109/TETCI.2022.3221483.

-
- [12] A. Gjorgjevikj, K. Mishev, L. Antovski, and D. Trajanov, "Requirements Engineering in Machine Learning Projects," *IEEE Access*, vol. 11, pp. 72186–72208, 2023, doi: 10.1109/ACCESS.2023.3294840.
- [13] A. Khan, A. Ali, J. Khan, F. Ullah, and M. Faheem, "Using Permutation-Based Feature Importance for Improved Machine Learning Model Performance at Reduced Costs," *IEEE Access*, 2025, doi: 10.1109/ACCESS.2025.3544625.
- [14] A. Khalid, G. Badshah, N. Ayub, M. Shiraz, and M. Ghouse, "Software Defect Prediction Analysis Using Machine Learning Techniques," *Sustain. 2023, Vol. 15, Page 5517*, vol. 15, no. 6, p. 5517, Mar. 2023, doi: 10.3390/SU15065517.
- [15] S. Hanifi, A. Cammarono, and H. Zare-Behtash, "Advanced hyperparameter optimization of deep learning models for wind power prediction," *Renew. Energy*, vol. 221, p. 119700, Feb. 2024, doi: 10.1016/J.RENENE.2023.119700.
- [16] L. Ferreira, A. Pilastrri, F. Romano, and P. Cortez, "Using supervised and one-class automated machine learning for predictive maintenance," *Appl. Soft Comput.*, vol. 131, p. 109820, Dec. 2022, doi: 10.1016/J.ASOC.2022.109820.
- [17] D. Kusumaningrum, N. Kurniati, and B. Santosa, "Machine learning for predictive maintenance," *Proc. Int. Conf. Ind. Eng. Oper. Manag.*, pp. 2348–2356, 2021, doi: 10.46254/sa02.20210717.
- [18] S. Haldar and L. Fernando Capretz, "Interpretable Software Maintenance and Support Effort Prediction Using Machine Learning," *IEEE/ACM 46th Int. Conf. Softw. Eng. Companion Proc. Interpret.*, 2024, doi: 10.1145/3639478.3643069.
- [19] C. Wan *et al.*, "Keeper: Automated Testing and Fixing of Machine Learning Software," *ACM Trans. Softw. Eng. Methodol.*, vol. 33, no. 7, 2024, doi: 10.1145/3672451.
- [20] R. Walia, "Application of Machine Learning for GUI Test Automation," *2022 28th Int. Conf. Information, Commun. Autom. Technol. ICAT 2022 - Proc.*, 2022, doi: 10.1109/ICAT54566.2022.9811187.
- [21] D. Reddy Seelam, "Automated Test Case Generation using Machine Learning," *Int. J. Adv. Res. Sci. Commun. Technol.*, 2024, doi: 10.48175/IJARSCT-22892.
- [22] Y. Zhang, "New Approaches to Automated Software Testing Based on Artificial Intelligence," *2024 5th Int. Conf. Artif. Intell. Comput. Eng. ICAICE 2024*, pp. 806–810, 2024, doi: 10.1109/ICAICE63571.2024.10863866.
- [23] Q. Wang *et al.*, "Graph Confident Learning for Software Vulnerability Detection," *Eng. Appl. Artif. Intell.*, vol. 133, p. 108296, Jul. 2024, doi: 10.1016/J.ENGAPPAL.2024.108296.
- [24] K. Napier, T. Bhowmik, and S. Wang, "An empirical study of text-based machine learning models for vulnerability detection," *Empir. Softw. Eng.*, vol. 28, no. 2, pp. 1–45, Mar. 2023, doi: 10.1007/S10664-022-10276-6.
- [25] X. Li, Y. Xin, H. Zhu, Y. Yang, and Y. Chen, "Cross-domain vulnerability detection using graph embedding and domain adaptation," *Comput. Secur.*, vol. 125, p. 103017, Feb. 2023, doi: 10.1016/J.COSE.2022.103017.
- [26] T. Narbaev, Ö. Hazir, B. Khamitova, and S. Talgat, "A machine learning study to improve the reliability of project cost estimates," *Int. J. Prod. Res.*, vol. 62, no. 12, pp. 4372–4388, Jun. 2024, doi: 10.1080/00207543.2023.2262051.
- [27] D. Hammann, "Big data and machine learning in cost estimation: An automotive case study," *Int. J. Prod. Econ.*, vol. 269, p. 109137, Mar. 2024, doi: 10.1016/J.IJPE.2023.109137.
- [28] M. O. Sanni-Anibire, R. Mohamad Zin, and S. O. Olatunji, "Developing a preliminary cost estimation model for tall buildings based on machine learning," *Int. J. Manag. Sci. Eng. Manag.*, vol. 16, no. 2, pp. 134–142, Apr. 2021, doi: 10.1080/17509653.2021.1905568.
- [29] K. Filippou, G. Aifantis, G. A. Papakostas, and G. E. Tsekouras, "Structure Learning and Hyperparameter Optimization Using an Automated Machine Learning (AutoML) Pipeline," *Inf.*, vol. 14, no. 4, 2023, doi: 10.3390/info14040232.
- [30] M. H. Rizky, M. R. Faisal, I. Budiman, D. Kartini, and F. Abadi, "Effect of Hyperparameter Tuning Using Random Search on Tree-Based Classification Algorithm for Software Defect Prediction," *IJCCS (Indonesian J. Comput. Cybern. Syst.)*, vol. 18, no. 1, p. 95, 2024, doi: 10.22146/ijccs.90437.
- [31] L. Liao, H. Li, W. Shang, and L. Ma, "An Empirical Study of the Impact of Hyperparameter Tuning and Model Optimization on the Performance Properties of Deep Neural Networks," *ACM Trans. Softw. Eng. Methodol.*, vol. 31, no. 3, Jul. 2022, doi: 10.1145/3506695.

-
- [32] P. Chen, J. Gong, and T. Chen, "Accuracy Can Lie: On the Impact of Surrogate Model in Configuration Tuning," *IEEE Trans. Softw. Eng.*, vol. 51, no. 2, pp. 548–580, 2025, doi: 10.1109/TSE.2025.3525955.
 - [33] G. Tripathy and A. Sharaff, "Hyperparameter elegance: fine-tuning text analysis with enhanced genetic algorithm hyperparameter landscape," *Knowl. Inf. Syst.*, vol. 66, no. 11, pp. 6761–6783, Nov. 2024, doi: 10.1007/S10115-024-02202-7.
 - [34] M. A. Serhani, H. Ismail, H. T. El-Kassabi, and H. Al Breiki, "Enhancing arrhythmia prediction through an adaptive deep reinforcement learning framework for ECG signal analysis," *Biomed. Signal Process. Control*, vol. 101, p. 107155, Mar. 2025, doi: 10.1016/J.BSPC.2024.107155.
 - [35] L. Liao, H. Li, W. Shang, and L. Ma, "An Empirical Study of the Impact of Hyperparameter Tuning and Model Optimization on the Performance Properties of Deep Neural Networks," *ACM Trans. Softw. Eng. Methodol.*, vol. 31, no. 3, 2022, doi: 10.1145/3506695.
 - [36] M. Ali, T. Mazhar, A. Al-Rasheed, T. Shahzad, Y. Y. Ghadi, and M. A. Khan, "Enhancing software defect prediction: a framework with improved feature selection and ensemble machine learning," *PeerJ Comput. Sci.*, vol. 10, p. e1860, Feb. 2024, doi: 10.7717/PEERJ-CS.1860/SUPP-13.
 - [37] W. Nugraha and A. Sasongko, "Hyperparameter Tuning on Classification Algorithm with Grid Search," *Sist. J. Sist. Inf.*, vol. 11, no. 2, pp. 391–401, May 2022, doi: 10.32520/STMSI.V11I2.1750.
 - [38] A. T. Nair and M. Arivazhagan, "Industrial activated sludge model identification using hyperparameter-tuned metaheuristics," *Swarm Evol. Comput.*, vol. 91, p. 101733, Dec. 2024, doi: 10.1016/J.SWEVO.2024.101733.
 - [39] P. Buczak, A. Groll, M. Pauly, J. Rehof, · Daniel Horn, and D. Horn, "Using sequential statistical tests for efficient hyperparameter tuning," vol. 108, pp. 441–460, 2024, doi: 10.1007/s10182-024-00495-1.
 - [40] G. Tripathy and A. Sharaff, "Hyperparameter elegance: fine-tuning text analysis with enhanced genetic algorithm hyperparameter landscape," *Knowl. Inf. Syst.*, vol. 66, no. 11, pp. 6761–6783, Nov. 2024, doi: 10.1007/S10115-024-02202-7/METRICS.
 - [41] M. Hassanali, M. Soltanaghaei, T. Javdani Gandomani, and F. Zamani Boroujeni, "Software development effort estimation using boosting algorithms and automatic tuning of hyperparameters with Optuna," *J. Softw. Evol. Process*, vol. 36, no. 9, p. e2665, Sep. 2024, doi: 10.1002/SMR.2665.
 - [42] J. Resti, H. D. Purnomo, T. Gonsalves, E. Mailoa, Y. Santoso, and M. R. Pribadi, "Metaheuristics Approach for Hyperparameter Tuning of Convolutional Neural Network," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 8, no. 3, pp. 340–345, Jun. 2024, doi: 10.29207/RESTI.V8I3.5730.
 - [43] Y. A. Ali, E. M. Awwad, M. Al-Razgan, and A. Maarouf, "Hyperparameter Search for Machine Learning Algorithms for Optimizing the Computational Complexity," *Processes*, vol. 11, no. 2, p. 349, Feb. 2023, doi: 10.3390/pr11020349.
 - [44] Y. Rimal, N. Sharma, and A. Alsadoon, "The accuracy of machine learning models relies on hyperparameter tuning: student result classification using random forest, randomized search, grid search, bayesian, genetic, and optuna algorithms," *Multimed. Tools Appl.*, vol. 83, no. 30, pp. 74349–74364, Sep. 2024, doi: 10.1007/S11042-024-18426-2/METRICS.
 - [45] J. Singh, J. K. Sandhu, and Y. Kumar, "Enhanced Metaheuristic Based Hyper-Parameters Tuning for Learning Models," *Proc. Int. Conf. Contemp. Comput. Informatics, IC3I 2023*, pp. 374–378, 2023, doi: 10.1109/IC3I59117.2023.10398069.
 - [46] A. R. Sampaio, I. Beschastnikh, D. Maier, D. Bourne, and V. Sundaresen, "Auto-tuning elastic applications in production," *Proc. - Int. Conf. Softw. Eng.*, pp. 355–367, 2023, doi: 10.1109/ICSE-SEIP58684.2023.00038.
 - [47] O. Kurniawan, Y. Alkhalifi, L. A. Fitriana, M. R. Firdaus, A. N. Rais, and S. W. Hadi, "Hyperparameter Tuning Optimization on Machine Learning Models to Predict Software Defects," *ICAISD 2024 - Int. Conf. Adv. Inf. Sci. Dev. AI Invest. Sustain. Dev. Hum. Living Digit. Proc.*, pp. 35–40, 2024, doi: 10.1109/ICAISD63055.2024.10895942.
 - [48] E. Christou, A. Parmaxi, and P. Zaphiris, "A systematic exploration of scoping and mapping literature reviews," *Univers. Access Inf. Soc.*, 2024, doi: 10.1007/s10209-024-01120-3.
 - [49] G. G. Giwangkoro and Y. S. Nugroho, "Unveiling Research Trends in Stack Overflow: A Comprehensive Analysis of General Discussion Theme," *2024 Int. Conf. Smart Comput. IoT Mach. Learn. SIML 2024*, pp. 130–136, 2024, doi: 10.1109/SIML61815.2024.10578280.
 - [50] A. Satar, "Examining the Determinants of Sustainable Competitive Advantage: A Systematic Literature

- Review,” *WSEAS Trans. Comput. Res.*, vol. 12, pp. 112–122, 2024, doi: 10.37394/232018.2024.12.11.
- [51] J. Ali, A. Jusoh, N. Idris, A. G. Airij, and R. Chandio, “Wearable Devices in Healthcare Services. Bibliometrix Analysis by using R Package,” *Int. J. online Biomed. Eng.*, vol. 18, no. 8, pp. 61–86, 2022, doi: 10.3991/IJOE.V18I08.31785.
- [52] S. Büyükkidik, “A Bibliometric Analysis: A Tutorial for the Bibliometrix Package in R Using IRT Literature,” *J. Meas. Eval. Educ. Psychol.*, vol. 13, no. 3, pp. 164–193, Sep. 2022, doi: 10.21031/EPOD.1069307.
- [53] T. Lutellier, H. V. Pham, L. Pang, Y. Li, M. Wei, and L. Tan, “CoCoNuT: Combining context-aware neural translation models using ensemble for program repair,” in *ISSTA 2020 - Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2020, pp. 101–114. doi: 10.1145/3395363.3397369.
- [54] J. Choe, S. J. Oh, S. Lee, S. Chun, Z. Akata, and H. Shim, “Evaluating weakly supervised object localization methods right,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3130–3139. doi: 10.1109/CVPR42600.2020.00320.
- [55] J.-H. Luo and J. Wu, “AutoPruner: An end-to-end trainable filter pruning method for efficient deep model inference,” *Pattern Recognit.*, vol. 107, 2020, doi: 10.1016/j.patcog.2020.107461.
- [56] N. D. Q. Bui, Y. Yu, and L. Jiang, “InferCode: Self-supervised learning of code representations by predicting subtrees,” in *Proceedings - International Conference on Software Engineering*, 2021, pp. 1186–1197. doi: 10.1109/ICSE43902.2021.00109.
- [57] M. M. Rahman, Y. Watanobe, and K. Nakamura, “A bidirectional lstm language model for code evaluation and repair,” *Symmetry (Basel)*, vol. 13, no. 2, pp. 1–15, 2021, doi: 10.3390/sym13020247.
- [58] Y. Liu, N. Zhu, K. Li, M. Li, J. Zheng, and K. Li, “An angle dominance criterion for evolutionary many-objective optimization,” *Inf. Sci. (Ny)*, vol. 509, pp. 376–399, 2020, doi: 10.1016/j.ins.2018.12.078.
- [59] C.-L. Zhang, Y.-H. Cao, and J. Wu, “Rethinking the route towards weakly supervised object localization,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13457–13466. doi: 10.1109/CVPR42600.2020.01347.
- [60] Z. Yang, J. Shi, J. He, and D. Lo, “Natural Attack for Pre-trained Models of Code,” in *Proceedings - International Conference on Software Engineering*, 2022, pp. 1482–1493. doi: 10.1145/3510003.3510146.
- [61] N. P. Lawrence, M. G. Forbes, P. D. Loewen, D. G. McClement, J. U. Backström, and R. B. Gopaluni, “Deep reinforcement learning with shallow controllers: An experimental application to PID tuning,” *Control Eng. Pract.*, vol. 121, 2022, doi: 10.1016/j.conengprac.2021.105046.
- [62] Z. Zou, S. Lei, T. Shi, Z. Shi, and J. Ye, “Deep Adversarial Decomposition: A Unified Framework for Separating Superimposed Images,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12803–12813. doi: 10.1109/CVPR42600.2020.01282.
- [63] I. Ashkenazi and O. Olsha, “Inappropriate Journal Authorship, Disputes, Plagiarism, and Mistrust in the Institution: Different Beasts ... Same Problem,” *Rambam Maimonides Med. J.*, vol. 14, no. 4, pp. 14–16, 2023, doi: 10.5041/RMMJ.10514.
- [64] J. A. Teixeira da Silva, “The Centrality of Trust in Academic Publishing Lies with the Corresponding Author,” *Rambam Maimonides Med. J.*, vol. 15, no. 2, pp. 1–2, 2024, doi: 10.5041/rmmj.10525.
- [65] X. Liu, J. Wu, and S. Chen, “Efficient Hyperparameters optimization through Model-based Reinforcement Learning and Meta-Learning,” *Proc. - 2020 IEEE 22nd Int. Conf. High Perform. Comput. Commun. IEEE 18th Int. Conf. Smart City IEEE 6th Int. Conf. Data Sci. Syst. HPCC-SmartCity-DSS 2020*, pp. 1036–1041, 2020, doi: 10.1109/HPCC-SmartCity-DSS50907.2020.00139.
- [66] A. H. Victoria and G. Maragatham, “Automatic tuning of hyperparameters using Bayesian optimization,” *Evol. Syst.*, vol. 12, no. 1, pp. 217–223, 2021, doi: 10.1007/s12530-020-09345-2.
- [67] P. S. N. Mindom, A. Nikanjam, and F. Khomh, “Harnessing pre-trained generalist agents for software engineering tasks,” *Empir. Softw. Eng.*, vol. 30, no. 1, 2025, doi: 10.1007/s10664-024-10597-8.
- [68] T. Elansari, M. Ouanan, and H. Bourray, “Mixed Radial Basis Function Neural Network Training Using Genetic Algorithm,” *Neural Process. Lett.*, vol. 55, no. 8, pp. 10569–10587, 2023, doi: 10.1007/s11063-023-11339-5.