

SOFTWARE DEFECT PREDICTION TRENDS: A BIBLIOMETRIC ANALYSIS OF MACHINE AND DEEP LEARNING

Harsih Rianto^{1*}; Omar Pahlevi²; Desmulyati³; Amrin⁴; Ade Surya Budiman³; Budi Supriyadi²

Information Technology¹

Information Systems²

Informatics³

Computer Technology⁴

Universitas Bina Sarana Informatika, Jakarta, Indonesia^{1,2,3,4}

<https://bsi.ac.id>^{1,2,3,4}

harsih.hhr@bsi.ac.id*, omar.opi@bsi.ac.id, desmulyati.dmy@bsi.ac.id, amrin.ain@bsi.ac.id,

ade.aum@bsi.ac.id, budi.bds@bsi.ac.id

(*) Corresponding Author

(Responsible for the Quality of Paper Content)



The creation is distributed under the Creative Commons Attribution-NonCommercial 4.0 International License.

Abstract— This study provides a comprehensive bibliometric mapping of global research trends and emerging frontiers in Software Defect Prediction (SDP), emphasizing the integration of machine learning (ML) and deep learning (DL) approaches. Unlike previous bibliometric surveys that focused narrowly on metric-based or short-term analyses, this work offers a broader and more integrated perspective on the intellectual evolution, collaboration patterns, and thematic directions in SDP research. Using data retrieved from the Scopus database and analyzed through Bibliometrix and VOSviewer, the study systematically applied the PRISMA protocol to ensure transparency and replicability. A total of 1,549 publications were examined, revealing a steady increase in scientific output dominated by China, India, and the United States. Thematic and keyword analyses identified five core clusters that trace the paradigm shift from traditional statistical models to advanced ML- and DL-driven predictive frameworks. Emerging topics such as transfer learning, cross-project prediction, and explainable AI (XAI) were identified as promising frontiers shaping the next phase of software quality prediction research. Beyond mapping academic progress, this study contributes strategic insights for researchers seeking to identify research gaps, industry practitioners developing intelligent defect prediction tools, and policymakers designing AI-driven software quality initiatives.

Keywords: Bibliometrik, Deep Learning, Machine Learning, Software Defect Prediction, Transfer Learning.

Intisari— Penelitian ini menyajikan pemetaan bibliometrik komprehensif mengenai tren penelitian global dan arah pengembangan terbaru (emerging frontiers) dalam bidang Software Defect Prediction (SDP), dengan penekanan pada integrasi pendekatan machine learning (ML) dan deep learning (DL). Berbeda dengan kajian bibliometrik sebelumnya yang berfokus sempit pada analisis berbasis metrik atau rentang waktu yang terbatas, penelitian ini menawarkan perspektif yang lebih luas dan terintegrasi terhadap evolusi intelektual, pola kolaborasi, serta arah tematik dalam riset SDP. Data penelitian diambil dari basis data Scopus dan dianalisis menggunakan perangkat Bibliometrix dan VOSviewer, dengan penerapan protokol PRISMA secara sistematis untuk menjamin transparansi dan replikabilitas. Sebanyak 1.549 publikasi dianalisis, menunjukkan peningkatan keluaran ilmiah yang stabil dengan dominasi kontribusi dari Tiongkok, India, dan Amerika Serikat. Analisis tematik dan kata kunci mengidentifikasi lima kluster inti yang meluruskan pergeseran paradigma dari model statistik tradisional menuju kerangka prediktif berbasis ML dan DL yang lebih canggih. Topik-topik baru seperti transfer learning, cross-project prediction, dan explainable AI (XAI) diidentifikasi sebagai area penelitian yang menjanjikan dalam membentuk fase berikutnya dari riset prediksi kualitas perangkat lunak. Di luar pemetaan perkembangan akademik, studi ini memberikan wawasan strategis bagi peneliti dalam mengidentifikasi kesenjangan penelitian, bagi praktisi industri dalam

mengembangkan alat prediksi cacat yang cerdas, serta bagi pembuat kebijakan dalam merancang inisiatif peningkatan kualitas perangkat lunak berbasis kecerdasan buatan.

Kata Kunci: Bibliometrik, Pembelajaran Mendalam, Pembelajaran Mesin, Prediksi Cacat Perangkat Lunak, Transfer Pembelajaran

INTRODUCTION

Ensuring software quality remains a persistent challenge and a cornerstone of modern software engineering. Software defects not only compromise reliability but also increase maintenance costs, reduce user satisfaction, and in critical cases, may lead to severe system failures. To address these challenges, the research community has extensively explored Software Defect Prediction (SDP) a strategic approach that identifies modules likely to contain defects prior to testing or deployment. By localizing high-risk modules early, SDP allows development teams to allocate resources more efficiently and improve the overall quality assurance process [1].

Over the years, the evolution of SDP research has mirrored the broader technological transformation in software engineering. Early studies relied on conventional statistical and metric-based techniques, but the field has since transitioned toward machine learning (ML) and deep learning (DL) approaches capable of modeling complex, nonlinear relationships within software data. Methods such as support vector machines, random forests, and convolutional neural networks have demonstrated substantial improvements in prediction accuracy [2], [3]. At the same time, ensemble learning, transfer learning, and natural language processing (NLP) have further enhanced the adaptability and generalization of predictive models across different projects and contexts [4], [5].

These methodological advancements signify more than technical progress they represent a paradigm shift from reactive defect detection to proactive, intelligent prediction. Recent studies have also integrated multiple data sources (e.g., abstract syntax trees, software metrics, dependency graphs) to capture richer software representations, while explainable AI (XAI) frameworks have begun to improve model transparency and interpretability [6], [7], [8]. However, despite these significant advancements, there is still limited bibliometric evidence that systematically maps how this intellectual evolution has unfolded globally.

Previous reviews of SDP have typically focused on specific algorithms, datasets, or evaluation techniques, offering valuable insights but lacking a comprehensive synthesis of the global

research landscape. In particular, prior bibliometric studies have either concentrated on narrow time spans or neglected to contextualize the recent convergence of ML and DL paradigms within SDP. This leaves open questions about the broader structural patterns such as collaboration networks, dominant research clusters, and emerging topics like transfer learning and explainable AI that characterize the evolution of this field [9], [10], [11], [12].

To fill this gap, the present study provides a holistic bibliometric mapping of global research trends and emerging frontiers in Software Defect Prediction (SDP). By employing Bibliometrix and VOSviewer on Scopus indexed publications, this work systematically visualizes intellectual structures, thematic evolutions, and collaborative relationships across countries, institutions, and authors. In doing so, it extends existing literature by integrating machine learning and deep learning perspectives into bibliometric analysis, offering an up-to-date understanding of how predictive modeling research in software engineering has evolved.

Beyond academic synthesis, this study also provides practical implications for various stakeholders. For researchers, it highlights unexplored clusters and potential research gaps; for industry practitioners, it offers insights that can guide the design of intelligent and interpretable defect prediction tools; and for policymakers, it informs strategic funding priorities toward AI-driven software quality improvement initiatives. Together, these contributions position this bibliometric mapping not merely as a descriptive survey, but as a strategic framework for navigating and shaping the future of SDP research.

MATERIALS AND METHODS

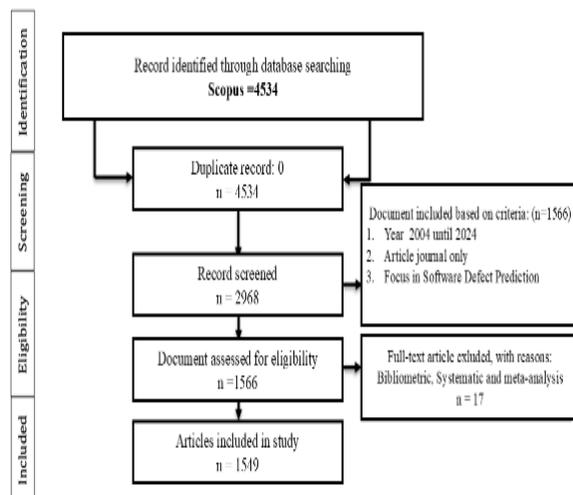
This study employed a bibliometric design following the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) framework to ensure methodological transparency, traceability, and replicability. The chosen time frame (2004–2024) was selected to capture two decades of research evolution, encompassing the transition from traditional metric-based approaches to the adoption of machine learning

(ML) and deep learning (DL) paradigms in Software Defect Prediction (SDP).

Bibliometric data were retrieved exclusively from the Scopus database using a refined search query containing key terms such as “software defect prediction,” “bug prediction,” “fault proneness,” and “defect prediction model.” This database was chosen for its broad coverage and detailed metadata suitable for citation and keyword analyses. The retrieved dataset was then processed using two complementary bibliometric tools: Bibliometrix (R package) for quantitative statistical analysis and VOSviewer for network visualization of co-authorship, co-citation, and keyword co-occurrence patterns. The use of both tools ensured analytical depth and visual interpretability.

The bibliometric process consisted of four systematic stages aligned with the PRISMA flow: identification, screening, eligibility, and inclusion. In the identification stage, 4,534 documents were retrieved from Scopus using the predefined search query. During the screening stage, the dataset was filtered by publication year (2004–2024), document type (journal articles), and topical relevance to software defect prediction. After applying these criteria, 2,968 documents were retained.

The eligibility stage involved manual review of abstracts and full texts to ensure alignment with the core topic and informational quality. At this stage, 1,566 publications passed the initial evaluation, while 17 documents were excluded because they represented systematic reviews, meta-analyses, or previous bibliometric studies—thus preventing duplication of synthesized content and ensuring the originality of this mapping.



Source: (Research Results, 2025)

Figure 1. PRISMA flow diagram adapted from previous bibliometric studies [10], [13], [14], [15]

Finally, 1,549 documents met all inclusion criteria and formed the final dataset for analysis. These publications represent primary empirical research in the field of SDP and were analyzed to examine publication trends, authorship collaborations, source distributions, and thematic developments over time. The PRISMA flow used in this study was adapted from several prior bibliometric analyses [10], [13], [14], [15] to maintain consistency with established practices and ensure clarity in reporting.

Overall, this methodological framework ensured that the dataset was comprehensive, the selection process transparent, and the analytical results both reproducible and reflective of global research trends in Software Defect Prediction (SDP).

RESULTS AND DISCUSSION

Document Information

A total of 1,549 documents related to Software Defect Prediction (SDP) were collected from the Scopus database, spanning the period 2004–2024 and distributed across 432 publication sources, including journals and proceedings. This wide distribution highlights the multidisciplinary nature of SDP, covering both computer science and software engineering domains. The inclusion of multiple outlets reflects the sustained global interest in software reliability, data-driven testing, and predictive modeling (Figure 2).



Source: (Research Results, 2025)

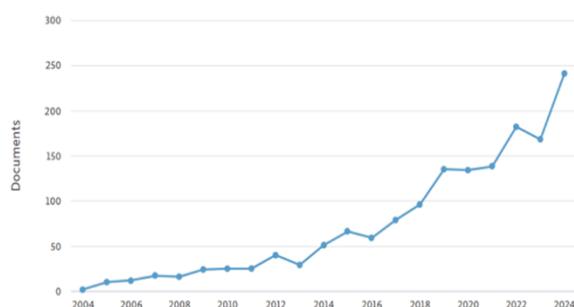
Figure 2. Document Information

The diversity of publication sources also illustrates how SDP has transitioned from a purely software-engineering concern toward a broader field integrating artificial intelligence and data science techniques. This evolution is further discussed in the following section on publication growth trends.

Publication Growth Trends

The annual publication output in Software Defect Prediction (SDP) research demonstrates a consistent upward trajectory throughout the past two decades. As illustrated in Figure 3, the number of publications has increased substantially,

reflecting both the expansion of research interest and the technological evolution of predictive software analytics. Between 2004 and 2010, growth remained modest, primarily consisting of early studies employing statistical or metric-based models. However, after 2015, a sharp increase in publication frequency can be observed, coinciding with the emergence of machine learning (ML) and deep learning (DL) frameworks such as TensorFlow and PyTorch, which significantly lowered the technical barriers to implementing AI models in software defect prediction [16].



Source: (Research Results, 2025)
Figure 3. Publication Growth Trends

This upward trend indicates a structural transformation in the research landscape, marking a paradigm shift from conventional metric-based prediction approaches toward intelligent, data-driven frameworks. The proliferation of open-source datasets most notably NASA's Metrics Data Program (MDP) and the PROMISE repository has further accelerated experimentation, reproducibility, and benchmarking among researchers. Additionally, the rise of cloud-based computational environments and accessible ML libraries has democratized predictive modeling, enabling broader participation from both academia and industry.

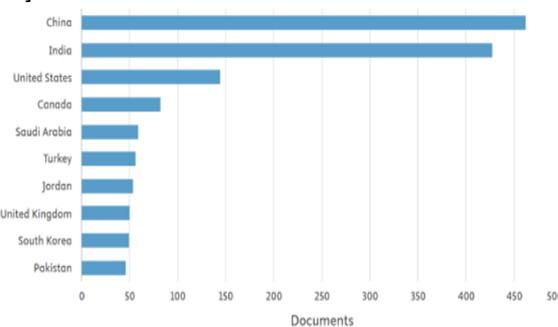
Overall, Figure 3 underscores the exponential trajectory of publication growth in SDP, confirming that this field has evolved into a critical subdomain of AI-driven software quality research. The consistent increase in scholarly output not only reflects growing academic curiosity but also responds to the industrial need for automated, intelligent defect detection and proactive software maintenance solutions.

Country Contribution Distribution

The global distribution of Software Defect Prediction (SDP) publications by country or region is depicted in Figure 4. The visualization clearly shows a concentration of research productivity in a few key nations. China leads with 462 documents

(29.8%), followed by India with 427 (27.6%), and the United States with 144 (9.3%). Collectively, these three countries account for nearly half of the total global research output on SDP, highlighting their central role in shaping the intellectual and technological landscape of the field.

The prominence of China, as seen in Figure 4, reflects long-term national initiatives that prioritize innovation in artificial intelligence and software engineering. Programs such as Made in China 2025 and the New Generation Artificial Intelligence Development Plan (2017) have significantly accelerated investment in AI-driven software quality research, promoting collaboration between academic institutions and the technology industry [17].



Source: (Research Results, 2025)
Figure 4. Country Contribution Distribution

Similarly, India's strong research output can be attributed to nationwide digital transformation policies such as Digital India and the proliferation of open-source development ecosystems that encourage data sharing, reproducibility, and collaboration between universities and software companies [18]. These initiatives have established India as a key hub for empirical research on AI-based software analytics and defect prediction.

In contrast, the United States, while ranking third in publication volume, continues to produce some of the most highly cited and methodologically advanced studies in the domain. This leadership is supported by strong partnerships between top-tier universities and industry leaders such as Microsoft, IBM, and Google, which have contributed to the development of open-source datasets and benchmark frameworks for predictive modeling [1].

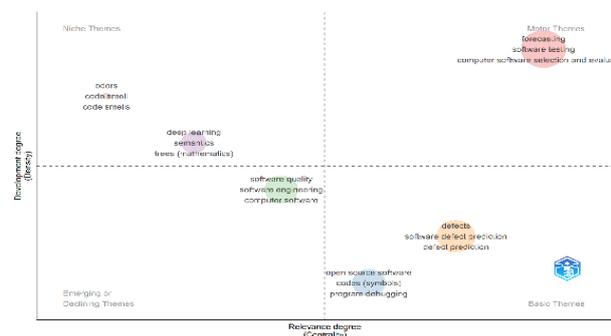
Overall, Figure 4 illustrates not only quantitative differences in research output but also the strategic alignment of national research priorities. These geographic patterns emphasize that policy frameworks, research funding models, and data infrastructure investments are decisive factors driving the global evolution of SDP research. The observed clustering of contributions thus mirrors both economic capacity and technological

Moreover, the visual distribution in Figure 6 underscores the convergence between Software Engineering and Artificial Intelligence (AI). The increasing adoption of machine learning (ML), deep learning (DL), and explainable AI (XAI) methods demonstrates how SDP research is being reframed through the lens of AI-driven analytics. Studies such as [19], [20] have highlighted that hybrid-ensemble and explainable models not only enhance prediction accuracy but also improve interpretability and trustworthiness key concerns in industrial AI deployment.

This interconnection between technical modeling and organizational impact confirms that SDP is evolving from a specialized software quality task into a strategic, interdisciplinary research domain. As visualized in Figure 6, the field now serves as a bridge linking software analytics, artificial intelligence, and decision science, illustrating a broader transformation toward intelligent, adaptive, and explainable software development ecosystems.

Thematic Map: Position and Development of Main Themes

The conceptual structure of Software Defect Prediction (SDP) research is depicted in Figure 7, which displays the thematic map derived from co-word analysis in Bibliometrix. The horizontal axis represents relevance (centrality), while the vertical axis represents development (density). This visualization groups research topics into four quadrants Motor Themes, Basic Themes, Niche Themes, and Emerging or Declining Themes indicating their maturity and strategic importance within the global SDP research landscape.



Source: (Research Results, 2025)

Figure 7. Thematic Map

As shown in Figure 7, the Motor Themes quadrant (upper right) contains topics such as forecasting, software testing, and computer software selection and evaluation. These themes exhibit both high density and high centrality, confirming that software testing and performance

evaluation remain the intellectual and methodological core of SDP studies. They correspond closely to Cluster 5 in Table 1, whose focus on core themes and predictive model generalization emphasizes performance evaluation, cross-project prediction, and transfer learning.

The Basic Themes quadrant (lower right) encompasses defect prediction, software defect prediction, and open-source software. These topics show strong relevance but moderate development, representing the foundational layer of SDP research where predictive frameworks and benchmark datasets are continuously refined. Their content aligns with Cluster 2, which centers on machine learning and optimization approaches such as random forest, support vector machine, and genetic algorithms used to enhance defect detection accuracy.

In the Niche Themes quadrant (upper left), terms such as code smell, code odors, and semantics appear. Although these topics are limited in centrality, they possess high density, signifying focused scholarly communities exploring code quality metrics, semantic analysis, and software maintainability. These correspond conceptually to Cluster 1, which in Table 1 highlights classical approaches in software engineering for example, fault prediction, object-oriented metrics, and regression analysis. Together they represent mature but specialized research niches grounded in traditional metric-based prediction.

The Emerging or Declining Themes quadrant (lower left) includes developing topics that are gaining relevance but remain less mature. In this mapping, deep learning occupies the transition area between emerging and niche themes, reflecting its steady move toward mainstream adoption in SDP. This theme is directly connected to Cluster 4 in Table 1, which represents cutting-edge technologies based on deep learning and AI including convolutional neural networks and long short-term memory models. The presence of deep learning in this transitional position confirms the ongoing shift of SDP toward advanced AI-driven paradigms.

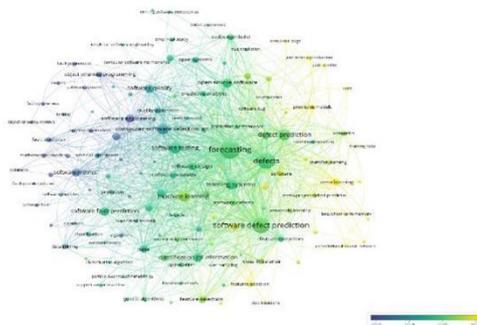
In addition, Cluster 3 in Table 1 focused on empirical research and open-source software is positioned conceptually near the boundary between basic and niche themes, consistent with Figure 7's placement of open-source software and software engineering. This reflects the growing reliance on mining real-world repositories and empirical validation in contemporary SDP research.

The thematic progression in Figure 7, when interpreted together with Table 1, demonstrates a clear evolutionary trajectory of SDP research. Early studies (Cluster 1) established the foundation

through metric-based and regression approaches; the next phase (Cluster 2 and 3) integrated machine-learning and empirical analyses; and the current frontier (Cluster 4 and 5) emphasizes deep learning, transfer learning, and performance generalization across projects. This progression evidences the field's transformation from traditional statistical modeling to adaptive, explainable, and AI-oriented predictive frameworks signifying that Software Defect Prediction has matured into a multidisciplinary bridge linking software engineering, data science, and artificial intelligence.

Keyword Mapping and Clustering Analysis

The intellectual structure of Software Defect Prediction (SDP) research is visualized in Figure 8, which presents the keyword co-occurrence network generated using VOSviewer. The accompanying Table 1 summarizes the five principal clusters identified through this analysis, describing each cluster's main focus, representative keywords, and research meaning. Together, Figure 8 and Table 1 provide a comprehensive overview of how SDP research themes are organized, interconnected, and evolving.



Source: (Research Results, 2025)
 Figure 8. visualization VOS Viewer

As shown in Figure 8, five color-coded clusters form a dense, interrelated network:

1. Cluster 1 (Red - Classical approaches in software engineering). This cluster includes keywords such as fault prediction, object-oriented metrics, and regression analysis, which highlight the origins of SDP in traditional software-engineering metrics and statistical modeling. It represents the methodological foundation upon which modern predictive frameworks were later developed.
2. Cluster 2 (Green - Machine learning and optimization approaches). Centered around terms like machine learning, random forest, support vector machine, genetic algorithms, and class imbalance, this cluster captures the

surge of research that applied ML algorithms to overcome the limitations of classical metric-based prediction. It reflects the consolidation of data-driven approaches that now dominate mainstream SDP studies.

3. Cluster 3 (Blue - Empirical research and open-source software). Comprising terms such as open-source software, mining software repositories, and empirical studies, this cluster emphasizes the importance of real-world datasets and repository mining for validating predictive models. Its position between Clusters 1 and 2 in Figure 8 indicates its bridging role—connecting traditional software-engineering practices with modern data-centric experimentation.
4. Cluster 4 (Yellow - Cutting-edge technologies based on deep learning and AI). This cluster, containing keywords like deep learning, convolutional neural networks, and long short-term memory, represents the newest technological frontier in SDP. The terms' placement at the periphery of Figure 8 illustrates their recent but rapidly growing prominence, marking the community's shift toward advanced, AI-driven prediction models that learn hierarchical representations of software code and defect patterns.
5. Cluster 5 (Purple - Core themes and predictive-model generalization). Dominated by defect prediction, prediction performance, cross-project prediction, and transfer learning, this cluster serves as the central hub of the network. Its strong connectivity with all other clusters underscores its integrative role—linking the theoretical roots (Cluster 1) and methodological advances (Clusters 2 and 4) with practical implementation and validation (Cluster 3).

The network topology visualized in Figure 8 reveals close relationships among these clusters, forming a continuous knowledge flow from classical to AI-driven paradigms. The proximity between Clusters 2 and 4 signifies that improvements in machine-learning optimization directly contribute to deep-learning-based defect prediction. Meanwhile, the connection between Clusters 3 and 5 highlights how empirical repository mining and open-source data increasingly support model evaluation, transfer learning, and cross-project generalization.

When interpreted together, Figure 8 and Table 1 align strongly with the thematic quadrants in Figure 7. The motor themes identified earlier (forecasting, software testing, and evaluation)

correspond to Cluster 5, the intellectual core of SDP, while deep learning and transfer learning from Cluster 4 represent emerging yet fast-developing research frontiers. Conversely, Cluster 1 overlaps with niche and foundational themes, capturing classical statistical perspectives that continue to inform baseline comparisons.

These linkages reveal a coherent research evolution from metric-based fault prediction (Cluster 1) to machine-learning optimization (Cluster 2), to empirical validation (Cluster 3), and finally toward deep and transfer learning (Clusters 4 and 5). The vibrant inter-cluster connectivity visualized in Figure 8 thus confirms that SDP research is no longer fragmented but has converged into an integrated ecosystem combining traditional software-engineering insight with modern AI analytics.

Table 1. Keyword Cluster Analysis

Cluster	Main Focus	Example Keywords	Research Meaning
Cluster 1	Classical approaches in software engineering	<i>fault prediction, object-oriented metrics, regression analysis</i>	Highlights the roots of prediction approaches based on software and traditional statistical models.
Cluster 2	Machine learning and optimization approaches	<i>random forest, support vector machine, genetic algorithms, class imbalance</i>	Depicts the increasing adoption of machine learning to address prediction challenges.
Cluster 3	Empirical research and open-source software	<i>open source software, mining software repositories, empirical studies</i>	Emphasizes the practical value of using real-world data in research.
Cluster 4	Cutting-edge technologies based on deep learning and AI	<i>deep learning, convolutional neural networks, long short-term memory</i>	Indicates the shift toward advanced AI-driven predictive approaches.
Cluster 5	Core themes and predictive model generalization	<i>defect prediction, performance, cross-project prediction, transfer learning</i>	Underlines the importance of performance evaluation and cross-project generalization in predictive models.

Source: (Research Results, 2025)

The emerging emphasis on transfer learning and explainable AI (XAI) within Clusters 4 and 5 indicates that the field is entering a new stage focused on interpretability, transparency, and cross-domain applicability—addressing long-standing challenges in model generalization and

industrial adoption [20]. Consequently, the keyword network shown in Figure 8 not only maps existing knowledge structures but also delineates the future trajectory of intelligent, explainable, and scalable software defect prediction research.

Future Research Trends and Hotspots

Building upon the thematic and keyword analyses presented in the previous sections, this study identifies several critical directions for future Software Defect Prediction (SDP) research. The summarized aspects are presented in Table 2, which categorizes the future landscape of SDP into five dimensions: Future Core Themes, Technological Approaches, Emerging Topics, Further Research Issues, and Development Directions.

Table 2 highlights that software testing, forecasting, and software evaluation will remain the future core themes of SDP. These areas continue to form the operational backbone of software quality assurance, now increasingly augmented by intelligent and automated analysis tools. Research in these domains is expected to focus on integrating predictive analytics within continuous integration (CI/CD) pipelines, enabling real-time defect detection and self-healing systems.

Table 2. Future Research Aspects in SDP

Research Aspect	Key Findings
Future Core Themes	Software testing, forecasting, software evaluation
Technological Approaches	Deep learning, transfer learning, convolutional neural networks
Emerging Topics	Cross-project defect prediction, ensemble learning, feature extraction
Further Research Issues	Class imbalance, model interpretability, cross-domain generalization
Development Directions	Advanced AI integration, explainable AI, robust modeling, practical applications

Source: (Research Results, 2025)

The technological approaches of future studies will be dominated by deep learning, transfer learning, and convolutional neural networks (CNNs). These methods represent the next phase in algorithmic sophistication, allowing SDP models to learn abstract representations of source code and software metrics directly from data [19]. The ongoing evolution toward transfer learning reflects a broader trend of leveraging pre-trained models across multiple domains to address limited data availability and project heterogeneity. This technological shift signifies a movement from handcrafted feature engineering to data-driven representation learning.

In terms of emerging topics, cross-project defect prediction, ensemble learning, and feature

extraction are gaining traction as hybrid strategies for improving model robustness and generalizability. By combining multiple classifiers and integrating knowledge across projects, researchers aim to overcome the bias and instability that often affect single-model approaches. These hybrid ensemble frameworks not only increase accuracy but also enhance the practical scalability of SDP in diverse industrial environments [18].

Further research issues remain pivotal challenges for the field. Problems such as class imbalance, model interpretability, and cross-domain generalization continue to limit the reliability of defect prediction in real-world applications. The growing focus on Explainable Artificial Intelligence (XAI) offers a promising pathway to address interpretability concerns by enabling stakeholders to understand the reasoning behind model predictions [20]. At the same time, handling class imbalance through synthetic sampling and cost-sensitive learning remains essential for building equitable and reliable defect prediction systems.

The development directions indicated in Table 2 point toward advanced AI integration, the incorporation of XAI principles, robust modeling, and stronger emphasis on practical applications. The next generation of SDP systems will likely blend predictive intelligence with explainability, producing tools that not only identify potential defects but also justify their decisions and suggest preventive actions. Such AI-driven frameworks could significantly enhance industrial adoption by aligning predictive analytics with software maintenance workflows, risk management strategies, and quality improvement policies.

From a broader perspective, these trends carry distinct practical implications for multiple stakeholders.

1. For academics, the future of SDP research lies in expanding datasets, improving model transparency, and exploring human-in-the-loop validation processes.
2. For industry practitioners, integrating interpretable AI tools into software engineering workflows can enable more accurate resource allocation, early fault detection, and continuous quality monitoring.
3. For policymakers and research funders, the identified directions emphasize the need to support open data repositories, cross-institutional collaboration, and reproducible research practices to accelerate innovation in software quality analytics.

Despite these promising directions, certain limitations should be acknowledged. The present

study relies solely on data indexed in the Scopus database, which may lead to underrepresentation of high-quality publications from conference proceedings and regional journals, particularly within Europe and North America [21]. Additionally, the bibliometric approach primarily captures quantitative relationships (e.g., co-occurrence patterns) rather than qualitative insights into theoretical contributions. Future research could integrate multiple databases (e.g., Web of Science, IEEE Xplore) and apply topic modeling or semantic analysis to provide deeper conceptual understanding of SDP evolution.

In conclusion, Table 2 and the associated analyses delineate the transition of SDP from traditional metric-based prediction toward AI-integrated, interpretable, and context-aware frameworks. The convergence of deep learning, transfer learning, XAI, and ensemble learning signifies a new paradigm where prediction accuracy, transparency, and practical usability are equally prioritized. This trajectory positions Software Defect Prediction as a central, continuously evolving pillar within the broader domain of intelligent software engineering.

DISCUSSION

This bibliometric study provides a comprehensive overview of the research landscape in Software Defect Prediction (SDP), based on an analysis of thousands of documents published and indexed in Scopus over the past two decades. The findings demonstrate that SDP research has experienced steady growth, with a notable increase since 2015, in line with the widespread adoption of machine learning and deep learning in predictive modeling. This indicates a growing awareness of the importance of software quality from the early stages of development [9]. The dominance of contributions from countries such as China, India, and the United States further highlights that this topic has become a strategic focus within the global information technology research ecosystem.

Nevertheless, several studies underline the challenges that remain, including class imbalance that affects the performance of prediction models [16], the limited effectiveness of ML-based approaches in practical contexts [22], and shifting trends toward the adoption of advanced models such as large language models (LLMs), which introduce both opportunities and methodological uncertainties [8]. Transparency of prediction models has also become an important issue, with recent research integrating explainable AI (XAI) frameworks to improve traceability and interpretability of defect prediction models [20].

Thus, although SDP research continues to grow rapidly, methodological and practical challenges remain a critical agenda in developing more accurate and reliable prediction models.

The subject area analysis shows that SDP-related research is dominated by Computer Science and Engineering, but has also expanded into other disciplines such as Mathematics, Decision Sciences, and Business, underscoring the multidisciplinary nature of the topic [10], [23]. At the thematic level, keyword mapping results reveal that themes such as software testing, forecasting, and evaluation fall within the motor theme category, signifying their central role in shaping SDP research. However, themes such as code smell or semantic analysis remain in niche or emerging positions, representing exploratory areas with potential for future development [1]. Several studies also highlight the limitations of machine learning approaches, including issues of effectiveness in industrial practice and the gap between academic trends and business needs [22], [23]. Accordingly, while SDP research has grown rapidly and become increasingly multidisciplinary, methodological challenges and practical applicability remain central concerns in fostering broader cross-disciplinary contributions.

The keyword network visualization and cluster analysis indicate that SDP research can be grouped into five thematic clusters. The initial clusters represent the historical foundation of software metrics based research and classical machine learning algorithms, while subsequent clusters emphasize empirical approaches leveraging open-source repositories. A further paradigm shift can be seen in clusters integrating deep learning, transfer learning, and cross-project prediction, which are considered capable of producing more flexible predictive models with cross-project generalizability [24], [25]. However, several studies show that despite their promise, these approaches still face serious challenges, such as imbalanced data distribution across projects and the risk of negative transfer, which can reduce model performance [26], [27]. Recent studies also stress that heterogeneous cross-project prediction continues to encounter feature compatibility issues across projects, although encoder networks and transfer learning approaches have yielded better results compared to classical methods [7]. Thus, current SDP research directions not only emphasize algorithmic innovation but also highlight the need for mitigation strategies to ensure reliable generalization across projects.

The mapping of future trends indicates that approaches such as ensemble learning, explainable

AI, and the utilization of big data to support robust modeling will become central foci in SDP research. This is supported by studies showing that ensemble-based hybrid models can effectively address class imbalance while improving prediction accuracy compared to classical methods [5], [19]. At the same time, the need to make prediction models more transparent and interpretable has fueled the development of explainable AI frameworks in the context of software defect prediction [20], [28]. Furthermore, the integration of advanced algorithms such as CNNs and LSTMs, along with sophisticated feature extraction techniques, has been proven to enhance model performance by capturing deeper semantic information from source code and software change histories [8]. Nonetheless, several studies emphasize that challenges remain, particularly related to data scalability, high computational requirements, and the limited interpretability of complex deep learning models. Therefore, future research directions emphasize balancing predictive accuracy and model transparency, ensuring that results are more practically applicable in the software industry.

That said, this study has several methodological limitations. The exclusive use of Scopus as the data source may limit the diversity of coverage and introduce bias in the representation of publications. Several studies recommend the integration of multiple databases such as Scopus, Web of Science, and Lens to provide a more comprehensive and less fragmented research landscape [21], [29]. Moreover, the quantitative nature of bibliometric analysis is often considered insufficient to fully capture the depth of theoretical contributions within individual articles, thus necessitating complementary qualitative or content analysis to enrich the interpretation of results [30], [31]. Nevertheless, numerous studies affirm that bibliometric methods provide a solid foundation for understanding the research landscape and can serve as a compass for both researchers and practitioners in navigating the future development of this research domain [32].

CONCLUSION

This study provides a comprehensive bibliometric mapping of global research trends and emerging frontiers in Software Defect Prediction (SDP), emphasizing the integration of machine learning (ML) and deep learning (DL) approaches. The findings reveal a clear paradigm shift from traditional metric-based models toward AI-driven predictive frameworks that are adaptive, data-intensive, and scalable. Thematic and keyword

analyses identified five interconnected research clusters, where mature themes such as software testing, forecasting, and evaluation remain central, while deep learning, transfer learning, and ensemble learning emerge as the next research frontier.

Future studies should focus on Explainable AI (XAI), cross-domain generalization, and class imbalance handling to enhance model transparency and industrial applicability. These advancements will bridge the gap between predictive accuracy and interpretability, an essential step for real-world adoption of intelligent defect prediction systems.

Overall, this research contributes by outlining the intellectual evolution and strategic roadmap of SDP. It offers valuable insights for researchers identifying new gaps, practitioners implementing interpretable predictive models, and policymakers promoting open, collaborative, and explainable AI ecosystems. Thus, SDP continues to evolve as a multidisciplinary field uniting software engineering, data science, and artificial intelligence toward smarter and more transparent software quality assurance.

REFERENCE

- [1] Z. Li, J. Niu, and X.-Y. Jing, "Software defect prediction: future directions and challenges," *Autom. Softw. Eng.*, vol. 31, no. 1, p. 19, May 2024, doi: 10.1007/s10515-024-00424-1.
- [2] M. J. Hernández-Molinos, A. J. Sánchez-García, R. E. Barrientos-Martínez, J. C. Pérez-Arriaga, and J. O. Ocharán-Hernández, "Software Defect Prediction with Bayesian Approaches," *Mathematics*, vol. 11, no. 11, p. 2524, May 2023, doi: 10.3390/math11112524.
- [3] C. Liu, S. Sanober, A. S. Zamani, L. R. Parvathy, R. Neware, and A. W. Rahmani, "Defect Prediction Technology in Software Engineering Based on Convolutional Neural Network," *Secur. Commun. Netw.*, vol. 2022, pp. 1–8, Apr. 2022, doi: 10.1155/2022/5058461.
- [4] X. Dong, Y. Liang, S. Miyamoto, and S. Yamaguchi, "Ensemble learning based software defect prediction," *J. Eng. Res.*, vol. 11, no. 4, pp. 377–391, Dec. 2023, doi: 10.1016/j.jer.2023.10.038.
- [5] S. Mehta and K. S. Patnaik, "Improved prediction of software defects using ensemble machine learning techniques," *Neural Comput. Appl.*, vol. 33, no. 16, pp. 10551–10562, Aug. 2021, doi: 10.1007/s00521-021-05811-3.
- [6] Q. M. U. Haq *et al.*, "Identification of Software Bugs by Analyzing Natural Language-Based Requirements Using Optimized Deep Learning Features," *Comput. Mater. Contin.*, vol. 78, no. 3, pp. 4379–4397, 2024, doi: 10.32604/cmc.2024.047172.
- [7] R. Haque, A. Ali, S. McClean, I. Cleland, and J. Noppen, "Heterogeneous Cross-Project Defect Prediction Using Encoder Networks and Transfer Learning," *IEEE Access*, vol. 12, pp. 409–419, 2024, doi: 10.1109/ACCESS.2023.3343329.
- [8] S. Zhang, S. Jiang, and Y. Yan, "A Hierarchical Feature Ensemble Deep Learning Approach for Software Defect Prediction," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 33, no. 04, pp. 543–573, Apr. 2023, doi: 10.1142/S0218194023500079.
- [9] A. Taskeen, S. U. R. Khan, and E. A. Felix, "A research landscape on software defect prediction," *J. Softw. Evol. Process*, vol. 35, no. 12, p. e2549, Dec. 2023, doi: 10.1002/smr.2549.
- [10] S. Zardari *et al.*, "A Comprehensive Bibliometric Assessment on Software Testing (2016–2021)," *Electronics*, vol. 11, no. 13, p. 1984, Jun. 2022, doi: 10.3390/electronics11131984.
- [11] P. N. Mata, J. M. Martins, and J. C. Ferreira, "New Software Product Development: Bibliometric Analysis," *J. Knowl. Econ.*, vol. 16, no. 1, pp. 4161–4184, Jun. 2024, doi: 10.1007/s13132-024-02095-5.
- [12] H. Maliha, "A Review on Bibliometric Application Software," *Scientometr. Lett.*, vol. 1, no. 1, Mar. 2024, doi: 10.58968/sl.v1i1.458.
- [13] F. Cobos-Alvarado, M. Peñaherrera-León, J. C. M. Alvarado, B. W. L. Valle, and J. E. Saavedra-Palma, "Bibliometric Analysis of Publications on Decentralization and Social Policy," *J. Ecohumanism*, vol. 3, no. 8, Nov. 2024, doi: 10.62754/joe.v3i8.4947.
- [14] P. Ngulube and N. F. V. Mosha, "Transparency in the Application of Theoretical Frameworks to the Advancement of Knowledge in Selected Library and Information Science Journals: A Systematic Review," *Afr. J. Libr. Arch. Inf. Sci.*, vol. 33, no. 2, pp. 131–150, Oct. 2023, doi: 10.4314/ajlais.v33i2.2.
- [15] L. O. Soares, A. D. C. Reis, P. S. Vieira, L. Hernández-Callejo, and R. A. M. Boloy, "Electric Vehicle Supply Chain Management: A Bibliometric and Systematic Review," *Energies*, vol. 16, no. 4, p. 1563, Feb. 2023, doi: 10.3390/en16041563.

- [16] S. R. Goyal, "Current Trends in Class Imbalance Learning for Software Defect Prediction," *IEEE Access*, vol. 13, pp. 16896–16917, 2025, doi: 10.1109/ACCESS.2025.3532250.
- [17] S. B. Jabeur, N. Stef, and W. B. Arfi, "ARTIFICIAL INTELLIGENCE FOR INNOVATION: A BIBLIOMETRIC ANALYSIS AND STRUCTURAL VARIATION APPROACH," *Int. J. Innov. Manag.*, vol. 28, no. 05n06, p. 2450020, Aug. 2024, doi: 10.1142/S1363919624500208.
- [18] Y. Z. Bala, P. A. Samat, K. Y. Sharif, and N. Manshor, "Current Software Defect Prediction: A Systematic Review," in *2022 Applied Informatics International Conference (AiIC)*, Serdang, Malaysia: IEEE, May 2022, pp. 117–121. doi: 10.1109/AiIC54368.2022.9914586.
- [19] M. Mustaqeem, T. Siddiqui, and S. Mustajab, "A hybrid-ensemble model for software defect prediction for balanced and imbalanced datasets using AI-based techniques with feature preservation: SMERKP-XGB," *J. Softw. Evol. Process*, vol. 37, no. 1, p. e2731, Jan. 2025, doi: 10.1002/smr.2731.
- [20] B. Gezici Geçer and A. Kolukısa Tarhan, "Explainable AI Framework for Software Defect Prediction," *J. Softw. Evol. Process*, vol. 37, no. 4, p. e70018, Apr. 2025, doi: 10.1002/smr.70018.
- [21] A. Tarazi, "Comparative Analysis of the Bibliographic Data Sources Using PubMed, Scopus, Web of Science, and Lens: Comparative Analysis of Bibliographic Data Sources," *High Yield Med. Rev.*, vol. 2, no. 1, Jun. 2024, doi: 10.59707/hymrUNHW4628.
- [22] D. Rai and J. Arcot Prashant, "Insights of effectivity analysis of learning-based approaches towards software defect prediction," *Int. J. Electr. Comput. Eng. IJECE*, vol. 14, no. 2, p. 1916, Apr. 2024, doi: 10.11591/ijece.v14i2.pp1916-1927.
- [23] S. Stradowski and L. Madeyski, "Machine learning in software defect prediction: A business-driven systematic mapping study," *Inf. Softw. Technol.*, vol. 155, p. 107128, Mar. 2023, doi: 10.1016/j.infsof.2022.107128.
- [24] R. Malhotra and S. Meena, "Defect prediction model using transfer learning," *Soft Comput.*, vol. 26, no. 10, pp. 4713–4726, May 2022, doi: 10.1007/s00500-022-06846-x.
- [25] T. Lei, J. Xue, D. Man, Y. Wang, M. Li, and Z. Kong, "SDP-MTF: A Composite Transfer Learning and Feature Fusion for Cross-Project Software Defect Prediction," *Electronics*, vol. 13, no. 13, p. 2439, Jun. 2024, doi: 10.3390/electronics13132439.
- [26] X. Fan, J. Mao, L. Lian, L. Yu, W. Zheng, and Y. Ge, "Software Defect Prediction Method Based on Stable Learning," *Comput. Mater. Contin.*, vol. 78, no. 1, pp. 65–84, 2024, doi: 10.32604/cmc.2023.045522.
- [27] B. Sotto-Mayor and M. Kalech, "A Survey on Transfer Learning for Cross-Project Defect Prediction," *IEEE Access*, vol. 12, pp. 93398–93425, 2024, doi: 10.1109/ACCESS.2024.3424311.
- [28] M. Begum *et al.*, "LCNN: Lightweight CNN Architecture for Software Defect Feature Identification Using Explainable AI," *IEEE Access*, vol. 12, pp. 55744–55756, 2024, doi: 10.1109/ACCESS.2024.3388489.
- [29] W. Chansanam and C. Li, "KKU-BiblioMerge: A novel tool for multi-database integration in bibliometric analysis," *Iberoam. J. Sci. Meas. Commun.*, vol. 5, no. 1, pp. 1–16, Feb. 2025, doi: 10.47909/ijsmc.157.
- [30] S. Z. Safin, N. F. Abdul Rahim, and H. Hanifah, "UNVEILING SUBJECTIVE CAREER SUCCESS: A COMPREHENSIVE BIBLIOMETRIC ANALYSIS," *Int. J. Entrep. Manag. Pract.*, vol. 6, no. 23, pp. 112–130, Dec. 2023, doi: 10.35631/IJEMP.623009.
- [31] Ibrahim Alshawabkeh, Fatma Zehra Tan, and Lee Sharolyn, "Artificial Intelligence and Human Resource Management: A Bibliometric Analysis," *Int. J. Bus. Manag.*, Apr. 2024, doi: 10.24940/theijbm/2024/v12/i2/BM2402-014.
- [32] A. P. Hanifah, M. Purnomo, and R. M. Dai, "Bibliometric Analysis of Entrepreneurial Competencies in the Last 10 Years (2013-2023)," *J. Samudra Ekon. Dan Bisnis*, vol. 15, no. 2, pp. 299–312, May 2024, doi: 10.33059/jseb.v15i2.9117.