ISSN 2007-9737

Exploring the Role of Artificial Intelligence in Software Security: A Comprehensive Systematic Review

Wilfredo Guia-Muñoz[1], Adrian Lizarbe-Estrada[1], Javier Gamboa-Cruzado[1], Alfonso Romero Baylón[1], Marcelino Carretero[2], Carlos Chávez-Herrera[3], Flavio Amayo-Gamboa[4], Francisco Antonio Castillo-Velásquez[5,*]

[1] Universidad Nacional Mayor de San Marcos, Peru

[2] Universidad Católica de Santo Toribio de Mogrovejo, Peru

[3] Universidad Nacional Tecnológica de Lima Sur, Peru

[4] Universidad Nacional de Trujillo, Peru

[5] Universidad Politécnica de Querétaro, Mexico

{wilfredo.guia, adrian.lizarbe , jgamboac, aromerob}@unmsm.edu.pe, mcarretero@usat.edu.pe, cchavezh@untels.edu.pe, famayo@unitru.edu.pe, francisco.castillo@upq.mx

Abstract. This paper examines the application of Artificial Intelligence (AI) in software security, within a context marked by increasingly sophisticated cyber threats that surpass the limitations of traditional methods, generating an urgent need for more effective solutions. For this purpose, a Systematic Literature Review (SLR) was conducted following the guidelines of Kitchenham [89] and PRISMA, initially retrieving 7,391 documents and refining the corpus to 70 relevant studies published between 2020 and 2025. The analysis focused on examining technologies, theoretical frameworks, and challenges associated with the impact of AI on software security. The results show that the most frequently used techniques are Machine Learning and Deep Learning, with a predominance of algorithms such as SVM, CNN, and Random Forest. In addition, there is a strong concentration of studies in Asian countries, particularly China, and notable development in areas such as Security Integration and Security Enumeration. The findings indicate an ongoing process of consolidation and evolution in the field, although gaps remain, such as the limited attention to emerging approaches like DeepSecAI and the scarce diversification of evaluation criteria. Consequently, future research should prioritize more transparent solutions, the integration of explainable frameworks, and the standardization of metrics that strengthen the comparability and applicability of results.

Keywords. Artificial intelligence, software security, systematic review, machine learning, application security.

1 Introduction

In the last decade, the increase in cyber threats, unauthorized access, and misuse of credentials has highlighted the vulnerability of numerous software systems to inadequately managed failures. In this context, the scientific community has shown growing interest in developing advanced solutions based on Artificial Intelligence (AI), aimed at automating the detection, classification, and prediction of code threats. The problem is evident: traditional mechanisms lack the necessary capacity to address the constant evolution of cyberattacks, making it imperative to

analyze the approaches, algorithms, and tools that the academic literature proposes from AI to address this challenge. Recent studies have shown increasing interest in applying AI to software security, particularly through machine learning approaches to detect attacks such as XSS and SQLi, where specialized frameworks have been developed that reduce false positives and enable the identification of critical vulnerabilities such as Vertical Broken Access Control in real time [1, 2, 3].

In parallel, datasets focused on Android applications have been created, enabling the training of models with more than 98% accuracy, highlighting the superiority of AI-based methods over conventional techniques [4,79,87]. Likewise, hybrid models and advanced architectures, including DistilBERT, CNN, and BiLSTM, have been proposed to improve attack classification efficiency, even in contexts with limited information, relying on semi-supervised and transfer learning approaches [5, 6, 12, 13].

Natural language processing has also been applied, with techniques such as Word2Vec and Universal Sentence Encoder, as well as the analysis of abstract syntax trees, which has made it possible to anticipate vulnerabilities in web applications with greater precision [7, 8, 10].

On the other hand, emerging practices such as DevSecOps, supported by large language models and chaos engineering, reinforce system resilience, while other studies emphasize the need to more systematically explore the integration of AI into system assurance (SSA) frameworks [9, 71, 78].

Practical tools such as AIBugHunter, integrated into IDEs, provide real-time assistance to developers, and systems such as IVul apply computer vision to code fragments, achieving accuracies close to 99% [16, 17, 18].

Vulnerability analysis has also benefited from the use of semantic embeddings, code graphs, and advanced statistical models to prioritize threats, complemented by comprehensive proposals against SQLi attacks based on SVM and neural networks, as well as deep analysis mechanisms applied to the cloud [11, 14, 15, 17].

Along the same lines, architectures such as Vul-Mixer stand out for optimizing generalization capacity without requiring excessive computational resources, consolidating the use of innovative techniques in static and dynamic analysis [19, 20]. In other critical domains, such as healthcare, artificial neural networks have demonstrated outstanding performance in predicting heart attacks, reinforcing the cross-sector applicability of AI [75].

Finally, recent reviews have identified a diverse set of machine learning techniques, including SVM, decision trees, Random Forest, and CNN, as effective predictors of vulnerabilities, while the impact of large language models and machine learning in software-defined networks is consolidating as one of the most promising lines of future research [77, 78, 88].

Reviewing the impact of AI on software security is essential in light of the accelerated growth of cyber threats and the critical dependence on software in strategic sectors.

This systematic review seeks to address relevant gaps in the literature, since, unlike previous studies focused on isolated or limited cases, it incorporates an updated time horizon (2020-2025), analyzes a wide spectrum of vulnerabilities, and comparatively evaluates different AI approaches. In addition, it aims to integrate fragmented or contradictory findings to provide a critical, structured, and evidence-based overview of the real effectiveness of AI in preventing, detecting, and mitigating software vulnerabilities.

The main objective of this study is to critically analyze recent scientific literature to identify, classify, and synthesize AI approaches applied to software security, assessing their effectiveness against different types of vulnerabilities and their contribution to improving threat detection, prediction, and mitigation.

Within this framework, the paper is organized as follows: Section 2 presents the background; Section 3 describes the methodology; Section 4 presents the main results and discussion; and Section 5 outlines the conclusions along with potential future research directions.

2 Background

The accelerated growth of software development and its integration into critical systems has

Table 1. Research questions and objectives

Question	Objective
RQ1: What criteria are used to evaluate the level of Software Security?	Identify and analyze the criteria employed to evaluate the level of software security.
RQ2: Which Artificial Intelligence technologies are most frequently applied to address Software Security problems?	Determine the AI technologies being applied in the field of software security.
RQ3: How are the studies that apply Artificial Intelligence in Software Security geographically distributed?	Analyze the geographical distribution of studies that apply AI in the field of software security.
RQ4: What definitions are used in the literature regarding Artificial Intelligence and Software Security?	Examine the definitions that have been used concerning AI and software security.
RQ5: What are the dominant thematic areas in the publications that apply Artificial Intelligence to Software Security?	Classify the main thematic categories addressed in research on AI and its influence on software security.

intensified concerns regarding security. In this scenario, Artificial Intelligence (AI), particularly Machine Learning (ML), emerges as a key resource to strengthen detection and prevention mechanisms, offering automated, adaptive, and scalable solutions.

2.1 Artificial Intelligence

Artificial Intelligence (AI) has had a decisive impact across multiple technological domains, including information security. Its ability to identify complex patterns, adapt to changing scenarios, and execute autonomous decisions enables a more dynamic response to cybersecurity challenges.

For example, a comprehensive framework has been proposed to describe both offensive and defensive dimensions of AI in hostile environments [37].

More specifically, AI has been used to enhance the resilience of security systems through adaptive and explainable techniques [40].

Likewise, hybrid proposals such as AIBugHunter demonstrate the practical potential of AI in predicting, classifying, and repairing software vulnerabilities [17].

Finally, in secure development contexts (DevSecOps), large language models (LLMs) have begun to play a relevant role by integrating with chaos engineering practices applied to security [10].

2.2 Software Security

Software security has become an essential priority in system development, driven by the increase in targeted attacks and the persistence of vulnerabilities in source code. Several studies have examined the causes of this problem, highlighting factors such as code quality, the absence of proper validations, and the use of outdated libraries [31].

In parallel, automated tools and methodologies have been proposed to classify bug reports as security-related or not, facilitating the prioritization of mitigation actions [5]. Similarly, approaches have been explored to strengthen web applications through the integration of traditional methods with intelligent techniques, such as machine learning-based firewalls [30][49]. More recently, blockchain and federated neural networks have been proposed for real-time protection of mobile applications [48].

2.3 Machine Learning

Machine Learning (ML) has proven to be a key technology for improving software security, thanks to its capacity for generalization and automatic learning from large volumes of data. In the domain of mobile applications, for example, efficient and accurate classification models have been developed to detect malware using specific features such as RGB vectors or customized feature vectors [33, 38].

In web applications, detection schemes for attacks such as XSS and SQLi have been proposed through hybrid approaches that integrate deep neural networks with supervised and unsupervised learning techniques [6,24,54]. Furthermore, frameworks such as Vul-Mixer and CODE-SMASH have been investigated, which combine different neural architectures to maximize accuracy in detecting vulnerabilities in source code [20,22]. These advances reflect how machine learning is consolidating as an effective tool to

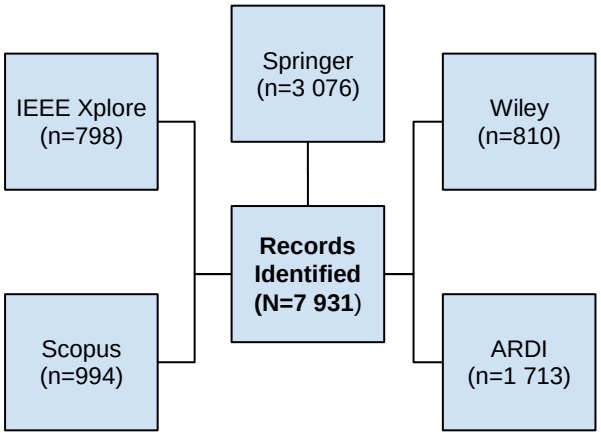

Fig. 1. Number of Results by Source

Table 2. Search terms and their synonyms

Name	Associated Terms
Artificial Intelligence	artificial intelligence, ai, generative artificial intelligence, generative ai, gen ai, machine learning, ml
Software Security	software security, application security, software protection, software vulnerability detection, software vulnerability prediction, code vulnerability detection, code vulnerability prediction

anticipate, classify, and prevent threats before they are exploited.

3 Methodology

This systematic literature review was conducted following the methodological guidelines proposed by Kitchenham in 2009 [89], which are widely recognized in the field of software engineering. Complementarily, the PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) framework was incorporated to ensure transparency, traceability, and rigor in the processes of searching, selecting, and analyzing the studies considered.

3.1 Research Questions and Objectives

A set of research questions was established (see Table 1) to guide the review and define the scope of the study. These questions are intended to examine how Artificial Intelligence (AI) has been applied to enhance software security.

3.2 Information Sources and Search Strategies

Scientific databases with recognized impact were selected (see Table 2), and specific search strings were formulated in order to systematically retrieve relevant and pertinent literature for the object of study.

The following represents the general search equation used as a reference for the review.

This expression served as the baseline formulation, which was then adapted and customized for each consulted database according to the specific characteristics of its search engine, in order to maximize the retrieval of relevant literature through the use of synonyms and Boolean operators:

("artificial intelligence" OR "ai" OR "generative artificial intelligence" OR "generative ai" OR "gen ai" OR "machine learning" OR "ml") AND ("software security" OR "application security" OR "software protection" OR "software vulnerability detection" OR "software vulnerability prediction" OR "code vulnerability detection" OR "code vulnerability prediction")

3.3 Identified Studies

Figure 1 presents the results obtained after applying the search strategies, showing the total number of studies collected in the initial phase of the identification process.

3.4 Study Selection

In order to ensure the relevance and methodological rigor of the review, six exclusion criteria (EC) were clearly defined and systematically applied during the screening process.

The implementation of these conditions allowed the refinement of the initial set of records and the concentration on the most relevant studies. This process is summarized in the PRISMA diagram

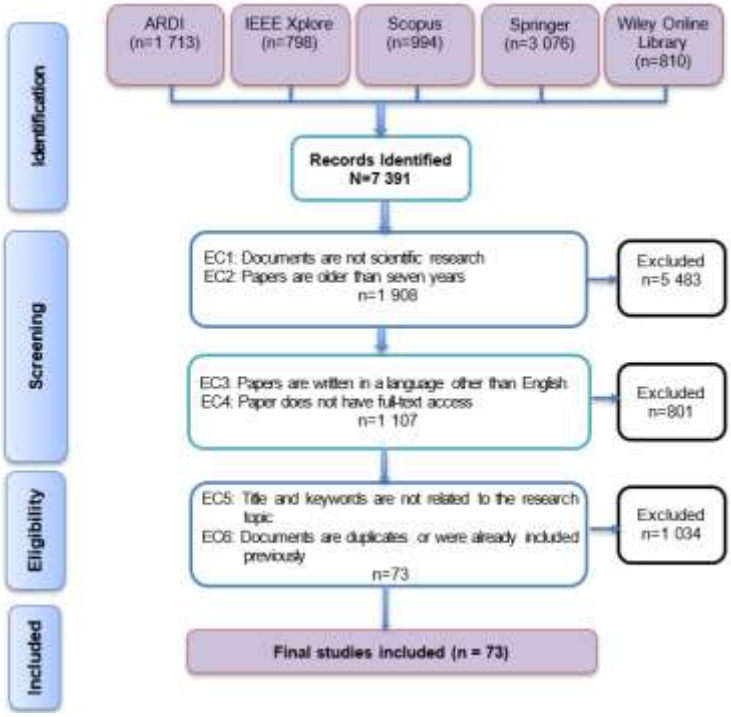

Fig. 2. PRISMA Flow Diagram

shown in Figure 2, which leads to the final set of included papers.

3.5 Quality Assessment

The quality evaluation of the included studies was conducted through a set of seven criteria (QA) designed to examine essential aspects such as the clarity of objectives, methodological coherence, and the robustness of conclusions. The defined criteria were:

QA1: Is the research objective formulated in a clear and precise manner?

QA2: Is the methodology applied in the study described with sufficient detail and transparency?

QA3: Does the paper demonstrate a logical and coherent structural organization?

QA4: Is the dataset employed clearly and verifiably specified?

QA5: Do the conclusions directly correspond to the established objectives?

QA6: Does the study adequately contextualize the environment in which the research was conducted?

QA7: Are the proposed experimental solutions clearly described and documented?

To ensure consistency and reliability in the process, each criterion was evaluated using a scale of 1 (poor), 2 (acceptable), and 3 (outstanding), establishing a minimum overall threshold of 11.5 out of 21 for inclusion.

Only the studies that met or exceeded this score were considered in the final analysis. The detailed results of this evaluation are presented in Table 3, which summarizes the scores obtained by each paper.

The quality assessment, applied to the 73 selected papers using the seven QA criteria, resulted in the exclusion of three studies that did not reach the minimum threshold score of 11.5, thereby ensuring the inclusion of research with sufficient rigor and methodological consistency.

ISSN 2007-9737

Table 3. Results of the Quality Assessment

Ref.	Type	QA1	QA2	QA3	QA4	QA5	QA6	QA7	Score
1	Journal	1	1	1	3	3	2	3	14
2	Journal	1	1	3	2	3	3	2	15
3	Journal	1	1	2	1	2	2	2	11
4	Journal	2	2	3	1	1	1	3	13
5	Journal	1	2	2	3	3	3	3	17
6	Journal	2	1	2	2	2	3	3	15
7	Journal	1	1	3	2	1	3	3	14
8	Journal	2	3	3	3	3	1	1	16
9	Journal	3	2	1	3	1	1	2	13
10	Confer.	2	2	3	3	2	1	1	14
11	Journal	2	1	1	2	2	1	1	10
12	Journal	3	2	1	3	1	2	2	14
13	Journal	2	1	3	1	1	2	2	12
14	Journal	1	3	3	1	1	3	3	15
15	Journal	2	2	2	2	2	1	3	14
16	Journal	3	3	2	1	2	1	3	15
17	Journal	1	1	3	3	1	1	2	12
18	Journal	2	1	3	3	1	3	1	14
19	Journal	3	1	3	2	1	2	3	15
20	Journal	3	2	2	3	2	3	2	17
21	Journal	2	2	2	1	1	1	3	12
22	Journal	3	1	2	3	2	3	2	16
23	Journal	3	2	3	1	2	3	2	16
24	Journal	2	2	2	1	2	2	3	14
25	Journal	2	3	3	3	1	2	2	16
26	Journal	1	3	3	2	3	2	1	15
27	Journal	2	3	3	1	1	1	2	13
28	Journal	3	2	2	2	1	1	1	12
29	Journal	1	3	3	1	2	1	3	14
30	Journal	2	3	1	2	1	1	2	12
31	Journal	3	1	2	2	2	1	2	13
32	Journal	1	3	1	2	2	1	3	13
33	Journal	2	1	2	3	2	2	3	15
34	Journal	1	1	1	3	2	1	3	12
35	Journal	2	3	1	1	1	3	3	14
36	Journal	3	1	1	2	3	1	1	12
37	Journal	2	2	1	3	2	3	3	16
38	Journal	1	2	2	2	2	2	3	14
39	Journal	1	3	2	1	3	3	2	15
40	Journal	1	3	1	3	1	3	1	13
41	Journal	1	2	1	2	1	3	3	13
42	Journal	1	1	3	1	3	2	3	14
43	Journal	3	2	3	1	1	3	1	14
44	Journal	2	2	3	2	3	2	1	15
45	Confer.	1	3	3	2	3	1	2	15
46	Journal	1	1	2	3	2	1	2	12
47	Journal	3	2	2	1	1	1	2	12
48	Journal	1	1	2	1	2	2	1	10
49	Journal	2	3	3	1	2	3	3	17
50	Journal	3	3	2	3	1	1	2	15
51	Journal	2	1	3	1	3	1	3	14
52	Confer.	1	2	1	3	1	3	3	14
53	Journal	2	3	1	1	2	3	3	15
54	Journal	2	2	3	3	1	3	2	16
55	Journal	2	1	2	2	2	2	1	12
56	Confer.	3	2	3	1	1	1	1	12
57	Journal	1	1	2	3	1	2	2	12
58	Journal	1	3	1	2	2	2	2	13
59	Journal	1	3	1	2	1	2	3	13
60	Journal	1	2	3	1	2	3	3	15
61	Journal	1	2	1	3	3	1	1	12
62	Journal	2	1	1	3	1	2	3	13
63	Journal	1	2	1	1	2	3	3	13
64	Journal	2	2	2	3	3	2	1	15
65	Journal	1	2	2	3	1	1	3	13
66	Journal	3	1	1	2	2	3	3	15
67	Journal	2	2	2	2	2	2	3	15
68	Journal	3	2	1	2	1	3	2	14
69	Journal	3	3	1	2	1	2	3	15
70	Journal	2	2	1	3	1	2	1	12
71	Journal	2	3	2	3	3	1	1	15
72	Journal	1	1	3	1	3	3	3	15
73	Journal	2	2	3	2	2	1	2	14

Fig. 3. Mendeley Desktop

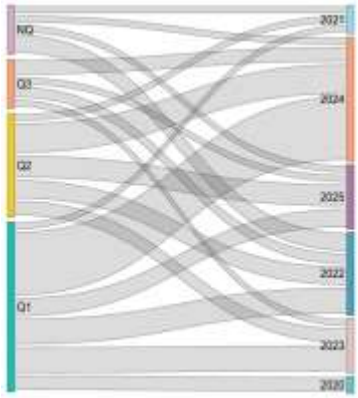

Fig. 4. Distribution of papers by quartile level and year

Table 4. Quartile levels of studies classified by year

Year	Q1	Q2	Q3	NQ	Total
2020	3	0	0	0	3
2021	1	1	0	1	3
2022	6	4	2	4	16
2023	6	3	1	0	10
2024	16	7	3	1	27
2025	4	5	1	1	11
Total	36	20	7	7	70

Table 5. Impact factors for quartile levels

Quartile	Total Citations	Number of Papers	Citations per Paper
Q1	295	36	8
Q2	135	20	7
Q3	6	7	1
NQ	88	7	13
Total	524	70	7

3.6 Data Extraction Strategy

The data extraction strategy was implemented using Mendeley Desktop, selected for its versatility and ease of use in managing references and metadata. This tool enabled the organization of the selected papers, the storage of key information, and the systematization of the analysis process. Figure 3 illustrates the interface and functionalities employed during this phase.

4 Results and Discussion

This section presents the most relevant findings derived from the analysis of the selected studies. The results are organized around the previously formulated research questions, providing a structured and coherent response to the objectives established in the study.

4.1. General Description of the Studies

The systematic review made it possible to identify a representative set of investigations addressing the application of artificial intelligence in software security. The analysis reveals significant patterns in academic production, reflecting both the growing interest in the topic and the concentration of results in high-impact journals-an essential aspect for assessing the robustness and relevance of the knowledge generated in the field.

Figure 4 and Tables 4 and 5 show the distribution of papers on artificial intelligence applied to software security, classified by quartile and year of publication. This approach allows observation of both the temporal evolution of academic production and the relative impact of each quality level.

The results show that the largest volume of publications is concentrated in Q1 journals, with a total of 36 papers, reflecting the preference for high-impact outlets. In temporal terms, 2024 stands out with the highest number of contributions (27), confirming sustained growth in the field. Papers in Q2 (20) also represent a significant contribution, although with a lower average citation rate compared to Q1. Works in Q3 and NQ are limited (7 each), but those in NQ present a citation rate per paper (13) higher than Q1 and Q2,

Table 6. Number of papers by affiliations and year

Publication Name	N°Papers	Citations	H-Index	Quartile
IEEE Access	9	112	2610	Q1
International Journal of Information Security	5	21	275	Q2
Scientific Reports	5	42	1735	Q1
Applied Sciences	4	28	648	Q2
Computers, Materials and Continua	3	3	201	Q2
Future Internet	3	41	255	Q2
Security and Communication Networks	3	50	0	NQ
Arabian Journal for Science and Engineering	2	9	162	Q1
Empirical Software Engineering	2	32	200	Q1
Information	2	13	144	Q2
Journal of Information Security and Applications	2	8	146	Q1
…	…	…	…	…
Total	70	524	8855	

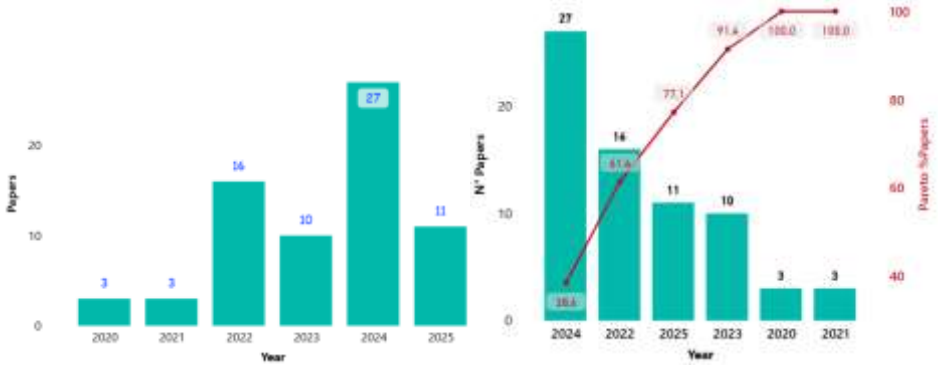

Fig. 5. Distribution of papers by year

suggesting a specific interest in certain studies outside the highest-ranked journals. Overall, the 70 papers accumulated 524 citations, with a general average of 7 citations per paper, a reasonable indicator of academic visibility.

The findings of this study contrast with those of Mishra and Pandya [73], who reported a peak in AI scientific production in 2020. Our results, focused specifically on software security, show that year to have the lowest volume of studies (n=3), with notable growth beginning in 2022.

This divergence is explained by the difference in the scope of the reviews (general vs. specific).

Furthermore, the concentration of publications in Q1 coincides with what has been reported in other areas such as medicine [80], confirming that this is a general trend in AI research rather than a particularity of this field.

These findings highlight the consolidation of the topic in elite journals, which strengthens its scientific legitimacy.

Likewise, the presence of cited studies in NQ suggests opportunities to diversify dissemination channels in emerging contexts. In the future, this trend may extend to other industrial sectors and geographic regions, where the combination of quality and impact will be decisive in guiding practical applications and new lines of research.

Table 6 presents the distribution of papers according to the journals in which they were published, also indicating their quartile, accumulated citations, and H-index.

This analysis makes it possible to identify the most impactful and relevant sources in the dissemination of scientific knowledge on artificial intelligence applied to software security.

Table 7. Evaluation criteria for software security level

Evaluation Criterion	Reference	Quantity (%)
Data Confidentiality	[1]	1 (5)
Data Integrity	[2], [5], [16], [26-27], [40]	6 (33)
Vulnerability Management	[26]	1 (6)
Access Control	[2], [14], [16], [24], [40], [43], [49], [53-54], [68]	10 (56)

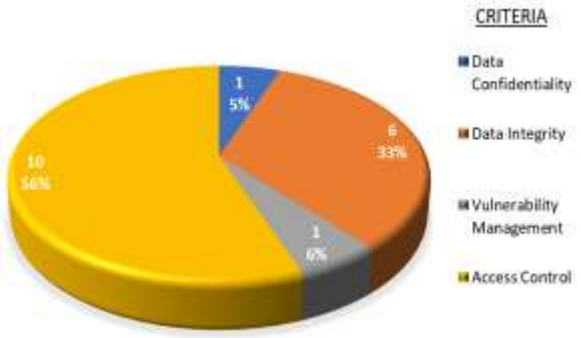

Fig. 6. Evaluation criteria for software security level

The results show that IEEE Access leads in volume with 9 papers and 112 citations, consolidating itself as the main dissemination platform in Q1. Journals such as International Journal of Information Security and Scientific Reports also stand out, with 5 and 4 publications respectively, reflecting diversity in thematic coverage. The highest H-index also corresponds to IEEE Access (2610), confirming its high visibility and academic rigor. The presence of publications in Q2 and NQ is noteworthy, although with lower citation and visibility, indicating a broad spectrum of dissemination. In total, the 70 papers reached 524 citations and a cumulative H-index of 8855, evidencing an expanding field with growing recognition.

While the study by Yaseen and colleagues [85] analyzes institutional productivity (affiliations such as Oxford and the Chinese Academy of Sciences), our study focuses on editorial productivity (journals such as IEEE Access). Both, however, converge on a fundamental finding: the concentration of knowledge in highly prestigious and internationally recognized actors. This convergence suggests that, regardless of the unit of analysis (institution vs. journal), research in artificial intelligence-whether general or applied to software security-is consolidating within clearly defined centers of excellence.

These findings suggest that the consolidation of the topic in Q1 journals guarantees scientific legitimacy and wide dissemination, but there is also a need to leverage Q2 and NQ journals to reach specific communities. At a practical level, this distribution opens the possibility of replicating publication strategies in other industrial sectors and different regions, fostering the transfer of knowledge to environments where software security is equally critical.

Figure 5 combines a bar chart with a Pareto analysis, showing both the annual distribution of papers between 2020 and 2025 and the cumulative weight of publications. This dual visualization not only allows the identification of temporal evolution but also highlights the most influential years in the total production.

The results show that 2024 concentrates the highest production with 27 papers, equivalent to 38.6% of the total, followed by 2022 with 16 publications (61.4% cumulative). Adding 2025 (11) and 2023 (10) reaches 91.4%, confirming that in just four years more than 90% of the literature is concentrated. In contrast, the early years 2020 and 2021 contributed only 3 papers each, reflecting the incipient start of this research line. This behavior evidences accelerated growth from 2022, peaking in 2024, with a slight contraction in 2025. The pattern follows the Pareto rule, where a few years concentrate the majority of publications.

The results of temporal distribution confirm the findings reported in recent literature. The study by Yitagesu and collaborators [72] had already highlighted the notable growing interest in this research area, which fully aligns with our evidence of accelerated growth starting in 2022. This trend is reinforced by the work of Miller and colleagues [74], whose analysis identifies precisely the same temporal pattern: a turning point in 2020 that triggered sustained growth, reaching its peak in 2023-2024. The coincidence of these findings across independent studies validates the robustness of the observed trend and confirms that the field of artificial intelligence applied to software

Table 8. AI techniques identified in the reviewed papers

Evaluation Criterion	Reference	Quantity (%)
Machine Learning	[1-32], [34-52], [54-58], [60-62], [64-69]	65 (40.6)
Deep Learning	[1], [5-10], [13-16], [18], [21-27], [29-34], [36], [38-41], [45-49], [51-65], [68-69]	52 (32.5)
Natural Language Processing	[5], [7-14], [18], [20], [22], [24-26], [29-31], [33-34], [39], [41], [45-47], [50], [52-54], [59-61], [63], [66-70]	38 (10)
Anomaly detection	[7-8], [10], [16], [24], [26-27], [40], [43], [51], [60], [64]	12 (7.5)
Reinforcement Learning	[1], [6], [8-9], [15], [26], [40], [45], [64], [70]	10 (4.4)
Static and Dynamic Code Analysis	[11], [14], [20-24], [29], [31], [34], [39], [42], [44], [51], [53], [56], [62]	17 (5)

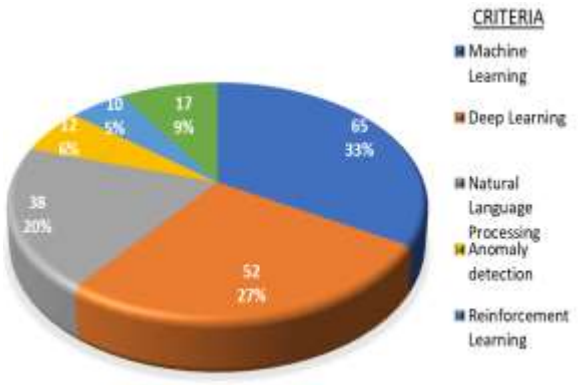

Fig. 7. AI techniques used in software security studies

security is undergoing a stage of maturation and academic consolidation.

The predominance of recent production suggests that the topic has reached a state of maturity and visibility that may extend to other sectors such as healthcare, transportation, or finance.

Moreover, this temporal pattern indicates that research has a strong situational character, opening space to explore its evolution in other regions and regulatory contexts.

Finally, the cumulative analysis constitutes a methodological benchmark for identifying windows of opportunity and projecting future scenarios for applied research.

4.2. Answers to the Research Questions

This section presents the RQs together with the main findings, their critical analysis, and the derived implications. The results are based on a systematic and exhaustive review of the literature on the use of artificial intelligence and its impact on software security. For this purpose, high-impact scientific databases were consulted.

RQ1: *What criteria are used to evaluate the level of software security?*

Table 7 and Figure 6 illustrate the distribution of criteria employed in the literature to evaluate the level of software security. This combined representation clearly highlights which dimensions receive the greatest research attention.

Access control emerges as the most studied criterion (56%), highlighting the importance assigned to protection against unauthorized access as the first line of defense. In second place, data integrity (33%) stands out as a pillar for ensuring that information remains reliable and free from malicious or accidental alterations. Vulnerability management accounts for only 6%, reflecting limited attention to processes of identification and preventive mitigation. Finally, data confidentiality (5%) appears with less weight, suggesting that despite its importance in critical sectors, it has not received the same level of analysis. Overall, the results reveal a bias toward immediate technical measures rather than a comprehensive vision of security.

When comparing our results with those of Wen, Shukla, and Katt [71], it is observed that while our study highlights access control and data integrity as the most cited criteria, their review focuses on vulnerability detection and risk analysis.

Criteria such as confidentiality, availability, and event logging are not explicitly addressed, which coincides with the limited attention identified in our analysis.

On the other hand, Yitagesu and colleagues [72] mention aspects such as vulnerability analysis and data quality, elements that are rarely addressed in this type of study. Their review focuses on the extraction of information about software security vulnerabilities, identifying key components and evaluation metrics. This contrasts

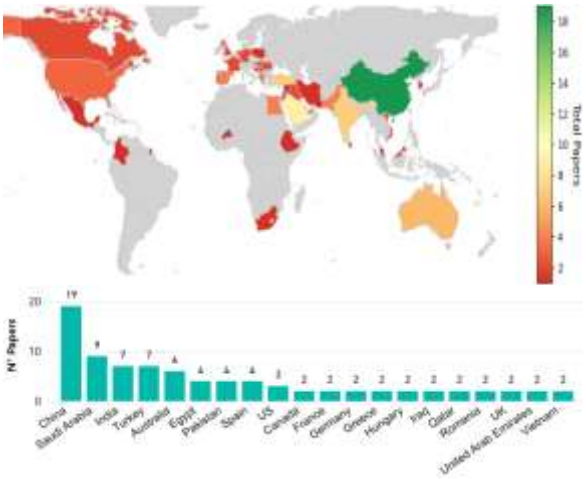

Fig. 8. Number of papers by country

Table 9. Papers by country and their impact

Country	Nº Papers	% Papers	No Cites	% Cites	Cites / Paper	H-Index
China	19	18%	192	25%	10	2,573
Saudi Arabia	9	9%	81	11%	9	1,663
India	7	7%	56	7%	8	1,065
Turkey	7	7%	30	4%	4	835
Australia	6	6%	21	3%	4	1,071
Egypt	4	4%	46	6%	12	765
Pakistan	4	4%	17	2%	4	820
Spain	4	4%	2	0%	1	240
us	3	3%	63	8%	21	453
...	...	...	...	...	...	...
Total	**104**	**100%**	**763**	**100%**	**7**	**14,436**

with other works that are more oriented toward general aspects of security. Similar to what has been observed in the medical field-where ANNs outperform traditional statistical methods in detecting heart attacks-AI applied to software security also exhibits clear advantages over conventional approaches [80].

The limited attention to criteria such as confidentiality and vulnerability management contrasts with the comprehensive approach proposed in recent AI risk management models, which require simultaneous evaluation of four critical dimensions: security (robustness and cybersecurity), accuracy (validity of outputs), fairness (non-discrimination and data representativeness), and explainability (human auditability). This framework suggests that future research in software security should broaden its evaluative scope to include not only technical controls but also ethical and social impacts [91].

These findings indicate the need to strengthen research on confidentiality and vulnerability management, expanding their application to areas such as healthcare, banking, and e-government. Likewise, the evaluation framework can be extrapolated to other geographical contexts to identify similarities or differences in security priorities. Finally, considering the temporal evolution of these criteria would allow anticipating trends and guiding public and corporate policies toward more balanced and sustainable security.

RQ2: *What Artificial Intelligence technologies are most frequently applied to address software security problems?*

Table 8 and Figure 7 summarize the artificial intelligence technologies applied in the literature to tackle software security issues. This synthesis highlights the predominant trends and the areas that remain underexplored.

Machine learning emerges as the most widely implemented technology (40.6%), highlighting its generalization capacity across different security scenarios. It is followed by deep learning (32.5%), consolidated as a key technique for advanced classification and detection tasks.

Natural language processing (10%) reflects the growing attention on semantic analysis and the interpretation of code or security reports.

Anomaly detection (7.5%) remains a complementary approach aimed at the early identification of irregularities. Other techniques, such as reinforcement learning (4.4%) and static and dynamic code analysis (5%), appear less frequently, although they represent an expanding field with strong potential to enhance preventive detection.

In contrast, Bassi and Singh [77] emphasize the predominance of supervised machine learning techniques such as Random Forest, Support Vector Machine (SVM), Naïve Bayes, and Decision Trees.

Similarly, Zhang and colleagues [78] highlight that LLMs are already being effectively applied to cybersecurity tasks such as threat detection and

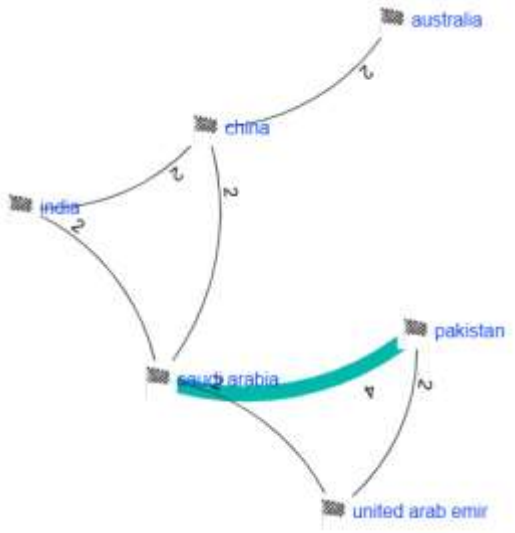

Fig. 9. Bibliometric network by country

vulnerability analysis, while also noting that more traditional machine learning and deep learning methods remain the most widely used. In the same vein, Senanayake and other researchers [79] confirm the frequent use of SVM, Random Forest, Naïve Bayes, Decision Trees, CNN, and LSTM as effective approaches for detecting vulnerabilities and classifying malicious code.

The predominance of machine learning and deep learning in this study aligns with findings from recent large-scale systematic reviews, where 88.4% of vulnerability detection research employed deep learning models, compared to only 7.2% using classical machine learning techniques. Within deep learning, recurrent architectures (especially BiLSTM) and graph-based models (with GCN as the leading approach) are the most widely adopted, showing a global convergence toward methods capable of capturing both sequential and structural dependencies in source code [83].

These results suggest that, although machine learning and deep learning dominate, further research is needed on emerging techniques such as reinforcement learning and hybrid code analysis. Moreover, the application framework can be transferred to sectors such as healthcare, finance, and Industry 4.0, where software security is equally critical. Finally, replicating these approaches across different regions and time periods would allow for the comparison of local priorities and adaptations to cyber threats.

RQ3: *How are studies applying Artificial Intelligence in the field of Software Security geographically distributed?*

Figure 8 and Table 9 present the geographical distribution of studies on the use of artificial intelligence in software security. This combination of maps, bar charts, and bibliometric metrics makes it possible to identify both the production and the academic impact by country.

China leads scientific production with 19 papers (18%) and accounts for 25% of the citations, with an outstanding h-index (2573), confirming its global leadership in the field. Saudi Arabia and India, with 9% and 7% of the papers respectively, also stand out in total citations and consolidate their position as emerging hubs in Asia and the Middle East. Australia and Egypt, despite having a smaller volume (6% and 4% of papers), show relevant impact, particularly Egypt with 12 citations per paper. The United States, with only 3 publications, excels with a high average of 21 citations per paper, evidencing superior quality and visibility despite its lower quantitative contribution. Other countries such as Turkey, Pakistan, and Spain show moderate production, although with notable differences in relative impact, reflecting disparities in research maturity.

Comparing the results of this study with those of Palash Uddin and colleagues [84], it is observed that studies on machine learning techniques, such as Federated Learning, are more concentrated in China, where the country's dominance also prevails in the findings of our research. Reinforcing this trend, Tymoteusz Miller and collaborators [74] identify China as the main contributor in terms of paper volume; however, unlike our study, they highlight the United States as the most representative country in North America, whereas in our review Canada showed the highest contribution in scientific publications. Complementarily, Yaseen and colleagues [85] emphasize the joint predominance of the United States and China in this research line. Likewise, Gamboa-Cruzado and his team [92] confirm China's leadership in scientific production on AI applied to software security, a pattern consistent

Table 10. Definitions of software security

Definition	Context	Reference
Software security consists of applying preventive and corrective measures to protect systems and data against threats, vulnerabilities, and attacks. Techniques such as machine learning, access control, and encryption are employed. Its objective is to guarantee the confidentiality, integrity, and availability of information.	Protection against threats and vulnerabilities	[1-3], [6-24], [26, 27], [31,32], [35, 36], [38-42], [44-48], [50-54], [56, 57], [59-61], [63, 64], [66, 67], [69, 70]
It consists of ensuring the confidentiality, integrity, and availability of data and information systems through the implementation of security controls.	CIA principles (Confidentiality, Integrity, Availability)	[4]
It consists of identifying and correcting errors in the design, development, or configuration of software that may compromise the security of a computer system.	Other approach	[34]

with other emerging technological fields such as cybersecurity in 5G networks, where Asia and Europe also dominate production, with China and the United Kingdom standing out for their high frequency of international collaborations. Finally, China's leadership in this field-particularly in intrusion detection and federated learning-aligns with recent reviews that identify China as a strategic actor in AI for cybersecurity, alongside the United States and the United Kingdom [76].

These results confirm that research on AI applied to software security is geographically concentrated, but with significant contributions from non-traditional science and technology countries. This pattern invites extending studies to other regions to diversify approaches and application scenarios. Likewise, the lessons learned can be transferred to strategic sectors such as health, finance, or transportation, where international cooperation and regional comparison are key to addressing future threats.

Figure 9 presents the bibliometric network among countries, highlighting the main collaboration links in research on artificial intelligence applied to software security. This analysis makes it possible to identify not only the leaders in scientific production but also the strategic alliances that enhance academic impact.

The strongest link is observed between Pakistan and Saudi Arabia, with a weight of 4, reflecting solid cooperation likely sustained across multiple projects. China establishes strategic connections with both India and Saudi Arabia (weight 2 each), confirming its role as a central node that articulates regional and global collaborations. India, in addition to its link with China, strengthens its cooperation with Saudi Arabia, evidencing its active integration into multinational networks. Saudi Arabia and the United Arab Emirates, as well as Pakistan with the United Arab Emirates, consolidate cooperation in the Middle East, an emerging hub for security research. Australia appears in the network associated with China, illustrating a key interregional bridge between Asia and Oceania. Taken together, the results suggest that the densest collaborations are concentrated in Asia and the Middle East, with China and Saudi Arabia as articulating nodes.

The co-authorship map obtained in this study was compared with the network diagram developed by Guembe and his team [81], who identified the United States and China as central nodes in research on AI applied to cyberattacks, with 27 collaborative papers. Similarly to the findings of this review, Koca and Çiftçi [82] also highlight a bibliometric network dominated by China, the United States, India, and Australia, countries that act as central nodes of scientific collaboration in artificial intelligence applied to cybersecurity, with the U.S.-China link standing out as the main axis.

The strengthening of intense bilateral alliances, such as the Saudi Arabia-Pakistan partnership, may serve as a model for other countries seeking to consolidate their presence in global research. These cooperation dynamics, if replicated in sectors such as health, energy, or transportation, could boost interdisciplinary innovation. Moreover, expanding these networks to underrepresented

Table 11. Definitions of artificial intelligence

Definition	Context	Reference
It consists of applying machine learning and deep learning models, such as neural networks, to automatically detect vulnerabilities or malware in software. These models allow analyzing code or representing it as images to extract relevant features. Architectures such as CNN and RNN are used to identify patterns associated with threats in computer systems.	Applied definition	[18], [46], [60]
Artificial intelligence involves the development of algorithms and models that enable machines to learn, make decisions, and recognize patterns as a human would. It is applied in areas such as speech recognition, data analysis, and computer security to autonomously detect threats.	Functional definition	[2, 3], [6-17], [19-24], [26, 27], [31, 32], [34-36], [39-41], [44, 45], [47, 48], [50-52], [54], [56, 57], [61, [63], [66, 67], [69, 70]
Artificial intelligence enables machines to learn from data, recognize patterns, and make decisions autonomously.	Technical definition	[1], [4], [20], [38], [42], [53], [64]
Artificial intelligence involves the use of pre-trained language models, such as ELMo, to capture deep contextual information in the words of a sentence and obtain embedded representations tailored to the current context.	Other approach	[59]

regions would foster a more equitable distribution of knowledge and generate solutions more contextualized to diverse geographical and temporal environments.

RQ4: *What definitions are employed in the literature regarding Artificial Intelligence and Software Security?*

Table 10 synthesizes the main definitions of software security identified in the literature, classified according to their application context. This systematization provides insight into how the field is conceptualized from different theoretical and practical perspectives.

The most widespread definition corresponds to software security as protection against threats, vulnerabilities, and attacks, supported by a broad set of references, and emphasizes the use of techniques such as machine learning, encryption, and access control. Another, more specific approach focuses on the CIA principles (confidentiality, integrity, and availability), highlighting the importance of the classical foundations of information security. Finally, some studies stress a corrective approach, oriented toward identifying and resolving design, development, or configuration errors that may compromise system security. Taken together, it is observed that definitions vary between a broad and proactive framework and more technical and operational views.

These definitions reflect the need for comprehensive approaches that combine prevention, detection, and correction, which can also be applied in areas such as digital health or banking, where security is critical. Moreover, adapting conceptual frameworks to different regions and historical periods will make it possible to evaluate how security priorities evolve with context. Finally, this conceptual diversity suggests an opportunity to develop a unified framework that facilitates comparisons across sectors and business applications.

Table 11 compiles the principal definitions of artificial intelligence (AI) identified in the literature, classified by focus. This analysis facilitates an understanding of how AI is conceptualized and applied in the context of software security.

The functional definition is the most widespread, linking AI with the ability to learn, recognize patterns, and make decisions, applied in computer security and other areas, with strong bibliographic support. The technical definition complements this framework by highlighting learning from data and autonomy in decision-making, representing a more structured and operational view. At the applied level, AI is defined through machine learning and deep learning models, such as CNNs and RNNs, used to detect software vulnerabilities and malware through code or image analysis. Finally, an alternative approach emerges, focused on the use of pre-trained

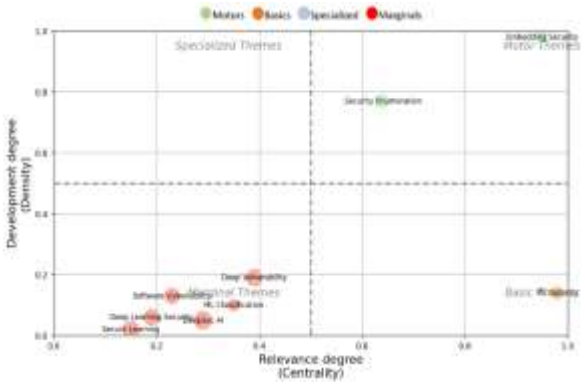

Fig. 10. Thematic map of the topic categories of the studies

Table 12. Density and centrality of the study topics

Topic	Density	Centrality	No. Cites	Category
Security Integration	0.98	0.95	66	Motor
Security Enumeration	0.77	0.64	151	Motor
Deep Vulnerability	0.19	0.39	315	Marginals
ML Security	0.14	0.98	123	Basics
Software Vulnerability	0.13	0.23	212	Marginals
ML Classification	0.1	0.35	116	Marginals
Deep Learning Security	0.06	0.19	220	Marginals
DeepSec AI	0.05	0.29	274	Marginals
Secure Learning	0.02	0.15	231	Marginals

language models such as ELMo, which enrich contextual semantic representation with direct applications in security. Taken together, these definitions reveal a broad spectrum, from theoretical foundations to advanced applications.

Researchers Wen, Shukla, and Katt [71] structure concepts from a functional approach, describing models such as Bayesian networks, MLPs, or CNNs in relation to their role in system assurance.

The diversity of definitions suggests the need for a unified framework that integrates the technical, functional, and applied dimensions, with potential for extrapolation to sectors such as health, finance, or education. Furthermore, the evolution toward advanced language models presents opportunities for transfer to different geographical and temporal contexts, adapting AI to local needs. Finally, the coexistence of approaches reinforces the importance of considering both theory and practice in future interdisciplinary research.

RQ5: *What are the dominant thematic areas in publications that apply Artificial Intelligence to Software Security?*

Figure 10 and Table 12 present the thematic distribution of the studies, where centrality indicates the relevance of the topics and density reflects their level of development. The analysis, based on keywords, makes it possible to identify both consolidated and emerging areas in research on artificial intelligence applied to software security.

The most consolidated topics correspond to Security Integration and Security Enumeration, both classified as Motors, with high centrality and density, evidencing their structuring role in the research agenda. In contrast, ML Security is positioned as a Basic theme, standing out for its

high centrality but lower density, which indicates growth potential. Most topics, such as Deep Vulnerability, Software Vulnerability, and DeepSec AI, appear as Marginals, with low density and centrality, reflecting areas still at an early stage but with high citation volumes that demonstrate academic interest. Deep Learning Security and Secure Learning also remain at the margins, though linked to trends toward hybrid approaches. Overall, the results suggest that research combines consolidated lines with emerging ones that require further maturation.

According to Khaleel and colleagues [90], the topics that stand out in the quadrant of high centrality and density are intrusion detection, machine learning, adversarial attacks, detection systems, attack detection, deep learning, and attacker examples. This indicates that the central topics in the field are strongly linked to the use of artificial intelligence in cybersecurity. In contrast, concepts such as random forest and specific detection models appear in the quadrant of low density and centrality, suggesting lower relevance or presence in the current literature. The classification of topics as marginal-including DeepSec AI and Deep Vulnerability-should not necessarily be interpreted as a sign of irrelevance, but rather as an indicator of conceptual emergence.

Recent bibliometric analyses in the field of Information Systems show a similar pattern: while machine learning dominates research (68 out of 98 studies), other categories such as robotics or hybrid approaches appear with low frequency despite their high disruptive potential. This “relative scarcity” in certain domains does not imply lack of value, but instead an opportunity to expand research toward more integral, ethical, and human-centered approaches-precisely the space occupied by emerging topics such as DeepSec AI in the domain of software security [86].

These findings suggest that future research should strengthen marginal topics to integrate them into the core of the discipline, especially deep learning–based security and complex vulnerabilities. At the applied level, this framework can be replicated in sectors such as healthcare and finance, where the identification of motor and marginal areas guides R&D investment. Finally, expanding the analysis to other geographical and temporal contexts will allow capturing the evolution of thematic priorities in software security.

5 Conclusions and Future Research

The results of this review reveal clear patterns in how the literature addresses software security through artificial intelligence. There is a marked tendency toward immediate technical measures, while more comprehensive approaches that combine multiple dimensions of protection have been less explored.

Within this framework, **RQ1** shows that the most frequently addressed criteria are access control and data integrity, in contrast to the limited attention given to confidentiality and vulnerability management.

Second, the evidence associated with **RQ2** shows that machine learning and deep learning are the most widely implemented technologies in the literature, confirming their central role in threat detection and classification. However, emerging techniques such as reinforcement learning and static/dynamic code analysis remain underexplored, which reveals opportunities to broaden the methodological spectrum of AI-based software security.

The findings related to **RQ4** show that there is no single definition of software security or artificial intelligence, but multiple approaches ranging from preventive and corrective frameworks to functional, technical, and applied perspectives.

This conceptual diversity reinforces the need to build integrative frameworks that allow greater theoretical coherence and facilitate the comparison of results across different studies and contexts.

With respect to **RQ5**, the results show that the motor thematic areas are centered on *Security Integration* and *Security Enumeration*, which exhibit high density and centrality. By contrast, most topics appear as marginal, such as *Deep Vulnerability* and *DeepSec AI*, reflecting research lines that are still incipient but with high academic interest, as shown by their citation figures. This configuration suggests a research agenda that combines consolidated fields with others that are still in the process of maturation.

Another important aspect is that the software evaluation criteria identified in RQ1 have a practical counterpart in the implementation of technologies from RQ2, confirming an alignment between security metrics and the AI tools applied. This coherence supports the design of more robust systems and provides a working framework applicable to different environments.

Likewise, the multiple definitions collected in RQ4 provide a theoretical foundation for the thematic areas highlighted in RQ5. Conceptual diversity translates into thematic diversity, reinforcing the notion that the field is still under construction and needs to consolidate its terminology and categories. Finally, the convergence of findings from the four RQs analyzed reveals that artificial intelligence applied to software security has reached a stage of consolidation in the academic literature, although gaps remain that must be addressed to achieve a more comprehensive and sustainable development of the field.

Future research should broaden the study of emerging AI techniques, integrating less explored security criteria such as confidentiality and vulnerability management. Moreover, it is recommended to deepen work on marginal topics identified in the thematic map in order to transform them into central research lines.

Finally, it is necessary to expand the geographical and sectoral coverage of studies, applying these approaches to fields such as healthcare, finance, transportation, and e-government, to enrich knowledge transfer and foster international comparison.

References

1. **Alhamyani, R., Alshammari, M. (2024).** Machine learning-driven detection of cross-site scripting attacks. Information, 15(7), pp. 420. DOI: 10.3390/ info15070420.
2. **Anas, A., Alhelbawy, A. A., Gamal, S. El, Youssef, B. (2024).** BACAD: AI-based framework for detecting vertical broken access control attacks. Egyptian Informatics Journal, 28, 100571. DOI: 10.1016/j.eij.2024.100571.
3. **Arasteh, B., Aghaei, B., Farzad, B., Arasteh, K., Kiani, F., & Torkamanian-Afshar, M. (2024).** Detecting SQL injection attacks by binary gray wolf optimizer and machine learning algorithms. Neural Computing and Applications, 36(12), pp. 6771–6792. DOI: 10.1007/s00521-024-09429-z.
4. **Arikan, K. E., Yilmaz, E. N. (2025).** AndroCom: A real-world Android applications' vulnerability dataset to assist with automatically detecting vulnerabilities. Applied Sciences, 15(5), pp. 2665. DOI: 10.3390/ app15052665.
5. **Arora, R., Kaur, A. (2025).** Automated categorization of bug reports as security related or non-security related: A machine learning-based solution. Vietnam Journal of Computer Science, 0, pp. 1–29. DOI: 10.1142/ S2196888825500022.
6. **Bacha, N. U., Lu, S., Ur Rehman, A., Idrees, M., Ghadi, Y. Y., Alahmadi, T. J. (2024).** Deploying hybrid ensemble machine learning techniques for effective cross-site scripting (XSS) attack detection. Computers, Materials and Continua, 81(1), pp. 707–748. DOI: 10.32604/cmc.2024.054780.
7. **Bahaa, A., Kamal, A., Fahmy, H., Ghoneim, A. S. (2024).** DB-CBIL: A DistilBert-based transformer hybrid model using CNN and BiLSTM for software vulnerability detection. IEEE Access, 12, pp. 64446–64460. DOI: 10.1109/ACCESS.2024. 3396410.
8. **Bakır, R. (2025).** UniEmbed: A novel approach to detect XSS and SQL injection attacks leveraging multiple feature fusion with machine learning techniques. Arabian Journal for Science and Engineering. DOI: 10.1007/s13369-024-09916-4.
9. **Bakır, R., Bakır, H. (2024).** Swift detection of XSS attacks: Enhancing XSS attack detection by leveraging hybrid semantic embeddings and AI techniques. Arabian Journal for Science and Engineering, 50(2), pp. 1191–1207. DOI: 10.1007/s13369-024-09140-0.
10. **Bedoya, M., Palacios, S., Díaz-López, D., Laverde, E., Nespoli, P. (2024).** Enhancing DevSecOps practice with large language models and security chaos engineering. International Journal of

Information Security, 23(6), pp. 3765–3788. DOI: 10.1007/s10207-024-00909-w.

11. **Bilgin, Z., Ersoy, M. A., Soykan, E. U., Tomur, E., Comak, P., Karacay, L. (2020).** Vulnerability prediction from source code using machine learning. IEEE Access, 8, pp. 150672–150684. DOI: 10.1109/ACCESS.2020.3016774.
12. **Charmanas, K., Mittas, N., Angelis, L. (2023).** Exploitation of vulnerabilities: A topic-based machine learning framework for explaining and predicting exploitation. Information, 14(7), pp. 403. DOI: 10.3390/info14070403.
13. **Chen, T., Chen, Y., Lv, M., He, G., Zhu, T., Wang, T., Weng, Z. (2021).** A payload based malicious http traffic detection method using transfer semi-supervised learning. Applied Sciences, 11(16), pp. 7188. DOI: 10.3390/app11167188.
14. **Ćirković, S., Mladenović, V., Tomić, S., Drljača, D., Ristić, O. (2025).** Utilizing fine-tuning of large language models for generating synthetic payloads: Enhancing web application cybersecurity through innovative penetration testing techniques. Computers, Materials and Continua, 82(3), pp. 4409–4430. DOI: 10.32604/cmc.2025.059696.
15. **Corona-Fraga, P., Hernandez-Suarez, A., Sanchez-Perez, G., Toscano-Medina, L. K., Perez-Meana, H., Portillo-Portillo, J., Olivares-Mercado, J., García Villalba, L. J. (2025).** Question–answer methodology for vulnerable source code review via prototype-based model-agnostic meta-learning. Future Internet, 17(1), pp. 33. DOI: 10.3390/fi17010033.
16. **Demilie, W. B., Deriba, F. G. (2022).** Detection and prevention of SQLI attacks and developing compressive framework using machine learning and hybrid techniques. Journal of Big Data, 9(1), pp. 92. DOI: 10.1186/s40537-022-00678-0.
17. **Elsayed, M. A., Zulkernine, M. (2020).** PredictDeep: Security analytics as a service for anomaly detection and prediction. IEEE Access, 8, pp. 45184–45197. DOI: 10.1109/ACCESS.2020.2977325.
18. **Fu, M., Tantithamthavorn, C., Le, T., Kume, Y., Nguyen, V., Phung, D., Grundy, J. (2024).** AIBugHunter: A practical tool for predicting, classifying and repairing software vulnerabilities. Empirical Software Engineering, 29(1). DOI: 10.1007/s10664-023-10346-3.
19. **Gear, J., Xu, Y., Foo, E., Gauravaram, P., Jadidi, Z., Simpson, L. (2023).** Software vulnerability detection using informed code graph pruning. IEEE Access, 11, 135626–135644. DOI: 10.1109/ACCESS.2023.3338162.
20. **González-Manzano, L., & Garcia-Alfaro, J. (2025).** Software vulnerability detection under poisoning attacks using CNN-based image processing. International Journal of Information Security, 24(2), pp. 1–22. DOI: 10.1007/s10207-025-00989-2.
21. **Grahn, D., Chen, L., Zhang, J. (2024).** VulMixer: Efficient and effective machine learning–assisted software vulnerability detection. Electronics, 13(13), pp. 2538. DOI: 10.3390/electronics13132538.
22. **Guo, Y., Bettaieb, S., Casino, F. (2024).** A comprehensive analysis on software vulnerability detection datasets: Trends, challenges, and road ahead. International Journal of Information Security, 23(5), pp. 3311–3327. DOI: 10.1007/s10207-024-00888-y.
23. **Han, S., Nam, H., Kang, J., Kim, K., Cho, S., Lee, S. (2024).** CODE-SMASH: Source-code vulnerability detection using Siamese and multi-level neural architecture. IEEE Access, 12, pp. 102492–102504. DOI: 10.1109/ACCESS.2024.3432323.
24. **Hussain, S., Nadeem, M., Baber, J., Hamdi, M., Rajab, A., Al Reshan, M. S., Shaikh, A. (2024).** Vulnerability detection in Java source code using a quantum convolutional neural network with self-attentive pooling, deep sequence, and graph-based hybrid feature extraction. Scientific Reports, 14(1), pp. 56871. DOI: 10.1038/s41598-024-56871-z.
25. **Kakisim, A. G. (2024).** A deep learning approach based on multi-view consensus for SQL injection detection. International Journal

of Information Security, 23(2), pp. 1541–1556. DOI: 10.1007/s10207-023-00791-y.

26. **Kekül, H., Ergen, B., Arslan, H. (2022).** A multiclass approach to estimating software vulnerability severity rating with statistical and word embedding methods. International Journal of Computer Network and Information Security, 14(4), pp. 27–42. DOI: 10.5815/ijcnis.2022.04.03.
27. **Khan, H. U., Khan, R. A., Alwageed, H. S., Almagrabi, A. O., Ayouni, S., Maddeh, M. (2025).** AI-driven cybersecurity framework for software development based on the ANN-ISM paradigm. Scientific Reports, 15(1). DOI: 10.1038/s41598-025-97204-y.
28. **Kim, T. H., Srinivasulu, A., Chinthaginjala, R., Dhakshayani, J., Zhao, X., Obaidur Rab, S. (2025).** Enhancing cybersecurity through script development using machine and deep learning for advanced threat mitigation. Scientific Reports, 15(1). DOI: 10.1038/s41598-025-92676-4.
29. **Koala, G., Bassolé, D., Tiendrebeogo, T., Sié, O. (2023).** Software vulnerabilities' detection by analysing application execution traces. International Journal of Advanced Computer Science and Applications, 14(6), pp. 1288–1294. DOI: 10.14569/IJACSA.2023.01406136.
30. **Kraker, W. De, Vranken, H. (2025).** MultiGLICE: Combining graph neural networks and program slicing for multiclass software vulnerability detection †. Computers, Materials and Continua, pp. 1–23.
31. **Leka, E., Lamani, L., Aliti, A., Hoxha, E. (2024).** Web application firewall for detecting and mitigation of based DDoS attacks using machine learning and blockchain. TEM Journal, 13(4), pp. 2802–2811. DOI: 10.18421/TEM134-17.
32. **Li, D., Liu, Y., Huang, J. (2024).** Assessment of software vulnerability contributing factors by model-agnostic explainable AI. Machine Learning and Knowledge Extraction, 6(2), pp. 1087–1113. DOI: 10.3390/make6020050.
33. **Li, X., Wang, T., Zhang, W., Niu, X., Zhang, T., Zhao, T., Wang, Y., Wang, Y. (2023).** An LSTM based cross-site scripting attack detection scheme for cloud computing environments. Journal of Cloud Computing, 12(1). DOI: 10.1186/s13677-023-00483-x.
34. **Li, X., Tang, Y., Christo, M. S., Zhao, Z., & Li, Y. (2022).** Android malware application detection method based on RGB image features in e-commerce. Journal of Internet Technology, 23(6), pp. 1343–1352. DOI: 10.53106/160792642022112306017.
35. **Liang, C., Wei, Q., Du, J., Wang, Y., Jiang, Z. (2025).** Survey of source code vulnerability analysis based on deep learning. Computers and Security, 148, pp. 104098. DOI: 10.1016/j.cose.2024.104098.
36. **Liu, C., Lu, J., Feng, W., Du, E., Di, L., Song, Z. (2023).** MOBIPCR: Efficient, accurate, and strict ML-based mobile malware detection. Future Generation Computer Systems, 144, pp. 140–150. DOI: 10.1016/j.future.2023.02.014.
37. **Luu, G. H., Duong, M. K., Pham-Ngo, T. P., Ngo, T. S., Nguyen, D. T., Nguyen, X. H., Le, K. H. (2024).** XSShield: A novel dataset and lightweight hybrid deep learning model for XSS attack detection. Results in Engineering, 24, 103363. DOI: 10.1016/j.rineng.2024.103363.
38. **Malatji, M., Tolah, A. (2024).** Artificial intelligence (AI) cybersecurity dimensions: A comprehensive framework for understanding adversarial and offensive AI. AI and Ethics, 5(2), pp. 883–910. DOI: 10.1007/s43681-024-00427-4.
39. **Manzil, H. H. R., Manohar Naik, S. (2023).** Android malware category detection using a novel feature vector-based machine learning model. Cybersecurity, 6(1). DOI: 10.1186/s42400-023-00139-y.
40. **Marashdih, A. W., Zaaba, Z. F., Suwais, K. (2022).** Predicting input validation vulnerabilities based on minimal SSA features and machine learning. Journal of King Saud University - Computer and Information Sciences, 34(10), pp. 9311–9331. DOI: 10.1016/j.jksuci.2022.09.010.
41. **Masud, M. T., Keshk, M., Moustafa, N., Linkov, I., Emge, D. K. (2024).** Explainable artificial intelligence for resilient security applications in the Internet of Things. IEEE

Open Journal of the Communications Society, 6, pp. 2877–2906.

42. **McLaughlin, C., Lu, Y. (2024).** Multi-class vulnerability prediction using value flow and graph neural networks. Neural Computing and Applications, 36(25), pp. 15869–15891. DOI: 10.1007/s00521-024-09819-3.
43. **Medeiros, N., Ivaki, N., Costa, P., Vieira, M. (2020).** Vulnerable code detection using software metrics and machine learning. IEEE Access, 8, pp. 219174–219198. DOI: 10.1109/ACCESS.2020.3041181.
44. **Misalkar, H. D., Harshavardhanan, P. (2024).** Assessing the efficacy of machine learning classifier for Android malware detection. Journal of Integrated Science and Technology, 12(4), pp. 1–17. DOI: 10.62110/sciencein.jist.2024.v12.788.
45. **Munonye, K., Péter, M. (2022).** Machine learning approach to vulnerability detection in OAuth 2.0 authentication and authorization flow. International Journal of Information Security, 21(2), pp. 223–237. DOI: 10.1007/s10207-021-00551-w.
46. **Pham, V. H., Hien, D. T. T., Chuong, N. P., Thai, P. T., Duy, P. T. (2024).** A coverage-guided fuzzing method for automatic software vulnerability detection using reinforcement learning-enabled multi-level input mutation. IEEE Access, 12, pp. 129064–129080. DOI: 10.1109/ACCESS.2024.3421989.
47. **Pooja, S., Chandrakala, C. B., Raju, L. K. (2022).** Developer's roadmap to design software vulnerability detection model using different AI approaches. IEEE Access, 10, pp. 75637–75656. DOI: 10.1109/ACCESS.2022.3191115.
48. **Rashid, M. U., Qureshi, S., Abid, A., Alqahtany, S. S., Alqazzaz, A., ul Hassan, M., Al Reshan, M. S., Shaikh, A. (2025).** Hybrid Android malware detection and classification using deep neural networks. International Journal of Computational Intelligence Systems, 18(1). DOI: 10.1007/s44196-025-00783-x.
49. **Senanayake, J., Kalutarage, H., Petrovski, A., Piras, L., Al-Kadri, M. O. (2024).** Defendroid: Real-time Android code vulnerability detection via blockchain federated neural network with XAI. Journal of Information Security and Applications, 82, 103741. DOI: 10.1016/j.jisa.2024.103741.
50. **Shaheed, A., Kurdy, M. H. D. B. (2022).** Web application firewall using machine learning and features engineering. Security and Communication Networks, 2022, 5280158. DOI: 10.1155/2022/5280158.
51. **Siewruk, G., Mazurczyk, W. (2021).** Context-aware software vulnerability classification using machine learning. IEEE Access, 9, pp. 88852–88867. DOI: 10.1109/ACCESS.2021.3075385.
52. **Sonnekalb, T., Heinze, T. S., Mäder, P. (2022).** Deep security analysis of program code: A systematic literature review. Empirical Software Engineering, 27(1). DOI: 10.1007/s10664-021-10029-x.
53. **Subhan, F., Wu, X., Bo, L., Sun, X., Rahman, M. (2022).** A deep learning-based approach for software vulnerability detection using code metrics. IET Software, 16(5), pp. 516–526. DOI: 10.1049/sfw2.12066.
54. **Szabó, Z., Bilicki, V. (2023).** A new approach to web application security: Utilizing GPT language models for source code inspection. Future Internet, 15(10), pp. 326. DOI: 10.3390/fi15100326.
55. **Tadhani, J. R., Vekariya, V., Sorathiya, V., Alshathri, S., El-Shafai, W. (2024).** Securing web applications against XSS and SQLi attacks using a novel deep learning approach. Scientific Reports, 14(1), pp. 48845. DOI: 10.1038/s41598-023-48845-4.
56. **Tang, G., Yang, L., Ren, S., Meng, L., Yang, F., Wang, H. (2021).** An automatic source code vulnerability detection approach based on KELM. Security and Communication Networks, 2021, 5566423. DOI: 10.1155/2021/5566423.
57. **Tang, X., Du, Y., Lai, A., Zhang, Z., Shi, L. (2023).** Deep learning-based solution for smart contract vulnerabilities detection. Scientific Reports, 13(1), pp. 47219. DOI: 10.1038/s41598-023-47219-0.

58. **Wang, X., Zhai, J., Yang, H. (2024).** Detecting command injection attacks in web applications based on novel deep learning methods. Scientific Reports, 14(1), pp. 74350. DOI: 10.1038/s41598-024-74350-3.
59. **Wu, B., Zou, F. (2022).** Code vulnerability detection based on deep sequence and graph models: A survey. Security and Communication Networks, 2022, 1176898. DOI: 10.1155/2022/1176898.
60. **Wu, G., Tang, H. (2023).** Binary code vulnerability detection based on multi-level feature fusion. IEEE Access, 11, 63904–63915. DOI: 10.1109/ACCESS.2023.3289001.
61. **Xing, X., Jin, X., Elahi, H., Jiang, H., Wang, G. (2022).** A malware detection approach using autoencoder in deep learning. IEEE Access, 10, 25696–25706. DOI: 10.1109/ACCESS.2022.3155695.
62. **Xu, R., Tang, Z., Ye, G., Wang, H., Ke, X., Fang, D., Wang, Z. (2022).** Detecting code vulnerabilities by learning from large-scale open source repositories. Journal of Information Security and Applications, 69, 103293. DOI: 10.1016/j.jisa.2022.103293.
63. **Yahya, H. M., Taha, D. B. (2024).** The development of the secure quality dataset (SQDS): Combining security and quality measures using deep machine learning for code smell detection. International Journal of Computing and Digital Systems, 16(1), pp. 995–1006. DOI: 10.12785/ijcds/160172.
64. **Yang, Y., Zhou, X., Mao, R., Xu, J., Yang, L., Zhangm, Y., Shen, H., Zhang, H. (2024).** DLAP: A deep learning augmented large language model prompting framework for software vulnerability detection. The Journal of Systems & Software, 219, 112234. DOI: 10.1016/j.jss.2024.112234.
65. **Yaser, A. L., Mousa, H. M., Hussein, M. (2022).** Improved DDoS detection utilizing deep neural networks and feedforward neural networks as autoencoder. Future Internet, 14(8), 240. https://doi.org/10.3390/fi14080240
66. **Yuan, X., Lin, G., Tai, Y., Zhang, J. (2022).** Deep neural embedding for software vulnerability discovery: Comparison and optimization. Security and Communication Networks, 2022, 5203217. https://doi.org/10.1155/2022/5203217
67. **Zaharia, S., Rebedea, T., Trausan-Matu, S. (2022).** Machine learning-based security pattern recognition techniques for code developers. Applied Sciences, 12(23), 12463. DOI: 10.3390/app122312463.
68. **Zaharia, S., Rebedea, T., Trausan-Matu, S. (2023).** Detection of software security weaknesses using cross-language source code representation (CLaSCoRe). Applied Sciences, 13(13), 7871. DOI: 10.3390/app13137871.
69. **Zhang, W., Li, Y., Li, X., Shao, M., Mi, Y., Zhang, H., Zhi, G. (2022).** Deep neural network-based SQL injection detection method. Security and Communication Networks, 2022, pp. 4836289. DOI: 10.1155/2022/4836289.
70. **Zhao, S., Zhu, J., Peng, J. (2024).** Software vulnerability mining and analysis based on deep learning. Computers, Materials and Continua, 80(2), pp. 3263–3287. DOI: 10.32604/ cmc.2024.041949.
71. **Wen, S.-F., Shukla, A., Katt, B. (2025).** Artificial intelligence for system security assurance: A systematic literature review. International Journal of Information Security, 24(43).
72. **Yitagesu, S., Xing, Z., Zhang, X., Feng, Z., Bi, T., Han, L., Li, X. (2025).** Systematic literature review on software security vulnerability information extraction. ACM Transactions on Software Engineering and Methodology.
73. **Mishra, N., Pandya, S. (2021).** Internet of Things applications, security challenges, attacks, intrusion detection, and future visions: A systematic review. IEEE Access, 9, pp. 59353–59377. DOI: 10.1109/ACCESS.2021.3073408.
74. **Miller, T., Durlik, I., Kostecka, E., Sokołowska, S., Kozlovska, P., Zwolak, R. (2025).** Artificial intelligence in maritime cybersecurity: A systematic review of AI-driven threat detection and risk mitigation strategies. Electronics, 14(9), 1844.

75. **León-Pérez, J., Bocangel-Rivera, E., Niño Montero, J., Gamboa-Cruzado, J., Soto Soto, L., Oseda Gago, D. (2021).** Revisión sistemática de la literatura sobre redes neuronales artificiales: Detección de ataques cardíacos. Revista Ibérica de Sistemas e Tecnologias de Informação (RISTI), E45, 303–317.
76. **Achuthan, K., Ramanathan, S., Srinivas, S., & Raman, R. (2024).** Advancing cybersecurity and privacy with artificial intelligence: Current trends and future research directions. Frontiers in Big Data, 7, 1497535.
77. **Bassi, D., Singh, H. (2023).** A systematic literature review on software vulnerability prediction models. IEEE Access, 11, 110289–110311. DOI: 10.1109/ACCESS.2023.3312613.
78. **Zhang, J., Bu, H., Wen, H., Liu, Y., Fei, H., Xi, R., Li, L., Yang, Y., Zhu, H., Meng, D. (2025).** When LLMs meet cybersecurity: A systematic literature review. Cybersecurity, 8(1). DOI: 10.1186/s42400-025-00361-w.
79. **Senanayake, J., Kalutarage, H., Al-Kadri, M. O., Petrovski, A., Piras, L. (2023).** Android source code vulnerability detection: A systematic literature review. ACM Computing Surveys, 55(9). DOI: 10.1145/3556974.
80. **Espinoza Villavicencio, H., Gamboa-Cruzado, J., López-Goycochea, J., Soto Soto, L. (2024).** The role of artificial intelligence in the diagnosis of neoplastic diseases: A systematic and bibliometric review. International Journal of Online and Biomedical Engineering (iJOE), pp. 43–68. DOI: 10.3991/ijoe.v20i04.45429.
81. **Guembe, B., Misra, S., Azeta, A., Lopez-Baldominos, I. (2025).** Bibliometric analysis of artificial intelligence cyberattack detection models. Artificial Intelligence Review, 58(6). DOI: 10.1007/s10462-025-11167-0.
82. **Koca, M., Çiftçi, S. (2025).** A comprehensive bibliometric analysis of Big Data and Cyber Security: Intellectual structure, trends, and global collaborations. Knowledge and Information Systems. DOI: 10.1007/s10115-025-02531-1.
83. **Shiri Harzevili, N., Boaye Belle, A., Wang, J., Wang, S., Jiang, Z. M., Nagappan, N. (2024).** A systematic literature review on automated software vulnerability detection using machine learning. ACM Computing Surveys, 57(3). DOI: 10.1145/3699711.
84. **Uddin, M. P., Xiang, Y., Hasan, M., Bai, J., Zhao, Y., & Gao, L. (2025).** A systematic literature review of robust federated learning: Issues, solutions, and future research directions. ACM Computing Surveys, 57(10), 1–62. https://doi.org/10.1145/3727643
85. **Yaseen, S. G., Albadrany, Q. H. A., Dajani, D., Al-Afaishat, M. M. (2025).** Artificial intelligence regulation: A bibliometric analysis. International Journal of Advances in Soft Computing and Its Applications, 17(1), pp. 217–232. DOI: 10.15849/IJASCA.250330.12.
86. **Collins, C., Dennehy, D., Conboy, K., Mikalef, P. (2021).** Artificial intelligence in information systems research: A systematic literature review and research agenda. International Journal of Information Management, 60, 102383. DOI: 10.1016/j.ijinfomgt.2021.102383.
87. **Cárdenas-Quispe, A., Vergaray-Mezarina, R., Gamboa-Cruzado, J. (2021).** Machine Learning para la detección de malware en Android: Revisión sistemática de la literatura. Revista Ibérica de Sistemas e Tecnologias de Informação, E45, pp. 318– 331.
88. **Aparcana-Tasayco, A. J., Gamboa-Cruzado, J. (2022).** Machine learning for management in software-defined networks: A systematic literature review. IEIE Transactions on Smart Processing and Computing, 11(6), pp. 400–411. DOI: 10.5573/IEIESPC.2022.11.6.400.
89. **Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., Linkman, S. (2009).** Systematic literature reviews in software engineering – A systematic literature review. Information and Software Technology, 51(1), pp. 7–15.
90. **Khaleel, Y. L., Habeeb, M. A., Albahri, A. S., Al-Quraishi, T., Albahri, O. S., Alamoodi, A. H. (2024).** Network and cybersecurity applications of defense in adversarial attacks:

A state-of-the-art using machine learning and deep learning methods. Journal of Intelligent Systems.

91. **Bernardelli, A. E., Giudici, P. (2025).** AI risk management: A bibliometric analysis. Risks, 13(7), pp. 131. DOI:10.3390/risks1307013.
92. **Gamboa-Cruzado, J., Cuya-Chuica, L., López-Goycochea, J., Nuñez-Meza, A., Del-Valle-Jurado, C. (2024).** Impact of 5G technology on cybersecurity: A comprehensive systematic and bibliometric review. Computación y Sistemas, 28(2), pp. 367–386. DOI: 10.13053/CyS-28-2-4734.

Article received on 03/07/2025; accepted on 11/10/2025.
**Corresponding author is Francisco A. Castillo-Velásquez.*