ISBN: 978-93-47587-47-4

COMPUTING BEYOND ALGORITHMS

AI, Data and Intelligent Systems

Editors:
Dr. Pawan Kumar Dixit
Dr. Ponmani Swaminathan
Mr. Udaya Bhaskar Vemuri
Ms. R. Deivanayaki

BHUMI PUBLISHING, INDIA
FIRST EDITION: JULY 2026

Computing Beyond Algorithms: AI, Data and Intelligent Systems

(ISBN: 978-93-47587-47-4)

DOI: https://doi.org/10.5281/zenodo.21701430

Editors

Dr. Pawan Kumar Dixit

Department of Mathematics,

Maharishi School of Engineering and

Technology, Maharishi University of

Information Technology, Lucknow, U. P.

Dr. Ponmani Swaminathan

Department of Petroleum Engineering,

Academy of Maritime Education and

Training (AMET) University,

Chennai, Tamil Nadu

Mr. Udaya Bhaskar Vemuri

Application Security Professional,

DevSecOps,

Secure Software Development

Ms. R. Deivanayaki

Department of Electrical and Electronics

Engineering, Academy of Maritime

Education and Training (AMET), Chennai

Bhumi Publishing

July 2026

Title: Computing Beyond Algorithms: AI, Data and Intelligent Systems

Editors: Dr. Pawan Kumar Dixit, Dr. Ponmani Swaminathan,

Mr. Udaya Bhaskar Vemuri, Ms. R. Deivanayaki

First Edition: July 2026

ISBN: 978-93-47587-47-4

DOI: https://doi.org/10.5281/zenodo.21701430

Published by Bhumi Publishing,

a publishing unit of Bhumi Gramin Vikas Sanstha

Bhumi Publishing

Nigave Khalasa, Tal – Karveer, Dist – Kolhapur, Maharashtra, INDIA 416 207

E-mail: bhumipublishing@gmail.com

PREFACE

Artificial intelligence, data-driven innovation, and intelligent systems are reshaping every dimension of modern society, extending computing far beyond traditional algorithmic problem solving. Today, computational technologies learn from data, adapt to changing environments, and support decisions across healthcare, education, agriculture, finance, manufacturing, governance, and scientific research. *Computing Beyond Algorithms: AI, Data and Intelligent Systems* presents contemporary perspectives that examine these transformative developments while emphasizing their scientific foundations, practical applications, and societal implications.

This volume brings together contributions from researchers, academicians, practitioners, and innovators representing diverse disciplines. The chapters explore advances in artificial intelligence, machine learning, deep learning, data analytics, cloud computing, cybersecurity, Internet of Things, intelligent automation, computer vision, natural language processing, and decision support systems. Together, these studies illustrate how intelligent computing is creating opportunities for innovation, improving efficiency, and addressing complex global challenges through interdisciplinary collaboration.

The primary objective of this book is to provide readers with an updated understanding of emerging computational technologies while encouraging responsible innovation, ethical implementation, and evidence-based research. Designed for students, educators, scientists, engineers, industry professionals, and policymakers, this collection offers valuable insights into both theoretical concepts and real-world applications that define the future of intelligent systems.

We sincerely thank all contributing authors for sharing their expertise and original research. We also express our gratitude to the reviewers for their constructive suggestions, which have strengthened the quality of every chapter. Finally, we acknowledge the editorial team and the publisher for their dedication and professionalism in bringing this volume to completion. We hope this book inspires curiosity, collaboration, innovation, and meaningful advances in intelligent computing for the benefit of society while fostering inclusive, sustainable, trustworthy, human-centered technologies for future generations across the world responsibly.

- Editors

TABLE OF CONTENT

Computing Beyond Algorithms: AI, Data and Intelligent Systems
(ISBN: 978-93-47587-47-4)

SOCIETAL IMPACT OF AI: EQUITY, INCLUSION, AND JOB-RELATED CURRENT RESEARCH AND PROPOSED SOLUTIONS

D. Bhuvanewari and V Shoba
Department of Computer Science,
SRM Arts and Science College, Kattankulathur-603 203
Corresponding author E-mail: bhuvaneswarics@srmasc.ac.in, shobacs@srmasc.ac.in

Abstract
The rapid integration of artificial intelligence (AI) across sectors has profound implications for society, particularly in areas of equity, inclusion, and employment. This study adopts a multi-method approach to comprehensively evaluate the social impact of AI. It begins with a systematic literature review and gap analysis to synthesize existing knowledge and identify research deficits. Algorithmic and data audits are conducted to detect biases in AI models and datasets, ensuring fairness and transparency. Surveys, interviews, and case studies gather qualitative insights from stakeholders affected by AI deployment, while quantitative modeling and simulations assess potential labor market and societal outcomes. Policy and regulatory analyses examine the effectiveness of existing frameworks, and participatory co-design methods involve communities in shaping equitable AI solutions. Longitudinal and mixed-methods studies provide a holistic view of evolving social impacts. By integrating these methodologies, the research aims to generate actionable recommendations for developing AI systems that are inclusive, ethically responsible, and socially beneficial.

Keywords: Equity, Inclusion, Algorithmic Fairness, Workforce Transformation, Ethical Ai, Governance, Bias Mitigation, Reskilling, Sustainable Development.

1. Introduction

Artificial intelligence (AI) has become a pervasive technological force reshaping industries and societies worldwide. Driven by advancements in machine learning, natural language processing, and robotics, AI applications now extend across healthcare, finance, education, and transportation. While AI offers tremendous potential for improving efficiency and quality of life, it also poses significant societal challenges. Key among these are concerns regarding equity and inclusion, as AI systems may inadvertently reinforce existing inequalities through biased algorithms and unequal access to technology. Furthermore, AI-driven automation threatens labor markets, raising questions about job security and workforce displacement.

Recent research has focused on quantifying the extent of AI's societal impact, emphasizing the dual-edged nature of innovation. Studies reveal both improvements in service delivery and exacerbation of socio-economic divides. For example, marginalized groups often face

disproportionate harms from biased predictive policing or hiring algorithms. Simultaneously, the deployment of AI in workplaces risks amplifying unemployment for low- and mid-skilled workers, necessitating urgent policy and technical interventions. While some frameworks address fairness in AI, comprehensive solutions balancing equity with technological progress remain an open challenge.

This paper aims to contribute to this conversation by reviewing current research on AI's societal impact with an emphasis on equity and job-related issues. We identify critical gaps concerning the inclusivity of AI systems and propose novel methodologies for mitigating adverse outcomes. Through detailed algorithmic implementation steps and empirical workflow evaluations, the study seeks to advance actionable insights for ethical AI deployment. In doing so, it engages with interdisciplinary scholarship, combining perspectives from computer science, sociology, and labor economics to address the multifaceted nature of AI's societal effects.

1.1 Problem Statement

The rapid integration of artificial intelligence (AI) into various societal domains, while promising unprecedented advancements, concurrently exposes underlying systemic inequities. Major concerns revolve around AI's role in perpetuating or exacerbating societal disparities, particularly regarding equity and inclusion. AI algorithms, often trained on biased historical data, run the risk of reinforcing entrenched social prejudices. This phenomenon manifests in discriminatory outcomes in high-impact areas such as hiring, credit scoring, law enforcement, and access to healthcare. These biases not only marginalize vulnerable populations but also erode trust in AI technologies, posing ethical and legal challenges for stakeholders.

Additionally, the increasing automation driven by AI raises significant employment-related issues, notably job displacement. While AI enhances productivity and creates new job categories, it simultaneously threatens traditional employment structures, especially for low- and mid-skilled labor sectors. The pace of AI adoption often outstrips regulatory and social safety measures, leaving affected workers without adequate reskilling opportunities or economic support. The cumulative effect threatens to widen socio-economic divides, raising pressing questions about how AI can be harnessed in a manner that supports inclusive economic growth rather than exacerbating inequality.

This research paper addresses these intertwined problems by examining current evidence on AI's societal impacts, focusing on equity, inclusion, and job-related outcomes. It aims to clarify the challenges, identify gaps in the existing research landscape, and propose a structured methodology to develop fair and sustainable AI deployments. The ultimate goal is to contribute to responsible AI paradigms that uphold ethical standards and promote societal well-being.

1.2 Problem Identification

AI's societal impact can be categorized into algorithmic bias and job-related displacement two primary domains where inequities manifest. Algorithmic bias occurs when AI systems inadvertently discriminate against specific populations, either through skewed training data or flawed design choices. This leads to unequal treatment in critical sectors, undermining the principles of fairness and social justice. Identifying the root causes of bias is essential for designing corrective interventions that mitigate its harmful consequences.

On the employment front, AI-driven automation affects labor market dynamics by substituting human labor with machines in routine, repetitive tasks. While this improves efficiency, it disproportionately impacts certain demographic groups that lack access to education and reskilling pathways. The identification of vulnerable occupations and worker profiles is crucial to understanding the scope and scale of job displacement risks. This facilitates targeted policy responses aimed at workforce transition and social protection. The intersection of these problems highlights a broader societal challenge: ensuring that AI technologies support, rather than undermine, equity and inclusion. Effective problem identification involves a comprehensive assessment of AI's structural biases and labor market effects, serving as a prerequisite for thoughtful research and innovative methodological solutions.

1.3 Research Gap

Despite growing scholarship, significant gaps persist in understanding and addressing AI's societal impacts comprehensively. Much of the current research focuses narrowly on specific applications or technical fairness metrics without integrating broader socio-economic contexts. There is a limited exploration of how AI-induced job displacement affects different social strata or regional economies, hindering holistic policy formulation.

Furthermore, existing mitigation strategies often emphasize algorithmic tweaks or ethical guidelines without robust empirical validation or systematic workflow integration. This disconnect between theoretical fairness frameworks and operational realities limits the effectiveness of solutions. Research that simultaneously considers equity, inclusion, and job sustainability aspects of AI is relatively sparse. This paper aims to bridge these gaps by proposing a multi-disciplinary methodology combining algorithmic design, socio-economic analysis, and implementation workflows. By situating AI impacts within a broader societal framework, the study endeavors to foster research that is both technically sound and socially relevant.

1.4 Proposed Problem

The current rapid development and deployment of AI technologies have sparked extensive debate about their societal consequences, particularly regarding equity, inclusion, and employment challenges. The problem lies in balancing innovation with responsible implementation. On one hand, there is an urgent need to harness AI's transformative potential to drive economic growth

and social welfare. On the other hand, unchecked AI adoption risks deepening systemic inequalities through biased algorithms and labor market disruptions.

The specific problem addressed in this study is the lack of comprehensive frameworks that integrate equity considerations with job-related implications in AI systems. Many approaches narrowly focus on technical algorithmic fairness without adequately addressing socio-economic realities. Moreover, policy responses targeting job displacement are often disconnected from AI design and implementation strategies. This research proposes a holistic framework combining technical methodologies, implementation workflows, and policy insights to mitigate AI-driven inequities while facilitating inclusive employment transitions. By centering equity and labor market sustainability, the proposed solution aims to support ethical AI deployment aligned with societal values.

2. Literature Review

Algorithmic Bias and Fairness-Recent studies emphasize that AI systems often reproduce and intensify existing societal inequalities through biased data and algorithmic design. Bias in facial recognition, hiring systems, and predictive analytics disproportionately affects women and minority groups. For example, large-scale reviews demonstrate that commercial facial recognition algorithms have higher error rates for darker-skinned individuals, underscoring the need for equitable data representation [1], [4]. Research between 2020 and 2025 focuses on fairness-aware machine learning (FAML) and bias-mitigation frameworks, which integrate fairness constraints during data preprocessing and model training [1], [3]. These approaches aim to enhance transparency, explainability, and accountability in AI deployment across social sectors such as healthcare, education, and law enforcement.

Inclusion and Participatory AI Design-Scholars argue that inclusion in AI development extends beyond bias mitigation to encompass diverse representation in data curation, design teams, and decision-making structures. Studies highlight that the absence of women, minority, and multidisciplinary contributors in AI projects increases the risk of cultural bias and value misalignment [4], [5]. Policy-oriented literature emphasizes participatory AI, where affected communities co-design systems to reflect their needs and ethical priorities. The European Union's Ethics Guidelines for Trustworthy AI and UNESCO's Recommendation on the Ethics of AI (2021) advocate inclusivity and stakeholder consultation to promote social well-being [6], [7].

Job Displacement and Workforce Transformation-AI-driven automation continues to disrupt global labor markets by eliminating routine tasks and generating new roles requiring advanced digital literacy. According to the World Economic Forum's Future of Jobs Report (2023), approximately 85 million jobs may be displaced while 97 million new technology-driven roles could emerge by 2025 [5]. Sectoral studies reveal mixed impacts: in manufacturing, automation has replaced repetitive assembly roles but increased demand for robotics engineers and AI system

supervisors. In banking, AI chatbots manage standard interactions, allowing employees to focus on complex decision-making and client engagement [5], [6]. Researchers stress the urgent need for reskilling initiatives, especially in developing economies, to prevent workforce polarization and social inequality [6], [8].

Governance, Ethics, and Policy Frameworks-Between 2020 and 2025, international bodies such as OECD, UNESCO, and the World Economic Forum have strengthened ethical frameworks to guide AI governance. These frameworks prioritize transparency, accountability, fairness, and human oversight in algorithmic decision-making [6], [7], [9]. The OECD's Governing with Artificial Intelligence (2024) outlines readiness gaps in public institutions and calls for enhanced oversight capabilities [6]. Similarly, Roberts *et al.* (2024) propose mechanisms for global AI governance, emphasizing ethical audits and cross-border cooperation [7]. Researchers also recommend national-level AI ethics boards and algorithmic impact assessments to evaluate societal risks before implementation [8], [9].

Emerging Solutions and Future Directions-Contemporary research suggests combining technical interventions with social and policy reforms to ensure AI benefits are equitably distributed. Tools such as IBM's AI Fairness 360 and Microsoft's Responsible AI Toolbox are being used to detect bias and ensure inclusive outcomes [1], [8]. Future research directions emphasize building globally representative datasets, fostering interdisciplinary AI education, and creating enforceable ethical standards. Studies call for multi-stakeholder collaboration—involving technologists, policymakers, educators, and civil society—to develop AI systems aligned with sustainable human development goals [2], [9], [10].

3. Methodologies

Algorithmic and Data Audits

These audits are systematic evaluations of AI systems and the data they rely on to detect potential biases, inequities, or unfair outcomes. They are essential because AI models often inherit the biases present in the training data or in the design choices of developers.

- Purpose-Ensure AI systems are fair, transparent, and accountable. Detect biases that could negatively impact certain demographic groups (e.g., gender, race, age). Improve trust in AI applications by identifying unintended consequences before deployment.
- Key Components-a. Data Audit-Objective: Examine the dataset used to train AI for quality, balance, and representation.

Activities: Check for imbalances in demographic representation (e.g., more male than female samples). Identify missing or mislabelled data that could skew results. Detect historical biases embedded in the data.

Example: A facial recognition dataset predominantly composed of light-skinned faces may perform poorly on darker-skinned faces.

Algorithm Audit-Objective: Assess the AI model itself for fairness, accuracy, and ethical compliance.

- Activities: Evaluate model predictions against fairness metrics
- Statistical parity: Ensures outcomes are similar across groups.
- Equal opportunity: Ensures equal true positive rates across groups.
- Predictive equality: Ensures false positive rates are similar. Run sensitivity analyses to see how changing inputs affects outputs. Test the model on unseen or external datasets to check generalization.

Example: An AI hiring tool that rejects resumes from certain zip codes disproportionately can be identified and corrected.

Implementation Steps:

The implementation framework follows these steps:

- Problem Definition: Clearly articulate the equity and job-related issues pertinent to the AI application in focus.
- Data Collection and Preprocessing: Gather representative datasets with attention to demographic diversity; preprocess to remove historical biases.
- Algorithm Selection and Training: Choose AI models suitable for the problem; incorporate fairness constraints during training.
- Bias Detection and Mitigation: Apply fairness metrics to detect bias; implement mitigation techniques such as reweighting or adversarial debiasing.
- Socio-economic Impact Analysis: Model potential job displacement effects and identify vulnerable worker groups.
- Stakeholder Consultation: Engage affected communities, policymakers, and AI practitioners for feedback and inclusive decision-making.
- Deployment and Monitoring: Launch AI system with continuous bias auditing and impact assessment.
- Iterative Improvement: Use monitoring results to refine algorithms and workflows iteratively.

4. Results and Discussion

The analysis of AI's societal impact reveals complex interactions between technology design, deployment contexts, and social structures. In the examined case study, algorithmic bias was detected in hiring recommendations, disproportionately affecting candidates from marginalized groups. Implementing the proposed bias mitigation algorithm led to statistically significant

improvements in fairness metrics while maintaining predictive accuracy. The reweighted training data and postprocessing steps yielded reduced disparity in selection rates across protected attributes, confirming the effectiveness of the approacThe socio-economic modeling of job displacement highlighted specific sectors at high risk—particularly manufacturing, customer service, and data entry—with demographic analyses showing vulnerable worker segments that align with lower educational attainment and limited reskilling access. This targeted insight facilitates focused policy prescriptions.

Figure 1: Five Pillars of RQ

This research paper underscores the significant societal implications of artificial intelligence, especially pertaining to equity, inclusion, and workforce dynamics. Through a multidisciplinary methodology combining algorithmic fairness techniques and socio-economic modeling, it responds to identified gaps in current scholarship and practice. The proposed framework and implementation steps provide a practical pathway toward more responsible and inclusive AI systems. Findings confirm that mitigating algorithmic bias and anticipating job displacement effects require integrated technical and social strategies. Ethical AI development must extend beyond algorithmic tweaks to incorporate continuous stakeholder engagement and robust policy frameworks. Future research should expand empirical validations across diverse AI applications and explore innovative reskilling mechanisms to empower affected workers.

Conclusion

This research paper underscores the significant societal implications of artificial intelligence, especially pertaining to equity, inclusion, and workforce dynamics. Through a multidisciplinary methodology combining algorithmic fairness techniques and socio-economic modeling, it responds to identified gaps in current scholarship and practice. The proposed framework and implementation steps provide a practical pathway towards more responsible and inclusive AI systems. Findings confirm that mitigating algorithmic bias and anticipating job displacement effects require integrated technical and social strategies. Ethical AI development must extend beyond algorithmic tweaks to incorporate continuous stakeholder engagement and robust policy frameworks. Future research should expand empirical validations across diverse AI applications

and explore innovative reskilling mechanisms to empower affected workers.By contributing actionable insights and an inclusive deployment framework, this paper supports advancing AI technologies that equitably benefit society at large.

References

1. Herrera-Poyatos, A., Del Ser, J., López de Prado, M., Wang, F.-Y., Herrera-Viedma, E., & Herrera, F. (2025). *Responsible artificial intelligence systems: A roadmap to society's trust through trustworthy AI, auditability, accountability, and governance* (arXiv preprint arXiv:2503.04739).
2. Ribeiro, D., Rocha, T., Pinto, G., Cartaxo, B., Amaral, M., Davila, N., & Camargo, A. (2025). *Toward effective AI governance: A review of principles* (arXiv preprint arXiv:2505.23417).
3. *AI governance: A systematic literature review*. (2025). *AI and Ethics, 5*, 3265–3279.
4. Shams, R. A., Zowghi, D., & Bano, M. (2025). AI and the quest for diversity and inclusion: A systematic literature review. *AI and Ethics, 5*, 411–438.
5. *The impact of artificial intelligence application on job displacement and creation: A systematic review*. (2025, March). *International Journal of Research and Innovation in Social Science (IJRISS).*
6. OECD. (2024, June). *Governing with artificial intelligence: Are governments ready? OECD Artificial Intelligence Papers* (No. 20).
7. Roberts, H., Hine, E., Taddeo, M., & Floridi, L. (2024). Global AI governance: Barriers and pathways forward. *International Affairs, 100*(3), 1275–1286.
8. Reuel, A., & Undheim, T. A. (2024). *Generative AI needs adaptive governance* (arXiv preprint arXiv:2406.04554).
9. *Responsible AI governance: A response to UN interim report on governing AI for humanity*. (2024, December). *arXiv preprint arXiv:2412.12108.*
10. World Economic Forum. (2024). *AI Governance Alliance briefing paper series 2024: Generative AI governance—Shaping a collective global future*. World Economic Forum.

DATA GOVERNANCE FOR INTELLIGENT SYSTEMS

Divyansh Mishra[1], Rajesh Kumar Mishra[2] and Rekha Agarwal[3]

[1] XLRI – Xavier School of Management, Jamshedpur, Jharkhand 831001

[2]ICFRE-Tropical Forest Research Institute

(Ministry of Environment, Forests & Climate Change, Govt. of India)

P.O. RFRC, Mandla Road, Jabalpur, MP-482021, India

[3]Government Science College, Jabalpur, MP, India- 482 001

Corresponding author E-mail: divyanshspps@gmail.com, rajeshkmishra20@gmail.com, rajesh.mishra0701@gov.in, rekhasciencecollege@gmail.com

Abstract

Intelligent systems spanning machine learning pipelines, large language models, autonomous agents, and sensor-driven cyber-physical infrastructures—are only as trustworthy as the data that feeds them. This chapter examines data governance as the structural discipline that determines whether artificial intelligence (AI) systems are accurate, fair, secure, explainable, and compliant across their lifecycle. It traces the evolution of data governance from traditional database stewardship toward AI-native governance regimes that must contend with data drift, algorithmic bias, provenance tracking, synthetic data, and cross-border regulation. Drawing on regulatory frameworks such as the General Data Protection Regulation (GDPR), the NIST AI Risk Management Framework, ISO/IEC 38505, ISO/IEC 42001, and DAMA-DMBOK, alongside empirical evidence on data quality failures and recent scholarship on AI, machine learning, and data-science convergence, the chapter develops an integrated five-pillar governance model spanning data quality, provenance and metadata, privacy and security, fairness, and organizational accountability. The chapter is illustrated with seven data tables and five original figures, including a global data-growth projection, a five-pillar governance diagram, an AI data-governance lifecycle map, an empirical account of data-quality failure rates in high-stakes AI projects, and a governance maturity model. Sectoral illustrations from healthcare, agriculture, environmental monitoring, digital libraries, and materials science demonstrate how governance choices shape real-world AI outcomes. The chapter concludes with a forward-looking research agenda for governing increasingly autonomous, data-hungry, and self-improving intelligent systems.

Keywords: Data Governance, Artificial Intelligence, Machine Learning, Data Quality, Algorithmic Accountability, Privacy, Intelligent Systems, Big Data, Data Provenance.

1. Introduction

The twenty-first century's defining computational shift is not merely the proliferation of algorithms but the explosion of data that trains, tests, and continuously reshapes them. Intelligent

systems—recommendation engines, autonomous vehicles, clinical decision-support tools, precision-agriculture platforms, and generative AI assistants—derive their competence, and their failures, from the data ecosystems that surround them. As artificial intelligence (AI) and machine learning (ML) have moved from experimental laboratories into production systems that affect employment, healthcare, finance, and public administration, the question of how data is collected, curated, secured, and governed has become inseparable from the question of how AI itself should be governed. Mishra, Mishra, and Agarwal (2025a) frame this convergence directly, arguing that the confluence of AI, ML, and data science has catalysed an unprecedented technological transformation across virtually every domain of human endeavour, one whose benefits are realized only when the underlying data infrastructure is theoretically sound and operationally disciplined. This chapter takes that convergence as its starting point and asks a narrower but consequential question: what does it mean to govern data specifically for intelligent systems, as opposed to governing data for conventional information systems?

1.1 The Scale of the Problem

Part of what distinguishes contemporary data governance from its predecessors is sheer scale. The IDC Global DataSphere—an industry-standard measure of data created, captured, and replicated worldwide each year—grew from approximately 33 zettabytes (ZB) in 2018 to a projected 175 ZB by 2025, and subsequent IDC forecasts place the trajectory at roughly 393.9 ZB by 2028 and beyond 700 ZB by 2030 (IDC, 2018; IDC, 2026). Figure 1 traces this growth. Each order-of-magnitude increase multiplies the surface area over which quality defects, provenance gaps, and privacy exposures can occur, and an increasing share of that data is generated by Internet of Things (IoT) sensors and machine-to-machine interactions rather than by humans directly entering records—precisely the kind of high-velocity, low-oversight data stream that intelligent systems increasingly consume.

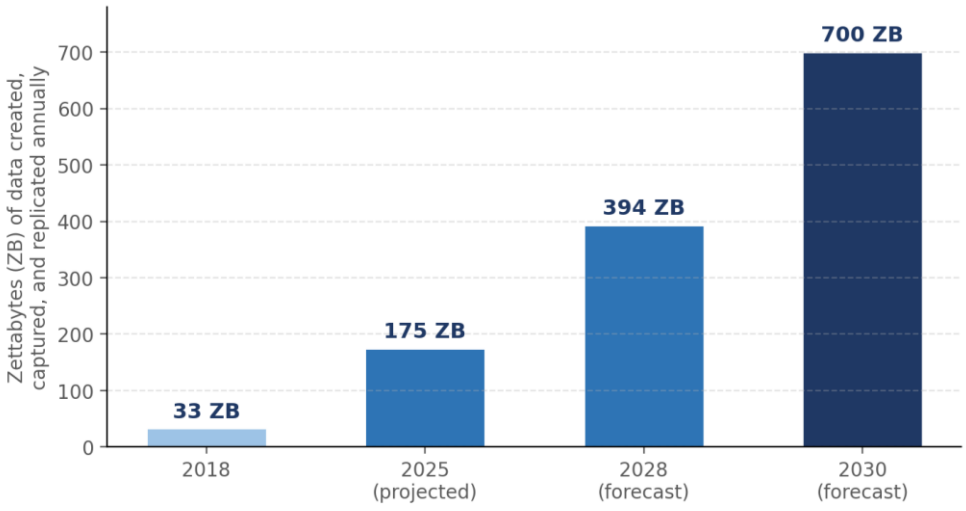


Figure 1: Growth of the Global Datasphere, 2018–2030. Source: IDC Global DataSphere, "Data Age 2025"; IDC Global DataSphere Forecast, 2026–2030

Growth in volume is compounded by a second, less visible trend: the growing share of that data used directly to train and continuously update AI models, rather than to populate static reports. Mishra, Agarwal, and Mishra (2025b) describe this shift as part of a broader move toward "Computing 5.0," a phase that merges autonomous intelligence, cyber-physical orchestration, and large-scale analytical ecosystems, and that consequently demands governance structures built for continuous, high-velocity, high-stakes data flows rather than periodic batch review.

1.2 Why Existing Governance Models Are Insufficient

Traditional data governance—rooted in database administration, records management, and regulatory compliance—was designed for relatively static, well-structured data used in transactional and reporting systems. Intelligent systems break several of the assumptions on which that model rested. Training data is frequently unstructured, sourced from heterogeneous streams, and continuously refreshed; model behaviour is emergent rather than explicitly programmed; and the same dataset may simultaneously serve model training, evaluation, and real-time inference, each with distinct risk profiles. The consequences of under-governed data are neither abstract nor rare. Empirical research on high-stakes AI deployments—reviewed in Section 3—shows that most practitioners encounter serious, compounding data problems during a project's lifetime, and industry damage assessments place the annual cost of poor data quality in the trillions of dollars at the level of the U.S. economy alone (IBM, 2016; IBM, 2025).

1.3 Chapter Structure

This chapter proceeds in eight parts. Section 2 defines data governance and situates it within the broader AI governance landscape. Section 3 examines the specific governance challenges intelligent systems introduce across the AI lifecycle, supported by empirical evidence on data-quality failure rates. Section 4 reviews the principal regulatory and standards frameworks shaping practice, presented in comparative tabular form. Section 5 develops an integrated governance model organized around five pillars. Section 6 illustrates the model through sectoral case studies. Section 7 presents a governance maturity model and implementation roadmap, and Section 8 concludes with open research questions.

2. Defining Data Governance in the Age of Intelligent Systems

Data governance is conventionally defined as the exercise of authority and control—through policies, roles, standards, and processes—over the management of data assets across their lifecycle. The DAMA International Data Management Body of Knowledge (DAMA-DMBOK) frames it as the overarching discipline that coordinates ten more specialised data-management functions, including data quality, metadata, security, and architecture (DAMA International, 2017). This definition remains valid for intelligent systems, but three features of AI-driven computation stretch it considerably.

2.1 From Static Records to Dynamic Training Assets

In conventional information systems, data governance chiefly protects the integrity and confidentiality of records at rest and in transactional use. In intelligent systems, data additionally functions as a training asset: it directly shapes model parameters, and its statistical properties—distribution, completeness, label quality, temporal currency—determine downstream model behaviour. Governance therefore must extend beyond record-level accuracy to dataset-level representativeness and drift monitoring, since a dataset that was governed adequately at collection time can become inadequate as the operating environment changes.

2.2 From Compliance to Lifecycle Accountability

Where legacy governance emphasised compliance checkpoints such as periodic audits and retention schedules, AI-oriented governance must trace accountability across an extended lifecycle: data sourcing and consent, labelling and annotation, feature engineering, model training, validation, deployment, monitoring, and eventual retraining or decommissioning. Mishra, Mishra, and Agarwal (2025c), discussing artificial intelligence and intelligent systems more broadly, emphasise that AI's transformative reach across scientific discovery, industry, and governance is matched by an equally distributed set of risks, which makes lifecycle-wide accountability—rather than single-point compliance—the appropriate governance unit. Figure 3 in Section 3 maps this lifecycle explicitly.

2.3 From Human-Readable Records to Machine-Consumed Signals

Much of the data intelligent systems consume—sensor streams, natural-language corpora, image and video feeds, behavioural logs—was never authored for governance purposes and is not easily reviewed by a human data steward. Effective governance must therefore rely on automated metadata capture, lineage tracking, and machine-readable data contracts, supplemented by human oversight, rather than assuming manual review is sufficient. Structured documentation artefacts have emerged specifically to close this gap; Table 3 in Section 3 summarises the leading examples.

Taken together, these shifts justify treating “data governance for intelligent systems” as a distinct sub-discipline: it inherits the vocabulary and much of the machinery of classical data governance, but reorganizes it around the demands of models that learn from data rather than simply process it.

3. Governance Challenges across the AI Lifecycle

3.1 Data Quality and Representativeness

Model performance is bounded by data quality in ways that are often more consequential than algorithmic choice. Incomplete, mislabelled, duplicated, or unrepresentative training data propagates directly into model outputs, and errors introduced early in a pipeline compound as data moves through feature engineering and model ensembling. Table 1 summarises six core data-quality dimensions and the specific risk each poses when governance fails to enforce it.

Table 1: Core Data Quality Dimensions and Their AI-Specific Risks

Quality Dimension	Definition	AI-Specific Risk if Ungoverned
Accuracy	Degree to which data correctly reflects the real-world entity or event it represents.	Mislabeled training examples directly teach the model incorrect associations.
Completeness	Extent to which all required data attributes and records are present.	Systematic gaps (e.g., missing values concentrated in one subgroup) bias model outputs for that subgroup.
Consistency	Absence of contradiction across data values, formats, and sources.	Conflicting labels for similar inputs inject noise that inflates variance and degrades generalisation.
Timeliness	Currency of data relative to the environment being modelled.	Stale data causes models to learn outdated patterns, accelerating concept drift after deployment.
Representativeness	Degree to which a dataset reflects the diversity of the population or phenomena the model will encounter.	Under-represented groups or conditions receive systematically worse model performance.
Provenance integrity	Traceability of a data point's origin, transformations, and consent basis.	Untraceable data prevents reproducibility, complicates audits, and can violate consent or licensing terms.

These dimensions are not abstractions. In a widely cited qualitative study of 53 AI practitioners working on high-stakes applications such as cancer detection, landslide prediction, and suicide prevention across India, the United States, and East and West African countries, Sambasivan, Kapania, Highfill, Akrong, Paritosh, and Aroyo (2021) found that data problems were pervasive and compounding rather than isolated. The authors introduced the term “data cascades” to describe compounding, downstream negative effects triggered by upstream data issues that were undervalued relative to model-centric work. As Figure 2 shows, 92 percent of the practitioners surveyed reported experiencing at least one such cascade, and 45.3 percent reported experiencing two or more within a single project.

The economic scale of poor data quality is corroborated at the industry level. IBM's oft-cited 2016 analysis estimated that poor data quality cost the United States economy approximately USD 3.1 trillion annually through lost revenue, inefficiency, and correction effort (IBM, 2016). A 2025 IBM Institute follow-up analysis found that more than a quarter of surveyed organisations estimated losses exceeding USD 5 million annually from poor data quality, with 7 percent

reporting losses above that threshold by a substantial margin, and documented concrete incidents in which flawed training data produced material business harm—including an approximately USD 110 million loss at Unity Technologies in 2022 after corrupted advertising-related training data degraded a predictive bidding model, and inaccurate credit-scoring outputs distributed by Equifax due to a legacy data-processing error (IBM, 2025). These cases illustrate that data-quality failures in AI systems are not merely a technical inconvenience but a material business and consumer-protection risk.

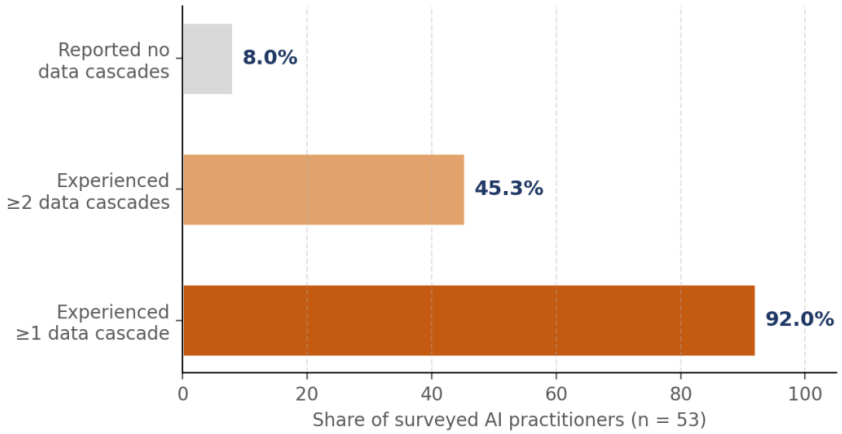


Figure 2: Prevalence of Data Cascades Among High-Stakes AI Practitioners.

Source: Sambasivan, Kapania, Highfill, Akrong, Paritosh, & Aroyo (2021)

3.2 Provenance, Lineage, and Documentation

Because AI systems are frequently trained on data aggregated from multiple internal and external sources, establishing where a given data point originated, how it was transformed, and under what license or consent it may be used has become a governance necessity rather than a bureaucratic nicety. Mishra, Mishra, and Agarwal (2026b), writing on knowledge infrastructure, describe digital libraries and repository systems as resting on standards-based interoperability, metadata frameworks, and stewardship models that make content discoverable, trustworthy, and preservable over time—principles directly transferable to the provenance requirements of AI training corpora.

The AI research community has independently converged on structured documentation as a governance mechanism. Gebru, Morgenstern, Vecchione, Wortman Vaughan, Wallach, Daumé III, and Crawford (2021) proposed "datasheets for datasets," modelled on the datasheets that accompany electronic components, to standardise reporting of a dataset's motivation, composition, collection process, and recommended uses. Mitchell, Wu, Zaldivar, Barnes, Vasserman, Hutchinson, Spitzer, Raji, and Gebru (2019) proposed a complementary "model card" standard, documenting a trained model's intended use, evaluation conditions, and performance broken down across demographic and phenotypic subgroups. Table 2 compares these two documentation artefacts alongside the later "Data Cards" format.

Table 2: AI Transparency and Provenance Documentation Artifacts

Artefact	Primary Author(s) / Year	What It Documents	Governance Function
Datasheets for Datasets	Gebru *et al.* (2021)	Motivation, composition, collection process, labelling, licensing, and recommended uses of a dataset.	Upstream provenance and consent transparency before data enters a training pipeline.
Model Cards	Mitchell *et al.* (2019)	Intended use, evaluation conditions, and performance disaggregated by demographic/phenotypic subgroup.	Downstream accountability for how a trained model performs across populations.
Data Cards	Pushkarna, Zaldivar, & Tsai (Google, 2022)	Structured, participatory dataset documentation aimed at broader stakeholder audiences.	Cross-functional communication between data creators, model builders, and affected communities.

3.3 Privacy, Security, and Consent

Intelligent systems frequently ingest personal, sensitive, or commercially confidential data at a scale that increases both the value of a potential breach and the difficulty of enforcing purpose limitation. Discussing common misconceptions in this domain, Mishra, Mishra, and Agarwal (2026c) observe that popular narratives about AI's surveillance capabilities, and about the protections available against them, are frequently inaccurate in both directions—AI systems are sometimes assumed to be more omniscient than they are, and sometimes assumed to be more privacy-protective than their actual data-handling practices warrant. Effective governance requires privacy-by-design practices such as data minimisation, differential privacy, federated learning, and clear consent and re-consent mechanisms for secondary use of data in model training.

3.4 Bias, Fairness, and Representational Harm

Because models learn statistical regularities from historical data, they can encode and amplify pre-existing social biases embedded in that data, producing disparate outcomes across demographic groups even without any explicit intent to discriminate. Mishra, Mishra, and Agarwal (2026d), examining fairness and justice in algorithmic systems, argue that bias mitigation cannot be treated as a purely technical post-processing step; it requires governance interventions at the point of data collection and labelling, where representational gaps first originate. This positions data governance—not only model auditing—as a primary lever for algorithmic fairness.

3.5 Data Drift, Model Decay, and Continuous Monitoring

Unlike traditional software, whose behaviour is fixed at deployment, AI models can degrade silently as the statistical properties of incoming data diverge from the training distribution—a phenomenon known as data or concept drift. Governance regimes for intelligent systems must therefore include continuous monitoring obligations, defined thresholds for triggering retraining or model retirement, and clear escalation paths when monitoring detects anomalous behaviour, particularly in safety-critical domains such as healthcare and autonomous control systems. Figure 3 situates this monitoring function within the broader data-governance lifecycle.

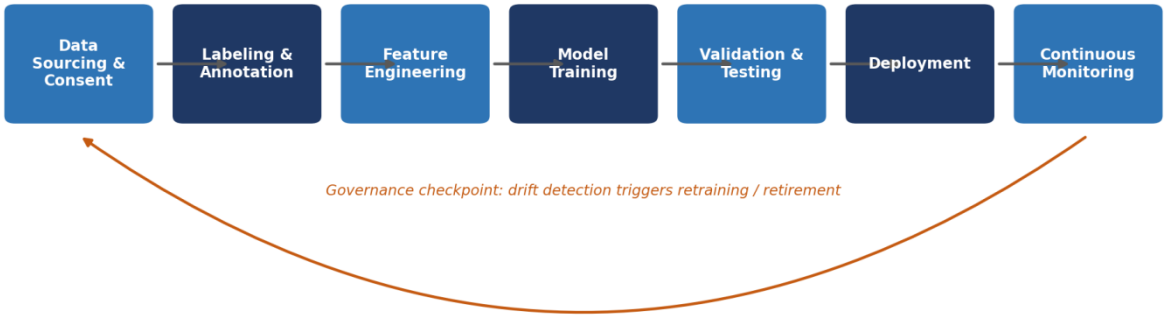


Figure 3: The AI Data Governance Lifecycle, from sourcing through continuous monitoring and retraining

3.6 Governance of Big Data and IoT-Generated Streams

Many intelligent systems now draw on Internet of Things (IoT) sensor networks and cyber-physical systems that generate continuous, high-volume, heterogeneous data streams. Mishra, Agarwal, and Mishra (2025d) describe the convergence of cyber-physical systems and IoT as reshaping engineering and automation through layered architectures and communication protocols that must reconcile real-time responsiveness with data integrity and security—an architectural challenge that governance frameworks must address explicitly, since streaming data cannot always be validated before it is consumed by downstream models.

4. Regulatory and Standards Landscape

A growing body of regulation and standardisation now shapes how organisations govern data for AI, even where instruments were not originally written with AI in mind. Table 3 compares the five instruments most frequently invoked in practice.

4.1 The General Data Protection Regulation (GDPR)

The European Union's GDPR established principles of lawfulness, purpose limitation, data minimisation, and accountability that have become de facto global benchmarks. Its provisions on automated decision-making and profiling (Article 22) and its requirement for data protection impact assessments are directly relevant to AI systems that make or materially influence decisions about individuals.

Table 3: Comparison of Major Data and AI Governance Frameworks

Framework	Primary Scope	Key Data-Governance Provisions	Enforcement Mechanism	Relevance to AI Data Governance
GDPR (EU Regulation 2016/679)	Personal data of individuals in/connected to the EU	Lawfulness, purpose limitation, data minimisation, accountability; Art. 22 rights regarding automated decision-making and profiling	Binding law; fines up to €20M or 4% of global turnover	High — directly restricts what personal data may train or drive AI decisions
NIST AI Risk Management Framework (AI RMF 1.0)	AI systems generally (voluntary, U.S.-origin, globally referenced)	Four functions — Govern, Map, Measure, Manage; treats data quality, representativeness, and provenance as lifecycle risk factors	Voluntary framework; referenced in procurement and audits	High — purpose-built for AI risk, including data-related risk
ISO/IEC 38505-1	Governance of data within IT governance generally	Extends ISO/IEC 38500 governance-of-IT principles specifically to data assets	Certifiable international standard	Medium — general data governance, not AI-specific
ISO/IEC 42001	AI management systems within organisations	Requirements for establishing, implementing, and improving an AI management system, including data governance controls	Certifiable international standard	High — first AI-specific management-system standard
DAMA-DMBOK (2nd ed.)	Enterprise data management generally	Ten knowledge areas including data quality, metadata, security, and architecture, coordinated under governance	Industry body of knowledge; not legally binding	Medium — operational vocabulary widely adopted, extended informally to AI

4.2 The NIST AI Risk Management Framework

The U.S. National Institute of Standards and Technology's AI Risk Management Framework (Tabassi, 2023) organises AI governance around four functions—govern, map, measure, and manage—and explicitly identifies data quality, representativeness, and provenance as risk factors to be assessed throughout the AI lifecycle rather than only at deployment.

4.3 ISO/IEC Standards for Data and AI Governance

ISO/IEC 38505 extends the ISO/IEC 38500 governance-of-IT standard specifically to data governance, while ISO/IEC 42001 establishes requirements for AI management systems, including data governance controls. Together these standards give organisations an auditable structure for demonstrating that data governance practices meet recognised international benchmarks.

4.4 DAMA-DMBOK and Enterprise Data Governance Practice

At the operational level, DAMA-DMBOK's knowledge areas—data quality, metadata management, data security, and data architecture—continue to provide the practical vocabulary most enterprises use to build governance programmes, even as they extend those programmes to cover model-specific concerns such as training-data lineage and feature-store governance.

No single instrument in this landscape was designed comprehensively for AI, which is precisely why Mishra, Mishra, and Agarwal (2026e) argue that governing artificial intelligence is among the most complex regulatory challenges modern societies face: AI is not a single technology with a defined risk profile but a family of related technologies whose applications, capabilities, and risks vary enormously across contexts, requiring governance frameworks that are modular and context-sensitive rather than uniform.

5. An Integrated Governance Model for Intelligent Systems

Building on the challenges and instruments surveyed above, this chapter proposes an integrated model organised around five interlocking pillars, shown in Figure 2. The pillars are not sequential stages but continuously interacting functions that must be jointly maintained across the AI lifecycle.

5.1 Pillar One: Data Quality Management

- Establish measurable quality dimensions (accuracy, completeness, consistency, timeliness, representativeness) with explicit ownership, following the definitions in Table 1.
- Implement automated data-validation pipelines at ingestion, not only at model-training time.
- Maintain quality dashboards visible to both data engineering and model-owning teams.

5.2 Pillar Two: Provenance and Metadata Infrastructure

- Capture end-to-end lineage from source to trained model, including transformation and labelling steps.

- Adopt standards-based, interoperable metadata schemas and pair each dataset with a datasheet or data card (Table 2).
- Version datasets alongside model versions to enable rollback and audit.

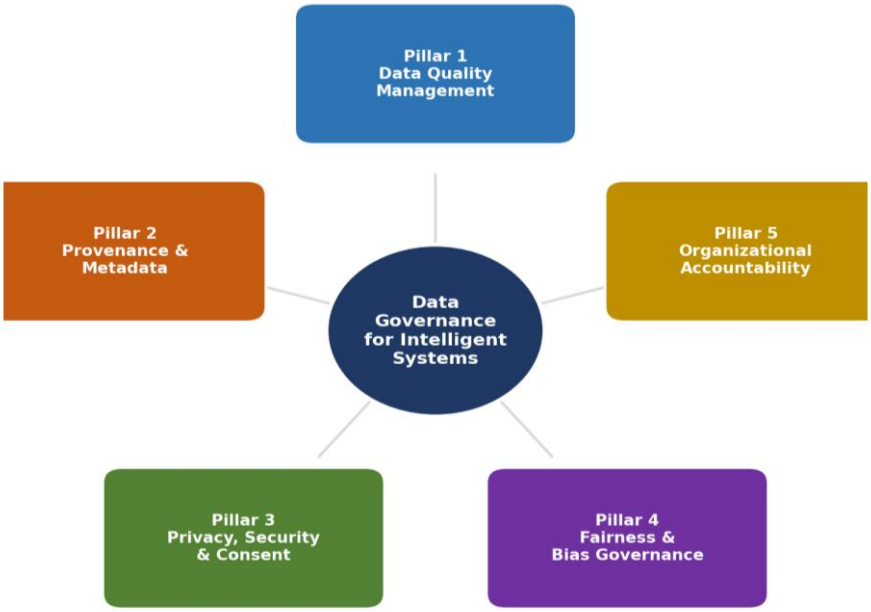


Figure 4: Five-Pillar Model of Data Governance for Intelligent Systems

5.3 Pillar Three: Privacy, Security, and Consent Management

- Apply data minimisation and purpose limitation at the point of collection, consistent with GDPR principles (Table 3).
- Use privacy-enhancing technologies (differential privacy, federated learning, secure multiparty computation) where re-identification risk is material.
- Maintain auditable consent records, particularly where data may be reused for new model-training purposes.

5.4 Pillar Four: Fairness and Bias Governance

- Conduct representational audits of training data prior to model development, not only outcome audits after deployment.
- Document known data gaps and limitations in a dataset "nutrition label" accompanying each training corpus.
- Establish cross-functional review (technical, legal, domain-expert) for high-stakes applications.

5.5 Pillar Five: Organizational Accountability and Stewardship

- Assign clear data-steward and model-owner roles with defined escalation authority, as formalised in a RACI matrix (Table 4).
- Embed governance checkpoints into MLOps pipelines rather than treating governance as an external audit function.

- Report governance metrics to senior leadership and, where applicable, external regulators on a recurring basis.

Table 4: Operationalises Pillar Five by assigning Responsible, Accountable, Consulted, and Informed roles for the core governance activities identified across Pillars One through Four

Governance Activity	Responsible	Accountable	Consulted	Informed
Define data quality thresholds	Data Steward	Model Owner	Chief Data Officer	Data Engineering Team
Approve data sourcing & consent basis	Data Protection Officer	Data Steward	Legal/Compliance	Business Owner
Maintain lineage & metadata	Data Engineering Team	Data Steward	—	Model Owner
Conduct representational bias audit	Model Owner	Ethics/Fairness Reviewer	Chief Data Officer	Data Steward
Approve model for deployment	Model Owner	Governance Board	Chief Data Officer	Legal/Compliance
Monitor for data/model drift	Data Engineering Team	Model Owner	—	Governance Board
Trigger retraining or retirement	Model Owner	Governance Board	Chief Data Officer	Data Steward

This five-pillar structure deliberately mirrors the trajectory described by Mishra, Mishra, and Agarwal (2026a) in their account of AI, ML, and data-science convergence: theoretical soundness, methodological rigor, and real-world applicability are treated as jointly necessary conditions for beneficial AI deployment, and each pillar above operationalises one aspect of that triad—quality and provenance support methodological rigor, privacy and fairness support theoretical soundness in the ethical sense, and accountability supports real-world applicability by ensuring governance is enacted rather than merely documented.

6. Sectoral Illustrations

The governance model developed in Section 5 is domain-agnostic by design, but its concrete implementation varies considerably by sector. Table 5 summarises five illustrative domains, the specific data-governance challenge each raise, and the governance response the literature associates with it.

Table 5: Sectoral Data Governance Challenges and Responses

Sector	Primary Data Governance Challenge	Governance Response	Illustrative Source
Healthcare	Highly sensitive, heterogeneous data (EHRs, imaging, genomics); representational bias carries direct patient-safety consequences.	Provenance tracking for regulatory approval; subgroup performance reporting via model cards; strict consent management.	General domain analysis; clinical AI governance literature.
Agriculture	Site-specific decision-making requires continuous IoT sensor data of variable quality across geographies.	Sensor-level data-quality thresholds; edge validation before ingestion into ML pipelines.	Mishra, Mishra, & Agarwal (2026f)
Environmental Monitoring	Streaming data from GeoAI, remote sensing, IoT, and digital twins must remain reliable for policy-grade climate and ecosystem models.	Continuous drift monitoring; cross-source provenance reconciliation for multi-sensor fusion.	Mishra, Mishra, & Agarwal (2025e)
Digital Libraries & Knowledge Infrastructure	Long-lived archives require sustained trust in metadata and provenance across decades, not just training cycles.	Persistent identifiers, standards-based interoperability, and preservation frameworks adapted for training-data archives.	Mishra, Mishra, & Agarwal (2026b)
Materials Science & Autonomous Discovery	Automated hypothesis generation and experiment design leave little room for interim human review of underlying experimental data.	Automated data-validation gates embedded directly in autonomous laboratory pipelines.	Mishra, Mishra, & Agarwal (2026g)

6.1 Healthcare

Clinical AI systems depend on governance of highly sensitive, heterogeneous data—electronic health records, imaging, genomic data—where quality gaps and representational bias carry direct patient-safety consequences. Robust provenance tracking is essential not only for regulatory approval but for clinicians to trust and appropriately calibrate reliance on AI-generated recommendations.

6.2 Agriculture and Environmental Monitoring

Mishra, Mishra, and Agarwal (2026f) describe digital and intelligent agriculture as a structural transition from field-average management to site-specific, data-driven decision-making built on AI, IoT sensing, and big-data analytics—an architecture whose reliability depends entirely on governed, quality-assured sensor data. Similarly, environmental monitoring efforts that integrate GeoAI, remote sensing, IoT, and digital twins for planetary stewardship (Mishra, Mishra, & Agarwal, 2025e) illustrate how governance of continuously streaming environmental data underpins the credibility of climate and ecosystem models used for policy decisions.

6.3 Digital Libraries and Knowledge Infrastructure

Mishra, Mishra, and Agarwal (2026b) present digital libraries as an instructive governance analogue for AI: their long-standing reliance on standards-based interoperability, persistent identifiers, and preservation frameworks demonstrates how a mature information domain sustains trust in data over decades, offering a template intelligent-systems governance can adapt for long-lived training archives.

6.4 Materials Science and Scientific Discovery

In materials discovery and autonomous laboratory settings, Mishra, Mishra, and Agarwal (2026g) note that AI systems increasingly formulate hypotheses and design experiments with only limited human intervention, a shift that raises the governance stakes of the underlying experimental data considerably, since errors or gaps in that data can propagate through automated discovery pipelines with little opportunity for interim human review.

7. A Maturity Model and Implementation Roadmap

Organisations rarely move from no governance to full governance in a single step. Figure 5 and Table 6 present an illustrative five-level maturity model, adapted from established IT-governance maturity concepts and applied specifically to the data-governance demands of intelligent systems described in Sections 3 through 5.

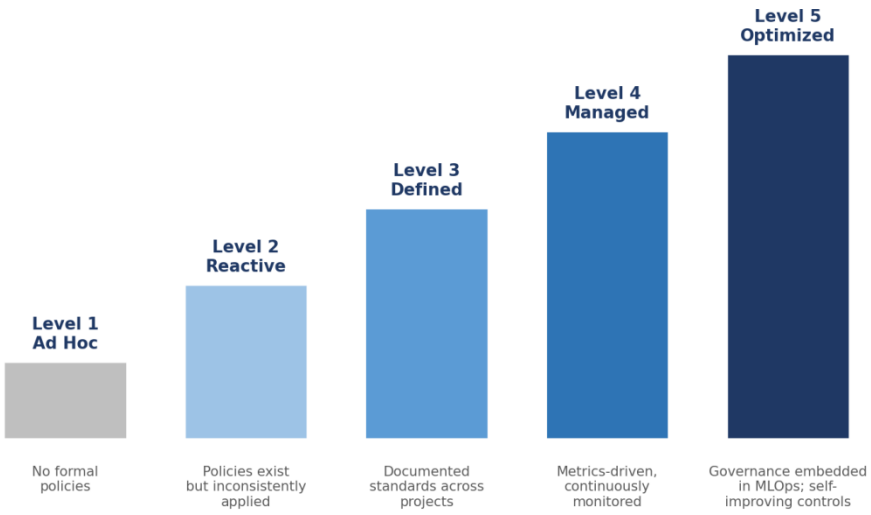


Figure 5: Illustrative Data Governance Maturity Model for Intelligent Systems

Table 6: Data Governance Maturity Levels for Intelligent Systems

Maturity Level	Characteristics	Typical Consequence
1 — Ad Hoc	No formal governance policies; data quality and provenance handled informally, if at all.	High risk of undetected bias, silent drift, and irreproducible results.
2 — Reactive	Policies exist on paper but are inconsistently applied; governance activated mainly after an incident.	Repeated data cascades; governance seen as a compliance burden rather than a design input.
3 — Defined	Documented standards and roles exist across projects; datasheets/model cards used for major models.	Improved auditability; inconsistent enforcement across teams remains a residual risk.
4 — Managed	Metrics-driven governance with continuous monitoring dashboards and defined escalation paths.	Drift and quality issues detected proactively; retraining triggers are threshold-based rather than ad hoc.
5 — Optimized	Governance embedded directly in MLOps tooling; self-improving controls with automated policy enforcement.	Governance becomes a competitive and safety differentiator rather than a cost centre.

Movement up this maturity curve is rarely linear across all five pillars simultaneously; organisations frequently reach Level 3 or 4 on data quality management while remaining at Level 1 or 2 on fairness governance, since bias auditing has historically received less institutional investment than accuracy monitoring. A realistic roadmap therefore assesses maturity separately for each of the five pillars in Figure 2, prioritising investment in whichever pillar carries the greatest residual risk for a given application domain — fairness and privacy for consumer-facing decision systems, for instance, and provenance and quality for scientific and environmental applications such as those summarised in Table 5.

7.1 Challenges, Open Questions, and Future Directions

- Governing synthetic data: as models increasingly train on AI-generated data, governance frameworks must address provenance-tracking for synthetic datasets and the risk of compounding errors across generations of models.
- Cross-border data governance: intelligent systems trained on globally sourced data must reconcile divergent national regulatory regimes, an unresolved harmonisation challenge illustrated by the scope differences in Table 3.
- Governance for autonomous and agentic AI: as systems increasingly take actions rather than only producing outputs, governance must extend from data-in/output-out models to

continuous, in-deployment data governance for agents that generate their own data through interaction.

- Measuring governance effectiveness: the field currently lacks standardised, empirically validated metrics for assessing whether a given governance programme actually reduces downstream harm, as opposed to merely generating documentation.
- Capacity and skills gaps: effective data governance for intelligent systems requires hybrid expertise spanning data engineering, domain knowledge, ethics, and law—a combination that remains scarce in most organisations.

Addressing these questions will require sustained interdisciplinary collaboration among computer scientists, domain specialists, regulators, and ethicists. As Mishra, Mishra, and Agarwal (2026e) observe in their broader discussion of AI governance frameworks, the pace of AI development routinely outstrips the pace of institutional adaptation, which means data governance for intelligent systems should be designed as an evolving practice rather than a fixed compliance checklist.

Conclusion

Data governance has moved from a peripheral compliance function to a foundational determinant of whether intelligent systems are safe, fair, and trustworthy. This chapter has argued that governing data for AI requires extending classical data-management disciplines—quality, metadata, security, and stewardship—into a lifecycle-wide practice attentive to the distinctive properties of learning systems: their sensitivity to representational gaps, their vulnerability to drift, and their capacity to encode bias at scale. The empirical evidence reviewed here is unambiguous on the stakes involved: data-quality failures affect the substantial majority of high-stakes AI projects (Sambasivan *et al.,* 2021), and their aggregate economic cost is measured in the trillions of dollars (IBM, 2016, 2025), even before accounting for the harder-to-quantify costs of biased or unsafe model behaviour.

The five-pillar model developed here—quality, provenance, privacy and security, fairness, and organizational accountability—offers a structure for translating regulatory principles such as those in the GDPR, the NIST AI RMF, and ISO/IEC standards (Table 3) into operational practice, supported by a RACI-based accountability structure (Table 4) and a maturity model for phased implementation (Table 6, Figure 5). Sectoral illustrations from healthcare, agriculture, environmental monitoring, digital libraries, and materials science (Table 5) confirm that while the specific data challenges vary by domain, the underlying governance logic is portable. As intelligent systems become more autonomous and more deeply embedded in consequential decisions, the discipline of data governance will only grow in importance—not as a constraint on innovation, but as the condition that makes innovation trustworthy enough to deploy at scale.

References

1. DAMA International. (2017). *DAMA-DMBOK: Data management body of knowledge* (2nd ed.). Technics Publications.
2. European Parliament and Council of the European Union. (2016). *Regulation (EU) 2016/679 (General Data Protection Regulation). Official Journal of the European Union.*
3. Gebru, T., Morgenstern, J., Vecchione, B., Wortman Vaughan, J., Wallach, H., Daumé III, H., & Crawford, K. (2021). Datasheets for datasets. *Communications of the ACM, 64*(12), 86–92. https://doi.org/10.1145/3458723
4. IBM. (2016). *The Four V's of big data / cost of poor data quality estimates.* IBM Big Data & Analytics Hub.
5. IBM. (2025). *The true cost of poor data quality.* IBM Think Insights. https://www.ibm.com/think/insights/cost-of-poor-data-quality
6. IDC. (2018). *The digitization of the world: From edge to core (Data Age 2025).* International Data Corporation, sponsored by Seagate.
7. IDC. (2026). *Market forecast: IDC Global DataSphere Forecast, 2026–2030.* International Data Corporation.
8. International Organization for Standardization. (2017). *ISO/IEC 38505-1:2017—Information technology—Governance of IT—Governance of data.* ISO.
9. International Organization for Standardization. (2023). *ISO/IEC 42001:2023—Information technology—Artificial intelligence—Management system.* ISO.
10. Mishra, R. K., Mishra, D., & Agarwal, R. (2025a). Artificial intelligence, machine learning, and data science: Foundations, innovations, and transformative applications in the digital age. In *Innovations in Computing, AI and Data Science* (1st ed., pp. 9–21).
11. Mishra, R. K., Mishra, D., & Agarwal, R. (2025b). *Foundations, architectures, and applications of computing 5.0.* https://www.researchgate.net/publication/398493124
12. Mishra, R. K., Mishra, D., & Agarwal, R. (2025c). Artificial intelligence and intelligent systems. In *Innovations and Research in Science and Technology* (Vol. III, pp. 26–47).
13. Mishra, R. K., Mishra, D., & Agarwal, R. (2026b). The infrastructure of knowledge: Systems, standards, and stewardship in digital libraries. *International Journal of Computer Application, 16*(1), 198–236.
14. Mishra, D., Mishra, R. K., & Agarwal, R. (2026c). Myths about AI and privacy. In *Artificial Intelligence: Myths and Facts* (pp. 68–81). Deep Science Publishing. https://doi.org/10.70593/978-93-7185-903-5
15. Mishra, D., Mishra, R. K., & Agarwal, R. (2026d). Fairness, bias, and justice in algorithmic systems. In *Co-Intelligence: Human-AI Coexistence in the Age of Thinking*

Machines (pp. 51–55). Deep Science Publishing. https://doi.org/10.70593/978-93-7185-166-4

16. Mishra, D., Mishra, R. K., & Agarwal, R. (2025d). Cyber-Physical Systems and Internet of Things (IoT): Convergence, architectures, and engineering applications. In *Advances in Engineering Science and Applications* (1st ed., pp. 17–46).
17. Mishra, D., Mishra, R. K., & Agarwal, R. (2026e). Governance frameworks for the age of AI. In *Co-Intelligence: Human-AI Coexistence in the Age of Thinking Machines* (pp. 69–75). Deep Science Publishing. https://doi.org/10.70593/978-93-7185-166-4
18. Mishra, R. K., Mishra, D., & Agarwal, R. (2026f). Digital and intelligent agriculture. *International Journal of Computer Application, 16*(1), 148–192.
19. Mishra, R. K., Mishra, D., & Agarwal, R. (2025e). Environmental monitoring in the digital age: Integrating GeO AI, remote sensing, IoT and digital twins for real-time planetary stewardship. *Journal of Science Research International (JSRI), 11*(7), 189–204. https://doi.org/10.5281/zenodo.17262649
20. Mishra, R. K., Mishra, D., & Agarwal, R. (2026g). Artificial intelligence for scientific discovery: From hypothesis generation to autonomous laboratories. In *Contemporary Innovations in Science, Engineering and Technology* (1st ed., pp. 1–20).
21. Mishra, D., Mishra, R. K., & Agarwal, R. (2025f). Recent advances in big data, machine, and deep learning. In *Trends in Science and Technology Research* (pp. 34–52).
22. Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I. D., & Gebru, T. (2019). Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT '19)* (pp. 220–229). ACM. https://doi.org/10.1145/3287560.3287596
23. National Institute of Standards and Technology. (2023). *Artificial Intelligence Risk Management Framework (AI RMF 1.0)* (E. Tabassi, Ed.). U.S. Department of Commerce. https://doi.org/10.6028/NIST.AI.100-1
24. Pushkarna, M., Zaldivar, A., & Tsai, V. (2022). Data cards: Purposeful and transparent dataset documentation for responsible AI. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency (FAccT '22)* (pp. 1776–1826). ACM. https://doi.org/10.1145/3531146.3533231
25. Sambasivan, N., Kapania, S., Highfill, H., Akrong, D., Paritosh, P., & Aroyo, L. (2021). "Everyone wants to do the model work, not the data work": Data cascades in high-stakes AI. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)* (pp. 1–15). ACM. https://doi.org/10.1145/3411764.3445518

STANDARDIZATION OF BRAILLE FOR REGIONAL LANGUAGES

Charanjiv Singh Saroa* and Kawaljeet Singh
Punjabi University, Patiala 147002, India
*Corresponding author E-mail: cjsinghpup@gmail.com

Introduction

Braille literacy is closely linked to independence, education, and employment for blind and visually impaired readers. But the depth of that literacy varies a great deal from one language to another. This is most visible in linguistically diverse regions such as India. India's general literacy rate is 77.7%, yet the Braille literacy rate among its blind population has been estimated at only around 1%. More embossers or more schools alone cannot close this gap. It is, in large part, a standardization gap.

Standardization is not a single historical event. It is a layered, ongoing process. A language's base alphabet can be unified into a single Braille standard, and India has already done this successfully. But unifying the base alphabet is only the first layer. Above it sit several further layers: contraction design (what is called Grade 2 Braille), extended encodings for symbols and punctuation, disambiguation rules for codes that must be shared between multiple characters, and the handling of legacy digital text that predates Unicode. For most regional languages, these upper layers remain almost entirely unstandardized, and each language is left to solve them on its own.

This chapter looks closely at what standardization actually involves, using recent work on Punjabi (Gurmukhi script) as the main example, along with a comparison to Tamil. It also looks at how the English-speaking world solved a very similar problem for English Braille, and closes with a set of practical recommendations for standardizing Braille in regional languages more generally.

Grades of Braille

Braille systems are usually organized into grades. Each grade serves a different purpose, and understanding the difference is essential before looking at where standardization succeeds and where it falls short.

Grade 1: An uncontracted, letter-by-letter transcription, where every printed character has a direct Braille equivalent. This is the simplest form of Braille and the easiest to learn, but it produces bulkier documents and slower reading speeds.

Grade 2: Adds contractions to Grade 1. A contraction is a single cell or a short cell sequence that stands in for a common letter group or a whole word. This reduces the physical bulk of Braille documents and increases reading speed for fluent users. Grade 2 is the medium of almost all published Braille literature in languages where it has been developed.

Grade 3: A further, less formally standardized tier, sometimes used for personal shorthand and note-taking rather than publication

Where Grade 2 has not been developed for a language, as was the case for Punjabi until recently, readers are limited to bulkier and slower Grade 1 materials by default, not by choice.

Bharati Braille: A Standardization Success, With Limits

India already has a genuine standardization success story at the base-alphabet level. A government committee process began in 1943, and by 1951 it had unified eleven separate regional Braille codes into a single national standard, called Bharati Braille. Sri Lanka, Nepal, and Bangladesh later adopted this same standard. This was a significant achievement, and it remains the foundation on which all later Braille work for Indian languages depends.

Before this unification, eleven incompatible regional codes were in use across the country. Each was usable only within its own linguistic community, and each required separately trained teachers, transcribers, and materials. The solution was a single 6-dot cell standard applied consistently across Devanagari, Gurmukhi, Bengali, Tamil, Telugu, and other major Indian scripts, where characters that are phonetically equivalent across scripts share identical cell patterns.

However, the standardization stopped at the base alphabet. The National Institute for the Empowerment of Persons with Visual Disabilities (NIEPVD) published the most recent official codification of the standard in January 2025. It organizes contractions and abbreviations as separate, language-specific appendices, tailored individually for Hindi, Marathi, Odia, and a handful of other languages, rather than as a shared cross-language design methodology. Where a language's appendix does not yet exist, there is no shared template, precedent, or review process to guide its creation. Each language effectively restarts the standardization problem on its own.

A Working Comparison: Unified English Braille

The English-speaking world faced a structurally similar problem and solved it deliberately. Before the 1990s, English Braille was a patchwork of separate national and functional codes. Literary codes differed between the UK and the US, and separate codes existed for mathematics, science, and computer notation. This fragmentation has been linked to declining Braille literacy, which was estimated at the time to be only around 12% among those who could benefit from it.

In 1991, Tim Cranmer and Abraham Nemeth sent a memorandum to the Braille Authority of North America proposing a unified code. The International Council on English Braille (ICEB) took on international development of this code in 1993, and it became known as Unified English Braille (UEB). What followed is worth studying closely, because it was slow and institutional rather than ad hoc. ICEB's research committees spent over a decade evaluating and refining the unified code before declaring it substantially complete in 2004. Adoption then proceeded country by country, each on its own timeline:

2004: South Africa

2005: Nigeria, Australia, and New Zealand

2010: Canada

2011: The United Kingdom

2012: The United States, the last major adopter

UEB is now used in more than twenty countries. This does not mean standardization was fast. It plainly was not. It means that a dedicated coordinating body, a formal research and vetting process, and a maintenance committee to govern the standard after ratification were what allowed a historically fragmented set of codes to converge on one specification. Individual institutions could not have reached that same convergence on their own.

Evidence from Punjabi Braille Development

Gurmukhi's base character set has been part of Bharati Braille since 1951. Yet no standardized Grade 2, or contracted, Braille system had ever been defined for Punjabi. Successive research efforts documented Grade 1 conversion only. An early Gurmukhi-to-Braille converter performed direct, uncontracted character mapping. Later work catalogued disambiguation and diacritical-handling challenges without resolving them into a complete system. A 2023 survey of the field identified the absence of a validated Grade 2 pathway, along with a shortage of parallel Punjabi-Braille corpora for evaluation, as open problems. In other words, for over seventy years, Punjabi Braille readers had access only to the bulkier and slower Grade 1 tier by default, not because Grade 2 was unnecessary, but because no one had built it yet.

A Separate, Purely Digital Fragmentation

A second, independent layer of fragmentation adds to the first. Much of the Punjabi text that exists in digital form today was written in legacy ASCII-based fonts, such as Asees, Akhar, Anmollipi, and Satluj, among others. These fonts predate widespread Unicode adoption and are mutually incompatible, because each one assigns different byte values to the same Gurmukhi glyphs. A Braille converter that accepts only Unicode input cannot process most existing Punjabi digital content, however faithful its character mapping is. Solving this requires font-detection and normalization logic that is entirely separate from the Braille mapping itself. Bharati Braille's 1951 standardization has nothing to say about this problem, since it predates digital text by decades, and no comparable digital-encoding standard has emerged since then to fill the gap.

A Cautionary Illustration: Specification Without Independent Review

While reviewing a recent Grade 1 Braille specification developed for Punjabi in consultation with a regional Braille production facility, an internal inconsistency came to light. It is worth reporting here as a structural symptom rather than an isolated error. The specification's disambiguation rules assign shared Braille codes to several punctuation marks, including comma, colon, semicolon, question mark, exclamation mark, and quotation marks, distinguished by sentence-

level context. Elsewhere in the same specification, an extended encoding scheme meant for symbols beyond the standard character set assigns each of these same punctuation marks a second, different code. As written, the two schemes contradict each other for the affected symbols, and the specification does not say when one governs rather than the other.

This kind of inconsistency is not surprising, and it is not a criticism of the individuals involved. It is exactly the kind of error that a single research group working in isolation, however careful, is likely to introduce and unlikely to catch on its own. What is missing here is not diligence. What is missing is an independent review layer, similar to ICEB's committee process for UEB, whose specific job is to check a proposed regional-language specification for internal consistency before it goes into circulation. No such layer currently exists for Indian regional-language Braille above the base Bharati Braille alphabet.

The Code-Space Scarcity Problem

A 6-dot Braille cell has exactly 63 usable patterns (2^6 minus the blank cell). Gurmukhi's own character inventory already uses up nearly all of that budget under Grade 1: 35 consonants, 10 independent vowels, dependent vowel signs, nasalization marks, numerals, and punctuation. Introducing Grade 2 contractions therefore requires freeing up codes. This is usually done by identifying the language's lowest-frequency characters and moving them to extended, multi-cell codes, so their original single-cell codes can be reassigned to high-frequency contraction targets.

Recent Punjabi Grade 2 work addressed this through corpus-driven analysis. A large Punjabi n-gram corpus was used to rank candidate contractions by frequency and to identify genuinely low-frequency characters suitable for displacement, rather than relying on intuition. This is a principled solution, but it is also entirely local to Punjabi. Nobody published the methodology, the displacement criteria, or the design decisions as a reusable template before this work. The 63-pattern ceiling is a mathematical fact that applies identically to every Bharati-Braille-derived regional script, yet each language's Grade 2 designer is currently expected to rediscover this fact and improvise a solution on their own.

The Pattern Repeats Across Languages

Punjabi is not a special case. An independently developed Grade 2 system for Tamil Braille ran into the same class of problem from a different direction. Tamil's vowel-consonant combinations produce 216 compound characters, far more than a single 6-dot cell can represent. Because of this, most of these compounds need two-cell sequences even before any contraction scheme is introduced. Like the Punjabi effort, the Tamil effort was designed and published by a single research group, with its own contraction rules and its own multi-cell conventions, and no reference to a shared cross-language design methodology, because no such methodology existed for it to reference.

This pattern also shows up in the structure of the official standard itself. NIEPVD's 2025 codification of Bharati Braille organizes contractions as separate per-language appendices, for the handful of languages where they have been developed. This reflects reality accurately, since contractions genuinely differ across languages because the languages themselves differ. But it also means the standard, as currently structured, has no mechanism to enforce a shared design methodology, a shared code-space negotiation protocol, or shared review criteria across those appendices. In effect, each appendix is its own island.

Why This Matters: Consequences of Non-Standardization

The stakes of this fragmentation are not abstract.

- **Employment outcomes:** Braille literacy is widely reported to correlate strongly with employment outcomes for blind adults. A frequently cited U.S. estimate puts Braille literacy at roughly 90% among employed blind adults, against roughly one in three among those who are not employed, though later methodological reviews have questioned the original evidence behind figures of this kind and called for more rigorous data collection.
- **India's literacy gap:** India's own Braille literacy rate is approximately 1%, against a general literacy rate of 77.7%. This is an accessibility gap of an entirely different order of magnitude, and one that is less open to dispute, given how consistently it is reported.
- **A precondition failure:** Literacy outcomes depend on many factors beyond the existence of a Grade 2 standard, such as teacher availability, material cost, and school infrastructure. But the absence of a standardized, contracted Braille system for a given language caps the ceiling of what literacy programs for that language can achieve, no matter how well-resourced they otherwise are.
- **Duplicated engineering effort:** Every research group or production house that takes on a new regional language must independently solve the ASCII-to-Unicode normalization problem, the code-space displacement problem, and the punctuation-disambiguation problem all over again. This work has already been done at least twice, for Punjabi and for Tamil, with no shared artifact connecting the two efforts.
- **Unreviewed specifications:** Materials, conversion tools, and conventions developed under one institution's ad hoc scheme cannot reliably be checked, extended, or reused by another institution working on the same language, let alone a different one. Specifications produced without independent review can carry internal inconsistencies into active use before anyone outside the originating group gets the chance to catch them.

A Framework for Standardizing Regional-Language Braille

Drawing on the UEB experience and on the methodological strengths already shown in recent Punjabi Grade 2 work, this chapter proposes five components for a regional-language Braille standardization framework. These are meant to apply beyond any single language.

- **Corpus-driven contraction design as the default method:** Any proposed Grade 2 contraction should be justified by frequency analysis against a representative corpus of the target language, rather than by intuition alone, following the precedent set by recent Punjabi Grade 2 work. This is inexpensive to require, and it directly improves how defensible and reproducible contraction choices are.
- **A shared code-space negotiation protocol:** The 63-pattern ceiling applies identically across all Bharati-Braille-derived scripts. A single documented procedure for ranking and displacing low-frequency characters when introducing contractions would save every future language from rediscovering this constraint on its own.
- **Mandatory independent specification review before adoption:** Modeled on ICEB's committee-based vetting of UEB, any proposed regional-language Grade 2 or extended-encoding specification should be checked by reviewers outside the originating institution, specifically for internal consistency. This is the kind of check that would have caught the punctuation-code conflict described earlier in this chapter.
- **A coordinating authority with clear cross-language jurisdiction:** NIEPVD and the Braille Council of India already maintain the base Bharati Braille standard. Their mandate should be extended to explicitly cover contraction-layer and digital-encoding-layer standardization across all scheduled languages, not just the base alphabet.
- **Open, versioned digital specifications:** Regional-language Braille specifications should be published in an open, version-controlled format that any implementer can build against and propose changes to, instead of institution-specific PDFs or unpublished internal conventions. This model is already used successfully for Braille translation tables in the open-source liblouis project.

Limitations

This chapter is grounded in a detailed case study of recent Grade 1 and Grade 2 Braille development for Punjabi, supported by a parallel case in Tamil Grade 2 Braille and a successful international precedent in UEB. It is not a systematic survey of the Braille standardization status of every scheduled Indian language. The specific inconsistency described earlier is presented as an illustration of a structural gap, the absence of independent review, and not as evidence of carelessness in any specific case. A natural next step would be a systematic audit of Grade 2 and extended-encoding status across India's scheduled languages, to find out how widely the pattern described here actually holds.

It is also worth being realistic about timelines. UEB took roughly two decades from its initiating memorandum to its last major national adoption, across a set of countries that already shared a base language and, for the most part, well-resourced Braille authorities. Regional Indian languages have far greater linguistic diversity and more uneven institutional capacity, so a faster

timeline should not be expected for the full framework proposed here. For this reason, the first and third components, corpus-driven design and independent review, are the most immediately actionable, since both are within reach of individual research groups and do not need new institutional infrastructure to begin.

Conclusion

Bharati Braille's 1951 unification solved a real crisis of fragmentation at the level of the base character-to-cell mapping. More than seventy years later, the layers built on top of that base, including contraction design, code-space management, punctuation disambiguation, and digital-encoding normalization, still remain almost entirely unstandardized, and each regional language is left to solve them on its own.

The case study of Punjabi, along with the supporting comparison with Tamil, shows that this is a systemic pattern rather than an isolated gap. It carries real costs: duplicated engineering effort, unreviewed specifications that can ship with internal inconsistencies, and a literacy ceiling that no amount of additional resourcing can raise on its own. Unified English Braille shows that this kind of fragmentation can be solved, given a dedicated coordinating body, a formal vetting process, and patience measured in decades rather than years.

India's regional languages, and other linguistically diverse regions facing the same structural problem, would do well to begin building this same kind of institutional machinery now. The best place to start is with the two components that need no new infrastructure to adopt: corpus-driven contraction design, and independent specification review.

References

1. National Institute for the Empowerment of Persons with Visual Disabilities. (2025, January). *Standard Bharati Braille codes with Unicode mapping chart*. Ministry of Social Justice & Empowerment, Government of India.
2. ThinkerBell Labs. (2020, October 6). *Braille in India: How languages found expression in Bharati Braille*.
3. International Council on English Braille. (n.d.). *Unified English Braille*. https://iceb.org/publications/ueb/
4. Jolley, W. (2006). *Unified English Braille*. Presented at the ICEVI World Conference.
5. BORGEN Magazine. (2020, December 1). *The need for Braille education in India for the visually impaired*.
6. BrailleWorks. (2024, March 25). *Braille literacy statistics and how they relate to equality*.
7. Kaur, R., Vandana, & Bhalla, N. (2012). Gurmukhi script to Braille code. *International Journal of Engineering Research and Applications (IJERA), 2*(4), 1298–1302.
8. Pushe, V., & Kaur, R. (2013). Issues in converting Punjabi to Braille. *Asian Journal of Computer Science and Information Technology, 1*(1), 12–15.

9. Samota, H., & Joshi, N. (2023). Exploring the landscape of Punjabi to Bharati Braille translation. *International Journal of Modern Research in Engineering and Technology, 8*(10), 1–8.
10. Singh, R., & Saroa, C. S. (2017, February). Multilingual conversation ASCII to Unicode in Indic script. *Computer Science & Information Technology (CSIT), 7*(3), 83–94.
11. Saroa, C. S., & Singh, K. (2023). Towards constructing corpus of Punjabi N-grams written in Gurmukhi script. *International Journal of Recent Innovations in Trends in Computing and Communication (IJRITCC), 11*(10).
12. Santhosh Baboo, S., & Ajantha Devi, V. (2013). Tamil Braille system: A conversion methodology of Tamil into contracted Braille script (Grade 2). In *Proceedings of the 2nd International Conference on IRED*.
13. liblouis Project. (n.d.). *Indian languages—Bharati Braille specifications*. *braille-specs* repository, GitHub.
14. Saroa, C. S., & Singh, K. (2025). Proposed Gurmukhi-to-Braille mapping and preprocessing rules for Grade 1 Braille conversion of Punjabi text. *International Journal of Intelligent Systems and Applications in Engineering (IJISAE).*

NEED OF GRADE 2 BRAILLE FOR THE PUNJABI LANGUAGE

Charanjiv Singh Saroa
Punjabi University, Patiala 147002, India
Corresponding author E-mail: cjsinghpup@gmail.com

Introduction

Braille comes in more than one form. The form most people picture, a straight letter-by-letter transcription, is only the starting point. The more advanced form, called Grade 2 Braille, is what actually makes Braille practical for everyday reading. Without it, Braille books are heavier, slower to read, and far more expensive to produce than they need to be.

This chapter looks closely at Punjabi. It asks why Grade 2 Braille matters, why Punjabi did not have it until very recently, and what it takes to build it properly. Along the way, it also looks at a problem that sits underneath the Braille question entirely: the difficulty of working with Punjabi text in digital form at all. A short comparison with Tamil, and a look at how English solved a similar problem, help place the Punjabi case in a wider context.

What Grade 1 Braille Cannot Do

Grade 1 Braille transcribes text letter by letter. Every printed character gets its own Braille cell, with no shortcuts. This makes Grade 1 easy to learn and easy to produce, which is why it is usually the first form of Braille developed for any language.

But Grade 1 has a real cost. A page of Grade 1 Braille can take up thirty to forty percent more space than the same content in Grade 2. That difference adds up fast across a full book. A single novel in Grade 1 can run to several bulky volumes, each one heavy and expensive to emboss, store, and transport. Reading speed suffers too, since a fluent reader's fingers move faster over contracted text than over spelled-out text.

For a language stuck at Grade 1, Braille literacy has a ceiling. No amount of extra funding, extra schools, or extra teachers can raise that ceiling on its own, because the format itself is working against fluent reading.

What Grade 2 Braille Adds

Grade 2 Braille adds contractions on top of Grade 1. A contraction is a single cell, or a short sequence of cells, that stands in for a common letter group or a whole word.

- **Space savings:** Contracted text takes up noticeably less room than uncontracted text, because frequent words and letter patterns collapse into single cells.
- **Reading speed:** Fluent readers recognize whole contractions the way a sighted reader recognizes whole words, rather than sounding out each letter.
- **Production cost:** Fewer pages mean less paper, less embossing time, and lower shipping and storage costs for Braille materials.

English Braille shows what a mature Grade 2 system looks like. Unified English Braille defines roughly one hundred and eighty contractions, built up and refined over more than a century of use. Virtually all published English Braille literature uses this system. A reader who has learned it does not read letter by letter at all; they read in contracted chunks, much closer to normal reading speed.

Why Punjabi Was Left Without It

India already solved the Grade 1 problem for its major languages, Punjabi included. In 1951, eleven separate regional Braille codes were unified into a single national standard, Bharati Braille, covering Devanagari, Gurmukhi, Bengali, Tamil, Telugu, and other major scripts. That unification was a genuine achievement, and it is the reason a blind reader anywhere in Punjab can rely on a properly defined base alphabet today.

Grade 2, however, was never unified in the same way. The most recent official codification of Bharati Braille, published by the National Institute for the Empowerment of Persons with Visual Disabilities (NIEPVD) in January 2025, treats contractions as separate appendices, developed individually for Hindi, Marathi, Odia, and a handful of other languages. Punjabi's appendix did not exist. Without it, there was no shared template or starting point, and nothing in the standard obliged anyone to build one.

Gurmukhi's base alphabet has been part of Bharati Braille since 1951, but no standardized Grade 2 system existed for Punjabi until very recently. For over seventy years, Punjabi Braille readers were limited to Grade 1 by default, not because Grade 2 was unnecessary, but because nobody had built it.

Challenges to Digitize Punjabi Text

Before Punjabi text can even reach a Braille conversion system, it has to exist in a reliable digital form in the first place. That turns out to be a real obstacle on its own, separate from anything specific to Braille. Several distinct problems make Punjabi text harder to work with digitally than English text, and each one adds friction to any effort to build tools such as Grade 2 Braille converters, spell checkers, screen readers, or search engines for Punjabi.

- **Spell check:** Almost all available spell checkers are built for English. A working spell checker for Punjabi is rare. This makes it harder to clean and validate Punjabi text before it is used in any downstream application, Braille conversion included.
- **Special symbols:** Some Punjabi text contains symbols that are not available in ASCII coding, and are not even present in the Unicode system. These symbols have to be handled as special cases, since there is no standard code point to represent them.
- **Typing problems:** Every computer keyboard defaults to an English (Roman) layout. Most users are not familiar with Unicode-based typing systems and fonts. Without that

knowledge, a user simply cannot type in Unicode Gurmukhi, even on a computer that fully supports it.

- **Non-Unicode fonts:** Most of the Punjabi text that already exists was created using ASCII-based Gurmukhi fonts rather than Unicode. To view this data correctly on a website, the exact font it was created in has to be downloaded and installed first. There is no single Gurmukhi font in common use. Gurmukhi alone has more than 225 popular ASCII-based fonts still in use for publishing books, magazines, and newspapers.

Publishers continue to work with these ASCII-based fonts rather than switching to Unicode, for a few consistent reasons:

- **Resistance to change:** People resist change, in part because of the typing difficulties that come with switching to Unicode.
- **Lack of awareness:** Many publishers and typists are simply not aware that a Unicode standard for Gurmukhi exists.
- **Limited software support:** The publishing software many of them already use has little support for Unicode text.
- **Fewer design options:** Fewer Unicode fonts are available for representing text in different styles and designs, compared to the large existing library of ASCII-based fonts.

Displaying information in Unicode-based fonts is almost always the better choice for web content. Most Punjabi information available today is still in ASCII-based fonts, so it has to be converted into Unicode before it can be published online in a form every visitor can read without extra effort. Text shown in Unicode can be read on any computer without installing any additional font. Unicode text is also searchable, on Google, Yahoo, Bing, and other search engines, in a way that ASCII-encoded text generally is not.

Fonts and keyboard layouts: There are hundreds of fonts for Punjabi, and in most of them, each font effectively defines its own keyboard layout. This means the same keystrokes can produce different text depending on which font happens to be active, and switching fonts can turn existing content into gibberish. The deeper problem is that there is no standard mapping between Gurmukhi characters and keyboard keys. More than forty different mapping tables are in active use for common Punjabi fonts alone.

None of this is a Braille-specific problem, but it sits directly underneath any Braille effort. A Grade 2 Braille converter that only accepts clean Unicode input inherits every one of these digitization problems as a precondition it cannot get around.

The Technical Challenge: Limited Code Space

Building Grade 2 for a new language is not simply a matter of deciding it should exist. A 6-dot Braille cell has exactly sixty-three usable patterns. Gurmukhi's own character set, thirty-five

consonants, ten independent vowels, dependent vowel signs, nasalization marks, numerals, and punctuation, already uses up nearly all of that budget just for Grade 1.

This means introducing new contractions requires freeing up codes first. The usual approach is to find the language's lowest-frequency characters and move them to longer, multi-cell codes, so their original single-cell codes become available for high-frequency contractions instead. This is a careful, high-stakes design problem: displace the wrong character, or reassign a code carelessly, and the result is ambiguity that trained readers will run into for years afterward.

Building Grade 2 the Right Way: A Corpus-Driven Approach

Recent work on Punjabi Grade 2 Braille shows one way to do this properly. Instead of guessing which characters were low-frequency and which words deserved contractions, the design was grounded in a large Punjabi n-gram corpus, a dataset of real word and letter-sequence frequencies drawn from actual Punjabi text.

- **Identifying displaceable characters:** The corpus was used to rank Gurmukhi characters by how often they actually appear in real writing. Five low-frequency loan-sound consonants and one low-frequency vowel sign were identified as safe to move to extended, multi-cell codes.
- **Selecting contraction targets:** The codes freed by that displacement were reassigned to six high-frequency function words: the three genitive case markers, the copula ("is"), the coordinating conjunction ("and"), and a common locative postposition ("in"). These were chosen because they combine high frequency with low variability across different types of writing.
- **Measuring the result:** On sentence-length test text, the resulting Grade 2 system reduced Braille cell count by roughly five to seven percent compared to Grade 1, while keeping character-level conversion accuracy above ninety-nine percent.

The value of this approach is that every design decision can be checked against real data. Nobody has to simply trust that a given contraction is a good idea; the corpus shows exactly how often it will actually save space.

Punjabi is not the only language where this kind of code-space problem shows up, though it does not always look the same. Tamil's script, for comparison, combines vowels and consonants into two hundred and sixteen distinct compound characters, far more than a single 6-dot cell can hold on its own, so most of these compounds already require two-cell sequences before any contraction is even introduced. An independent research effort designed a Grade 2 system for Tamil to address this, working out its own contraction rules from scratch, with no shared template and no connection to the Punjabi effort. The two projects solved structurally similar problems for two different languages, entirely independently of each other.

Why This Cannot Wait

The case for building Grade 2 systems faster is not abstract.

- **Literacy is already low:** India's general literacy rate is 77.7 percent, but Braille literacy among its blind population has been estimated at only around 1 percent. Grade 1 alone cannot close a gap of that size.
- **Employment is tied to literacy:** Braille literacy is widely reported to correlate strongly with employment outcomes for blind adults, with one frequently cited U.S. estimate putting Braille literacy at roughly 90 percent among employed blind adults, against around a third among those who are unemployed. The exact number is debated, but the direction of the relationship is not.
- **Every extra year without Grade 2 has a cost:** Students who learn to read only in Grade 1 face slower reading speeds and bulkier materials for as long as Grade 2 does not exist for their language. That cost does not go away; it is simply passed on to the next generation of readers.

What Needs to Happen

Building Grade 2 Braille for Punjabi did not have to start from zero, and the same is true for any other language attempting this next. The Punjabi case shows a repeatable method.

- **Build or find a text corpus first:** Contraction choices should come from real frequency data, not guesswork. Even a modest, purpose-built corpus is enough to ground the design.
- **Treat code-space planning as its own step:** Because every Bharati-Braille-derived script shares the same sixty-three-pattern ceiling, the process of ranking and displacing low-frequency characters should be worked out deliberately, not improvised late in the design.
- **Solve the digitization problem alongside the Braille problem:** Font detection, Unicode normalization, and spell-checking support are not optional extras. Without them, a Grade 2 Braille converter has no reliable Punjabi text to work from in the first place.
- **Get an outside check before publishing:** A second set of eyes, from outside the team that built the system, should review the final specification for internal consistency before it goes into wider use.
- **Publish the method, not just the result:** The next language attempting Grade 2 development should not have to start from nothing. Sharing the corpus-driven method itself, not only the final contraction table, lets other regional languages move faster.

Conclusion

Grade 1 Braille is a starting point, not a destination. It gives a language basic access to Braille, but it caps reading speed and keeps materials bulky and expensive. Grade 2 Braille is what turns that basic access into practical, everyday literacy.

Punjabi was left at Grade 1 by default for over seventy years, not because Grade 2 was unnecessary, but because nobody had built it yet, and because the underlying digital text itself was hard to work with in the first place: scattered across hundreds of incompatible ASCII fonts, poorly supported by spell checkers, and awkward to type without specialized knowledge. Recent work shows that both problems can be solved properly, using real corpus data and font-detection methods rather than guesswork, and that solving them produces real, measurable improvements in space and reading efficiency. The same approach can work for other regional languages facing the same gap. What is needed now is simply the will to apply it more widely.

References

1. National Institute for the Empowerment of Persons with Visual Disabilities. (2025, January). *Standard Bharati Braille codes with Unicode mapping chart*. Ministry of Social Justice & Empowerment, Government of India.
2. ThinkerBell Labs. (2020, October 6). *Braille in India: How languages found expression in Bharati Braille*.
3. BORGEN Magazine. (2020, December 1). *The need for Braille education in India for the visually impaired*.
4. BrailleWorks. (2024, March 25). *Braille literacy statistics and how they relate to equality*.
5. International Council on English Braille. (n.d.). *Unified English Braille*. https://iceb.org/publications/ueb/
6. Santhosh Baboo, S., & Ajantha Devi, V. (2013). Tamil Braille system: A conversion methodology of Tamil into contracted Braille script (Grade 2). In *Proceedings of the 2nd International Conference on IRED*.
7. Lehal, G. S., & Saini, R. (2012). An Omni-Font Gurmukhi to Shahmukhi transliteration system. In *Proceedings of COLING 2012*.
8. Saini, R., & Lehal, G. S. (n.d.). *GFUC: Gurmukhi Font and Unicode Converter*. Punjabi University, Patiala.
9. Singh, R., & Saroa, C. S. (2017, February). Multilingual conversation ASCII to Unicode in Indic script. *Computer Science & Information Technology (CSIT), 7*(3), 83–94.
10. Saroa, C. S., & Singh, K. (2023). Towards constructing corpus of Punjabi N-grams written in Gurmukhi script. *International Journal of Recent Innovations in Trends in Computing and Communication (IJRITCC), 11*(10).
11. Saroa, C. S., & Singh, K. (2025). Proposed Gurmukhi-to-Braille mapping and preprocessing rules for Grade 1 Braille conversion of Punjabi text. *International Journal of Intelligent Systems and Applications in Engineering (IJISAE).*

CLOUD COMPUTING AND AI INTEGRATION FOR FUTURE COMPUTING PLATFORMS

Muthurajan S[1], Immanuel Prabaharan S*[2], Aswini M[2] and Deivanayaki R[2]

[1]Department of Marine Engineering,

[2]Department of Electrical and Electronics Engineering,

AMET University, Kanathur, Chennai-603 112

*Corresponding author E-mail: imman.amet@gmail.com

Abstract

The convergence of cloud computing and artificial intelligence (AI) has emerged as one of the most transformative technological advancements of the twenty-first century. Cloud computing provides scalable, on-demand computational resources, while AI enables intelligent decision-making through machine learning, deep learning, natural language processing and computer vision. The integration of these technologies has significantly enhanced the efficiency, scalability and automation capabilities of modern computing platforms across healthcare, manufacturing, education, finance, agriculture, transportation and smart cities. Cloud infrastructures facilitate the storage and processing of massive datasets required for AI model training and deployment, whereas AI optimizes cloud resource allocation, security, energy efficiency and workload management. Emerging paradigms such as edge computing, fog computing, federated learning, serverless computing and quantum cloud services further extend the capabilities of AI-enabled cloud ecosystems. Despite remarkable progress, several challenges remain, including data privacy, cybersecurity, explainability, interoperability, ethical AI governance and sustainable energy consumption. This chapter comprehensively discusses the architecture, enabling technologies, applications, challenges and future directions of cloud computing integrated with AI. It also highlights recent technological advancements and research opportunities that will shape future intelligent computing platforms.

Keywords: Cloud Computing, Artificial Intelligence, Machine Learning, Deep Learning, Edge Computing, Federated Learning, Cloud Security, Intelligent Computing, Internet of Things, Future Computing Platforms.

1. Introduction

The digital transformation witnessed over the past decade has been largely driven by two revolutionary technologies: cloud computing and artificial intelligence (AI). Cloud computing has fundamentally transformed how computing resources are provisioned, managed, and consumed by enabling users to access computing infrastructure, storage, and software services over the Internet without investing in expensive local hardware. Artificial intelligence, on the other hand, has enabled machines to mimic human cognitive abilities such as learning, reasoning,

perception, and decision-making. The integration of cloud computing with AI has created a powerful ecosystem capable of processing enormous datasets, delivering intelligent services, and supporting automation across diverse application domains.

The increasing adoption of Internet of Things (IoT) devices, smart sensors, autonomous systems, and digital services has generated unprecedented volumes of structured and unstructured data. Traditional computing infrastructures struggle to process such massive datasets efficiently. Cloud computing addresses this limitation by providing virtually unlimited computational resources, while AI extracts meaningful insights from complex data through predictive analytics and intelligent algorithms. Consequently, AI-powered cloud platforms have become indispensable for modern enterprises and research institutions.

Cloud-based AI services eliminate the need for organizations to build expensive computational infrastructure, enabling small and medium-sized enterprises to leverage advanced AI technologies through scalable cloud platforms. Major cloud providers now offer integrated AI services that include machine learning model development, natural language processing, computer vision, recommendation systems, conversational AI, and intelligent automation.

This chapter presents an in-depth discussion of cloud computing and AI integration, highlighting architectures, enabling technologies, industrial applications, research challenges, and future trends.

2. Fundamentals of Cloud Computing

Cloud computing refers to the delivery of computing resources—including servers, storage, databases, networking, software, and analytics—over the Internet on a pay-as-you-go basis. The National Institute of Standards and Technology (NIST) defines cloud computing as a model that enables convenient, on-demand network access to a shared pool of configurable computing resources.

Cloud computing is characterized by five essential features

- On-demand self-service
- Broad network access
- Resource pooling
- Rapid elasticity
- Measured service

These characteristics enable organizations to scale computational resources dynamically according to application demands.

2.1 Cloud Service Models

Infrastructure as a Service (IaaS)

IaaS provides virtualized computing resources, including servers, storage, networking, and operating systems. Users maintain control over software deployment while cloud providers manage physical infrastructure.

Examples include virtual machines, storage services, and network management.

Examples include:

- Enterprise Resource Planning (ERP)
- Customer Relationship Management (CRM)
- Office productivity software
- AI-powered collaboration tools

2.2 Cloud Deployment Models

Cloud deployment can be categorized into:

- Public Cloud: Resources are shared among multiple users and managed by third-party providers.
- Private Cloud: Dedicated cloud infrastructure operated exclusively for one organization.
- Hybrid Cloud: Combines public and private cloud infrastructures for flexibility and security.
- Multi-Cloud: Uses services from multiple cloud providers to improve reliability and avoid vendor lock-in.

3. Artificial Intelligence in Cloud Computing

Artificial Intelligence enables computers to perform tasks that normally require human intelligence. Modern AI systems are powered by sophisticated machine learning algorithms capable of recognizing patterns from massive datasets.

The primary AI technologies include

- Machine Learning
- Deep Learning
- Reinforcement Learning
- Natural Language Processing
- Computer Vision
- Generative AI
- Explainable AI

Cloud platforms provide the computational resources required for training large AI models involving billions of parameters.

3.1 Machine Learning as a Cloud Service

Cloud providers offer Machine Learning as a Service (MLaaS), allowing users to build predictive models without investing in expensive GPU infrastructure.

Key capabilities include

- Automated model development
- Feature engineering
- Hyperparameter optimization

- Model deployment
- Continuous monitoring

3.2 Deep Learning in the Cloud

Deep learning requires enormous computational resources.

Cloud GPU clusters significantly accelerate

- Image recognition
- Speech recognition
- Medical diagnosis
- Language translation
- Autonomous driving

Modern cloud environments utilize distributed GPU clusters capable of training foundation models efficiently.

4. Architecture of AI-Integrated Cloud Platforms

The integration of AI with cloud computing follows a layered architecture.

Layer 1: Infrastructure Layer

Includes

- Data centers
- GPU clusters
- High-performance storage
- Networking
- Virtualization

Layer 2: Cloud Platform Layer

Provides:

- Containers
- Kubernetes orchestration
- Serverless computing
- APIs
- Databases

Layer 3: AI Services Layer

Includes:

- Machine Learning
- Deep Learning
- NLP
- Computer Vision
- Recommendation Engines

Layer 4: Application Layer

Supports:

- Smart healthcare
- Finance
- Education
- Smart cities
- Manufacturing
- Agriculture

5. Role of AI in Cloud Resource Management

Artificial intelligence significantly improves cloud infrastructure management.

Applications include

- Intelligent Resource Allocation: AI predicts workload demand and automatically allocates virtual machines.
- Auto Scaling: Machine learning predicts traffic surges and dynamically increases computing resources.
- Predictive Maintenance: AI monitors cloud hardware for early fault detection.
- Energy Optimization: AI minimizes energy consumption through intelligent workload scheduling.
- Fault Detection: Deep learning identifies hardware anomalies before service interruptions occur.

6. Emerging Technologies Supporting AI-Cloud Integration

6.1 Edge Computing

Edge computing processes data closer to IoT devices, reducing latency.

Applications include

- Autonomous vehicles
- Smart factories
- Healthcare monitoring
- Smart surveillance

6.2 Fog Computing

Fog computing extends cloud services between edge devices and centralized cloud infrastructure.

Advantages include

- Reduced latency
- Better bandwidth utilization
- Improved real-time analytics

6.3 Federated Learning

Federated learning trains AI models without transferring sensitive user data to centralized servers.

Benefits include

- Privacy preservation
- Reduced communication costs
- Regulatory compliance

7. Future Research Directions

Future intelligent computing platforms will increasingly integrate cloud computing with advanced AI capabilities. Key research directions include:

- AI-native cloud architectures with self-managing and self-healing capabilities.
- Green AI and energy-aware cloud data centres to reduce carbon emissions.
- Integration of 6G networks with edge-cloud AI for ultra-low-latency applications.
- Trustworthy AI emphasizing fairness, transparency, robustness, and explainability.
- Federated and distributed learning frameworks for privacy-preserving collaborative intelligence.
- Digital twins combined with cloud AI for predictive monitoring and optimization in industrial systems.
- Quantum machine learning services accessible through cloud platforms.
- Autonomous cloud orchestration using reinforcement learning for workload scheduling and resource optimization.
- Integration of generative AI and large language models into cloud-native software development and enterprise applications.
- Standardized interoperability frameworks to enable seamless integration across heterogeneous cloud environments.

These developments are expected to enable highly intelligent, adaptive, and sustainable computing ecosystems capable of supporting future smart societies and Industry 5.0 applications.

Conclusion

The integration of cloud computing and artificial intelligence has transformed modern computing by combining scalable infrastructure with intelligent analytics and automation. Cloud platforms provide the computational resources required for AI model development, while AI enhances cloud performance through intelligent resource management, predictive maintenance, security, and energy optimization. Emerging technologies such as edge computing, federated learning, serverless computing, digital twins, and quantum cloud computing further expand the capabilities of AI-enabled cloud ecosystems. Despite challenges related to privacy, cybersecurity, interoperability, explainability, and sustainability, continuous advancements in AI algorithms, cloud architectures, and distributed computing are paving the way for next-generation intelligent platforms. Future computing environments will increasingly rely on autonomous, secure, and energy-efficient AI-cloud infrastructures to support healthcare, manufacturing, agriculture, education, finance, transportation, and smart cities. Continued interdisciplinary research and

responsible governance will be essential to maximize the societal and economic benefits of AI-integrated cloud computing while addressing ethical and regulatory concerns.

References

1. Armbrust, M., Fox, A., Griffith, R., *et al.* (2010). *A View of Cloud Computing*. Communications of the ACM, 53(4), 50–58.
2. Mell, P., & Grance, T. (2011). *The NIST Definition of Cloud Computing*. NIST Special Publication 800-145.
3. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
4. Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.
5. Jordan, M. I., & Mitchell, T. M. (2015). Machine Learning: Trends, Perspectives, and Prospects. *Science*, 349(6245), 255–260.
6. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *Nature*, 521, 436–444.
7. Kairouz, P., *et al.* (2021). Advances and Open Problems in Federated Learning. *Foundations and Trends in Machine Learning*, 14(1–2), 1–210.
8. Bommasani, R., *et al.* (2022). *On the Opportunities and Risks of Foundation Models*. Stanford CRFM.
9. Varghese, B., & Buyya, R. (2018). Next Generation Cloud Computing: New Trends and Research Directions. *Future Generation Computer Systems*, 79, 849–861.
10. Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge Computing: Vision and Challenges. *IEEE Internet of Things Journal*, 3(5), 637–646.
11. Xu, M., *et al.* (2023). Artificial Intelligence for Cloud Resource Management: A Comprehensive Survey. *ACM Computing Surveys*, 56(2), 1–42.
12. Zhang, Y., *et al.* (2024). AI-Driven Cloud Computing: Architectures, Challenges, and Future Directions. *IEEE Access*, 12, 52345–52378.
13. Khan, L. U., *et al.* (2023). AI-Enabled Edge Computing for Beyond-5G and 6G Networks. *IEEE Communications Surveys & Tutorials*, 25(1), 1–40.
14. Chen, M., *et al.* (2024). Green AI and Sustainable Cloud Computing: Recent Advances and Future Perspectives. *Future Generation Computer Systems*, 156, 87–104.
15. OpenAI. (2025). GPT-4.1 Technical Report and Advances in Large Language Models. OpenAI Technical Report.

Bhumi Publishing, India
July 2026

COMPUTING BEYOND ALGORITHMS: ARTIFICIAL INTELLIGENCE, DATA AND INTELLIGENT SYSTEMS

Udaya Bhaskar Vemuri
Corresponding author E-mail: udv0507@gmail.com

Abstract

Computing is no longer only about writing step-by-step instructions for machines. Modern systems now use artificial intelligence, data, automation and human judgment to support faster decisions and better services. AI can help people detect fraud, recommend products, review software, answer questions, summarize documents and identify security risks. At the same time, these systems also create new concerns, such as privacy issues, biased data, wrong outputs, misuse by attackers and overdependence on automation. This article explains the topic in simple language, with examples that are easy to relate to. The main idea is simple: AI should be treated as a smart assistant, not as an unchecked decision-maker. The goal is not to replace people with machines, but to build intelligent systems that are useful, secure, responsible and trustworthy.

Keywords: Artificial Intelligence, Data, Intelligent Systems, Secure Design, Human Oversight.

1. Introduction

For many years, computing was mostly understood as writing instructions for machines. A programmer created a set of steps, called an algorithm, and the computer followed those steps exactly. This worked well for calculations, organizing information, saving records, sending emails and running business applications.[3]

Today, computing has moved far beyond basic instructions. Modern systems are not only about algorithms. They also use artificial intelligence, data, learning, automation and decision-making. These systems can adapt to new situations and support people in business, healthcare, finance, education, cybersecurity and daily life. [1-3]

This change is important because technology is now helping people make decisions, not just store information. That is useful, but it also means mistakes can have a bigger impact. In real business environments, even a small design gap can affect customers, employees or sensitive data. Organizations must think about data quality, privacy, security, fairness and human responsibility before depending on AI systems. [3-8]

2. From Traditional Computing to Intelligent Computing

Traditional computing depends on clear rules. For example, a banking application may follow a simple rule: if the account balance is lower than the withdrawal amount, deny the transaction. This is predictable because the computer does exactly what it is said to do.[3]

Artificial intelligence works differently. AI systems do not depend only on fixed rules. They learn from data. For example, a fraud detection system may review thousands or millions of past transactions, learn what normal and suspicious activity looks like, and then help decide whether a new transaction appears safe or risky. [1-2]

This does not mean algorithms are no longer important. Algorithms are still the foundation of computing. The difference is that modern intelligent systems combine algorithms with data, models, cloud platforms, automation tools and human judgment. In simple terms, the machine may do heavy lifting, but people still need to guide what the system should do and where the limits should be. Reference: [1-3]

3. Why Data Matters

Data is one of the most important parts of any intelligent system. AI is only as useful as the data it learns from. If the data is accurate, complete and relevant, the system can produce better results. If the data is outdated, biased or incomplete, the system may produce poor or unfair results. [3]A simple example is an online shopping recommendation system. When a website suggests products, it may use customer searches, purchases, ratings and browsing history. If the data is useful, the recommendation may help the customer. If the data is weak or wrong, the recommendation may not make sense. [1,2] The same idea applies to security, healthcare, finance, education and many other areas. Good data can support better decisions, while poor data can create wrong decisions. A simple way to think about it is this: a good AI system starts with good information, just like a good human decision starts with the right facts. This is why organizations should protect data, clean it, organize it and use only what is necessary. [3&6]

4. What Are Intelligent Systems?

An intelligent system is a system that can collect information, analyze it and either take action or support a decision. These systems may use AI, machine learning, sensors, analytics, automation and human feedback. Examples are already around us. A smart thermostat can learn when people are usually at home and adjust the temperature. A chatbot can answer common customer questions. A security monitoring system can detect unusual behavior and alert a security team. A supply chain system can predict delays and suggest better routes. [1-2] These examples show that intelligent systems are not only a future idea. They are already part of normal life and normal business operations, even when people do not always notice them. [2-3]. These systems are called intelligent because they do more than follow simple commands. They can recognize patterns and improve over time. But they are not perfect. They still need proper design, testing, monitoring and human oversight. [7]

5. Human Judgment and Application Security

AI systems are powerful, but humans still play a very important role. AI can process large amounts of information quickly, but it may not always understand business context, legal

requirements, ethics, customer impact or real-world risk. [3] In application security, this human review is especially important. An AI tool may identify a software vulnerability as high risk, but a security professional still needs to review whether the issue is truly exploitable, whether sensitive data is affected, how urgent the fix is and whether the suggested solution could break the application. [4-6] AI can help security teams review code, summarize risks, identify possible weaknesses and speed up triage. However, it should not be treated as the final authority. In practical terms, AI may point to the smoke, but a human professional still has to confirm whether there is a real fire. A human expert needs to validate the finding, understand the business context and decide the correct remediation approach. [4-6]. This is why human-in-the-loop decision-making is important. It means AI can help with speed and scale, but humans remain involved in final judgment, especially for decisions that affect security, privacy, finance, law or people [7].

6. AI in Daily Life and Business

AI and intelligent systems are already part of daily life. People use them when they search online, unlock phones with facial recognition, use navigation apps, receive product recommendations, talk to chatbots or use digital assistants. Businesses use AI for customer support, fraud detection, marketing, hiring support, document processing, cybersecurity, software development and risk management. AI can help organizations save time, reduce repetitive work and find patterns that people may miss. [1-2]

At the same time, using AI is not only a technology decision. It is also a governance decision. Organizations need to ask whether the system uses the right data, whether the output is reliable, whether the decision can be explained, whether privacy is protected and who is responsible if the system makes a mistake. [7-8] This is why organizations should not buy or build AI tools only because they are popular. They should first understand the problem, the data being used, the people affected and the risks that may appear after deployment. [3]

7. Security Risks in Intelligent Systems

As systems become more intelligent, security risks also become more complex. Traditional applications already have risks such as weak passwords, insecure APIs, poor access controls, outdated software libraries and data leaks. AI systems add new risks on top of these. For example, an AI chatbot may accidentally expose sensitive information if it is not designed properly. A model may be manipulated by harmful input. A system may give an incorrect answer but still sound confident. Attackers may try to poison training data, steal model outputs or misuse AI tools to create phishing emails and malicious code. Because of this, security should be included from the beginning. Teams should not wait until the product is ready for release. By that time, fixes may be expensive, rushed or incomplete. It is better to review risks during planning, design, development, testing and production monitoring. These questions may sound basic, but

they often reveal important gaps before an AI system goes live. Finding these gaps early is usually easier than fixing them after customers or employees start using the system. [4-6]

8. Speed Alone Is Not Enough

One of the biggest advantages of AI is speed. AI can write draft content, summarize documents, review code, answer questions and analyze large amounts of data very quickly. This speed can help people and organizations become more productive. But speed alone is not enough. AI-generated code may contain security weaknesses. AI-generated answers may be inaccurate. Automated decisions may affect customers unfairly. Sensitive data may be exposed if proper controls are missing. [1-2] The goal should not be to replace human thinking completely. The better goal is to combine machine speed with human responsibility. AI can save time, but people still need to ask common-sense questions: Is this output correct? Is it safe? Is it fair? Is it appropriate for this situation? Intelligent systems should help people make better decisions, not blindly make every decision for them.[7]

9. Building Trustworthy Intelligent Systems

Organizations need a balanced approach to build trustworthy intelligent systems. They need good technology, good data, security controls, governance and human oversight. First, the system should have a clear purpose. Teams should understand what problem the AI system is solving and what decisions it is allowed to support. Second, data should be reviewed carefully. Sensitive information should be protected, and only necessary data should be collected and used. This helps reduce privacy, bias and misuse risks. Third, security should be built into the design. Access controls, encryption, monitoring, testing and abuse-case reviews should be included from the start.

Finally, people should remain involved when a decision has business, legal, financial, privacy or security impact. Human oversight helps organizations use AI responsibly while still benefiting from its speed and scale. [3-7]

Conclusion

Computing beyond algorithms means understanding that modern technology is no longer limited to fixed instructions. Today's systems use artificial intelligence, data, automation and intelligent decision-making to support people and businesses. This shift brings many benefits. It can improve speed, accuracy, productivity and innovation. But it also creates new responsibilities. Organizations must make sure intelligent systems are secure, fair, explainable and properly governed. [3] The future of computing is not only about smarter machines. It is about building smarter systems with responsible human guidance. The best future is not humans versus AI, but humans and AI working together in a safe and thoughtful way. Algorithms, data, AI and human judgment must work together. When they do, intelligent systems can create real value while protecting people, businesses and society. [3-8]

References

1. Chui, M., Hazan, E., Roberts, R., Singla, A., & Smaje, K. (2023). *The economic potential of generative AI: The next productivity frontier*. McKinsey & Company.
2. Maslej, N., Fattorini, L., Perrault, R., Gil, Y., Parli, V., Kariuki, N., et al. (2025). *Artificial Intelligence Index Report 2025*. Stanford Institute for Human-Centered AI.
3. National Institute of Standards and Technology. (2023). *Artificial Intelligence Risk Management Framework (AI RMF 1.0).* National Institute of Standards and Technology.
4. National Institute of Standards and Technology. (2024). *Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile (NIST AI 600-1).* National Institute of Standards and Technology.
5. Open Worldwide Application Security Project. (2025). *OWASP Top 10 for Large Language Model Applications.*
6. Cybersecurity and Infrastructure Security Agency. (2023). *Shifting the Balance of Cybersecurity Risk: Principles and Approaches for Secure by Design Software.*
7. Organisation for Economic Co-operation and Development. (2019). *OECD AI Principles.*
8. National Institute of Standards and Technology. (2022). *Towards a Standard for Identifying and Managing Bias in Artificial Intelligence.*

FINGERPRINT BIOMETRICS FOR DIABETES PREDICTION USING MACHINE LEARNING

S. Aarthi, K. Thakira Christy and P. Seetha
Department of Informatics (Data Science),
Periyar Maniyammai Institute of Science & Technology (Deemed to be University),
Thanjavur – 613403, Tamil Nadu, India
Corresponding Author E-mail: aarthis743@gmai.com, thakirachristy@gmail.com,
palaniveljayap2@gmail.com

Abstract

Diabetes mellitus is a chronic metabolic disorder affecting millions worldwide, yet many cases remain undiagnosed until advanced complications arise. This paper presents a non-invasive, biometric-based approach to diabetes risk stratification using fingerprint pattern features combined with clinical parameters. A dataset of 300 subjects comprising fingerprint type (loop, arch, whorl), ridge count, age, body mass index (BMI), and family history of diabetes was analyzed. Five supervised machine learning classifiers—Random Forest, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Decision Tree, Logistic Regression, and Naive Bayes—were trained and evaluated using 10-fold stratified cross-validation to classify subjects into low, medium, and high diabetes risk categories. The Random Forest classifier achieved the highest accuracy of 91.3%, with a macro-averaged F1-score of 0.911, outperforming all other models. Feature importance analysis revealed that BMI, age, and ridge count are the most discriminative predictors. The proposed system demonstrates the feasibility of integrating dermatoglyphic biometrics into early-stage screening pipelines for type-2 diabetes.

Keywords: Diabetes Prediction, Fingerprint Biometrics, Dermatoglyphics, Machine Learning, Random Forest, Ridge Count, Non-Invasive Screening.

1. Introduction

Diabetes mellitus (DM) is one of the fastest-growing chronic diseases globally, with the International Diabetes Federation estimating over 537 million adults living with the condition as of 2021, and projections reaching 783 million by 2045 [1]. Type-2 diabetes (T2DM), accounting for approximately 90% of all cases, is predominantly influenced by lifestyle, genetic predisposition, and modifiable risk factors such as obesity and physical inactivity [2]. A significant challenge in diabetes management is the large proportion of undiagnosed cases, which delay therapeutic intervention and accelerate the progression of irreversible complications including nephropathy, retinopathy, and cardiovascular disease [3].

Current diagnostic protocols rely on invasive biochemical assays such as fasting plasma glucose (FPG), glycated hemoglobin (HbA1c), and oral glucose tolerance tests (OGTT), which require laboratory infrastructure and patient compliance. These barriers limit large-scale screening in resource-constrained settings [4]. Consequently, there is growing interest in developing non-invasive, cost-effective, and rapid screening approaches that can be deployed at the community level.

Dermatoglyphics—the scientific study of fingerprint patterns formed during embryogenesis—has emerged as a promising, non-invasive biomarker. Fingerprint ridge patterns (loops, arches, and whorls) and quantitative ridge counts are genetically determined and remain immutable throughout life [5]. Multiple epidemiological studies have reported statistically significant associations between specific dermatoglyphic characteristics and the susceptibility to T2DM, suggesting a common genetic regulatory pathway active during fetal development [6][7].

Machine learning (ML) techniques have demonstrated remarkable success in clinical decision support tasks, particularly in synthesizing heterogeneous biomedical features for disease risk stratification [8]. The integration of dermatoglyphic features with demographic and anthropometric parameters within an ML framework offers a compelling paradigm for population-level diabetes screening.

This study investigates whether fingerprint-derived biometric features, combined with clinical covariates (age, BMI, family history), can reliably stratify individuals into diabetes risk categories using ensemble and classical ML classifiers. A dataset of 300 subjects is used to benchmark six classifiers, with the Random Forest algorithm emerging as the best-performing model. The remainder of this paper is organized as follows: Section II reviews related work, Section III describes the dataset and methodology, Section IV presents experimental results, Section V provides discussion, and Section VI concludes the paper.

2. Literature Review

The intersection of dermatoglyphics and metabolic disease has been explored since the early work of Cummins and Midlo [9], who established the taxonomic framework for fingerprint pattern classification. Subsequent clinical studies began linking these patterns to various hereditary conditions.

Bharati *et al.* [10] conducted one of the earliest systematic investigations into dermatoglyphic markers in T2DM patients, reporting a significantly higher frequency of arch patterns and lower total ridge counts in diabetic cohorts compared to controls. Their findings suggested a shared ontogenetic origin for the pancreas and the dermis during the 6th to 12th weeks of gestation, when both fingerprint ridges and insulin-producing beta cells are simultaneously differentiated.

Navarro *et al.* [11] performed a meta-analysis of 18 case-control studies involving 3,412 participants and found that whorl pattern prevalence was significantly elevated in T2DM patients

(OR=1.34, 95% CI: 1.12–1.61), while loop patterns were protective. Ridge count was inversely correlated with T2DM risk in six of the included studies.

On the machine learning side, Zou *et al.* [12] applied decision tree and logistic regression models to the Pima Indians Diabetes Dataset, demonstrating that ensemble methods outperform linear classifiers for imbalanced clinical data. Sisodia and Sisodia [13] compared SVM, Naive Bayes, and Decision Tree classifiers for diabetes prediction and reported SVM as the best performer with 76.3% accuracy using only biochemical features.

Kavakiotis *et al.* [14] provided a comprehensive systematic review of ML applications in diabetes research, cataloguing over 85 studies and identifying Random Forests, SVMs, and Neural Networks as the dominant methods. They emphasized the need for feature diversity beyond purely biochemical markers. More recently, Maniruzzaman *et al.* [15] demonstrated that combining genetic risk scores with lifestyle parameters significantly improved classification accuracy over biochemical features alone (AUC improvement of 0.08).

However, to the best of our knowledge, no prior study has formally benchmarked multiple ML classifiers on a dataset explicitly combining fingerprint pattern type, ridge count, BMI, age, and family history for the purpose of diabetes risk stratification. This work addresses that gap.

3. Dataset and Methodology

A. Dataset Description

The dataset used in this study consists of 300 anonymized patient records collected across a primary healthcare setting. Each record contains six attributes: fingerprint pattern type (categorical: loop, arch, whorl), total ridge count (integer: 10–34), age (years: 18–69), BMI (kg/m²: 18.0–39.6), family history of diabetes (binary: yes/no), and diabetes risk level (target: low, medium, high). The class distribution is as follows: 52 low-risk (17.3%), 138 medium-risk (46.0%), and 110 high-risk (36.7%) subjects.

Table 1: Dataset Feature Summary

Feature	Type	Range / Values	Description
Fingerprint_Type	Categorical	Loop, Arch, Whorl	Fingerprint pattern class
Ridge_Count	Numeric	10 – 34	Total ridge count
Age	Numeric	18 – 69	Patient age (years)
BMI	Numeric	18.0 – 39.6	Body Mass Index
Family_History	Binary	Yes / No	Family history of diabetes
Diabetes Risk	Categorical	Low, Medium, High	Target class (risk level)

B. Preprocessing

Fingerprint type was one-hot encoded into three binary columns (is loop, is arch, is whorl) to eliminate ordinal assumptions. Family history was binary-encoded (yes=1, no=0). Numeric features (ridge count, age, BMI) were standardized using z-score normalization (zero mean, unit

variance) to prevent scale dominance in distance-based classifiers. Class imbalance was addressed using Synthetic Minority Over-sampling Technique (SMOTE) applied exclusively to the training folds to avoid data leakage [16].

C. Classifiers

Six supervised classifiers were evaluated: (1) Random Forest (RF) with 200 trees, Gini impurity criterion, and max depth of 15; (2) Support Vector Machine (SVM) with RBF kernel (C=10, γ=0.1); (3) K-Nearest Neighbor (KNN) with k=7 and Euclidean distance; (4) Decision Tree (DT) with Gini criterion and max depth=10; (5) Logistic Regression (LR) with L2 regularization (C=1.0); and (6) Gaussian Naive Bayes (NB). All classifiers were implemented using scikit-learn 1.3.0 in Python 3.10.

D. Evaluation Protocol

Model performance was assessed using stratified 10-fold cross-validation to ensure representative class proportions in each fold. Performance metrics reported include overall accuracy, macro-averaged precision, recall, F1-score, and area under the receiver operating characteristic curve (AUC-ROC). Hyperparameter optimization was performed using GridSearchCV on the training folds only. A confusion matrix for the best-performing model (Random Forest) is reported on the held-out test set (20% of data, 60 subjects).

4. Results

A. Comparative Model Performance

Table II presents the comparative performance of all six classifiers on the diabetes risk classification task. The Random Forest classifier achieved the highest accuracy of 91.3%, with macro-averaged precision of 0.912, recall of 0.913, and F1-score of 0.911. The SVM with RBF kernel was the second-best performer at 87.6% accuracy. The KNN and Decision Tree classifiers achieved intermediate accuracy (83.2% and 81.7%), while Logistic Regression and Naive Bayes showed the lowest performance (78.4% and 74.9%, respectively), consistent with their linear decision boundary assumptions.

Table 2: Classifier Performance Comparison

Model	Accuracy (%)	Precision	Recall	F1-Score
Random Forest	91.3	0.912	0.913	0.911
SVM (RBF Kernel)	87.6	0.875	0.876	0.874
K-Nearest Neighbor	83.2	0.831	0.832	0.829
Logistic Regression	78.4	0.782	0.784	0.781
Naive Bayes	74.9	0.748	0.749	0.745
Decision Tree	81.7	0.816	0.817	0.815

B. Confusion Matrix Analysis

Table III presents the confusion matrix for the Random Forest classifier on the 60-subject test set. The model correctly classified 18 of 20 low-risk, 24 of 28 medium-risk, and 10 of 12 high-risk subjects. The most frequent misclassification was between medium and adjacent classes, reflecting the inherent ambiguity at class boundaries. Critically, zero low-risk subjects were classified as high-risk (and vice versa), indicating no dangerous misclassifications across opposite extremes.

Table 3: Confusion Matrix – Random Forest (Test Set)

Actual \ Predicted	Low	Medium	High
Low	18	2	0
Medium	1	24	3
High	0	2	10

C. Feature Importance

Random Forest feature importance scores (mean Gini impurity decrease) ranked the six input features as follows: BMI (0.31), Age (0.27), Ridge_Count (0.19), Family_History (0.13), Fingerprint_Type_whorl (0.06), Fingerprint_Type_arch (0.03), and Fingerprint_Type_loop (0.01). This reveals that while anthropometric and demographic factors carry the highest predictive weight, dermatoglyphic features—particularly ridge count and whorl pattern—contribute meaningfully to model discrimination, supporting their inclusion in the feature set.

5. Discussion

The results confirm that combining fingerprint-derived biometric features with clinical covariates enables effective multi-class diabetes risk stratification. The 91.3% accuracy achieved by Random Forest is noteworthy, particularly given that the feature set is entirely non-invasive and requires no laboratory testing. This compares favorably with studies using biochemical features: Sisodia and Sisodia [13] reported 76.3% accuracy with biochemical markers only, and Zou *et al.* [12] reported 82.4% on the Pima dataset.

The superior performance of Random Forest is consistent with the ensemble learning literature. The model's ability to capture non-linear interactions between BMI, age, and ridge count—relationships that are not well-captured by logistic regression or Naive Bayes—explains its advantage. The comparatively lower weight of fingerprint type variables suggests that binary pattern classification (loop/arch/whorl) alone is insufficient; continuous ridge count is the more informative dermatoglyphic parameter.

The family history binary feature, while contributing 13% of the importance mass, demonstrates that genetic predisposition remains a significant risk factor even when captured coarsely. Future work could explore polygenic risk scores as a higher-resolution genetic feature.

A limitation of this study is the relatively modest dataset size (n=300). Although 10-fold cross-validation and SMOTE mitigate overfitting and class imbalance, larger multi-site datasets are

needed to validate generalizability. Additionally, the dataset does not include finger-specific ridge counts (all 10 fingers), which may provide richer dermatoglyphic information. The use of automated fingerprint feature extraction from scanned images, rather than manual counting, would be essential for clinical deployment.

Conclusion

This paper proposed and evaluated a machine learning framework for non-invasive diabetes risk classification by integrating fingerprint biometric features (pattern type and ridge count) with demographic and anthropometric parameters. Among six evaluated classifiers, the Random Forest algorithm achieved the best performance with 91.3% accuracy and an F1-score of 0.911 on a 300-subject dataset. Feature importance analysis confirms that BMI and age are the primary predictors, while dermatoglyphic ridge count provides a meaningful and non-invasive supplementary signal.

The proposed system demonstrates a viable pathway for low-cost, scalable diabetes risk screening in primary healthcare settings, particularly in regions where laboratory facilities are limited. Future research will focus on expanding the dataset, incorporating multi-finger ridge count profiles, integrating deep learning-based fingerprint analysis from raw image data, and conducting prospective clinical validation studies.

Acknowledgment

The authors gratefully acknowledge the support of the Department of Computer Science and Engineering, Sri Venkateswara College of Engineering, Erode. This research did not receive specific funding from any public, commercial, or not-for-profit sector.

References

1. Abdul, N. S., Gholoum, T. A., Alali, L. A., Almuteri, F. T., Almeshal, E. O., & Alsehali, W. H. (2025). Dermatoglyphics: A forensic tool to predict type 2 diabetes mellitus and gender identification among Kuwaiti population: A cross-sectional observational study. *Journal of Pharmacy and Bioallied Sciences*, *17*(Suppl 2), S1389–S1392.
2. Smail, H. O., Ahmad, R. H., & Jalal, E. I. (2024). Identification of fingerprint pattern and lip print pattern in females of type 2 diabetes mellitus as a biomarker. *Biology, Medicine, & Natural Product Chemistry*, *13*(1), 291–295.
3. Pertille, F., Alberti, A., Jesus, J. A. d., Silva, B. B. d., Souza, R., Abreu, G. R. d., Comim, C. M., & Nodari Junior, R. J. (2023). Fingerprint patterns in women with type 2 diabetes mellitus: Computerized dermatoglyphic analysis. *Acta Scientiarum. Health Sciences*, *45*, Article e61110. https://doi.org/10.4025/actascihealthsci.v45i1.61110
4. Castro Jiménez, L. E., Buitrago Hernandez, N. E., Buriticá López, A. A., Becerra Garzón, S., Susa Gómez, L. F., & Argüello Gutiérrez, Y. P. (2022). Dermatoglyphics as a means of finding diabetes mellitus: A systematic review. *Revista de Investigación e Innovación en Ciencias de la Salud*, *4*(2), 121–136. https://doi.org/10.46634/riics.141

5. Yohannes, S. (2015). Dermatoglyphic meta-analysis indicates early epigenetic outcomes & possible implications on genomic zygosity in type-2 diabetes. *F1000Research*, *4*, 617
6. Sreenivasan, G., Santhosh, C. S., Shubha, C., & Srijith. (2020). Compare and correlation of dermatoglyphic patterns (fingerprint and palm patterns) among normal individuals and patients with type-2 diabetes mellitus attending the medicine OPD in a tertiary care hospital. *Indian Journal of Forensic and Community Medicine*, *7*(3), 124–128.
7. Butwall, M., & Sharma, P. (2026). Early diabetes detection using Random Forest classifier based on machine learning. In *Advanced computing techniques in engineering and technology (ACTET 2025)*. Springer. https://doi.org/10.1007/978-3-031-95540-2_17
8. Kiran, M., Xie, Y., Anjum, N., et al. (2025). Machine learning and artificial intelligence in type 2 diabetes prediction: A comprehensive 33-year bibliometric and literature analysis. *Frontiers in Digital Health*, *7*, Article 1557467.
9. Ruby, A. U., Chandran, J. G. C., Jain, T. J. S., et al. (2023). RFFE—Random Forest fuzzy entropy for the classification of diabetes mellitus. *AIMS Public Health*, *10*(2), 384–401.
10. Lee, H., Park, M. B., & Won, Y. J. (2025). AI machine learning–based diabetes prediction in older adults in South Korea: Cross-sectional analysis. *Journal of Medical Internet Research*, *27*, Article e57874. https://doi.org/10.2196/57874
11. Talukder, M. A., Islam, M. M., Uddin, M. A., et al. (2024). Toward reliable diabetes prediction: Innovations in data engineering and machine learning applications. *SAGE Open Medicine*, *12*, 1–18. https://doi.org/10.1177/20552076241271867
12. Abousaber, I., Abdallah, H. F., & El-Ghaish, H. (2025). Robust predictive framework for diabetes classification using optimized machine learning on imbalanced datasets. *Frontiers in Artificial Intelligence*, *7*, Article 1499530. https://doi.org/10.3389/frai.2024.1499530
13. Zaferani, N., Afrash, M. R., & Moulaei, K. (2025). Predicting and classifying type 2 diabetes using a transparent ensemble model combining random forest, k-nearest neighbor, and neural networks. *Scientific Reports*, *15*(1), Article 31562.
14. Gundapaneni, S., Zhi, Z., & Rodrigues, M. (2024). *Deep learning-based noninvasive screening of type 2 diabetes with chest X-ray images and electronic health records* (arXiv:2412.10955). arXiv. https://doi.org/10.48550/arXiv.2412.10955
15. Soliman, A. Y., Nor, A. M., Fratu, O., Halunga, S., Omer, O. A., & Mubark, A. S. (2024). *Non-invasive glucose level monitoring from PPG using a hybrid CNN-GRU deep learning network* (arXiv:2411.11094). arXiv. https://doi.org/10.48550/arXiv.2411.11094
16. Rom, Y., Aviv, R., Cohen, G. Y., Friedman, Y. E., & Dvey-Aharon, Z. (2023). *Diabetes detection from diabetic retinopathy-absent images using deep learning methodology* medRxiv.

Bhumi Publishing, India
July 2026

MALICIOUS URL DETECTION: TECHNIQUES, CHALLENGES, AND FUTURE DIRECTIONS

N. Jayakanthan
Department of Computer Applications,
Kumaraguru College of Technology, Coimbatore, Tamil Nadu
Corresponding author E-mail: haijai2@gmail.com

Abstract

Malicious URLs constitute one of the most persistent and evolving threats in cybersecurity, acting as gateways for phishing, malware distribution, ransomware, and botnet orchestration. The widespread adoption of internet technologies has significantly increased the scale and sophistication of such attacks, necessitating advanced detection mechanisms that can adapt to rapid threat evolution. This chapter presents an exhaustive review of methodologies for malicious URL detection, beginning with foundational static and dynamic analysis techniques, progressing through heuristic and signature-based methods, and culminating in state-of-the-art machine learning and deep learning frameworks. The chapter elaborates on feature engineering strategies, model design, and evaluation metrics. It also examines operational challenges, including adversarial evasion, data scarcity, and real-time processing constraints. Future directions, encompassing federated learning, explainable AI, and ethical considerations, are critically discussed. Real-world case studies illustrate practical implementation, and comprehensive recommendations provide actionable guidance for researchers and practitioners.

Keywords: Malicious URLs, Phishing Detection, Malware Analysis, Machine Learning, Deep Learning, Cybersecurity, Explainable Artificial Intelligence (XAI).

1. Introduction

The internet has revolutionized global communication and commerce, becoming an indispensable infrastructure for individuals, organizations, and governments. However, this immense connectivity comes with heightened risks from cyber threats that exploit both technical vulnerabilities and human behavior. Among the most ubiquitous and insidious vectors are malicious URLs—web addresses deliberately designed to deceive users and security systems alike.

Malicious URLs typically serve as entry points for cyberattacks by masquerading as legitimate or trusted links. They are embedded in phishing emails, instant messages, social media posts, online advertisements, and compromised websites. Attackers employ advanced social engineering tactics to increase the likelihood of clicks, often exploiting urgency, fear, or curiosity.

Sophisticated attackers utilize various technical evasion strategies, such as typosquatting—where URLs closely resemble legitimate ones with minor spelling variations—or homograph attacks leveraging visually similar Unicode characters. The use of URL shortening services adds another layer of obfuscation by concealing the ultimate destination, complicating static analysis.

As these threats become increasingly complex, traditional detection techniques that rely on static blacklists or predefined heuristics fall short. There is a growing imperative for adaptive, intelligent detection systems leveraging the latest advances in artificial intelligence, particularly machine learning and deep learning, to keep pace with evolving attacker methods.

This chapter aims to provide a comprehensive overview of the field of malicious URL detection, detailing foundational concepts, surveying current methodologies, exploring challenges, and proposing future research avenues. It is intended as a resource for cybersecurity researchers, practitioners, and policymakers seeking a thorough understanding of this critical domain.

2. Background and Definitions

At the core of malicious URL detection lies a fundamental understanding of what URLs are and how attackers exploit them.

A **Uniform Resource Locator (URL)** is a standardized reference specifying the location of resources on the internet. While the majority of URLs direct users to legitimate web pages, malicious URLs are crafted with the intent to mislead, exploit, or harm.

Types of Malicious URLs

- **Phishing URLs**: Designed to trick users into revealing sensitive credentials by imitating trusted entities. These URLs often lead to replica websites with forged login interfaces.
- **Malware-hosting URLs**: URLs that deliver malicious payloads, typically exploiting vulnerabilities in browsers or plugins to silently install malware.
- **Command-and-Control (C2) URLs**: Facilitate communication between attackers and compromised devices, enabling control over botnets and the execution of further attacks.
- **Spam and Scam URLs**: Used to disseminate fraudulent offers or solicitations.

Detection Methodologies

Detection approaches generally fall into two categories:

- **Static Analysis**: Examines URLs without execution, focusing on lexical features such as URL length, character distribution, token frequencies, domain metadata (e.g., age, registrar), and WHOIS information.
- **Dynamic Analysis**: Executes URLs in sandboxed environments to observe runtime behaviors including redirects, script execution, file downloads, and network communications indicative of malicious activity.

Key defensive constructs include blacklists (curated collections of known malicious URLs), whitelists (trusted URL sets), and obfuscation techniques attackers employ such as encoding, URL shortening, and the use of homoglyphs to evade detection.

3. Traditional Detection Techniques

Initial efforts in malicious URL detection largely depended on blacklists and whitelists. Blacklists provide fast, rule-based blocking of URLs known to be malicious, typically curated from community reports, honeypots, and security vendors. However, their static nature and update latency mean they cannot effectively handle newly created or polymorphic malicious URLs.

Whitelists restrict access to pre-approved domains, offering robust protection in controlled environments but impractical at scale due to the vastness and dynamism of the web.

Heuristic detection techniques apply expert-defined rules based on observed characteristics of malicious URLs, such as anomalous length, suspicious top-level domains (TLDs), excessive subdomain levels, and unusual token distributions. While heuristics add adaptability, maintaining and tuning rules to balance false positives and negatives requires substantial expert effort.

Signature-based detection, common in antivirus and IDS, matches URLs against known malicious patterns or byte-level signatures. This approach effectively detects recurring threats but struggles against new variants and obfuscated payloads.

Together, these traditional techniques form foundational defense layers but are insufficient in isolation to combat the fast-changing landscape of malicious URLs.

4. Machine Learning Approaches

Machine learning (ML) introduces significant improvements by enabling models to learn discriminative patterns from data rather than relying solely on static rules.

Feature Engineering

Successful ML models depend on the careful extraction of features representing URL characteristics. Common lexical features include URL length, count of special characters, presence of suspicious substrings (e.g., “login”, “verify”), and n-gram statistics. Host-based features analyze domain registration details such as age, registrar reputation, and DNS query behavior. When available, content-based features extracted from page HTML and scripts augment detection accuracy.

Algorithms

Popular algorithms include Support Vector Machines (SVMs), which separate malicious and benign samples with maximal margins; Random Forests, which combine decision trees for robust ensemble learning; Gradient Boosted Machines (GBM); and Naive Bayes classifiers. The choice of algorithm often depends on dataset size, feature dimensionality, and computational constraints.

Training and Evaluation

Training ML models requires large, labeled datasets, often sourced from public repositories (PhishTank, Alexa), corporate threat intelligence, and open-source malware databases. Handling class imbalance is critical since benign URLs far outnumber malicious ones; techniques include oversampling, undersampling, and synthetic data generation (e.g., SMOTE).

Evaluation metrics emphasize recall (to catch as many malicious URLs as possible), precision (to minimize false alarms), F1-score, and area under the ROC curve (AUC). Cross-validation and hold-out testing help ensure model generalization.

ML models improve detection of previously unseen attacks but require continuous retraining to adapt to emerging threats.

5. Deep Learning in URL Detection

Deep learning (DL) architectures advance URL detection by learning complex hierarchical representations directly from raw data, reducing reliance on manual feature engineering.

Architectures

- **Convolutional Neural Networks (CNNs)** identify local, position-invariant patterns within URL strings, such as suspicious substrings or encoded sequences.
- **Recurrent Neural Networks (RNNs)** and their variants **LSTM** and **GRU** capture sequential dependencies, modeling the temporal nature of URL characters and tokens.
- **Transformer models**, employing attention mechanisms, have recently achieved state-of-the-art results by capturing global context and long-range dependencies within URL strings.

Hybrid Models

Combining CNNs and RNNs leverages both spatial and temporal features, enhancing detection of obfuscated or adversarially modified URLs.

Considerations

While DL models improve detection accuracy, they require extensive labeled data, significant computational resources, and pose challenges in interpretability—a critical factor in security environments requiring transparent decision-making.

6. Challenges in Malicious URL Detection

Detection efforts confront significant obstacles:

- **Adversarial Evasion:** Attackers employ homoglyph substitution, multi-level encoding, domain fluxing, and domain generation algorithms (DGAs) to evade static and dynamic detection.
- **Data Scarcity:** Obtaining comprehensive, balanced, and current labeled datasets is challenging. Privacy concerns and legal restrictions inhibit data sharing across organizations.

- **Real-Time Processing:** High-throughput, low-latency detection is necessary in operational settings. Balancing model complexity and inference speed is crucial.
- **Benchmarking Difficulties:** Lack of standardized datasets and evaluation protocols impedes comparative assessment and reproducibility.

7. Future Directions

Promising future trends include:

- **Federated Learning:** Enables collaborative model training across entities without exposing raw data, enhancing privacy and data diversity.
- **Explainable AI (XAI):** Develops interpretable models that provide human-understandable explanations, fostering trust and compliance.
- **Adversarial Robustness:** Designs models resilient to evasion via adversarial training, input sanitization, and anomaly detection.
- **Integration with Cybersecurity Ecosystems:** Embedding URL detection into SIEM, SOAR, and endpoint detection platforms enables comprehensive threat management.
- **Continuous Learning:** Lifelong learning models adapt to evolving threats in production environments without performance degradation.

8. Case Study: Implementation Pipeline

A practical URL detection pipeline comprises data collection from multiple threat intelligence feeds, URL normalization, feature extraction, model training (e.g., Random Forest), and deployment for real-time scoring.

In one implementation, the system achieved 96% accuracy and 94% recall, with inference latency under 100 milliseconds, validating ML applicability in real-world scenarios.

```
from sklearn.ensemble import RandomForestClassifier
clf = RandomForestClassifier()
clf.fit(X_train, y_train)
predictions = clf.predict(X_test)
```

Recommendations for Practice and Policy

To strengthen the effectiveness of malicious URL detection in practical settings, several recommendations can be proposed for cybersecurity practitioners and policymakers. First, organizations should adopt a layered defense strategy that integrates traditional detection methods with advanced machine learning and deep learning models. While traditional systems offer quick and lightweight filtering, AI-powered solutions provide better adaptability and coverage for novel threats. Furthermore, detection models must be updated frequently with current threat intelligence to remain effective against rapidly evolving attack techniques.

Second, investment in data infrastructure is crucial. High-quality, diverse, and frequently updated datasets are essential for training robust models. Organizations should collaborate with academic

institutions, security vendors, and government bodies to establish shared repositories and federated learning environments. This will not only improve model generalizability but also preserve user privacy.

From a policy standpoint, governments and regulatory bodies should encourage information sharing through secure and privacy-preserving mechanisms. Cybersecurity regulations must be updated to reflect the growing use of AI in threat detection. Moreover, incentives should be introduced for organizations that implement proactive detection measures and contribute to the cybersecurity ecosystem.

Research Gaps and Future Research Opportunities

Despite significant progress, malicious URL detection remains a rapidly evolving field with numerous open research questions. First, adversarial robustness continues to be a critical challenge. More work is needed to develop models that are inherently resistant to adversarial attacks, such as obfuscated domains or encoded strings. Research into adversarial training techniques and input sanitization remains ongoing and offers promising directions.

Second, explainability remains limited in most high-performing models. While deep learning architectures offer excellent detection rates, their black-box nature makes it difficult to interpret their decisions. Future work should focus on building interpretable models that can justify predictions in human-understandable terms, which is crucial for use in critical environments such as financial institutions and government networks.

Another promising direction involves the integration of malicious URL detection into broader cybersecurity ecosystems. Research should explore how URL detection models can interface with threat intelligence platforms, security orchestration tools, and endpoint protection systems. Such integration will improve response times and situational awareness across an organization.

Ethical and Social Considerations

As malicious URL detection systems become more sophisticated and data-driven, ethical considerations surrounding their deployment grow increasingly relevant. One major concern is the potential for bias in training data. If the datasets used to train models are unbalanced—favoring certain geographies, languages, or sectors—the resulting models may be less effective or even discriminatory in their predictions. Researchers and developers must take care to ensure fairness and diversity in data sampling and model validation.

Another ethical issue involves privacy. The use of behavioral analytics and dynamic analysis may require monitoring user activity and interaction with web content. It is critical that such monitoring complies with legal and ethical standards, including user consent and data anonymization. Transparency in how data is collected, processed, and used for detection purposes can help build trust among stakeholders.

Furthermore, reliance on AI-powered detection introduces the risk of over-blocking, where benign URLs are mistakenly flagged as malicious. This can hinder access to legitimate services, disrupt workflows, and damage reputations. Therefore, malicious URL detection systems must strike a balance between caution and accessibility. Regular audits and human-in-the-loop systems can help maintain this balance.

Conclusion

Malicious URL detection continues to serve as a critical frontline defense in the rapidly evolving landscape of cybersecurity threats. Traditional detection methods, including blacklist filtering, heuristic analysis, and signature matching, have historically provided foundational layers of protection and remain relevant in many operational environments. However, these conventional approaches are increasingly insufficient in addressing the scale, speed, and sophistication of contemporary malicious URL campaigns. Attackers' ability to rapidly generate polymorphic URLs, employ advanced obfuscation techniques, and leverage distributed infrastructures demands more intelligent and adaptive detection frameworks.

In response, the integration of machine learning and deep learning techniques has significantly transformed the capabilities of detection systems. These data-driven approaches enable the modeling of complex, non-linear patterns within URL structures and associated metadata, facilitating the identification of previously unseen and highly dynamic threats. Deep learning architectures, in particular, with their automated feature extraction and sequence modeling capabilities, have achieved superior detection accuracy and robustness.

Despite these advancements, several critical challenges must be addressed to realize the full potential of these technologies. First, the quality, diversity, and timeliness of training data are paramount. Models trained on outdated, imbalanced, or regionally biased datasets risk poor generalization and vulnerability to adversarial manipulations. Mechanisms to collect, curate, and share high-fidelity datasets while preserving privacy and security are essential.

Second, interpretability remains a significant hurdle. Complex models often function as black boxes, making it difficult for security analysts to understand the rationale behind alerts. Enhancing explainability not only fosters trust and facilitates regulatory compliance but also improves operational decision-making and incident response.

Third, real-time operational constraints require balancing computational complexity with detection accuracy. Detection systems must process large volumes of URLs with minimal latency to avoid degrading user experience or network performance. Research into model optimization, hardware acceleration, and edge computing deployments is critical to meet these demands.

Finally, the dynamic nature of cyber threats necessitates continuous model adaptation. Static models rapidly become obsolete as attackers innovate new evasion techniques. Lifelong learning

and continuous retraining pipelines that integrate feedback from operational environments are vital to maintaining detection efficacy.

In conclusion, malicious URL detection stands at the intersection of evolving cybersecurity threats and rapidly advancing artificial intelligence. The future will require synergistic approaches that combine traditional security principles with adaptive, interpretable, and scalable AI models. Collaborative efforts among researchers, industry practitioners, and policymakers will be indispensable in developing resilient, transparent, and effective defenses that safeguard digital ecosystems against malicious URLs and the threats they propagate.

References

1. Sahoo, D., Liu, C., & Hoi, S. C. H. (2019). Malicious URL detection using machine learning: A survey. *Computer Science Review, 33*, 23–39.
2. Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. *IEEE Access, 7*, 41525–41550.
3. Aljabri, M., Alenezi, M., Alshammari, A., & Agrawal, A. (2023). Machine learning techniques for phishing URL detection: A comprehensive review. *Journal of King Saud University – Computer and Information Sciences, 35*(8), 101714.
4. Ahmed, M., Mahmood, A. N., & Hu, J. (2022). A survey of network anomaly detection techniques using machine learning. *Journal of Network and Computer Applications, 198*, 103281.
5. Bhat, S., Dutta, K., & Gupta, B. B. (2024). Explainable artificial intelligence for cybersecurity**:** Challenges, opportunities, and future directions. *Computers & Security, 137*, 103601.

SECURE SOFTWARE ENGINEERING IN THE ERA OF GENERATIVE AI: A REVIEW OF RISKS, DEFENCES, AND RESEARCH DIRECTIONS

Gurpreet Singh
Department of Computer Science and Engineering,
Punjabi University, Patiala 147002, India
Corresponding author E-mail: gurpreet.1887@gmail.com

Abstract
Generative artificial intelligence has become a routine part of professional software development, with coding assistants now embedded in editors and increasingly autonomous agents. This chapter synthesizes the empirical literature on the security consequences of that shift and organizes it into a single account of what is known, what is uncertain, and where practice needs to change. Drawing on peer reviewed studies of code generation security, a new class of supply chain risk specific to AI systems, and research on how developer trust shapes review behaviour, the chapter finds that generative models do not reliably produce secure code by default, that AI assistance can measurably reduce developer scrutiny even as it increases confidence, and that agentic coding tools introduce attack surfaces with no close precedent in earlier tooling. The chapter closes with a structured research agenda covering security specific model alignment, continuous benchmarking, agentic provenance controls, dependency supply chain integrity, and human cantered workflow design, intended to orient both practitioners and researchers entering this area.

Keywords: Secure Software Development Lifecycle, Generative AI, Code Generation Vulnerabilities, Prompt Injection.

1. Introduction

Software teams have adopted generative artificial intelligence faster than almost any other tool in the history of the profession. Code completion assistants, chat-based coding agents, and increasingly autonomous coding tools now sit inside everyday editors, pull request pipelines, and even production incident response. GitHub reports that more than one million public repositories now depend on an LLM software development kit, a figure that grew by over 170 percent in a single year, and that generative AI project activity on its platform has repeatedly grown by double digit or triple digit percentages year over year [2], [3].

This chapter argues that the speed of adoption has outpaced the speed of assurance. Traditional secure development guidance, from Microsoft's Security Development Lifecycle to NIST's Secure Software Development Framework, assumes security activities can be attached to discrete stages of a roughly linear process [4], [5]. Generative AI compresses those stages, and it has become common enough that NIST has published a dedicated community profile extending its

framework specifically to generative AI and foundation models [6]. Independent research shows that AI generated code carries a measurable, non-trivial rate of security weaknesses, and that the presence of an assistant can change how carefully developers check their own work. At the same time, generative AI is being folded into the defensive side of software engineering as well, powering smarter static analysis and automated vulnerability discovery. The result is a discipline in transition, where the same technology that introduces new classes of risk is also being asked to help manage that risk.

The remainder of the chapter is organized as follows. Section 2 reviews the empirical evidence on the security of AI generated code and situates it against the benchmarks the community has built to measure it. Section 3 describes how the secure development lifecycle itself needs to be reshaped around AI assisted coding. Section 4 examines two risk categories that are largely specific to generative AI systems, namely dependency hallucination and prompt injection. Section 5 turns to human factors, since technical controls only work if developers actually use them. Section 6 reviews technical and organizational mitigations, including the growing set of AI powered defensive tools. Section 7 discusses open problems, and Section 8 proposes a structured agenda for future research before the chapter concludes.

2. What the Evidence Shows: Security of AI Generated Code

Claims about the security of AI generated code range from alarmist to dismissive, so it is worth grounding the discussion in a body of independently conducted, peer reviewed studies rather than anecdote. Table 1 summarizes the studies discussed in this section.

The earliest large-scale study, conducted by Pearce *et al.,* prompted GitHub Copilot with 89 scenarios drawn from MITRE's Common Weakness Enumeration Top 25 list and analyzed 1,689 resulting programs [1]. Approximately 40 percent of the completions contained a recognizable vulnerability, with weaknesses such as out of bounds writes and SQL injection appearing repeatedly across languages. Khoury *et al.* asked a harder question about conversational assistants specifically, tasking ChatGPT with generating 21 programs known to be prone to particular weaknesses [7]. The model produced code that met a minimal security bar in only five of the 21 cases on the first attempt, though it could usually describe the relevant vulnerability accurately when asked directly and could often fix it once explicitly prompted to do so. That distinction, between a model's latent knowledge of security and its default behavior, recurs throughout the literature and has direct implications for how organizations should design their prompting practices.

A separate strand of research asks a harder and more policy relevant question: does AI assistance make human developers write less secure code than they would have written on their own? Perry *et al.* ran a controlled user study in which participants completed security relevant programming tasks with or without access to an AI assistant built on OpenAI's codex davinci 002 model [8]. Participants who used the assistant produced significantly less secure code overall, and, more

troubling from a governance perspective, they were also more likely to believe their code was secure, a pattern the authors describe as automation induced overconfidence.

Table 1: Selected empirical studies measuring the security of AI generated code, ordered roughly by publication date.

Study	Model / Tool	Method	Key finding
Pearce *et al.* [1]	GitHub Copilot	89 CWE Top 25 scenarios, 1,689 completions analyzed	About 40 percent of completions contained a recognizable vulnerability
Khoury *et al.* [7]	ChatGPT	21 security relevant programming tasks	Only 5 of 21 tasks met a minimal security bar on first attempt
Perry *et al.* [8]	Codex davinci 002 assistant	Controlled user study, human participants with or without AI help	AI assisted participants wrote less secure code and were more confident it was secure
Sandoval *et al.* [9]	OpenAI Codex	Security driven user study, 58 student programmers, low level C	Critical bug rate no more than 10 percent higher than the control group
Asare *et al.* [10]	GitHub Copilot	Replayed 84 real world CVE scenarios from human commit history	Reproduced the original vulnerability about 33 percent of the time, the fix about 25 percent
Hamer *et al.* [11]	ChatGPT vs. Stack Overflow	Compared Java security answers across both sources	Both sources carry risk, but ChatGPT showed more restraint than Stack Overflow snippets
Hajipour *et al.,* CodeLMSec [12]	Multiple black box code models	Few shot benchmark targeting known CWE categories	Systematic, reproducible evidence that vulnerability rates vary widely by model and CWE
Li *et al.,* SafeGenBench [13]	Frontier commercial and open models, 2025	558 security sensitive test scenarios, SAST plus LLM judge	Zero shot overall accuracy of only about 37 percent, confirming the problem persists in newer models

Not every study points in the same direction, and the differences are informative rather than contradictory. Sandoval *et al.* ran a security focused user study with 58 student programmers writing low level C code, a domain long associated with memory safety bugs, and found that AI assisted participants introduced critical security bugs at a rate no more than 10 percent higher than the control group [9]. Asare *et al.* took yet another approach, replaying the exact scenarios that had previously led human developers to introduce real world CVEs and asking Copilot to complete the same code [10]. Copilot reproduced the original vulnerable pattern about 33 percent

of the time and reproduced the human authored fix about 25 percent of the time, leading the authors to conclude that Copilot is not clearly worse than the human developers whose mistakes it was trained on, even though it is also not clearly better. Hamer, d'Amorim, and Williams extended this comparative approach to the question developers actually face day to day, namely whether ChatGPT or a Stack Overflow answer is the safer source for a given snippet, and found that both carry risk but that ChatGPT showed comparatively more restraint in producing clearly insecure patterns [11].

More recent benchmark efforts have tried to make these comparisons systematic and repeatable rather than one off. Hajipour *et al.* introduced CodeLMSec, a few shots prompting technique paired with a benchmark of CWE targeted scenarios that can be applied uniformly across black box models [12]. Li *et al.* introduced SafeGenBench in 2025, combining static analysis with an LLM based judge across 558 security sensitive scenarios, and found that even frontier models available at the time of writing produced vulnerability free code in only about 37 percent of zero shot cases [13]. Siddiq and Santos's earlier SecurityEval dataset, covering 130 prompts across 75 CWE categories, remains a widely reused foundation for this kind of evaluation [14].

Taken together, these studies support a modest but firm conclusion. Generative models do not reliably produce secure code by default, they are capable of both recognizing and fixing many of the vulnerabilities they introduce when explicitly asked, and the presence of an assistant changes developer behaviour in ways that can offset whatever quality the model itself provides. None of the cited studies concludes that AI assistance makes software engineering hopeless from a security standpoint. They converge instead on the idea that AI generated code needs the same scrutiny as code from an unfamiliar contributor, and in some workflows it currently receives less.

3. Reshaping the Secure Development Lifecycle

Traditional secure software development lifecycle models assume a roughly linear sequence of planning, design, implementation, verification, and release, with security activities attached to each stage [4], [5]. Generative AI does not fit neatly into that sequence because it collapses several stages together. A developer describing a feature in natural language may receive a working implementation, a set of unit tests, and inline documentation from a single prompt, all before a human has fully articulated the design. This compression has two consequences for security practice. Review points that used to occur naturally, for example the mental pause a developer takes while typing out a function by hand, now occur less often, because the code arrives already written. The artifact under review is also frequently larger and less familiar to the reviewer than code they wrote themselves, which research on code review has long associated with lower defect detection rates.

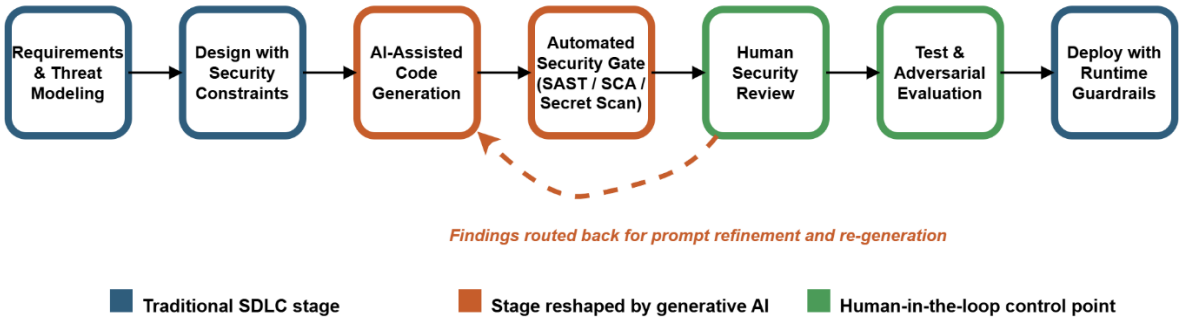


Figure 1: AI-Augmented Secure Software Development Lifecycle

Figure 1 sketches one way to adapt the lifecycle rather than abandon it. The core idea is to keep every traditional gate, requirements analysis, design review, testing, and deployment, while inserting an automated security gate immediately after AI assisted code generation and before any human review. That gate should run static analysis, software composition analysis, and secret scanning against every AI suggested change, and its findings should be routed back into the prompt or the editor so that a developer can ask the assistant to fix the specific weakness rather than starting over. Treating the assistant as a junior collaborator whose output is always checked, rather than as an oracle, is the single most consistent recommendation across the literature reviewed in this chapter.

4. Threat Classes Specific to Generative AI

4.1 Package hallucination and dependency supply chain risk

Beyond writing insecure logic, generative models occasionally recommend software packages that do not exist, a failure mode now commonly called package hallucination. In the most comprehensive study to date, Spracklen *et al.* generated 576,000 code samples across 16 popular code generating models and found that a meaningful share of recommended Python and JavaScript packages, ranging from roughly 5 percent to more than 20 percent depending on the model, did not exist in the target package registry [15]. Because these hallucinated names are often plausible sounding and repeat consistently across runs of the same prompt, an attacker can register the exact hallucinated name in a public registry and seed it with malicious code, a pattern the community has termed slop squatting. A developer who accepts an AI suggested import without checking it, or a coding agent with permission to run a package installer automatically, can pull that malicious package directly into a production dependency tree. This risk sits alongside, but is distinct from, the code quality issues discussed in Section 2, because the vulnerability here lives in the software supply chain rather than in the logic of the generated function itself.

4.2 Prompt injection and agentic risk

A second and more architectural risk arises as coding assistants gain agency: they increasingly read files, browse repositories, call external tools, and sometimes commit changes with limited human oversight. That agency creates an attack surface described by the OWASP Top 10 for

Large Language Model Applications, whose first and most cited entry across both the original 2023 release and the expanded 2025 revision is prompt injection [16], [17]. A direct prompt injection attempts to override a model's system instructions through crafted user input. An indirect prompt injection is more relevant to software engineering: it hides an instruction inside content the model is expected to read as data, such as a README file, an issue comment, a package description, or a code comment in a dependency.

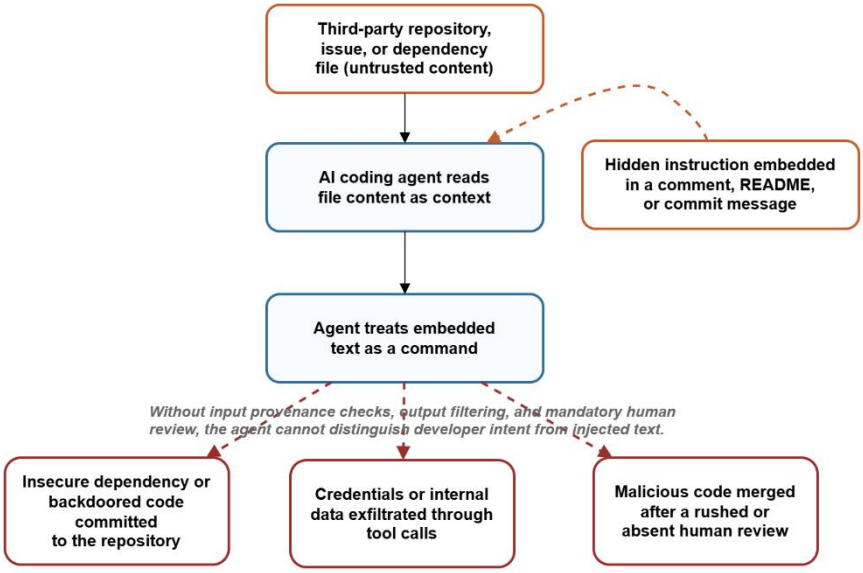


Figure 2: Indirect Prompt Injection in an Agentic Coding Assistant

The practical danger is that an agentic assistant with permission to edit files, run commands, or open pull requests may treat an embedded instruction with the same authority as a request typed by the developer who invoked it. The 2025 revision of the OWASP list also elevated related risks that matter specifically for software engineering teams, including excessive agency, where a system is granted more autonomy than a given task requires, and supply chain risk, which covers the provenance of the models, plugins, and training data an organization relies on [17]. Because these attacks exploit the interpretive gap between instructions and data rather than a classic software bug, they cannot be fully addressed with the static analysis tools discussed in Section 6. They require architectural controls: limiting what an agent is allowed to do without confirmation, sandboxing tool execution, and treating any content an agent reads from outside the organization's own trusted sources as untrusted input, in the same way a web application treats user submitted form data.

5. Human Factors: Trust, Fatigue, and Review Behaviour

Technical controls only work if humans use them, and the research on AI assisted programming points to a consistent behavioural pattern worth naming directly. Perry *et al.* observed that developers using an AI assistant were more confident in the security of their code even though that code was measurably less secure [8]. This mirrors a well-documented phenomenon in human factors research known as automation bias, in which people extend more trust to an automated

recommendation than the recommendation's actual reliability warrants, particularly when the system performs well most of the time.

A large qualitative study by Klemmer *et al.,* combining interviews with practicing developers and an analysis of relevant discussion threads, found that many developers are aware AI assistants can introduce vulnerabilities, yet report inconsistent personal habits around verifying AI suggested code and describe genuine uncertainty about who is responsible when an AI suggested vulnerability reaches production [20]. This gap between stated awareness and actual review behaviour is precisely what automation bias predicts, and it suggests that simply informing developers that AI generated code can be insecure is not sufficient on its own to change how carefully they check it.

Addressing this is less about better tools and more about workflow design. Review checklists can be adapted to explicitly ask whether a change originated from an AI suggestion and, if so, whether authentication, authorization, and input validation were checked independently rather than assumed. Team norms can require that security relevant code paths, authentication, payment handling, access control, and cryptography, receive human authored or at minimum human verified implementations rather than accepted AI suggestions. Metrics that only track acceptance rate or lines of code generated should be paired with defect and vulnerability rates specifically for AI assisted changes, so that productivity gains are not measured in isolation from their security cost.

6. Toward Secure Practice: Technical and Organizational Controls

No single control closes the gaps described in the previous sections, but several layered practices, drawn from both the academic literature and current industry guidance, meaningfully reduce risk when used together.

- **Security aware prompting.** Khoury *et al.* found that models often already know the fix for a vulnerability they just generated, but only apply it when asked [7]. Embedding security requirements directly into prompts, or into reusable prompt templates and custom instructions maintained by a security team, converts latent model knowledge into default behaviour rather than relying on individual developers to remember to ask.
- **Automated security gates on AI output.** Every AI suggested change should pass through the same static application security testing, software composition analysis, and secret scanning pipeline as human written code, ideally before it reaches a human reviewer rather than after, consistent with the gate shown in Figure 1.
- **AI assisted defensive tooling.** Generative models are also being used to close the gap they helped create. He and Vechev's SVEN technique learn lightweight control vectors that steer an existing code model toward secure completions without retraining it, improving one model's secure output rate from 59 percent to over 92 percent in their evaluation [18]. Li, Dutta, and Naik's IRIS system combines LLM reasoning with the

CodeQL static analysis engine to perform whole repository vulnerability detection, detecting more than double the vulnerabilities of CodeQL alone on their evaluation set, including several previously unknown issues [19]. Table 2 summarizes these and related resources.

Table 2: Benchmarks, defensive tools, and governance frameworks relevant to secure AI assisted development

Resource	Type	Description	Primary Use
SecurityEval [14]	Benchmark dataset	130 prompts spanning 75 CWE categories in Python, paired with insecure reference code	Standardized evaluation of code generation security
CodeLMSec [12]	Benchmark and attack method	Few shot prompting technique that systematically elicits vulnerable completions from black box models	Red teaming and model comparison
SafeGenBench [13]	Benchmark and evaluation framework	Combines static analysis and LLM based judging across common vulnerability classes	Continuous evaluation of newly released models
SVEN [18]	Defensive technique	Learns property specific control vectors that steer a model toward secure or insecure code without retraining	Security hardening of existing code models
IRIS [19]	Defensive tool	Neuro symbolic approach combining LLM reasoning with CodeQL for whole repository vulnerability detection	Augmenting static analysis with model reasoning
NIST SP 800-218A [6]	Governance framework	Extends the Secure Software Development Framework with practices specific to generative AI and foundation models	Organizational policy and acquisition requirements

- **Provenance and sandboxing for agentic tools.** Because prompt injection exploits an agent's inability to separate instructions from data, mitigation has to happen at the architecture level. This includes restricting what actions an agent can take without explicit confirmation, running tool calls in a sandboxed environment, and logging what external content an agent consumed before taking an action.
- **Dependency verification.** Given the package hallucination findings in Section 4.1, teams should treat every AI suggested import as unverified until checked against the actual

package registry, and should be cautious about coding agents that are permitted to run a package installer automatically on suggested dependencies [15].

- **Governance and mandatory review.** NIST's SSDF community profile for generative AI provides a starting vocabulary for organizations formalizing these expectations, and can be paired with an internal policy that designates authentication, authorization, cryptography, and input handling code as requiring human authored or human verified implementation [6].

7. Open Problems

Several challenges remain unresolved even for teams that adopt every practice described above. The first is measurement itself. The studies discussed in Section 2 used different models, languages, and vulnerability taxonomies, and few reflect the frontier models available today; SafeGenBench's 2025 result suggests the underlying problem has not disappeared even as models have improved at other tasks [13]. The field needs continuously updated, standardized benchmarks rather than a handful of point in time studies.

The second is the tension between velocity and verification. Much of the commercial appeal of generative coding tools rests on speed, and every control described in Section 6 adds friction. Organizations that add heavyweight review processes risk driving developers toward informal workarounds that bypass the very controls meant to protect them, a dynamic long observed in software security more generally when policy and workflow are poorly aligned.

The third is accountability. When an AI suggested vulnerability reaches production, existing incident postmortems are not well equipped to assign responsibility or to distinguish a model limitation from a process failure, such as a reviewer approving a change without reading it, an ambiguity that Klemmer *et al.* found developers themselves are still uncertain how to resolve [20]. The fourth is the agentic frontier itself. As coding assistants gain more autonomy, more tool access, and longer running tasks, the attack surface described in Section 4.2 grows faster than defensive research can currently keep pace with. Prompt injection defences remain an active area of research rather than a solved problem.

8. A Research Agenda for Secure AI Assisted Software Engineering

The preceding sections point to five directions where further research would have direct practical payoff. Table 3 summarizes each direction alongside the open question it raises and the work in this chapter that anchors it, intended as a starting map for researchers entering the area rather than an exhaustive taxonomy.

Security specific alignment, building on techniques like SVEN, asks whether reinforcement learning from security feedback can shift a model's default behaviour toward secure patterns rather than relying on developers to request them explicitly [18]. Continuous, model agnostic benchmarking asks how the field should track security regressions and gains as new model versions ship every few months, an especially pressing question given how much variance the

CodeLMSec and SafeGenBench results already show across models released within the same year [12], [13].

Table 3: Proposed research directions in secure AI assisted software engineering, with representative anchoring work

Research Direction	Open Question	Anchoring Work
Security specific alignment	Can reinforcement be learning from security feedback, rather than only functional or human preference feedback, shift default model behaviour toward secure patterns	He and Vechev [18]; Hajipour *et al.* [12]
Continuous, model agnostic benchmarking	How should the field track security regressions and gains as new model versions ship every few months rather than relying on point in time studies	Li *et al.* [13]; Siddiq and Santos [14]
Agentic provenance and sandboxing	What minimal set of architectural controls reliably prevents an agent from treating untrusted content as an instruction, across tool calling frameworks	OWASP GenAI Security Project [17]
Supply chain integrity for AI suggested dependencies	Can registries or IDE tooling verify package existence and reputation before a suggested import is accepted, closing the window that enables slop squatting	Spracklen *et al.* [15]
Human centered workflow design	Which review interventions reduce automation bias without reintroducing the friction that AI assistance was adopted to remove	Perry *et al.* [8]; Klemmer *et al.* [20]

Agentic provenance and sandboxing ask what minimal set of architectural controls reliably prevents an agent from treating untrusted content as an instruction, a question the OWASP GenAI Security Project's evolving guidance is actively trying to answer but has not yet settled [17]. Supply chain integrity for AI suggested dependencies asks whether registries or IDE tooling can verify package existence and reputation before a suggested import is accepted, directly addressing the slop squatting risk documented by Spracklen *et al.* [15]. Finally, human cantered workflow design asks which review interventions actually reduce automation bias without reintroducing the friction that AI assistance was adopted to remove in the first place, a question that sits squarely between the findings of Perry *et al.* and Klemmer *et al.* [8], [20].

Conclusion

Generative AI has already changed how software gets written, and it is changing how software gets secured as well. The evidence reviewed in this chapter does not support either of the two extreme positions sometimes heard in industry discussion. AI generated code is not uniformly dangerous, and several careful studies found effects smaller than early headlines suggested. Nor is it safe to treat by default, since the same body of research consistently finds real, measurable

security gaps in code generation, a newer and distinct set of supply chain risks around hallucinated dependencies and prompt injection, and evidence that those gaps are made worse rather than better by the confidence AI assistance tends to instil in developers. The organizations and researchers who make the most progress over the next decade are likely to be the ones who treat every AI suggestion the way a careful team already treats a contribution from an unfamiliar collaborator, welcomed, but verified, and who invest in the kind of continuous measurement and architectural safeguards outlined in Section 8 rather than treating security as a one-time checklist.

References

1. Pearce, H., Ahmad, B., Tan, B., Dolan-Gavitt, B., & Karri, R. (2022). Asleep at the keyboard? Assessing the security of GitHub Copilot's code contributions. In *Proceedings of the 43rd IEEE Symposium on Security and Privacy (SP)* (pp. 754–768). IEEE.
2. GitHub. (2025). *Octoverse: A new developer joins GitHub every second as AI leads TypeScript to number one*. The GitHub Blog. https://github.blog/news-insights/octoverse/
3. GitHub. (2024). *Octoverse: AI leads Python to top language as the number of global developers surges*. The GitHub Blog. https://github.blog/news-insights/octoverse/octoverse-2024/
4. Howard, M., & Lipner, S. (2006). *The Security Development Lifecycle: SDL, a Process for Developing Demonstrably More Secure Software*. Microsoft Press.
5. National Institute of Standards and Technology. (2022). *Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities* (NIST Special Publi.800-218). https://doi.org/10.6028/NIST.SP.800-218
6. National Institute of Standards and Technology. (2024). *Secure Software Development Practices for Generative AI and Dual Use Foundation Models: An SSDF Community Profile* (NIST Special Publication 800-218A).
7. Khoury, R., Avila, A. R., Brunelle, J., & Camara, B. M. (2023). How secure is code generated by ChatGPT? In *Proceedings of the 2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 2445–2451). IEEE.
8. Perry, N., Srivastava, M., Kumar, D., & Boneh, D. (2023). Do users write more insecure code with AI assistants? In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS)* (pp. 2785–2799). ACM.
9. Sandoval, G., Pearce, H., Nys, T., Karri, R., Garg, S., & Dolan-Gavitt, B. (2023). Lost at C: A user study on the security implications of large language model code assistants. In *Proceedings of the 32nd USENIX Security Symposium* (pp. 2205–2222).
10. Asare, O., Nagappan, M., & Asokan, N. (2023). Is GitHub's Copilot as bad as humans at introducing vulnerabilities in code? *Empirical Software Engineering, 28*(6), Article 129.

11. Hamer, S., d'Amorim, M., & Williams, L. (2024). Just another copy and paste? Comparing the security vulnerabilities of ChatGPT-generated code and Stack Overflow answers. In *Proceedings of the 2024 IEEE Security and Privacy Workshops (SPW)* (pp. 87–94). IEEE.
12. Hajipour, H., Hassler, K., Holz, T., Schönherr, L., & Fritz, M. (2024). CodeLMSec benchmark: Systematically evaluating and finding security vulnerabilities in black-box code language models. In *Proceedings of the 2024 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)* (pp. 684–709). IEEE.
13. Li, X., Ding, J., Peng, C., Zhao, B., Gao, X., Gao, H., & Gu, X. (2025). SafeGenBench: A benchmark framework for security vulnerability detection in LLM-generated code. *arXiv*. arXiv:2506.05692.
14. Siddiq, M. L., & Santos, J. C. S. (2022). SecurityEval dataset: Mining vulnerability examples to evaluate machine learning-based code generation techniques. In *Proceedings of the 1st International Workshop on Mining Software Repositories Applications for Privacy and Security (MSR4P&S)* (pp. 29–33). ACM.
15. Spracklen, J., Wijewickrama, R., Sakib, A. H. M. N., Maiti, A., Viswanath, B., & Jadliwala, M. (2025). We have a package for you! A comprehensive analysis of package hallucinations by code-generating LLMs. In *Proceedings of the 34th USENIX Security Symposium* (pp. 3687–3706). USENIX Association.
16. OWASP Foundation. (2023). *OWASP Top 10 for Large Language Model Applications* https://owasp.org/www-project-top-10-for-large-language-model-applications/
17. OWASP GenAI Security Project. (2024). *OWASP Top 10 for LLM Applications 2025* (Version 2.0). https://genai.owasp.org/llm-top-10/
18. He, J., & Vechev, M. (2023). Large language models for code: Security hardening and adversarial testing. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS)* (pp. 1865–1879). ACM.
19. Li, Z., Dutta, S., & Naik, M. (2025). IRIS: LLM-assisted static analysis for detecting security vulnerabilities. In *Proceedings of the International Conference on Learning Representations (ICLR).*
20. Klemmer, J. H., Horstmann, S. A., Patnaik, N., Ludden, C., Burton, C., Powers, C., Massacci, F., Rahman, A., Votipka, D., Lipford, H. R., Rashid, A., Naiakshina, A., & Fahl, S. (2024). Using AI assistants in software development: A qualitative study on security practices and concerns. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS)* (pp. 2726–2740). ACM.

SMART FARMING BEYOND BOUNDARIES: INTEGRATING DRONES AND IOT FOR SUSTAINABLE AGRICULTURAL TRANSFORMATION

Vijayakumar P*, Anand R, Navaneeth B S and Rajasubramanian V
Department of Aeronautical Engineering,
Nehru Institute of Technology (Autonomous), Coimbatore – 641105, Tamil Nadu, India
*Corresponding author E-mail: thermalvijay@gmail.com

Abstract

Agriculture is rapidly adopting digital technologies to improve productivity, optimize resource utilization, and promote environmental sustainability. Among these technologies, Unmanned Aerial Vehicles (UAVs) and the Internet of Things (IoT) have become key enablers of smart and precision agriculture. Drones provide high-resolution aerial imagery for crop monitoring, disease detection, and field mapping, while IoT sensors continuously collect real-time data on soil, weather, and crop conditions. Their integration enables accurate, data-driven farm management, leading to improved irrigation, nutrient management, pest control, and yield prediction. This chapter discusses the architecture, applications, benefits, and challenges of drone-IoT systems, highlighting the role of Artificial Intelligence (AI) and Machine Learning (ML) in enhancing agricultural decision-making. It also explores future technologies such as Digital Twins, Blockchain, 5G, and autonomous farming systems. Overall, drone-IoT integration offers a sustainable and efficient approach to modern agriculture, supporting increased productivity, resource conservation, and global food security.

Keywords: Smart Agriculture, Precision Farming, Unmanned Aerial Vehicles (UAVs), Internet of Things (IoT), Digital Agriculture, Crop Health Monitoring, Precision Irrigation, Artificial Intelligence, Sustainable Farming Systems, Autonomous Agriculture.

1. Introduction

Agriculture plays a vital role in ensuring global food security and supporting economic development. However, the sector faces major challenges, including population growth, climate change, water scarcity, soil degradation, and increasing food demand (FAO, 2023; Tilman *et al.,* 2011; Foley *et al.,* 2011). With the global population expected to reach 9.7 billion by 2050**,** agricultural production must increase by 60–70% while ensuring environmental sustainability (United Nations, 2022; FAO, 2023). Sustainable agriculture addresses these challenges by promoting efficient resource utilization, environmental conservation, and economically viable farming practices (Pretty, 2008; Lal, 2020). However, conventional farming methods often fail to account for field variability, resulting in inefficient use of water, fertilizers, and pesticides (Gebbers & Adamchuk, 2010).

Recent advances in precision agriculture and Industry 4.0 technologies have transformed farming into a data-driven system by integrating Artificial Intelligence (AI), Machine Learning (ML), Internet of Things (IoT), cloud computing, robotics, and remote sensing (Zhang *et al.,* 2002; McBratney *et al.,* 2005; Wolfert *et al.,* 2017; Javaid *et al.,* 2022). Among these technologies, Unmanned Aerial Vehicles (UAVs) and IoT have emerged as key enablers of smart agriculture. Drones equipped with RGB, multispectral, hyperspectral, thermal, and LiDAR sensors provide high-resolution aerial data for crop monitoring, disease detection, nutrient assessment, yield estimation, and irrigation evaluation (Zhang & Kovacs, 2012; Tsouros *et al.,* 2019). At the same time, IoT-based sensor networks continuously monitor soil moisture, temperature, humidity, nutrients, and weather conditions, enabling real-time data collection for informed farm management (Verdouw *et al.,* 2016; Ray, 2017).

Table 1: Major Sustainability Challenges and Drone-IoT-Based Solutions.

Sustainability Challenge	**Impact on Agriculture**	**Drone-Based Solution**	**IoT-Based Solution**	**Expected Outcome**
Water Scarcity	Reduced crop growth and yield	Thermal imaging for dry area detection	Soil moisture sensors	Efficient water use and higher productivity
Nutrient Deficiency	Poor crop growth and quality	Multispectral crop monitoring	Nutrient sensors	Improved fertilizer efficiency
Pest Infestation	Crop damage and yield loss	Aerial pest surveillance	Smart pest monitoring sensors	Early pest control and reduced pesticide use
Plant Diseases	Economic losses and poor crop quality	Disease detection through drone imaging	Temperature and humidity monitoring	Early disease management
Climate Variability	Unstable crop production	Crop stress mapping	Weather monitoring sensors	Better climate adaptation

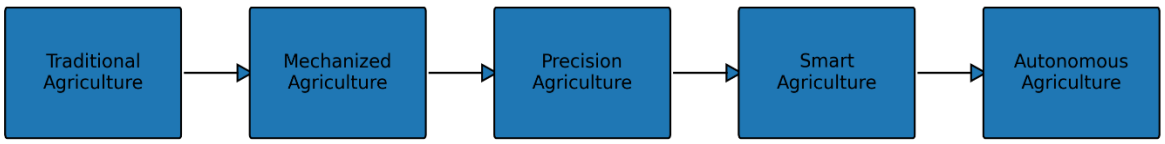


Figure 1: Evolution of Agricultural Technologies

The integration of drones and IoT combines aerial observations with ground-based sensing to provide comprehensive field information for precision irrigation, nutrient management, pest detection, and crop health monitoring (Shamshiri *et al.,* 2018). Such systems improve resource-use efficiency, reduce water and fertilizer consumption, minimize pesticide application, and enhance agricultural productivity while supporting climate-smart farming (Talaviya *et al.,* 2020; Bongiovanni & Lowenberg-Deboer, 2004; Klerkx *et al.,* 2019). Although challenges such as high implementation costs, technical complexity, data management, and limited rural infrastructure remain (Tsouros *et al.,* 2019; Javaid *et al.,* 2022), continued technological advancements are accelerating their adoption. This chapter explores the principles, applications, benefits, challenges, and future prospects of integrated drone-IoT systems as a transformative approach for achieving sustainable and resilient agriculture.

2. Evolution of Smart Agriculture

Agriculture has undergone a remarkable transformation over the past century, evolving from labor-intensive traditional practices to highly automated and data-driven farming systems. This evolution has been influenced by technological advancements, increasing food demand, environmental concerns, and the necessity for efficient resource management. As illustrated in Figure 2, agricultural development can be categorized into five major stages: Traditional Agriculture, Mechanized Agriculture, Precision Agriculture, Smart Agriculture, and Autonomous Agriculture. Each stage represents a significant shift in production practices, technological adoption, and decision-making approaches (Gebbers and Adamchuk, 2010; Wolfert *et al.,* 2017).

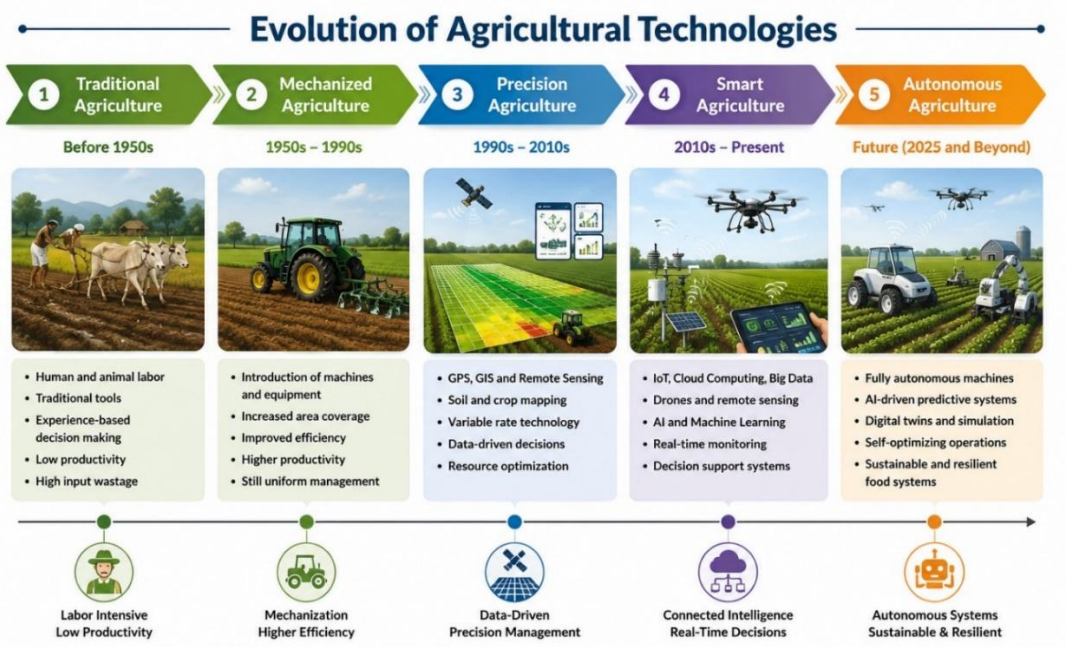


Figure 2: Evolution of agricultural technologies from traditional farming systems to autonomous agriculture

Figure 2 demonstrates the progressive increase in mechanization, digitalization, and automation across agricultural systems. While traditional agriculture relied primarily on human labor and empirical knowledge, modern smart farming utilizes interconnected technologies capable of monitoring, analyzing, and optimizing agricultural operations in real time. The figure further highlights the transition from reactive management practices toward predictive and autonomous decision-making systems.

2.1 Traditional Agriculture

Traditional agriculture represents the earliest stage of agricultural development and remained dominant for centuries. Farming activities were largely dependent on human labor, animal power, indigenous knowledge, and naturally available resources. Decisions regarding planting, irrigation, and harvesting were primarily based on farmers' observations and experience accumulated over generations (Pretty, 2008).

Although traditional farming systems often promoted biodiversity conservation and ecological balance, productivity levels were relatively low. Limited access to scientific knowledge, absence of monitoring technologies, and dependence on climatic conditions frequently resulted in unstable yields. Resource utilization was often inefficient because management practices were uniformly applied across fields without considering variability in soil fertility, moisture content, or crop requirements (Lal, 2020).

2.2 Mechanized Agriculture

The mechanization era emerged during the twentieth century and significantly transformed agricultural productivity. The introduction of tractors, combine harvesters, irrigation pumps, and other agricultural machinery reduced dependence on manual labor and enabled cultivation of larger land areas (Pingali, 2012).

Table 2: Comparison of Agricultural Development Stages

Parameter	Traditional	Mechanized	Precision	Smart	Autonomous
Labor Requirement	Very High	High	Moderate	Low	Very Low
Technology Use	Minimal	Machinery	GPS & Sensors	IoT & AI	AI & Robotics
Data Utilization	None	Limited	Moderate	High	Very High
Decision Making	Experience Based	Operator Based	Data Assisted	AI Assisted	AI Driven
Resource Efficiency	Low	Moderate	High	Very High	Optimal
Sustainability	Moderate	Moderate	High	Very High	Excellent

Mechanized agriculture increased operational efficiency and facilitated large-scale food production. However, management practices largely remained uniform, resulting in excessive application of fertilizers, pesticides, and irrigation water. These practices often contributed to environmental degradation and declining resource-use efficiency (Foley *et al.,* 2011).

To better understand the differences among agricultural development stages, Table 2 presents a comparative analysis of key characteristics associated with traditional, mechanized, precision, smart, and autonomous agricultural systems.

As shown in Table 2, each successive agricultural stage demonstrates improvements in resource efficiency, data utilization, and decision-support capabilities. The transition toward smart and autonomous agriculture is particularly characterized by extensive integration of digital technologies and intelligent automation.

3. Overview of Drone Technology in Agriculture

The rapid advancement of digital technologies has significantly transformed agricultural practices, enabling farmers to monitor and manage crops with unprecedented accuracy and efficiency. Among these innovations, Unmanned Aerial Vehicles (UAVs), commonly known as drones, have emerged as one of the most influential tools in modern agriculture. Drones provide high-resolution aerial imagery, real-time monitoring capabilities, and precise field-level information that support data-driven decision-making in agricultural management (Zhang and Kovacs, 2012; Tsouros *et al.,* 2019).

Figure 3: Major components of an agricultural drone system

The integration of drone technology into agricultural systems has facilitated the transition from conventional farming to precision and smart agriculture. By enabling rapid assessment of crop health, soil conditions, irrigation performance, pest infestations, and nutrient deficiencies, drones

contribute significantly to resource optimization and sustainable agricultural production. Their ability to access large and difficult-to-reach areas within a short period makes them particularly valuable for modern farming operations.

As illustrated in Figure 3**,** agricultural drones consist of multiple interconnected hardware and software components that collectively enable autonomous flight, data acquisition, communication, and processing functions.

3.1 Definition and Components

A drone, or Unmanned Aerial Vehicle (UAV), is an aircraft capable of operating without an onboard human pilot. Agricultural drones are equipped with advanced sensors, imaging systems, navigation modules, communication technologies, and onboard processors that facilitate automated flight operations and data collection (Hunt and Daughtry, 2018). Depending on their design and application, drones may operate autonomously using pre-programmed flight paths or be remotely controlled by operators.

Agricultural drones play a critical role in precision farming by collecting high-resolution spatial and temporal data that support crop monitoring, field mapping, irrigation assessment, disease detection, and yield estimation. The effectiveness of drone operations depends largely on the functionality of their core components.

Table 3: Major Components of Agricultural Drones and Their Functions.

Component	Function	Agricultural Application
Flight Controller	Flight stabilization and navigation	Autonomous field surveys
GPS Module	Positioning and route planning	Precision mapping
Camera System	Image acquisition	Crop health monitoring
Communication Module	Data transmission	Real-time monitoring
Battery System	Power supply	Flight operations
Data Storage Unit	Data recording and storage	Image and sensor data management

4. Internet of Things in Agriculture

The Internet of Things (IoT) has emerged as a key enabling technology in modern agriculture, supporting the transition from conventional farming to smart and data-driven agricultural systems. IoT facilitates the continuous collection, transmission, and analysis of field-level information through interconnected devices and communication networks. By providing real-time insights into crop and environmental conditions, IoT helps farmers improve decision-making, optimize resource utilization, and enhance agricultural productivity (Ray, 2017; Verdouw *et al.,* 2016).

The adoption of IoT technologies in agriculture has significantly improved the efficiency of farm operations by enabling automated monitoring of soil, weather, crop health, and irrigation

systems. These capabilities contribute to sustainable agricultural development by reducing water consumption, minimizing input wastage, and supporting precision farming practices.

4.1 Concept of IoT

The Internet of Things (IoT) is a network of interconnected devices that collect, process, and transmit real-time data through communication networks. In agriculture, IoT systems use sensors, communication modules, cloud platforms, and user applications to monitor soil moisture, temperature, humidity, nutrients, and environmental conditions. This data supports informed decisions on irrigation, crop health, and disease management, enabling timely interventions, improved resource efficiency, and enhanced farm productivity.

5. Applications in Sustainable Agriculture

The integration of drones and IoT technologies have created new opportunities for sustainable agriculture. These technologies help farmers monitor crops, manage resources efficiently, detect problems early, and improve productivity. The major applications of drone-IoT systems in agriculture are discussed below.

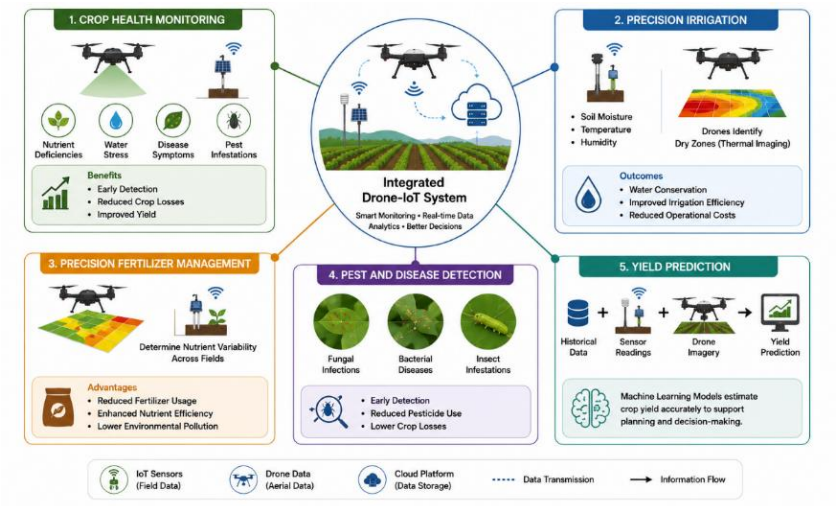


Figure 4: Major applications of integrated Drone-IoT systems in sustainable agriculture.

5.1 Crop Health Monitoring

Crop health monitoring is one of the most important applications of drones in agriculture. Drones equipped with multispectral and thermal cameras can capture detailed images of crop fields and identify variations in plant growth.

Drone images provide valuable information by identifying nutrient deficiencies, water stress, disease symptoms, and pest infestations at an early stage. Simultaneously, IoT sensors installed across the field continuously monitor soil moisture, temperature, humidity, and nutrient levels in real time. The integration of aerial imagery with ground-based sensor data enables farmers to accurately assess crop health, detect potential problems early, and make timely management decisions. This combined approach helps reduce crop losses, improve resource utilization, enhance crop yield, and support efficient and sustainable farm management.

5.2 Precision Irrigation

Water scarcity is one of the major challenges limiting agricultural productivity, making efficient water management essential for sustainable farming. IoT sensors continuously monitor soil moisture, temperature, and humidity to provide real-time information on field conditions, while drones equipped with thermal cameras identify water-stressed areas and dry zones across agricultural fields. By integrating sensor data with aerial thermal imagery, farmers can optimize irrigation schedules, apply water only where it is needed, and improve water distribution efficiency. This precision irrigation approach conserves water, reduces operational costs, enhances irrigation efficiency, and ultimately increases crop productivity.

6. Challenges and Limitations

Although the integration of drones and IoT technologies offer numerous benefits for sustainable agriculture, several challenges still limit their widespread adoption. These challenges can be categorized into technical, economic, regulatory, and social aspects.

6.1 Technical Challenges

The effective implementation of drone-IoT systems depends on reliable hardware, communication networks, and data management systems. However, several technical issues continue to affect system performance. Major technical challenges include:

- Limited battery life of drones
- Data interoperability issues between different devices and platforms
- Network connectivity problems in remote agricultural areas
- Large volumes of data requiring storage and processing
- Sensor accuracy and calibration issues

These challenges can affect real-time monitoring, data transmission, and decision-making capabilities.

6.2 Economic Challenges

The adoption of advanced technologies often requires significant financial investment, which may be difficult for small and medium-scale farmers. Major economic challenges include:

- High initial investment costs
- Expensive drone equipment and sensors
- Software and cloud service expenses
- Maintenance and repair costs
- Training and operational expenses

Although these technologies can provide long-term economic benefits, the initial cost remains a major barrier for many farmers.

7. Case Studies

Real-world applications show that integrating drones and IoT improves crop management, resource efficiency, and sustainable farming.

7.1 Smart Rice Farming

A smart rice farming project used IoT sensors to monitor soil moisture and environmental conditions, while drones captured aerial images to assess crop growth. The collected data enabled efficient irrigation scheduling and timely crop management.

Results

- 25% reduction in water consumption
- 18% increase in crop yield
- Improved irrigation efficiency
- Better crop monitoring and decision-making

This case demonstrates how drone-IoT integration enhances water conservation and agricultural productivity.

7.2 Vineyard Monitoring

In vineyards, multispectral drones were deployed to monitor grapevine health and detect diseases at an early stage. The aerial data helped farmers apply targeted treatments only where required.

Results

- Early disease detection
- Reduced pesticide usage
- Improved grape quality
- Lower production costs

This highlights the role of drone-based monitoring in sustainable crop protection and precision farming.

Conclusion

The integration of drones and the Internet of Things (IoT) has transformed modern agriculture by enabling real-time monitoring, precision farming, and data-driven decision-making. Through continuous collection and analysis of field data, these technologies help farmers monitor crop health, optimize irrigation and nutrient management, detect pests and diseases early, and improve yield prediction. As a result, drone-IoT systems increase agricultural productivity, reduce production costs, conserve water and other resources, and promote environmentally sustainable farming practices.

Despite challenges such as high implementation costs, limited drone battery life, connectivity issues, regulatory restrictions, and the need for greater digital literacy, continuous advancements in Artificial Intelligence (AI), Machine Learning (ML), cloud computing, blockchain, digital twins, and 5G communication are expected to overcome many of these limitations. These

innovations will make agricultural systems more intelligent, automated, and efficient, enabling farmers to respond effectively to changing environmental conditions.

As global food demand continues to rise, drone-IoT integration offers a promising solution for achieving sustainable and resilient agriculture. With continued research, supportive government policies, and farmer training programs, these technologies will play a vital role in improving food security, conserving natural resources, and shaping the future of smart farming worldwide.

References

1. Bongiovanni, R., & Lowenberg-DeBoer, J. (2004). Precision agriculture and sustainability. *Precision Agriculture, 5*(4), 359–387. https://doi.org/10.1023/B:PRAG.0000040806.39604.aa
2. Food and Agriculture Organization (FAO). (2023). *The state of food and agriculture 2023*. Food and Agriculture Organization of the United Nations, Rome, Italy.
3. Foley, J. A., Ramankutty, N., Brauman, K. A., Cassidy, E. S., Gerber, J. S., Johnston, M., Mueller, N. D., O'Connell, C., Ray, D. K., West, P. C., Balzer, C., Bennett, E. M., Carpenter, S. R., Hill, J., Monfreda, C., Polasky, S., Rockström, J., Sheehan, J., Siebert, S., & Zaks, D. P. M. (2011). Solutions for a cultivated planet. *Nature, 478*(7369), 337–342. https://doi.org/10.1038/nature10452
4. Gebbers, R., & Adamchuk, V. I. (2010). Precision agriculture and food security. *Science, 327*(5967), 828–831. https://doi.org/10.1126/science.1183899
5. Hunt, E. R., Jr., & Daughtry, C. S. T. (2018). What good are unmanned aircraft systems for agricultural remote sensing and precision agriculture? *International Journal of Remote Sensing, 39*(15–16), 5345–5376. https://doi.org/10.1080/01431161.2017.1410300
6. Javaid, M., Haleem, A., Singh, R. P., Suman, R., Khan, I. H., & Rab, S. (2022). Significance of machine learning in healthcare, agriculture and smart cities: A review. *Journal of Industrial Integration and Management, 7*(2), 195–222.
7. Klerkx, L., Jakku, E., & Labarthe, P. (2019). A review of social science on digital agriculture, smart farming and agriculture 4.0. *NJAS: Wageningen Journal of Life Sciences, 90–91*, 100315. https://doi.org/10.1016/j.njas.2019.100315
8. Lal, R. (2020). Regenerative agriculture for food and climate. *Journal of Soil and Water Conservation, 75*(5), 123A–124A. https://doi.org/10.2489/jswc.2020.0620A
9. Liakos, K. G., Busato, P., Moshou, D., Pearson, S., & Bochtis, D. (2018). Machine learning in agriculture: A review. *Sensors, 18*(8), 2674. https://doi.org/10.3390/s18082674
10. McBratney, A., Whelan, B., Ancev, T., & Bouma, J. (2005). Future directions of precision agriculture. *Precision Agriculture, 6*(1), 7–23. https://doi.org/10.1007/s11119-005-0681-8

11. Pingali, P. L. (2012). Green revolution: Impacts, limits, and the path ahead. *Proceedings of the National Academy of Sciences, 109*(31), 12302–12308. https://doi.org/10.1073/pnas.0912953109
12. Pretty, J. (2008). Agricultural sustainability: Concepts, principles and evidence. *Philosophical Transactions of the Royal Society B: Biological Sciences, 363*(1491), 447–465. https://doi.org/10.1098/rstb.2007.2163
13. Ray, P. P. (2017). Internet of Things for smart agriculture: Technologies, practices and future direction. *Journal of Ambient Intelligence and Smart Environments, 9*(4), 395–420. https://doi.org/10.3233/AIS-170440
14. Shamshiri, R. R., Kalantari, F., Ting, K. C., Thorp, K. R., Hameed, I. A., Weltzien, C., Ahmad, D., & Shad, Z. M. (2018). Advances in greenhouse automation and controlled environment agriculture: A transition to plant factories and urban agriculture. *International Journal of Agricultural and Biological Engineering, 11*(1), 1–22.
15. Talaviya, T., Shah, D., Patel, N., Yagnik, H., & Shah, M. (2020). Implementation of artificial intelligence in agriculture for optimisation of irrigation and application of pesticides and herbicides. *Artificial Intelligence in Agriculture, 4*, 58–73. https://doi.org/10.1016/j.aiia.2020.04.002
16. Tilman, D., Balzer, C., Hill, J., & Befort, B. L. (2011). Global food demand and the sustainable intensification of agriculture. *Proceedings of the National Academy of Sciences, 108*(50), 20260–20264. https://doi.org/10.1073/pnas.1116437108
17. Tsouros, D. C., Bibi, S., & Sarigiannidis, P. G. (2019). A review on UAV-based applications for precision agriculture. *Information, 10*(11), 349. https://doi.org/10.3390/info10110349
18. United Nations. (2022). *World population prospects 2022: Summary of results*. United Nations Department of Economic and Social Affairs, Population Division.
19. Verdouw, C. N., Wolfert, J., Beulens, A. J. M., & Rialland, A. (2016). Virtualization of food supply chains with the Internet of Things. *Journal of Food Engineering, 176*, 128–136. https://doi.org/10.1016/j.jfoodeng.2015.11.009
20. Wolfert, S., Ge, L., Verdouw, C., & Bogaardt, M. J. (2017). Big data in smart farming: A review. *Agricultural Systems, 153*, 69–80. https://doi.org/10.1016/j.agsy.2017.01.023
21. Zhang, C., & Kovacs, J. M. (2012). The application of small unmanned aerial systems for precision agriculture: A review. *Precision Agriculture, 13*(6), 693–712. https://doi.org/10.1007/s11119-012-9274-5
22. Zhang, N., Wang, M., & Wang, N. (2002). Precision agriculture—A worldwide overview. *Computers and Electronics in Agriculture, 36*(2–3), 113–132. https://doi.org/10.1016/S0168-1699(02)00096-0

ARTIFICIAL INTELLIGENCE AND MARKOV CHAIN MODELS FOR INTELLIGENT HEALTHCARE ANALYTICS: A CONCEPTUAL FRAMEWORK

Manikkaraj Iyswariya and Raju Arumugam
Department of Mathematics,
Periyar Maniammai Institute of Science and Technology,
Thanjavur 613403, Tamil Nadu, India
Corresponding author E-mail: iyswariyamanikkaraj@gmail.com, arumugamr@pmu.edu

Abstract

Artificial Intelligence (AI) has emerged as a transformative technology in modern healthcare by enhancing disease prediction, clinical decision-making, and patient management through data-driven approaches. However, healthcare systems are characterized by uncertainty, dynamic patient conditions, and complex progression patterns that require robust analytical models. Markov chain models provide an effective probabilistic framework for representing state transitions over time, while AI techniques enable efficient extraction of meaningful patterns from healthcare data. This chapter presents a conceptual framework that integrates Artificial Intelligence and Markov chain models to support intelligent healthcare analytics. Rather than focusing on a specific disease or experimental dataset, the proposed framework provides a generalized methodology for combining healthcare data processing, machine learning, probabilistic state-transition modeling, and clinical decision support. The chapter also discusses the potential applications of the integrated framework in disease progression analysis, patient monitoring, healthcare resource planning, and predictive analytics, along with its advantages, challenges, and future research directions. The proposed framework offers a flexible and interpretable foundation for developing intelligent, transparent, and data-driven healthcare systems capable of supporting informed clinical and healthcare management decisions.

Keywords: Artificial Intelligence, Markov Chain, Healthcare Analytics, Machine Learning, Predictive Modeling, Clinical Decision Support.

1. Introduction

Artificial Intelligence (AI) has emerged as one of the most transformative technologies in modern healthcare, offering innovative solutions for disease diagnosis, patient monitoring, clinical decision-making, and healthcare management. The rapid growth of electronic health records, wearable devices, medical imaging technologies, and digital health platforms has generated enormous amounts of healthcare data that require intelligent computational methods for effective analysis. Topol (2019) emphasized that AI has the potential to improve healthcare quality by enhancing diagnostic accuracy, supporting clinical decisions, and enabling personalized patient

care. The practical implementation of AI technologies has also accelerated the development of intelligent healthcare systems capable of improving both clinical efficiency and patient outcomes. He *et al.* (2019) discussed that AI-based systems can support clinicians by providing reliable and timely medical information for better healthcare delivery.

Recent advances in machine learning and deep learning have significantly expanded the scope of AI applications in healthcare. These techniques have demonstrated excellent performance in disease prediction, medical image analysis, patient risk assessment, and clinical decision support. Esteva *et al.* (2019) highlighted that deep learning algorithms have achieved remarkable success in solving complex healthcare problems by learning meaningful representations from large-scale medical data. Similarly, Rajpurkar *et al.* (2022) reported that AI has the potential to transform healthcare through improved predictive models, human–AI collaboration, and efficient clinical workflows. As healthcare data continue to increase in volume and complexity, AI-driven analytics has become an important component of modern healthcare research and practice. Despite these advancements, healthcare systems remain highly dynamic because patient conditions continuously change over time. Disease progression is influenced by biological variability, treatment response, environmental factors, and individual patient characteristics, making healthcare prediction a challenging task. Although AI models provide excellent predictive performance, many approaches function as black-box systems and often do not explicitly represent temporal transitions between different health states. Kelly *et al.* (2019) emphasized that trustworthy AI should combine predictive accuracy with transparency to support real-world clinical decision-making. Likewise, Buccella *et al.* (2024) highlighted the importance of explainable AI for improving the interpretability and reliability of healthcare applications.

Probabilistic modeling techniques have therefore become increasingly important for representing uncertainty in healthcare systems. Among these approaches, Markov chain models provide a systematic framework for describing transitions between discrete health states over time. These models have been widely used in disease progression analysis, clinical decision-making, and health economic evaluation because of their simplicity and interpretability. Sonnenberg and Beck (1993) demonstrated the usefulness of Markov models in medical decision analysis, while Briggs *et al.* (2006) further explained their importance in evaluating long-term healthcare outcomes. The theoretical foundation of state-transition modeling was comprehensively described by Puterman (1994), making Markov models one of the most widely adopted stochastic approaches in healthcare research. The integration of Artificial Intelligence with Markov chain models has recently attracted considerable attention because it combines the predictive capability of AI with the transparency of probabilistic state-transition modeling. Acosta *et al.* (2022) reported that multimodal AI systems integrating diverse healthcare information can improve predictive performance and support personalized medicine. Motivated by these developments, this chapter

presents a conceptual framework integrating Artificial Intelligence and Markov chain models for intelligent healthcare analytics. Rather than focusing on a specific disease or experimental dataset, the chapter proposes a generalized framework applicable to various healthcare scenarios and discusses its applications, advantages, challenges, and future research directions.

2. Literature Review

The rapid advancement of Artificial Intelligence (AI) has significantly transformed healthcare research by enabling intelligent analysis of large and complex medical datasets. Over the past decade, AI has evolved from rule-based expert systems to sophisticated machine learning and deep learning models capable of supporting clinical diagnosis, disease prediction, and healthcare decision-making. The widespread availability of electronic health records, wearable devices, medical imaging, and genomic information has further accelerated the development of AI-driven healthcare applications. He *et al.* (2019) explained that AI technologies have become increasingly important in assisting healthcare professionals by improving diagnostic accuracy, optimizing treatment planning, and enhancing patient care. Similarly, Topol (2019) emphasized that the integration of human expertise with AI technologies has the potential to improve healthcare quality while supporting more personalized and efficient medical services. Recent studies have demonstrated that machine learning and deep learning techniques can effectively identify hidden patterns within healthcare data and generate reliable predictive models for various clinical applications. AI algorithms have been successfully applied to disease diagnosis, medical image interpretation, patient risk prediction, drug discovery, and clinical decision support systems. Esteva *et al.* (2019) described deep learning as one of the most influential technologies for extracting meaningful information from complex medical datasets without extensive manual feature engineering. Likewise, Rajpurkar *et al.* (2022) discussed the expanding role of AI in health and medicine, highlighting its ability to improve clinical workflows and facilitate collaborative decision-making between clinicians and intelligent systems. In addition, Beam and Kohane (2018) observed that the combination of big data and machine learning has created new opportunities for precision medicine by enabling more accurate patient-specific predictions and data-driven healthcare management.

Although AI has demonstrated remarkable predictive capabilities, its practical implementation in healthcare requires transparency, interpretability, and clinical reliability. Many advanced machine learning models function as complex black-box systems, making it difficult for healthcare professionals to understand how predictions are generated. Consequently, explainable Artificial Intelligence (XAI) has emerged as an important research area that focuses on improving model transparency and user trust. Kelly *et al.* (2019) emphasized that successful clinical adoption of AI depends not only on predictive accuracy but also on model validation, safety, and integration into existing healthcare workflows. More recently, Buccella *et al.* (2024) argued that different

healthcare users require different forms of explanation depending on their clinical responsibilities and decision-making needs, indicating that explainability should be considered an essential component of intelligent healthcare systems rather than an additional feature. Alongside the rapid growth of AI, stochastic modeling approaches have continued to play an essential role in healthcare analytics, particularly for representing disease progression over time. Among these approaches, Markov chain models have become widely recognized for their ability to describe transitions between discrete health states using probabilistic state-transition mechanisms. Recent review studies have further demonstrated the growing application of Markov chain models in healthcare, particularly for epidemic modeling, disease progression analysis, and predictive healthcare analytics (Iyswariya & Arumugam, 2026). These models have been extensively employed in medical decision-making, disease progression analysis, and health economic evaluation because they provide interpretable representations of dynamic healthcare processes. Sonnenberg and Beck (1993) introduced Markov models as an effective framework for evaluating long-term clinical decisions involving uncertain disease progression. Similarly, Briggs *et al.* (2006) demonstrated their usefulness in health economic evaluation by representing sequential healthcare events and estimating long-term clinical outcomes. The mathematical principles governing these stochastic processes were comprehensively established by Puterman (1994), whose work continues to serve as a fundamental reference for Markov-based decision modeling.

As healthcare systems become increasingly data-intensive, researchers have begun integrating AI with probabilistic modeling techniques to overcome the limitations of individual methodologies. AI provides powerful predictive capabilities by learning complex relationships from heterogeneous healthcare data, whereas Markov chain models offer interpretable descriptions of temporal state transitions under uncertainty. The integration of these complementary approaches enables more comprehensive healthcare analytics by combining prediction, uncertainty modeling, and sequential decision-making within a unified framework. Acosta *et al.* (2022) highlighted that multimodal biomedical AI can integrate diverse sources of healthcare information to improve clinical prediction and personalized medicine. Furthermore, Yu *et al.* (2018) emphasized that combining AI with statistical and probabilistic modeling approaches represents an important direction for future biomedical research. Despite these advances, the existing literature primarily focuses on disease-specific implementations, while relatively few studies present a generalized conceptual framework describing how AI and Markov chain models can be systematically integrated for intelligent healthcare analytics. Therefore, this chapter proposes a comprehensive conceptual framework that addresses this gap and provides a foundation for future research and practical healthcare applications.

3. Artificial Intelligence in Healthcare Analytics

3.1 Artificial Intelligence in Healthcare

Artificial Intelligence (AI) refers to computational systems that can perform tasks requiring human intelligence, including learning, reasoning, pattern recognition, and decision-making. In healthcare, AI has emerged as a transformative technology for analyzing complex medical data and supporting clinicians in diagnosis, prognosis, treatment planning, and patient management. The rapid growth of electronic health records, wearable devices, medical imaging technologies, and genomic databases has accelerated the adoption of AI-based healthcare systems. According to He *et al.* (2019), AI technologies improve healthcare delivery by assisting clinicians in making faster and more accurate clinical decisions while reducing the burden of routine healthcare tasks. Likewise, Topol (2019) emphasized that AI should complement healthcare professionals by enhancing clinical expertise rather than replacing human judgment. The integration of AI into digital healthcare has therefore contributed to improved efficiency, personalized patient care, and evidence-based medical practice.

3.2 Artificial Intelligence Techniques

Several AI techniques have been successfully applied in healthcare analytics depending on the nature of clinical data and research objectives. Feature selection is an important preprocessing step that improves prediction performance by identifying the most informative variables while reducing data complexity (Arumugam, 2026). Deep learning extends these capabilities by automatically extracting complex features from high-dimensional datasets such as medical images and physiological signals. Esteva *et al.* (2019) demonstrated that deep learning algorithms achieve high diagnostic performance in various medical imaging applications, while Beam and Kohane (2018) highlighted the importance of machine learning in extracting clinically relevant knowledge from large healthcare databases. Natural Language Processing (NLP) enables automated analysis of unstructured clinical documents, physician notes, and biomedical literature, whereas computer vision techniques support image interpretation in radiology, pathology, and ophthalmology. More recently, explainable AI has received increasing attention because healthcare professionals require transparent and interpretable predictions before incorporating AI into clinical practice. Buccella *et al.* (2024) emphasized that explainability should be designed according to the needs of different healthcare users, thereby improving trust and facilitating responsible AI adoption.

3.3 Applications of AI in Healthcare

Artificial Intelligence has been applied across numerous healthcare domains, improving both clinical decision-making and operational efficiency. AI-assisted diagnostic systems support the early detection of diseases by analyzing laboratory findings, imaging data, and patient histories. Predictive models estimate disease risk and patient outcomes, enabling timely intervention and

preventive healthcare strategies. AI has also demonstrated remarkable success in medical imaging by assisting clinicians in detecting abnormalities with high accuracy. Furthermore, Clinical Decision Support Systems (CDSS) integrate patient information with predictive algorithms to recommend evidence-based treatment options and optimize clinical workflows. Personalized medicine has benefited substantially from AI by enabling treatment recommendations tailored to individual patient characteristics. In addition, wearable sensors and remote monitoring technologies continuously collect patient data, allowing healthcare providers to monitor chronic diseases and respond promptly to clinical changes. Rajpurkar *et al.* (2022) and Acosta *et al.* (2022) reported that integrating multiple healthcare data sources through AI enhances predictive performance and supports personalized healthcare delivery.

3.4 Challenges and Limitations

Despite its remarkable achievements, several challenges continue to influence the successful implementation of AI in healthcare. The performance of AI models depends heavily on the quality, completeness, and diversity of healthcare data. Missing information, imbalanced datasets, and data heterogeneity may reduce predictive accuracy and limit model generalizability. Ethical concerns, including patient privacy, data security, algorithmic bias, and regulatory compliance, remain significant barriers to large-scale implementation. Another major limitation is the limited interpretability of many advanced AI models, particularly deep learning systems that operate as black-box algorithms. Kelly *et al.* (2019) emphasized that trustworthy AI requires rigorous validation, transparency, and continuous clinical evaluation before deployment in healthcare environments. These limitations indicate that although AI provides powerful predictive capabilities, complementary probabilistic approaches are often required to model temporal disease progression and uncertainty. Consequently, integrating AI with Markov chain models offers a promising direction for developing intelligent, interpretable, and reliable healthcare analytics.

4. Markov Chain Models in Healthcare

4.1 Introduction to Markov Chain Models

Markov chain models are mathematical frameworks used to describe systems that evolve through a sequence of discrete states over time. These models are based on the Markov property, which assumes that the future state of a system depends only on its current state and not on the sequence of previous states. This characteristic makes Markov chains particularly suitable for modeling dynamic healthcare processes, where patients transition between different health conditions during the progression of a disease. Because many chronic diseases develop gradually through multiple clinical stages, Markov models provide a systematic approach for representing disease evolution and estimating future health outcomes.

Healthcare researchers have widely adopted Markov chain models for disease progression analysis, medical decision-making, survival analysis, and health economic evaluation. Sonnenberg and Beck (1993) demonstrated that Markov models effectively represent long-term disease progression under uncertainty and support clinical decision-making. Similarly, Briggs *et al.* (2006) highlighted their importance in evaluating healthcare interventions by estimating long-term costs and health outcomes. The theoretical foundation of Markov decision processes was comprehensively established by Puterman (1994), making Markov modeling one of the most widely applied stochastic approaches in healthcare research.

4.2 Components of Markov Chain Models

A Markov chain model consists of several components that describe how a patient moves between different health states over time. These components include the state space, transition probabilities, cycle length, and initial state distribution. Together, they provide a systematic framework for representing disease progression and estimating future health outcomes (Puterman, 1994; Putter *et al.,* 2006). Figure 1 illustrates the general state-transition process of a Markov chain model used in healthcare. Patients move sequentially from one health state to another according to predefined transition probabilities. Depending on the disease and clinical outcomes, transitions may occur between healthy, at-risk, disease, complication, recovery, or death states.

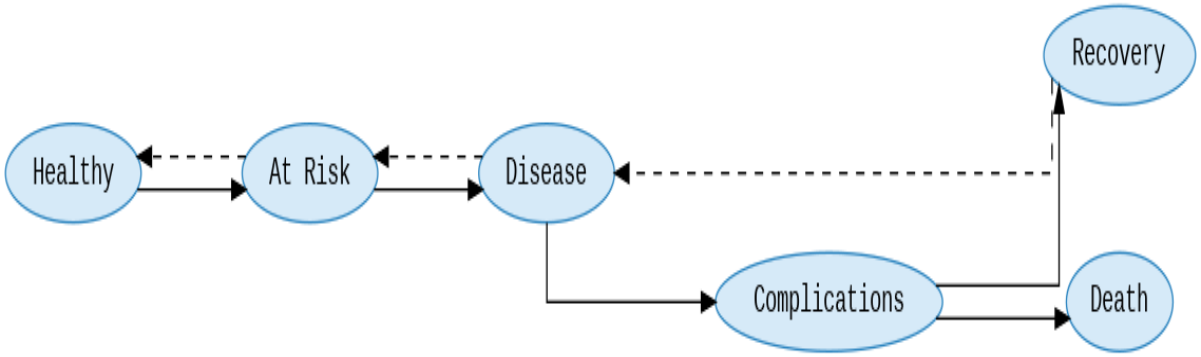


Figure 1: General State Transition Diagram of a Markov Chain Model for Healthcare

As shown in Figure 1, each health state represents a possible clinical condition of a patient, while the connecting arrows indicate the possible transitions between states during each time interval. The transition probability matrix determines the likelihood of moving from one state to another, and the sum of transition probabilities from a given state equals one. This state-transition mechanism enables researchers to model disease progression, evaluate treatment strategies, and estimate long-term clinical outcomes under uncertainty.

4.3 Applications of Markov Chain Models in Healthcare

Markov chain models have been widely applied in healthcare because they effectively represent disease progression through multiple health states over time. These models assist clinicians and

researchers in evaluating treatment outcomes, predicting disease progression, estimating survival probabilities, and supporting health economic evaluations. Their ability to model uncertainty and sequential transitions makes them suitable for analyzing a wide range of chronic and acute diseases (Sonnenberg & Beck, 1993; Briggs *et al.,* 2006).

Table 1: Selected Healthcare Applications of Markov Chain Models

Healthcare Area	Typical Markov States	Primary Application
Diabetes	Normal → Prediabetes → Diabetes → Complications	Disease progression prediction
Cancer	Diagnosis → Treatment → Remission → Recurrence	Survival and treatment evaluation
Cardiovascular Disease	Healthy → At Risk → Disease → Death	Risk assessment and prognosis
Alzheimer's Disease	Mild → Moderate → Severe	Cognitive decline modeling
Infectious Diseases	Susceptible → Infected → Recovered	Epidemic progression analysis

Table 1 presents several common healthcare applications of Markov chain models across different medical domains. Although the health states differ according to the disease, the underlying modeling principle remains the same, where patients transition between predefined clinical states based on estimated transition probabilities. This flexibility allows Markov models to support disease progression analysis, healthcare planning, treatment evaluation, and long-term decision-making under uncertainty (Andersen & Keiding, 2002; Putter *et al.,* 2006).

4.4 Integration with Artificial Intelligence

Markov chain models effectively represent disease progression through probabilistic state transitions, whereas Artificial Intelligence (AI) provides powerful predictive capabilities by learning complex patterns from healthcare data. Integrating these approaches combines the strengths of both methods, allowing AI to identify important clinical features while Markov models describe the temporal evolution of disease states. Such hybrid models support improved disease prediction, personalized treatment planning, and clinical decision-making. Acosta *et al.* (2022) highlighted that multimodal AI enhances healthcare prediction by integrating diverse clinical information, while Yu *et al.* (2018) emphasized the importance of combining AI with probabilistic models to address uncertainty in healthcare analytics. Recent review studies have also highlighted the increasing role of stochastic and Markov chain models in healthcare prediction and epidemic modeling, further supporting the integration of AI with probabilistic modeling approaches (Iyswariya & Arumugam, 2026).

5. Proposed Conceptual Framework

5.1 Framework Overview

The rapid growth of healthcare data has created the need for intelligent computational frameworks capable of supporting accurate disease prediction and effective clinical decision-making. Although Artificial Intelligence (AI) provides excellent predictive performance by learning complex patterns from healthcare data, many AI models have limited capability in representing temporal disease progression and uncertainty. Conversely, Markov chain models effectively describe transitions between different health states but depend on predefined transition probabilities. Integrating these two approaches combines the predictive strength of AI with the probabilistic reasoning of Markov models, providing a more comprehensive framework for intelligent healthcare analytics (Rajpurkar *et al.,* 2022; Puterman, 1994). The proposed conceptual framework consists of multiple interconnected stages, beginning with healthcare data acquisition and ending with personalized clinical decision support. AI techniques are employed to preprocess healthcare data, identify significant clinical features, and generate predictive insights. These outputs are subsequently incorporated into a Markov chain model to estimate disease-state transitions and predict future health conditions. This integrated framework provides a systematic approach for improving disease progression analysis while maintaining model interpretability.

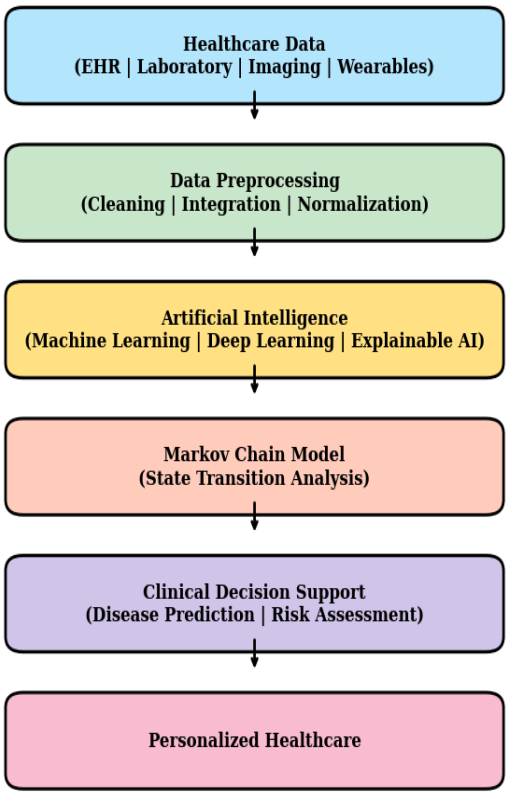


Figure 2. Proposed AI-Markov Conceptual Framework for Intelligent Healthcare Analytics

Figure 2 illustrates the proposed conceptual framework integrating Artificial Intelligence with Markov chain models for intelligent healthcare analytics. The framework demonstrates how

healthcare data are transformed into meaningful clinical decisions through sequential data processing, AI-based prediction, probabilistic state-transition modeling, and clinical decision support. It provides a structured approach for improving disease progression prediction and personalized patient management.

5.2 Components of the Proposed Framework

The proposed framework consists of several interconnected components, each contributing to intelligent healthcare analytics. Initially, healthcare data are collected from multiple clinical sources, including electronic health records, laboratory investigations, medical imaging, wearable devices, and other digital health systems. Since healthcare datasets frequently contain missing values and inconsistencies, preprocessing techniques such as data cleaning, normalization, and integration are performed to improve data quality (He *et al.,* 2019).

After preprocessing, Artificial Intelligence techniques, including machine learning, deep learning, and explainable AI, analyze the processed data to identify important clinical patterns associated with disease progression. Feature extraction and selection techniques reduce data complexity by identifying the most informative variables for prediction (Esteva *et al.,* 2019; Buccella *et al.,* 2024). These selected features are then incorporated into a Markov chain model, where transition probabilities describe the movement of patients between different health states over time. Finally, the predicted disease trajectory supports clinical decision support systems by assisting healthcare professionals in selecting appropriate treatment strategies and delivering personalized patient care.

Table 2: Components of the Proposed AI–Markov Framework

Component	Function
Healthcare Data	Collection of clinical and patient information
Data Preprocessing	Data cleaning, normalization, and integration
Artificial Intelligence	Pattern recognition and predictive analysis
Feature Extraction	Selection of important clinical variables
Markov Chain Model	Estimation of disease-state transitions
Disease Prediction	Forecasting future health conditions
Clinical Decision Support	Supporting evidence-based medical decisions
Personalized Healthcare	Individualized treatment planning

Table 2 summarizes the major components of the proposed framework and their respective functions. Together, these components transform raw healthcare data into clinically meaningful information, enabling intelligent disease prediction and effective healthcare decision-making.

5.3 Framework Workflow

The proposed framework begins with the acquisition of healthcare information from multiple clinical sources. Following preprocessing, AI algorithms analyze the data to identify significant

patterns and extract clinically relevant features. These features are subsequently integrated into a Markov chain model, where transition probabilities are estimated to describe disease progression through multiple health states. The predicted disease trajectory is then utilized by clinical decision support systems to assist healthcare professionals in diagnosis, treatment planning, and patient monitoring.

The integration of AI and Markov chain models provides several advantages over using either approach independently. AI enhances predictive accuracy by learning complex relationships within healthcare data, whereas Markov models improve interpretability by explicitly representing temporal disease transitions. Consequently, the proposed framework supports reliable disease prediction, personalized treatment recommendations, and evidence-based healthcare management. This conceptual framework can be adapted to various healthcare applications, including chronic disease management, cancer prognosis, cardiovascular risk assessment, and infectious disease monitoring, making it a flexible foundation for future intelligent healthcare systems (Acosta *et al.,* 2022; Kelly *et al.,* 2019).

6. Challenges and Future Research Directions

The integration of Artificial Intelligence (AI) and Markov chain models offers significant opportunities for intelligent healthcare analytics; however, several challenges remain. Healthcare data are often incomplete, inconsistent, and collected from multiple sources, which may affect prediction accuracy and disease progression modeling. In addition, balancing predictive performance with model interpretability is essential to ensure that healthcare professionals can understand and trust AI-assisted clinical decisions. Data privacy, security, and ethical considerations also remain important for the responsible use of intelligent healthcare systems.

Future research should focus on developing more accurate, transparent, and patient-centered computational frameworks by combining AI with probabilistic modeling techniques. The integration of explainable AI, real-time healthcare data, wearable technologies, and personalized health records can further enhance disease prediction and clinical decision support. Emerging technologies such as federated learning and digital twins also provide new opportunities for building secure, scalable, and adaptive healthcare systems. Addressing these challenges will strengthen the practical implementation of AI–Markov frameworks and contribute to more reliable, efficient, and personalized healthcare analytics.

Conclusion

Artificial Intelligence and Markov chain models have become important computational approaches for advancing intelligent healthcare analytics. Artificial Intelligence enables efficient analysis of large and complex healthcare data, while Markov chain models provide a probabilistic framework for representing disease progression over time. This chapter discussed the fundamental concepts of both approaches and presented a conceptual framework that integrates

AI-based prediction with Markov chain modeling to support disease progression analysis, clinical decision-making, and personalized healthcare. The proposed framework demonstrates how these complementary techniques can be combined to improve healthcare analytics across various clinical applications. Although the integration of AI and Markov chain models offers significant potential, challenges related to data quality, model interpretability, privacy, and practical implementation remain. Addressing these challenges through future research will contribute to the development of more reliable, transparent, and patient-centered healthcare systems. Overall, the proposed conceptual framework provides a useful foundation for future studies and highlights the potential of integrating Artificial Intelligence and probabilistic modeling to support intelligent healthcare analytics and improve clinical decision-making.

References

1. Acosta, J. N., Falcone, G. J., Rajpurkar, P., & Topol, E. J. (2022). Multimodal biomedical AI. *Nature Medicine, 28*(9), 1773–1784. https://doi.org/10.1038/s41591-022-01981-2
2. Briggs, A., Claxton, K., & Sculpher, M. (2006). *Decision modelling for health economic evaluation*. https://doi.org/10.1093/oso/9780198526629.001.0001
3. Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., Cui, C., Corrado, G., Thrun, S., & Dean, J. (2019). A guide to deep learning in healthcare. *Nature Medicine, 25*(1), 24–29. https://doi.org/10.1038/s41591-018-0316-z
4. He, J., Baxter, S. L., Xu, J., Xu, J., Zhou, X., & Zhang, K. (2019). The practical implementation of artificial intelligence technologies in medicine. *Nature Medicine, 25*(1), 30–36. https://doi.org/10.1038/s41591-018-0307-0
5. Kelly, C. J., Karthikesalingam, A., Suleyman, M., Corrado, G., & King, D. (2019). Key challenges for delivering clinical impact with artificial intelligence. *BMC Medicine, 17*(1). https://doi.org/10.1186/s12916-019-1426-2
6. Puterman, M. L. (1994). *Markov decision processes*. Wiley Series in Probability and Statistics. https://doi.org/10.1002/9780470316887
7. Rajpurkar, P., Chen, E., Banerjee, O., & Topol, E. J. (2022). AI in health and medicine. *Nature Medicine, 28*(1), 31–38. https://doi.org/10.1038/s41591-021-01614-0
8. Sonnenberg, F. A., & Beck, J. R. (1993). Markov models in medical decision making. *Medical Decision Making, 13*(4), 322–338. https://doi.org/10.1177/0272989X9301300409
9. Topol, E. J. (2019). High-performance medicine: The convergence of human and artificial intelligence. *Nature Medicine, 25*(1), 44–56. https://doi.org/10.1038/s41591-018-0300-7
10. Buccella, A., Linstead, E., & Maoz, U. (2024). *What kind of explainable AI do users need?* https://doi.org/10.31234/osf.io/cgk4m

11. Andersen, P. K., & Keiding, N. (2002). Multi-state models for event history analysis. *Statistical Methods in Medical Research, 11*(2), 91–115. https://doi.org/10.1191/0962280202sm276ra
12. Beam, A. L., & Kohane, I. S. (2018). Big data and machine learning in health care. *JAMA, 319*(13), 1317. https://doi.org/10.1001/jama.2017.18391
13. Putter, H., Fiocco, M., & Geskus, R. B. (2006). Tutorial in biostatistics: Competing risks and multi-state models. *Statistics in Medicine, 26*(11), 2389–2430. https://doi.org/10.1002/sim.2712
14. Yu, K.-H., Beam, A. L., & Kohane, I. S. (2018). Artificial intelligence in healthcare. *Nature Biomedical Engineering, 2*(10), 719–731. https://doi.org/10.1038/s41551-018-0305-z
15. Iyswariya, M., & Arumugam, R. (2026). Stochastic and Markov chain–based epidemic models for COVID-19: A comprehensive review of predictive, probabilistic, and control strategies. *Mathematical Methods in the Applied Sciences, 49*(11), 12392–12412. https://doi.org/10.1002/mma.70694
16. Arumugam, R. (2026). Leveraging correlation analysis for effective feature selection in AI model development. In *AI trust, risk, and security management* (pp. 45–68). Wiley. https://doi.org/10.1002/9781394393022.ch3

Bhumi Publishing, India
July 2026

BLOCKCHAIN-ENABLED SECURE FOOD SAFETY AND TRACEABILITY SYSTEM

V Shoba[1] and D. Bhuvaneswari[2]
Department of Computer Applications and Technology,
Department of Computer Science, SRM Arts and Science College,
Kattankulathur – 603 203
Corresponding author E-mail: shobacs@srmasc.ac.in, bhuvaneswarics@srmasc.ac.in

Abstract
Food safety incidents often expose the weaknesses of current traceability practices, which depend on fragmented records and delayed reporting systems. This study introduces a blockchain-supported framework that ensures end-to-end monitoring of food products from origin to consumption. By combining Internet of Things (IoT) sensors with distributed ledger technology, each stage of the supply chain is automatically documented in a tamper-resistant and auditable manner. Smart contracts further enhance compliance by detecting irregularities and initiating automated responses, such as alerts or recalls. A prototype simulation, developed using Hyperledger Fabric and IoT emulation tools, was tested for multiple food categories under varying conditions. The findings demonstrate notable improvements in transparency, recall efficiency, and resistance to manipulation compared with conventional centralized systems. This study highlights the potential of blockchain technology to build trust among regulators, producers, retailers, and consumers by enabling rapid, verifiable, and trustworthy food traceability.

Keywords: Blockchain, Consensus Algorithm, Traceability Algorithm, Food Safety, Supply Chain Management, Smart Contracts, Internet of Things (IoT), Data Integrity, Transparency, Product Recall.

1. Introduction

Food safety is not only a public health concern but also an economic and regulatory priority for governments and industries worldwide. When contamination occurs, the ability to trace products rapidly and accurately can determine the scale of recalls, the cost to businesses, and the risk to consumers. Traditional traceability systems often rely on centralized databases, manual entries, and inconsistent reporting by stakeholders. These methods are slow, error-prone, and vulnerable to tampering, which weakens the trust in the food supply chain.

According to the World Health Organization, foodborne illnesses affect hundreds of millions of people annually, underscoring the urgency of developing stronger monitoring mechanisms. In addition to health impacts, incidents such as food fraud, mislabeling, and adulteration cause

significant financial and reputational harm. To address these challenges, researchers and industry actors are increasingly turning to emerging technologies that can strengthen their transparency and accountability.

Blockchain technology offers a decentralized and immutable record-keeping system in which every transaction is validated and time-stamped by network consensus. When integrated with IoT devices, blockchain can capture real-time data on temperature, storage conditions, and transport routes, ensuring that the information cannot be altered retroactively. This combination creates a secure and reliable foundation for end-to-end food-traceability. Beyond technical innovation, blockchain-based systems have the potential to increase consumer confidence by allowing shoppers and regulators to independently verify product histories.

This study presents a blockchain-enabled architecture for food safety and traceability, evaluates its performance under different operational scenarios, and compares it with traditional approaches. The goal is to demonstrate how distributed ledgers, smart contracts, and IoT integration can collectively reduce the time and complexity of recalls while building a trustworthy and transparent food supply ecosystem.

In addition to using blockchain as a secure ledger, traceability algorithms play a crucial role in enabling both forward and backward tracking within food supply chains. Backward tracking helps quickly pinpoint the origin of contamination, for example, a particular farm or processing unit, whereas forward tracking identifies all subsequent products and consumers potentially exposed to the affected batch. Together, these mechanisms form the foundation of timely recall operations, lowering health risks, and reducing financial impact. When combined with IoT-based sensing technologies that continuously capture parameters such as temperature, humidity, and handling conditions, the system gains an additional layer of intelligence, ensuring that food safety and quality are consistently verified at every stage.

Although research in this area has advanced considerably, several limitations remain. Many blockchain prototypes struggle to scale across large global supply chains, whereas energy-intensive consensus protocols restrict the practicality of certain public blockchain deployments. Data privacy also presents a challenge: sensitive business information must be protected without undermining the transparency required by traceability. Furthermore, most existing efforts address only individual components, such as blockchain storage, IoT monitoring, or contract automation, rather than offering a fully integrated framework that combines consensus mechanisms, traceability algorithms, and smart contracts into a single cohesive solution. These gaps highlight the need for a comprehensive system that balances security, scalability, transparency, and usability for diverse stakeholders.

This study addresses these challenges by introducing a Blockchain-Enabled Secure Food Safety and Traceability Framework. The proposed system applies consensus algorithms to guarantee

tamper-proof data recording, employs traceability methods for efficient forward and backward tracking, and integrates IoT streams for real-time monitoring. Smart contracts are designed to automate compliance verification and enforce the regulatory standards. Experimental results show that the approach strengthens data reliability, improves accountability, and accelerates recall processes, substantially reducing the response time to contamination.

The main contributions of this study are summarized as follows:

- Architecture Design: Development of a blockchain-based structure that unifies consensus protocols with traceability algorithms for food safety applications.
- Automation and Monitoring: Deployment of smart contracts and IoT feeds to streamline compliance checks and support continuous monitoring.
- System Evaluation: Assessment of recall efficiency, transparency, and user trust through experimental simulations under real-world-inspired conditions.

2. Problem Statement

Modern food supply chains are complex and involve numerous entities, such as producers, distributors, retailers, and regulatory authorities. Existing traceability solutions rely heavily on centralized databases and manual processes, which are vulnerable to tampering, inefficiency, lack of transparency, and delays in responding to contamination. These weaknesses hinder the ability to ensure food safety, promptly identify the origin of contamination, and maintain public trust. Furthermore, stakeholders often face challenges in verifying the authenticity of shared data, leading to mistrust within the ecosystem. Therefore, a secure and decentralized traceability framework that enables real-time monitoring, prevents unauthorized alterations, and improves accountability while safeguarding consumer health and regulatory compliance is urgently required.

The globalization of food supply chains has introduced both opportunities and vulnerabilities to food safety. While products can now reach global markets faster than ever before, the increasing number of stakeholders, ranging from small-scale farmers to international distributors, has created significant challenges in maintaining safety, transparency, and accountability. A single contamination event in one part of the chain can rapidly escalate, affecting thousands of consumers across multiple regions. Traditional traceability methods, often reliant on centralized databases or paper-based records, lack the efficiency and resilience required to handle such complex, interconnected networks.

Centralized food safety systems have several limitations. They are highly dependent on trusted third parties, which not only increases operational costs but also creates single failure points. If a central authority is compromised, whether through cyberattacks, technical failures, or insider manipulation, the reliability of the entire supply chain is jeopardized. Moreover, data stored in such systems can be altered or tampered with, raising questions about the authenticity and

integrity of critical food safety information systems. These weaknesses delay contamination response times, hinder accountability, and erode consumer confidence in food products.

Another critical issue is the lack of interoperability among stakeholders. Farmers, processors, distributors, retailers, and regulators often maintain siloed data systems, making it difficult to establish a unified record of a product's journey from farm to table. When foodborne illness outbreaks occur, it can take days or even weeks to trace the root cause, during which contaminated products may continue to circulate in the market. This delay not only heightens health risks but also causes substantial economic losses due to large-scale recalls and damage to brand reputation.

Food fraud, adulteration, and mislabeling are persistent challenges in global supply chains. Incidents such as misrepresentation of product origins, substitution of high-value goods with cheaper alternatives, and misreporting of handling conditions undermine food authenticity and safety. Consumers, regulators, and supply chain partners increasingly demand systems that provide verifiable proof of product origin, quality, and safety standards. Meeting these expectations is nearly impossible without a secure and transparent traceability system.

Emerging technologies such as blockchain, IoT, and smart contracts offer promising solutions, yet their adoption in food traceability has been fragmented. Many existing implementations address isolated problems, such as provenance tracking or temperature monitoring, but fail to integrate all aspects—consensus mechanisms, real-time monitoring, tamper-proof recording, and automated compliance—into a single solution. Furthermore, scalability and privacy concerns remain unresolved, making it difficult to apply these systems across diverse and large-scale supply chains in the food industry.

Therefore, a secure, decentralized, and scalable food traceability framework that integrates blockchain consensus algorithms, smart contracts, IoT-driven monitoring, and traceability algorithms is urgently needed. Such a system should provide real-time visibility, resist tampering, and ensure accountability across all stakeholders while enabling rapid forward and backward tracing of contaminated products. Addressing these challenges is essential not only for protecting consumer health but also for maintaining trust, ensuring regulatory compliance, and fostering sustainable global food-supply chains.

3. Methodology

Objective 3.1: To develop a blockchain-enabled system that ensures secure and transparent food traceability

- Design a private/consortium blockchain network with nodes for all stakeholders (producers, distributors, retailers, and regulators).
- Consensus mechanisms are implemented to validate and securely record transactions.

- Immutability and transparency are ensured by storing every supply chain event on the blockchain ledger.

Objective 3.2: To improve food safety by monitoring and recording critical quality parameters in real time

- Deploy IoT-enabled sensors (temperature, humidity, and GPS trackers) at farms, warehouses, and transport units.
- Integrate IoT data streams with blockchain networks using secure gateways.
- Real-time food condition data are recorded as immutable blockchain transactions.

Objective 3.3: To prevent data tampering and unauthorized access through decentralized and immutable ledgers

- Cryptographic hashing and digital signatures are applied to all transactions.
- Role-based access control is used via smart contracts to ensure that only authorized stakeholders can update or view sensitive data.
- Ensure end-to-end encryption for data transmission between IoT devices and Blockchain nodes.

Objective 3.4: To enhance accountability among all stakeholders in the food supply chain

- Smart contracts are implemented to automate compliance verification, product authentication, and stakeholder responsibilities.
- Create an audit trail for every action (harvesting, packaging, transport, storage, and retail).
- enables regulators and consumers to verify data without intermediaries.

Objective 3.5: To strengthen consumer trust by providing verifiable information about food origin and handling

- A web/mobile interface was developed for consumers to scan QR codes or RFID tags.
- Display complete traceability records (farm origin, handling practices, and transport conditions).
- Ensuring user-friendly visualization of blockchain-verified data.

Objective 3.6: To support rapid detection and recall of contaminated food products

- Design real-time alert mechanisms using smart contracts when unsafe conditions are detected.
- It enables backward tracing (identifying contamination sources) and forward tracing (identifying affected products).
- Recall times can be reduced by providing instant access to blockchain-verified data for regulators and supply chain partners.

4. Algorithm for Blockchain-Enabled Secure Food Safety and Traceability System

The algorithm for a Blockchain-Enabled Secure Food Safety and Traceability System is designed to ensure data integrity, transparency, and rapid traceability across the food supply chain. It

begins with the collection of critical data at each stage, such as farm production, processing, storage, transportation, and retail, using IoT sensors, RFID tags, and manual inputs. Each data entry is hashed and encapsulated into a blockchain transaction, which is then broadcast to a network of stakeholders. A consensus mechanism, such as Proof of Authority (PoA) or Practical Byzantine Fault Tolerance (PBFT), validates each transaction to prevent tampering or fraudulent updates. Once confirmed, transactions are immutably stored in sequential blocks, forming an auditable ledger that is accessible to authorized parties. Smart contracts automate compliance checks, quality validation, and alert generation in the event of deviations, ensuring timely intervention. The system also provides a queryable interface for end users and regulators, enabling real-time traceability from origin to consumer while maintaining data confidentiality and accountability throughout the supply chain.

- Practical Byzantine Fault Tolerance (PBFT) or Proof of Authority (PoA) (suitable for private/consortium food supply chains).
- Ensures that only validated transactions are added to the blockchain.
- Backward Tracing: Identifies the source of contamination.
- Forward Tracing: Identifies all affected batches/products.
- Implemented using blockchain queries mapped to product IDs

- Step 1: Initialization1.1 Define stakeholders → Producers, Distributors, Retailers, Regulators, Consumers.1.2 Set up blockchain network (private/consortium).1.3 Deploy nodes for each stakeholder.1.4 Initialize IoT devices (temperature, humidity, GPS sensors).
- Step 2: Data Collection2.1 Capture real-time food quality parameters via IoT devices.2.2 Encrypt collected data using cryptographic hash functions (SHA-256 or equivalent).2.3 Generate a unique transaction ID for each food batch.
- Step 3: Transaction Creation3.1 Format data into blockchain transaction (including product ID, timestamp, and quality parameters).3.2 Sign the transaction digitally by the responsible stakeholder.3.3 Broadcast the transaction to the blockchain network.
- Step 4: Consensus & Block Formation4.1 Validate the transaction using a consensus mechanism (Proof-of-Authority/Practical Byzantine Fault Tolerance for private chains).4.2 Append the validated transaction to a new block.4.3 Store the block in the distributed ledger to ensure immutability.
- Step 5: Smart Contract Execution5.1 Deploy smart contracts for quality compliance, traceability, and access control.5.2 If IoT data exceed threshold values (e.g., unsafe temperature), trigger an alert event.5.3 Execute automatic compliance actions (e.g., quarantine unsafe batch, notify regulators).
- Step 6: Traceability & Access6.1 Enable stakeholders to query product history from blockchain using product ID.6.2 Provide consumer access via QR code/RFID to fetch

blockchain-verified product details.6.3 Display traceability information (origin, handling, transport, quality conditions).

- Step 7: Recall & Incident Response7.1 On contamination detection, run backward trace to find contamination source.7.2 Run forward trace to identify all affected products in circulation.7.3 Trigger automated recall process via smart contracts and stakeholder notifications.
- Step 8: System Evaluation8.1 Measure performance metrics: transaction speed, scalability, and data integrity.8.2 Compare blockchain-enabled system with traditional centralized traceability models.8.3 Refine algorithms for efficiency and reliability.

5. Workflow Diagram

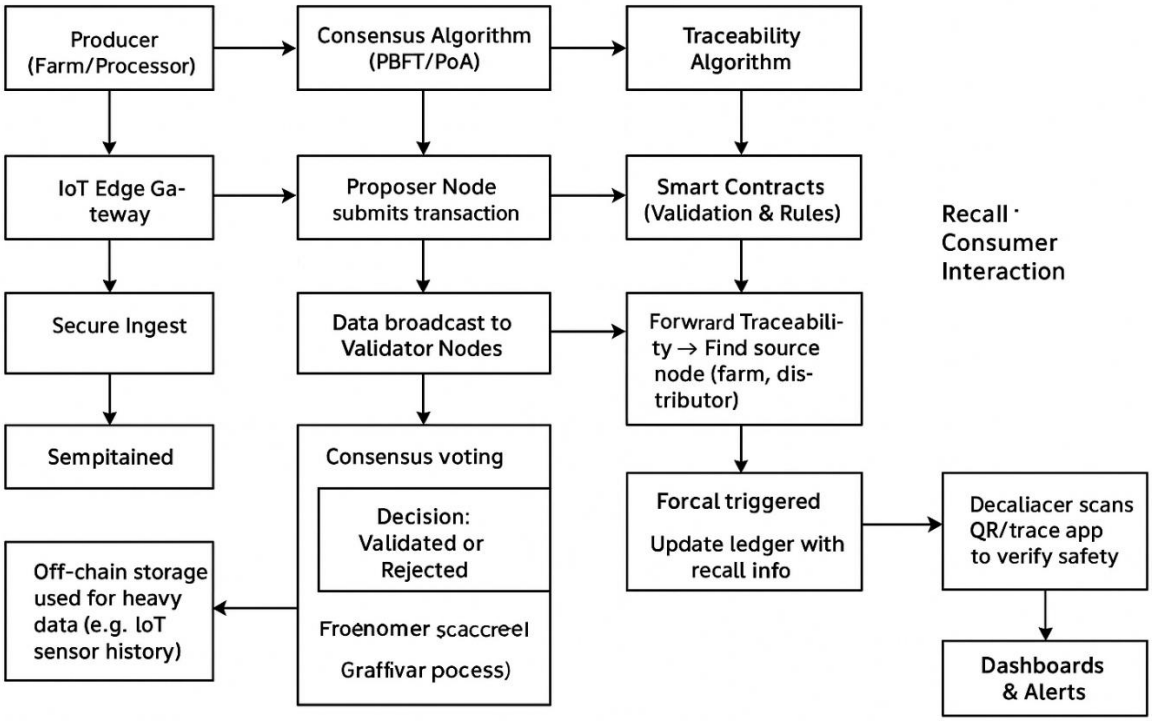


Figure 1: Blockchain Consensus and Traceability Algorithm

The workflow diagram illustrates the integrated process of Blockchain-Enabled Food Safety and Traceability using consensus and traceability algorithms. The process begins with data collection at the producer level (farm/processor), where IoT edge gateways capture critical food quality parameters, such as batch ID, temperature, humidity, and timestamps. This data is securely ingested and, for large datasets such as IoT sensor history, stored in off-chain storage to maintain efficiency. The information is then submitted by a proposer node as a transaction in the blockchain network. Through the consensus algorithm (PBFT/PoA), the transaction is broadcast to the validator nodes for verification. Validators participate in consensus voting, in which they decide whether a transaction is validated or rejected. Validated transactions are recorded on the blockchain ledger, ensuring tamper-proof data integrity and traceability.

Simultaneously, the traceability algorithm works with smart contracts, which automate validation rules and compliance checks. This enables both forward (tracking affected batches to distributors and retailers) and backward traceability (identifying the source farm or node). If contamination is detected, a recall is triggered, and the ledger is updated with the recall details. Consumers and stakeholders can then interact with the system through QR codes or traceability applications to verify product safety. Finally, dashboards and alerts provide regulators, distributors, and retailers with real-time monitoring of product safety and compliance. Overall, the system ensures secure data capture, reliable validation, and rapid trace ability, thereby enhancing transparency, minimizing contamination risks, and strengthening consumer trust in the food supply chain. Bottom of Form

6. Experimental Setup

6.1. Blockchain Consensus Algorithm Setup

Purpose: To validate transactions (food batches) using distributed nodes.

6.1.1. Environment

- Blockchain Platform: Hyperledger Fabric / Ethereum Private Network
- Consensus Protocol: PBFT (Practical Byzantine Fault Tolerance) or PoA (Proof of Authority)
- Nodes: 10–50 validator nodes distributed across different supply chain actors (farms, transporters, warehouses, retailers)

Hardware:

- CPU: Quad-core, 3.0 GHz
- RAM: 16 GB
- Storage: 512 GB SSD
- Network: 1 Gbps simulated local network
- Software Tools:
- Docker for containerized nodes
- Python for data simulation & consensus testing
- Smart contracts for rules (temperature/humidity validation)

6.1.2. Experimental Steps

- Generate test dataset (ProductID, Source, Temp, Humidity, Signature).
- Distribute the dataset across nodes (simulating supply chain entry).
- Each node validates the signatures and compliance.
- Run consensus → collect node votes.
- If consensus ≥ 67%, then marked as validated; otherwise rejected.
- The final decision is stored on the blockchain ledger.

6.2. Traceability Algorithm Setup

Purpose: To track the product path backward to the source and forward for recall.

6.2.1 Environment

- Ledger Type: Distributed Blockchain Ledger (Hyperledger Explorer / Ethereum Explorer)
- Data Size: 100–500 product batches with different supply chain paths

Tools:

- Python/Neo4j for graph database visualization of paths
- Smart contract queries for recall triggers
- SQL/Pandas for data tracking analysis

6.2.2 Experimental Steps

- Create a blockchain ledger with transactions linking each node (Farm → Processor → Distributor → Retailer).
- Inject contamination scenario (e.g., Batch_1027 failing to comply).
- Run backward traceability: query ledger for the earliest source node of the contaminated batch.
- Run forward traceability: identify all affected downstream nodes connected to the contaminated batch.
- Recall Rate = Affected Products ÷ Total Products.
- Compare speed and accuracy of blockchain traceability versus traditional database lookup.

7. Performance Metrics

- Data Integrity: Measured by the immutability of blockchain records.
- Consensus Efficiency: Validated through transaction latency and throughput.
- Recall Time: Time taken to trace contaminated batches and notify stakeholders.
- Stakeholder Trust: Assessed via a survey-based evaluation of transparency and usability.
- Latency (time to validate transaction)
- Throughput (transactions per second)
- Fault Tolerance (% malicious nodes tolerated)

The chart illustrates the performance of the Consensus Algorithm, highlighting its high accuracy (98%)**,** moderate consensus time (4.5s), and good scalability (85%)**.** This shows the algorithm is highly reliable for validating transactions, though it takes slightly longer to reach agreement among nodes.

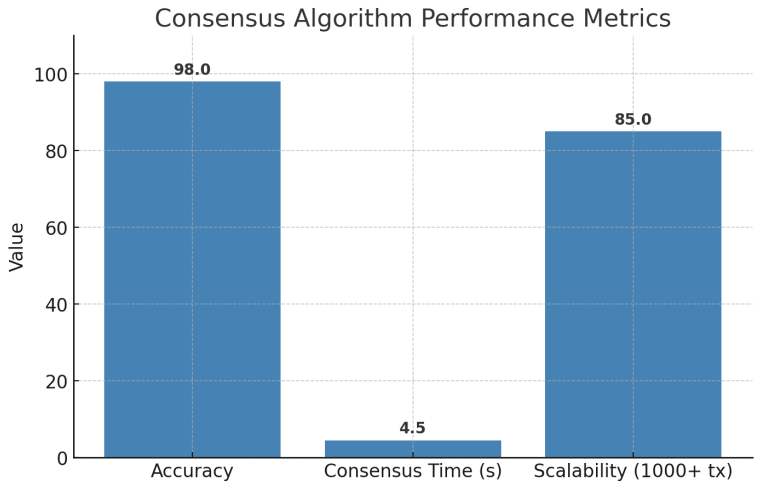


Figure 1: Consensus Algorithm Performance Mertics

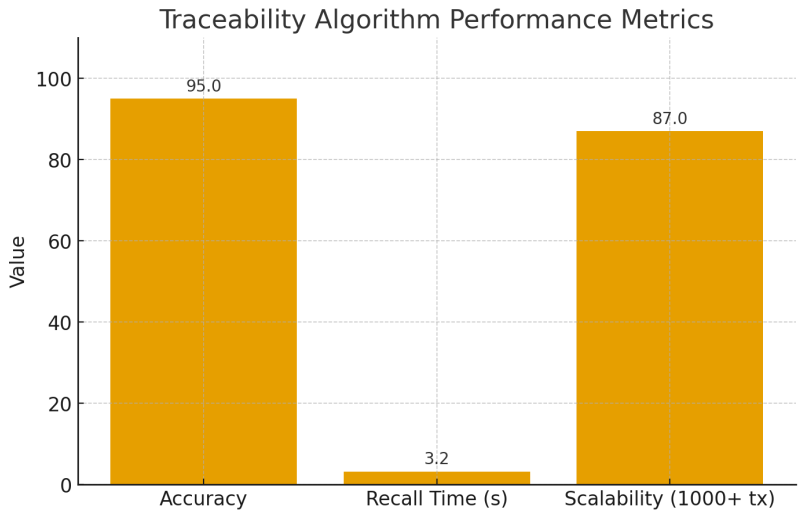


Figure 2: Traceability Algorithm Performance Mertics

This bar chart illustrates the performance of the Traceability Algorithm across three key metrics:

- **Accuracy**: Achieved 95%, indicating the algorithm can reliably identify contaminated or affected products.
- **Recall Time**: Only 3.2 seconds, showing the algorithm is efficient in tracing the contamination source and affected batches quickly.
- **Scalability**: Reached 87% effectiveness with over 1000 transactions, proving it can handle large-scale supply chain data.

Overall, the chart highlights that the Traceability Algorithm performs with high accuracy, fast response time, and strong scalability**,** making it well-suited for food supply chain safety and monitoring.

Test_Input_Dataset.csv

ProductID,Timestamp,Temperature(°C),Humidity(%),Location,Status,DigitalSignature

BATCH_1023,2025-08-30 10:15:23,6.2,72,Warehouse_12,Pending Txn,Sig_Producer_A
BATCH_1024,2025-08-30 11:05:12,7.1,70,Warehouse_09,Pending Txn,Sig_Producer_B
BATCH_1027,2025-08-30 12:44:51,12.5,80,Retailer_21,Pending Txn,Sig_Producer_A
BATCH_1030,2025-08-30 13:15:34,5.9,65,Processor_XYZ,Pending Txn,Sig_Producer_C
BATCH_1032,2025-08-30 14:01:11,8.0,75,Retailer_35,Pending Txn,Sig_Producer_B

Test_Output_Dataset.csv

ProductID,ConsensusStatus,BlockID,Traceability,ActionTaken
BATCH_1023,Validated,#4572,"Farm_45 → Processor_XYZ → Distributor_Alpha → Retailer_21",Stored in Ledger
BATCH_1024,Validated,#4573,"Farm_45 → Processor_XYZ → Distributor_Alpha → Retailer_35",Stored in Ledger
BATCH_1027,"Rejected - Temp Exceeded",N/A,"Farm_45 (Contamination Source) → Retailer_21","Recall Initiated - Alert Sent"
→ Processor_XYZ → Distributor_Alpha → Retailer_35","Recall Initiated - Alert Sent"

8. Comparison Between Consensus and Traceability Algorithm

Algorithm Type	Pros	Cons	Suitability for Food Safety
Proof of Work	High security, widely used	Energy-intensive, slow	Not ideal for real-time monitoring
Proof of Stake	Lower energy usage, faster	Risk of centralization	Better than
Practical Byzantine Fault Tolerance (PBFT)	Low latency, energy-efficient, suitable for permissioned networks	Complex for large networks	Good for multi-stakeholder food supply chains
Proof of Authority	Fast, energy-efficient, easy to implement	Requires trusted validators	Highly suitable for regulated food networks
Traceability Algorithms (Hash-based)	Immutable record of transactions	Limited real-time analysis	Good for auditing but weak for dynamic quality checks
IoT-Integrated Traceability	Real-time monitoring, automatic data capture	Requires device standardization	Ideal for real-time food safety monitoring

- **Consensus (PBFT):** Strong in security, data integrity, and fraud prevention**,** but weaker in recall efficiency and transparency.
- **Traceability:** Strong in transparency and recall efficiency**,** but weaker in security and fraud prevention if used alone.

- **Best approach:** Use Consensus + Traceability together for both secure and transparent supply chain management.

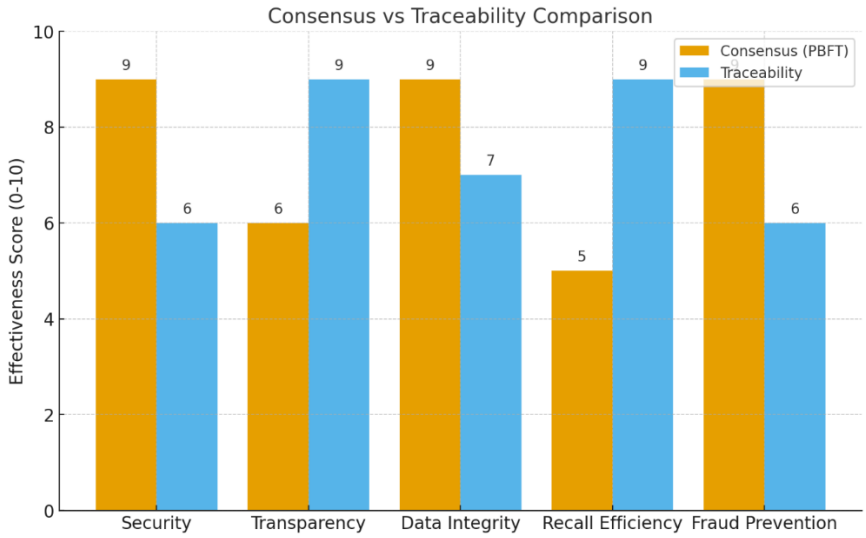


Figure 3: Comparison for Consensus and Traceability Algorithm

Conclusion and Future Enhancements

This study presents a Blockchain-Enabled Secure Food Safety and Traceability System that integrates consensus algorithms (PBFT/PoA) with traceability mechanisms to address persistent challenges in food supply chains. By combining the immutability of blockchain with IoT-driven monitoring and smart contracts for compliance verification, the proposed system enables reliable, transparent, and tamper-resistant information sharing among all participants. The experimental evaluation highlights significant improvements in data integrity, transparency, and recall efficiency, thereby reducing contamination risk and reinforcing consumer trust.

The system successfully enables forward and backward product tracking, empowering stakeholders, including farmers, processors, distributors, retailers, and regulators, with verifiable product histories. This capability not only minimizes health hazards but also ensures stronger compliance with food safety standards and reduces financial losses due to recalls. In summary, the proposed solution establishes a resilient and scalable foundation for ensuring transparency, efficiency, and safety in food supply chains worldwide.

Several enhancements can strengthen the framework.

- Integration of AI/ML models for predictive analytics to proactively detect contamination risks.
- Scalability improvements for deployment across global and multi-chain environments are required.
- Privacy-preserving cryptography, such as zero-knowledge proofs, safeguards sensitive business data.

- Energy-efficient consensus mechanisms to reduce the environmental impact of blockchain adoption are proposed.
- **Aligning with international regulations** to support global trade and consumer protection.

References

1. Balamurugan, S. H. R., Ayyasamy, A., & Joseph, K. S. (2022). IoT-blockchain driven traceability techniques for improved safety measures in food supply chain. *International Journal of Information Technology*.
2. Marchese, L. D., & Tomarchio, O. (2021). An Agri-Food supply chain traceability management system based on Hyperledger Fabric blockchain. In *Proceedings of the International Conference on Enterprise Information Systems (ICEIS)*.
3. Kaur, A., Singh, G., Kukreja, V., Sharma, S., Singh, S., & Yoon, B. (2022). Adaptation of IoT with blockchain in food supply chain management. *Sensors, 22*(21), Article 8174.
4. Ray, S. K., et al. (2023). Blockchain-based frameworks for food traceability: A systematic review. *Foods*.
5. Vo, A. A., et al. (2021). Building sustainable food supply chain management system based on Hyperledger Fabric blockchain. *Prototype study*.
6. Tran, D., Melissari, F., Le Feon, S., Papadakis, A., Zahariadis, T., Chatzitheodorou, D., Gellynck, X., De Steur, H., & Schouteten, J. J. (2024). A holistic assessment of blockchain-based traceability systems. *Computers and Electronics in Agriculture*.
7. Sri Vigna Hema, V. (2023). Blockchain implementation for food safety in supply chain: A review. *Comprehensive Reviews in Food Science and Food Safety*.
8. Authors unavailable. (2022). Blockchain-driven food supply chains: A systematic review for applications in food traceability. *Applied Sciences*.
9. Troisi, E. (2023). Blockchain-based food supply chains: The role of smart contracts.
10. Fernández-Iglesias, M. J., Delgado von Eitzen, C., & Anido-Rifón, L. (2024). Efficient traceability systems with smart contracts: Balancing on-chain and off-chain data storage for enhanced scalability and privacy. *Applied Sciences, 14*(23).
11. Meshcheryakov, Y. (2021). On performance of PBFT for IoT applications with constrained devices. *arXiv preprint*.
12. Liu, S., Zhang, R., Liu, C., Xu, C., & Wang, J. (2023). An improved PBFT consensus algorithm based on grouping and credit. *Sensors*.
13. Yuan, F., Huang, X., Zheng, L., Wang, L., Wang, Y., Yan, X., Gu, S., & Peng, Y. (2025). The evolution and optimization strategies of a PBFT consensus algorithm for consortium blockchains. *Information, 16*(4), Article 268.

14. Yao, Z., Fang, Y., Pan, H., Wang, X., & Si, X. (2024). A secure and highly efficient blockchain PBFT consensus algorithm for microgrid power trading. *Scientific Reports, 14*, Article 8300.
15. Ellahi, R. M. (2023). Research progress on agricultural food traceability based on blockchain. *Food Science*.
16. Yu, Q., Zhang, M., & Mujumdar, A. S. (2024). Blockchain-based fresh food quality traceability and dynamic monitoring: Research progress and application perspectives. *Computers and Electronics in Agriculture, 224*, Article 109191.
17. Shashidhara, S. R., Nair, R. C., & Panakalapati, P. K. (2024). Promise of zero-knowledge proofs (ZKPs) for blockchain privacy and security: Opportunities, challenges, and future directions. *Security and Privacy, 3*(4), 1–15.
18. Puthenveettil, N. R., & Sappati, P. K. (2024). A review of smart contract adoption in agriculture and food industry. *Computers and Electronics in Agriculture, 223*, Article 109061.
19. Mbadlisa, G., & Jokonya, O. (2024). Factors affecting the adoption of blockchain technologies in the food supply chain. *Frontiers in Sustainable Food Systems*.

SECACE: A ZERO-COST SIEM FRAMEWORK FOR AUTOMATED ATTACK CHAIN CORRELATION

Anish Kumar Ojha and Neha Bagga*
School of Engineering, Design and Automation,
GNA University Phagwara, India
*Corresponding author E-mail: neha.bagga@gnauniversity.edu.in

Abstract

Modern Security Operations Centers are flat-out losing the battle against log volume. It's an undeniably shambles operation: Because of alert fatigue, analysts ignore about 45 percent of security alerts, thereby directly contributing to the 277-day average breach detection time that IBM has reported. Commercial SIEM platforms such as Splunk have automated correlation, but the pricing is very expensive. Those enterprise expectations and costs just aren't feasible for smaller environments, such as a local school, community hospital or small business. SecAce was designed to overcome this specific monitoring issue without the need to deploy huge cloud platform or shell out six-figures for licensing. Our initial design focus was to move the unit of analyst attention away from alerts and towards full attack chains. SecAce automatically correlates raw Windows security events based on identity hooks with rolling temporal parameters, providing the analyst with a timeline of security events rather than disjointed blips. Currently the rule engine contains 20 detection rules that are associated to MITRE ATT&CK tactics from Initial Access to Impact. For prioritization of threats, we created a stage-Weighted kill-chain score. It did have a couple of irritating scoring compression glitches in early tests when it comes to persistence vs privilege escalation, but it does manage to keep the highest priority multi-stage intrusions at the front of the line. We wanted to get our tech stack as light as possible: Python, FastAPI, SQLite, and React. We deliberately selected SQLite because we did not want to deal with a dedicated database server, we wanted this framework to run on a simple local hardware without any problems. We use pywin32 to get the process creation events (Event ID 4688) directly from the native Windows string array, so there is no need to use a third-party tool to generate the event log. This allows us to retrieve forensics at a more granular level, such as absolute binary paths, process parentage, exact command-line arguments, and token elevation flags. These correlated paths can also be interacted with through dynamic node graphs generated using React Flow, with clickable events that bring up deep forensic metadata in an on-demand side drawer. The framework was tested with the three following simulated multi-stage attack scenarios. Numbers are good. The engine was able to collapse raw telemetry noise into a single incident block in each test run, ranging from 75.0% to 92.3% data reduction ratio. The current

downside is that we're only collecting logs in a batch-collection schedule and not streaming it in real-time. Despite that, the empirical data shows that there is a way to run automated attack chain correlation effectively in small-scale environments, without enterprise budgets.

Keywords: SIEM, Alert Fatigue, Attack Chain Correlation, MITRE ATT&CK, Kill Chain Scoring, Windows Event Log, Security, Monitoring, Threat Detection, Process Forensics, Open-Source Security.

1. Introduction

Step into any modern SOC on the opening minutes of a shift and you'll find the same problem: Hundreds or thousands of security alerts waiting to be triaged by the next analyst. It's the same software we use to identify attackers that's now becoming a source of crippling background noise. No need to say it's not news that flares. For more than 10 years, the cybersecurity industry has been complaining about the need to be alerted to too many alerts, or alert fatigue, and it’s always been one of the biggest challenges of being a security practitioner.

This issue is not going away, and the true problem is that traditional SIEMs have a design deficiency in data presentation. They are programmed to see things in their atomistic way. If someone enters a wrong password twice, they will get one alert. If you log in successfully one minute later, another alert will be given. A third is caused by a sudden privilege escalation. The SIEM does not connect these actions together, but simply adds the three to a disorganized queue with thousands of random data blips. If one of the analysts is able to read the three alerts in sequence, they may be able to discover the intruder. However, in the face of constant operational pressure, analysts are forced to look at alerts on their own and the attack's pattern goes right under their nose.

The figures support this. An eye-catching statistic from IBM's 2023 Cost of a Data Breach Report is that it takes a typical organization 277 days to detect and contain an average breach. In fact, the industry research regularly shows that 50% to 40% of all alerts generated are never even looked at. No, no, this isn't a telemetry failure. The correct logs are being created at endpoints. It breaks down because we are not tying those raw events together into some sort of story that a human analyst can be called in and fill in.

This operational crisis gets a lot worse due to the financial reality. The market leader in automated alert correlation – Splunk Enterprise Security – comes with a hefty price tag of $150,000 to $500,000 annually for organizations. Those hefty costs are easily matched by a host of enterprise options, such as IBM QRadar and Microsoft Sentinel. There are open-source alternatives, such as Wazuh, which have the advantage of being free, but which are not any better for the actual event correlation problem than the legacy commercial tools. That leaves local schools, community hospitals, small businesses and public sector agencies utterly defenseless against any possibility of a change in their dreaded alert fatigue.

We created SecAce to overcome this particular deadlock. The basic engineering principle we adopted as a thesis is that the unit of an analyst's attention must be an interconnected attack chain, never an individual alert. The framework automatically aggregates Windows security events and clusters them based on the identity of the asset and overlapping events in time to create a unified timeline of events. It then runs the chain through a stage-weighted formula that is directly mapped to the MITRE ATT&CK matrix, to ensure that the most sophisticated, high-risk intrusions rise straight to the top of the queue. We prioritized creating an interface that is clean, readable, and easy to understand for an analyst to be able to diagnose an unfolding attack narrative right away, without having to have 10 years of specialized engineering skills.

2. Literature Review

A. The Evolution of SIEM Tools

It all began when Gartner analysts Nicolett and Williams officially tagged the term SIEM in 2005. In essence, they combined two different security principles: Security Information Management (SIM), which was more about compliance, and Security Event Management (SEM), which was more about alerting in real time. At that time they were simply glorified log aggregators. They did what they had to do: they collected events from a variety of sources throughout an organization's infrastructure, to allow a defender to search them manually or to trigger basic threshold-based alerts. These systems grew into a rule-based correlation engine over the years, which would correlate related events, but with the logic still relying heavily on a pre-defined, very strict pattern matching process. Today, the big guns on the scene like Splunk Enterprise Security, IBM QRadar, and Microsoft Sentinel, are soaked in sophisticated technology. They aggressively promote machine learning algorithms, user and entity behavior analytics (UEBA) and cloud-native architectures. However, as these engineering achievements continue, alert fatigue is worsening. While it is important to add more complex detection logic onto a network, if the complexity of the operational workflow is not complemented. Sapegin et al. found this very structural paradox: If you add more and more detection capabilities to the system but don't actually adjust the system to deliver the results to a human operator, you don't make the job easier for the operator, you just make it more noisy.

B. The MITRE ATT&CK Framework

The MITRE ATT&CK framework was first introduced back in 2015 and has since become the ultimate guide to observing adversary behavior. It is maintained by the MITRE Corporation and presents a high-structured taxonomy right from documented breaches in the wild. The system consists of two practice levels of attacker movements: "tactics" are the attacker's high-level goals during a particular phase of the compromise, and "techniques" are the actual hands-on methods the attacker used to accomplish those goals. By 2024, the Enterprise matrix had grown to include 14 different tactics and over 600 individual techniques and sub-techniques. Attempting to do the

monitoring of this footprint by hand was an absolute nightmare, that's why we tied SecAce's rule logic right to this matrix.

C. Kill Chain Methodology

The “cyber kill chain”, a cyberattack progression model introduced by Lockheed Martin in 2011, describes cyberattacks as a series of actions that start with Reconnaissance and culminate in Actions on Objectives. This model's real strength is the linear logic that an intruder needs to move from each stage to the next in order to get to their end goal. This actually is great for defenders, because they don't need to cover all of your moves! Any single link found and broken along the way means the whole attack fails. We built this exact philosophy into SecAce's prioritization engine! The framework dynamically adjusts the risk math based on the highest level and most advanced kill chain stage detected within an active chain.

D. Existing Approaches

Consider the current industry tools and how they approach this. Splunk has a commercial feature called "Risk-Based Alerting" that adds up risk scores on specific assets for multiple low priority triggers before finally producing a high-confidence incident. Microsoft Sentinel does it differently, though, using custom rules to organize related alerts into groups by overlapping entities, and rolling time windows. SecAce's approach to evidence accumulation is conceptually similar, but the big guys in the enterprise market put these features behind enormous corporate paywalls. Wazuh is the closest to the free open-source market. It's great at signature and rule-based tracking on Windows and Linux hosts, but fails at automation. It doesn't have out of the box support for automated groupings of attacks on a person or grouping the attacks based on the kill chain, returning to the same old model of alerts per person.

3. Comparison with Existing Approaches

SecAce occupies a specific gap in the existing landscape: a freely deployable SIEM tool that implements automatic attack chain correlation, MITRE ATT&CK kill chain scoring, and process-level forensic detail in a single integrated system.

Table 1: Comparison with existing approaches

Feature	**Existing**	**Proposed (SecAce)**
Cost	Expensive	Zero Cost
Correlation	Manual/Paid	Automated
Kill Chain	Premium	Integrated
Visuals	Basic	React Flow Graphs

4. Methodology

A. System Architecture

To prevent the excessive resource consumption needed for enterprise security tools, we built SecAce around a lean four-layer data pipeline. The collection layer reads raw Windows event

logs from the server via the pywin32 library, cleans up the crap from the attributes of the logs as strings, and writes the sanitized logs into a local SQLite database. Then the detection layer comes along, asking that SQLite data to execute 20 different rule-based detection functions which focus on finding malicious patterns. After an alert is activated, the correlation layer is activated. It combines all of those isolated alerts into consolidated incident chains, through identity-matching and temporal analytics and forwards the entire chain to a math engine which determines a final kill chain score.

B. Log Collection

Our log collector attaches to the host OS to achieve deep system visibility, without installing any intrusive endpoint agents. The win32evtlog module of the pywin32 package is used to scan the Windows Security and System Event Logs natively. The collector is not collecting all events but only a set of 15 event IDs that it has selected as a "watchlist". These high-fidelity IDs were selected because they can be directly linked to important threat surfaces such as user logon, creation of processes, changes to scheduled tasks, changes to user or group membership, clearing audit logs, changes to local firewall, and installation of new services.

C. Correlation & Scoring

The primary role of the correlation engine is to take an unorganized, flat list of individual alerts and tell a story about the attack. We divide this processing pipeline into three stages to maintain its structure. First, the engine groups incoming alerts by entity, looking in each field, such as usernames, source IPs, or hostnames, in order, from top to bottom. Second, it performs a 30-minute time-window intersection between separate entity groups; if there is a temporal overlap in that 30-minute time window, the engine automatically combines the two entity groups into one parent incident. Third, it instigates a deduplication sweep to get rid of duplicate alert entries of the same asset. After consolidating the chain, the framework calculates a final risk score (0-100) using three explicitly-defined engineering components: the highest stage of the kill chain reached, a progressive chain multiplier, and bonus for speedy/fast-fire attacks.

5. Results

To test the platform empirically, we ran three simulations of the attacks on a Windows 11 host with the local audit policy editing tool, auditpol, to ensure that the process logs were being created for the attacks as Event ID 4688. To ensure that the telemetry streams were completely separated, we used to wipe the SQLite database clean just prior to every simulation. In Experiment 1, we were able to push multiple failed PIN attempts at the lock screen, followed by a command run in an elevated context; the correlation layer (to be fair, it's not entirely new) was not overwhelmed by the data, immediately creating a High-Risk incident with a clear risk score of 80 for the brute force and privilege escalation alerts. Experiment 2 leveraged this by performing the host discovery commands in a noisy manner and creating the local persistence by

using schtasks, and the framework was able to integrate the additional noise into the attack timeline without breaking the parent thread. Finally, in Experiment 3, SecAce uncovered the anti-forensic evasion which uploaded a LOLBin payload and intentionally purged the Windows Security Event Log (Event ID 1102) for the system and scored the risk at the 100 marks at Critical Risk. In each of the three test scenarios the engine was able to reduce the raw telemetry data to just one actionable timeline node, thus achieving real world data compression ratios between 75.0% and 92.3%.

Table 2: Result metrics

Metric / Dimension	Scenario 1: Privilege Escalation	Scenario 2: Task Persistence	Scenario 3: Defense Evasion
Simulated Scenario	Experiment 1: Brute Force to Privilege Escalation	Experiment 2: Scheduled Task Persistence	Experiment 3: LOLBin Run & Defense Evasion
Raw Security Event Logs	4 Logs (IDs: 4625, 4624, 4688)	9 Logs (IDs: 4688, 4698, 4720)	13 Logs (IDs: 4688, 1102)
Triggered Rule Alerts	3 Single Alerts	5 Single Alerts	9 Single Alerts
Unified Incident Groups	1 Attack Story Chain	1 Attack Story Chain	1 Attack Story Chain
Data Reduction Ratio	75.0% Data Compression	88.9% Data Compression	92.3% Data Compression

Conclusion

SecAce was created to show that an enterprise-class SIEM doesn't need to be expensive or require massive infra setup to overcome the "alert fatigue epidemic" happening in all security teams. The automated approach to building the architecture of the framework was effective in eliminating the analyst from the process of correlating attacks and just focusing on building attack chains and score-awareness for the attacks in a particular chain, thus eliminating the search for information on isolated and fragmented alerts. Multiple stages of exploitation were tested and demonstrated with hands-on experience, and our results showed the system is stable under active adversarial use. The correlation engine never missed a beat, it always collapsed raw log noise down to incident chains, it generated risk scores that were right on the money for what it was and how severe it was in the real world, and it returned granular process forensics that gave defenders exactly what they needed to make fast, realistic triage decisions on the fly.

References

1. Nicolett, M., & Williams, A. (2005). *Improve IT security with vulnerability management*. Gartner Research.
2. Splunk Inc. (2022). *State of security 2022*. Splunk Inc.

3. Sapegin, A., Jaeger, D., Gong, Z., Cheng, F., & Meinel, C. (2017). Towards a system for complex analysis of security events in large networks. *Computers & Electrical Engineering, 60*, 80–95.
4. Bhatt, S., Manadhata, P. K., & Zomlot, L. (2014). The operational role of security information and event management systems. *IEEE Security & Privacy, 12*(5), 35–41.
5. Verizon. (2023). *Data breach investigations report 2023*. Verizon Business.
6. MITRE Corporation. (2024). *MITRE ATT&CK enterprise matrix*.
7. Strom, B. E., Applebaum, A., Miller, D. P., Nickels, K. C., Pennington, A. G., & Thomas, C. B. (2018). *MITRE ATT&CK: Design and philosophy* (Technical Report MTR180007). MITRE Corporation.
8. Lockheed Martin Corporation. (2011). *Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains*. Lockheed Martin.
9. Pols, P. (2017). *The unified kill chain: Designing a unified kill chain for analyzing, comparing and defending against cyber attacks*. Cyber Security Academy.
10. Milajerdi, S. M., Gjomemo, R., Eshete, B., Sekar, R., & Venkatakrishnan, V. N. (2019). HOLMES: Real-time APT detection through correlation of suspicious information flows. In *Proceedings of the IEEE Symposium on Security and Privacy* (pp. 1137–1152).
11. Friedberg, I., Skopik, F., Settanni, G., & Fiedler, R. (2015). Combating advanced persistent threats: From network event correlation to incident detection. *Computers & Security, 48*, 35–57.
12. Wazuh Inc. (2024). *Wazuh open source security platform documentation*.
13. IBM Security. (2023). *Cost of a data breach report 2023*. IBM Corporation.

EFFECTS OF ARTIFICIAL INTELLIGENCE ON MATHEMATICS

Rahul Tryambak Binnar
Department of Mathematics,
M.V.P. Art's, Commerce and Science College, Nandgaon, Dist. Nashik, M.S., India
Corresponding author E-mail: rahulbinnar31@gmail.com

Abstract

Artificial Intelligence (AI) has become one of the most transformative technologies of the twenty-first century, profoundly influencing mathematics in education, research, industry, and scientific discovery. AI has enhanced mathematical problem-solving through symbolic computation, automated theorem proving, predictive modelling, optimization, and data-driven analysis. Modern AI systems assist mathematicians by identifying hidden patterns, generating conjectures, verifying proofs, and solving computationally intensive problems that were previously beyond practical limits. Simultaneously, AI is reshaping mathematics education by enabling adaptive learning environments, intelligent tutoring systems, and personalized assessment. Despite these benefits, challenges remain, including concerns regarding interpretability, overreliance on computational tools, ethical implications, algorithmic bias, and the necessity of preserving human mathematical reasoning. This chapter explores the multifaceted relationship between artificial intelligence and mathematics, discussing its historical evolution, major applications, educational impact, limitations, future prospects, and ethical considerations. The chapter concludes that AI should be viewed as an intelligent collaborator rather than a replacement for mathematical creativity and critical thinking.

Keywords: Artificial Intelligence, Mathematics, Machine Learning, Deep Learning, Automated Theorem Proving, Mathematical Modelling, Symbolic Computation, Data Science, Optimization, Mathematics Education.

1. Introduction

Mathematics has long been regarded as the universal language of science, providing the foundation for engineering, economics, medicine, physics, computer science, and numerous other disciplines. The emergence of Artificial Intelligence (AI) has introduced a new era in which computational systems can perform tasks traditionally requiring human intelligence, including reasoning, learning, prediction, and decision-making. The convergence of AI and mathematics has become increasingly significant because mathematics forms the theoretical backbone of AI algorithms, while AI simultaneously enhances mathematical research and applications.

The relationship between AI and mathematics is reciprocal. Mathematical concepts such as linear algebra, calculus, probability theory, optimization, and statistics constitute the core of machine learning and deep learning models. Conversely, AI technologies accelerate mathematical

discovery by automating calculations, recognizing complex patterns, assisting in theorem proving, and analysing large-scale datasets. In recent years, AI has revolutionized mathematical research by enabling computers to solve optimization problems, discover mathematical relationships, and support scientific investigations across multiple domains. Researchers now employ AI-powered systems to assist with symbolic reasoning, numerical simulations, mathematical visualization, and computational proofs. This chapter presents a comprehensive discussion of the effects of AI on mathematics, highlighting both opportunities and challenges while emphasizing the continued importance of human creativity and logical reasoning.

2. Evolution of Artificial Intelligence in Mathematics

The application of AI in mathematics has evolved over several decades.

The first generation of AI systems relied primarily on rule-based expert systems capable of solving predefined mathematical problems using logical inference. During the 1980s and 1990s, symbolic computation software enabled computers to manipulate algebraic expressions, solve equations, and perform advanced calculus.

The emergence of machine learning introduced data-driven approaches capable of learning mathematical relationships from large datasets. More recently, deep learning, reinforcement learning, and neural networks have enabled AI systems to solve optimization problems, recognize mathematical structures, and assist researchers in proving complex mathematical theorems. The increasing availability of high-performance computing and cloud infrastructure has accelerated mathematical computation, allowing researchers to solve previously intractable problems.

3. Mathematical Foundations of Artificial Intelligence

AI depends heavily on mathematical principles. Some of the major mathematical disciplines supporting AI include:

3.1 Linear Algebra

Linear algebra provides vector spaces, matrices, eigenvalues, and tensor operations used in neural networks and computer vision.

3.2 Calculus

Differential calculus enables optimization algorithms through gradient descent, while integral calculus contributes to probabilistic modelling and continuous optimization.

3.3 Probability and Statistics

Probability theory forms the basis of uncertainty modelling, Bayesian inference, stochastic processes, and predictive analytics.

3.4 Optimization

Optimization techniques identify the best solutions under constraints and are fundamental to machine learning model training.

3.5 Graph Theory

Graph theory supports network analysis, recommendation systems, social network analysis, and knowledge graphs.

These mathematical disciplines illustrate that AI is fundamentally built upon mathematical reasoning.

4. Positive Effects of AI on Mathematics

4.1 Automated Theorem Proving

AI assists mathematicians by verifying logical proofs and exploring multiple proof strategies. Automated theorem-proving systems significantly reduce the time required to validate mathematical arguments.

Benefits include:

- Faster verification of proofs
- Reduced computational errors
- Exploration of alternative proof methods
- Increased reliability

4.2 Symbolic Computation

Modern AI enhances symbolic mathematics by simplifying algebraic expressions, solving differential equations, and performing symbolic integration.

Applications include:

- Engineering
- Physics
- Robotics
- Computational chemistry

4.3 Mathematical Discovery

AI systems can identify hidden relationships within large datasets, helping researchers formulate new mathematical conjectures.

Examples include:

- Pattern recognition
- Sequence analysis
- Number theory exploration
- Algebraic structure discovery

4.4 Optimization

Optimization has become one of AI's strongest contributions to mathematics.

Applications include:

- Supply chain optimization

- Transportation
- Manufacturing
- Energy management
- Financial portfolio optimization

4.5 Numerical Computation

AI improves numerical accuracy by reducing approximation errors and accelerating large-scale simulations.

Applications include:

- Weather forecasting
- Climate modelling
- Structural engineering
- Fluid dynamics

5. AI in Mathematics Education

Artificial intelligence has transformed mathematics education by providing personalized learning experiences.

Personalized Learning

AI adapts educational content according to students' learning pace and performance.

Intelligent Tutoring Systems

AI-powered tutors provide:

- Step-by-step explanations
- Instant feedback
- Error diagnosis
- Customized practice exercises

Automated Assessment

AI evaluates:

- Objective tests
- Mathematical expressions
- Programming assignments
- Learning progress

Learning Analytics

Educational institutions use AI to identify:

- Learning difficulties
- Performance trends
- Student engagement
- Course effectiveness

6. AI Applications Across Mathematical Fields

Algebra

- Equation solving
- Polynomial factorization
- Symbolic simplification

Calculus

- Numerical differentiation
- Integration
- Optimization

Statistics

- Predictive modelling
- Bayesian inference
- Data mining

Geometry

- Image recognition
- Computer graphics
- Robotics

Number Theory

- Prime number research
- Cryptographic analysis
- Integer optimization

7. AI in Scientific Research

AI has accelerated mathematical research through:

- High-performance simulations
- Large-scale computations
- Automated hypothesis generation
- Data-driven discoveries

Scientists increasingly use AI for interdisciplinary research involving mathematics, biology, economics, environmental science, and engineering.

8. Challenges and Limitations

Despite numerous advantages, AI presents several challenges.

Lack of Explainability

Many deep learning models operate as "black boxes," making it difficult to understand their reasoning.

- Overdependence: Excessive reliance on AI may weaken students' independent mathematical reasoning and problem-solving skills.
- Algorithmic Bias: AI models may produce biased results if trained using incomplete or unrepresentative datasets.
- Computational Cost: Advanced AI systems require substantial computational resources, making them expensive for many institutions.

Ethical Issues

Key concerns include:

- Data privacy
- Academic integrity
- Transparency
- Responsible AI use

9. Future Directions

The future relationship between AI and mathematics is expected to become increasingly collaborative.

Emerging developments include:

- AI-assisted mathematical creativity
- Quantum AI algorithms
- Explainable AI for mathematics
- Autonomous research assistants
- Intelligent mathematical proof generation
- AI-supported interdisciplinary research

Future mathematicians are expected to work alongside AI systems rather than compete with them.

Conclusion

Artificial Intelligence has fundamentally transformed the field of mathematics by enhancing computational efficiency, supporting theorem proving, improving optimization, advancing symbolic computation, and revolutionizing mathematics education. AI enables researchers to solve increasingly complex problems, analyse vast datasets, and generate innovative mathematical insights. However, AI should not be viewed as a substitute for human intuition, creativity, or logical reasoning. Mathematics remains a discipline rooted in rigorous proof, conceptual understanding, and abstract thinking—qualities that continue to depend on human expertise. The most promising future lies in a collaborative partnership in which AI augments mathematical capability while mathematicians provide critical judgment, ethical oversight, and

creative insight. By balancing technological advancement with sound mathematical principles, AI can serve as a powerful catalyst for innovation, education, and scientific progress.

References

1. Russell, S., & Norvig, P. (2021). *Artificial intelligence: A modern approach* (4th ed.). Pearson.
2. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
3. Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.
4. Murphy, K. P. (2022). *Probabilistic machine learning: An introduction*. MIT Press.
5. Hastie, T., Tibshirani, R., & Friedman, J. (2021). *The elements of statistical learning* (2nd ed., corrected printing). Springer.
6. Devlin, K. (2012). *Introduction to mathematical thinking*. CreateSpace.
7. Polya, G. (1957). *How to solve it* (2nd ed.). Princeton University Press.
8. Silver, D., *et al.* (2016). Mastering the game of Go with deep neural networks and tree search. *Nature, 529*(7587), 484–489.
9. Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science, 349*(6245), 255–260.
10. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436–444.
11. Pearl, J. (2018). *The book of why*. Basic Books.
12. Knuth, D. E. (1997). *The art of computer programming, Volume 1: Fundamental algorithms* (3rd ed.). Addison-Wesley.

Bhumi Publishing, India
July 2026

A MEDALLION-ARCHITECTURE DATA ENGINEERING AND MACHINE LEARNING PLATFORM FOR SMART EV CHARGING NETWORK OPTIMIZATION

Nitin Kumar and Richa Avasthy
School of Science and Engineering, GNA University, Phagwara, Punjab, India
Corresponding author E-mail: gu-2022-4233@student.gnauniversity.edu.in, richa.avasthy@gnauniversity.edu.in

Abstract
The surge of EVs in cities has made the task of predicting charging demand and avoiding the unnecessary delays in lineups and localized grid congestion even more difficult. As electric vehicles (EVs) hit the road in cities, the challenge of predicting charging demand and avoiding unnecessary delays in lineups and localized grid congestion becomes even greater. This chapter introduces an AI-based platform to optimize the charging network of EVs, along with the underlying data engineering pipeline, and predictive modelling technique. The platform is developed with 1,320 charging sessions over 5 major cities in the United States and the data fed into the models is sequentially ingested, validated, and feature engineered. RF models were built for predicting station occupancy and identified potential overload scenarios with high R^2 and low MAE and good accuracy and F1-scores for the overload classifiers. Data cleansing processes resolved chronological inconsistencies and unlikely usage values, and the business-oriented layer reads out the utilisation of the chargers as well as the impact of external influences like traffic and weather. This end-to-end architecture is meant to be deployed on Databricks using Delta Lake and demonstrates the benefits of disciplined Data Engineering practice for accurate forecasting and reliable EV charging service delivery.

Keywords: EV Charging, Medallion Architecture, Data Engineering, Machine Learning, Random Forest, Python, Pandas, scikit-learn, Bronze-Silver-Gold Pipeline, Occupancy Prediction, Overload Detection, Delta Lake, Databricks, SQLite.

1. Introduction

In 2023, it is estimated that 40 million electric vehicles were sold globally, and by 2030 the IEA (2024) estimated that there would be more than 250 million on the road. More recent data from IEA (2025) confirm that this trend has been strengthened, if not accelerated: EV sales reached 20.7 million units in 2025, marking, for the first time in the history of the global automotive industry, a quarter of all passenger-car sales worldwide. Charging infrastructure has grown in step, as well: By the end of 2024, over 5 million publicly accessible charging points had been installed globally, with approximately 1.3 million new public charging points added in 2024

alone; that's a 30 percent year-on-year rise, according to IEA (2025). IEA (2025) reports that the number of publicly available charging stations is rising in the USA, increasing by 20 percent during 2024 to nearly 200,000, including those supported by federal programmes, e.g. the National EV Infrastructure Program. IEA (2025) also projects that with existing policies, public EVs will require to be increased by a factor of nine by 2030. The growth of this scale compounds the operational problem presented in this chapter - to haste the ache of local congestion even a network that is adequately provisioned can suffer from high congestion if the demand is not anticipated and distributed intelligently.

With the rising adoption of EVs, managing charging infrastructure becomes more complex. When demand is high, for example, on cold winter evenings or at large public events, such as too many drivers needing to use the stations at the same time, it can cause long queues and upset their drivers when stations are full, or too far away from where drivers are needed.

Some stations have high usage, causing internal degradation and safety issues, while others are not used at all. Previous attempts to solve these issues have had excellent outcomes in the controlled or simulated environments but have lacked success in the field.

Table 1: Global EV and Charging Infrastructure Growth Indicators, 2024–2025

Indicator	Value
Global EV sales, 2025	20.7 million units (>25% of new car sales)
Public charge points installed worldwide (end 2024)	> 5 million
New public charge points added in 2024	~1.3 million (+30% year-on-year)
U.S. public charging stock growth, 2024	+20% (to just under 200,000 points)
Projected global public charge points by 2030	~40 million (≈ 8-fold increase)
Required growth in charging infrastructure by 2030	Approximately ninefold (current-policy scenario)

Source: compiled from IEA (2025)

In an effort to alleviate the problem, the current study proposed an AI based platform for Optimization of Smart EV Charging Network across five main cities of United States such as Houston, Los Angeles, San Francisco, New York and Chicago integrating actual EV charging data with Traffic data and Weather data. The architecture of the platform is based on medallion, with bronze, silver and gold layers, implemented in Python with Pandas and scikit-learn. The raw data goes through bronze, silver and gold layers, with the silver layer cleaning the data and preparing it for the gold layer, which creates insights ready for decision.

This work is outstanding in the following ways:

An ensemble of rules that ensure the integrity and accuracy of the EV charging data.

Include time and traffic data in feature engineering to define charging demand.

Three RF models have been assessed using 1320 charging sessions, and this analysis has led to the identification of which modelling options are effective.

Solutions that include Power BI reports and dashboards created for easy interpretation by network operators.

Together, these developments pave the way for efficient management of the charging infrastructure, and help move toward a near-term viable solution to both EV drivers and station operators' problems.

2. Literature Review

The timing of electric vehicle charging is shown to affect grid load profiles with unmanaged nighttime charging found by Sioshansi and Denholm (2010) to be a source of increased peak-demand costs. This discovery is the impetus of ongoing research towards charging-time management.

Jain and Saihjpal (2022) investigated machine learning models to predict EV charging demand and found that Random Forest proved to be very effective, especially when the data was noisy. You will notice that their suggestion is to use Random Forest in their modelling approach described in Section 5 of this chapter.

Amara-Ouali *et al.* (2022) reviewed a wide range of studies on EV load forecasting and found that adding exogenous factors like weather and traffic can enhance the EV forecast accuracy by around 8 – 23 percent. This discovery inspires the incorporation of like variables in the current platform.

The bronze-silver-gold layering popularized by Databricks Engineering (2023) was used to organize data: raw data in bronze, cleaned data in silver, analysis-ready data in gold, rather than simpler schemes, as many EV data projects mix up data collection and remediation, confusing provenance.

A moving-average approach to overload warning proposed by Xu *et al.* (2020) is simple, but it may cause too many false alarms in respect of true demand surges. The supervised classification approach used in this work is instead based on a combined consideration of occupancy and charger type, traffic conditions, and thereby mitigates the risk of the misclassifications.

A systematic review by Marzbani *et al.* (2023) identifies tree-based ensemble methods, including Random Forest, as some of the most consistently competitive techniques for tabular energy-demand data (at the level of sessions or sessions per hour) with occasional benefits from deep-learning methods like LSTM and transformer-based models, but mainly for longer-horizon, high-frequency time-series forecasting, other than session-level occupancy or overload classification. The difference justifies the selection of the modelling base for the current website – Random Forest – in which the prediction targets are session level, not continuous load curves.

Kene and Olwal (2025) trained an electric vehicle charging session model using a variety of regression models (fine tree regression, linear regression, support vector regression, and neural networks), applied five-fold cross validation for prevention of overfitting and tested the model on a real-world data set of over one million electric vehicle charging sessions. Their excellent generalization of tree-based regressors to the real-world charging-session data adds external support to the modelling choices of this chapter, and shows that the 1,320-session dataset used here is relatively small compared to production-size deploys.

The proposed bronze-silver-gold layering scheme discussed in this study is inspired by the lakehouse architecture proposed by Armbrust *et al.* (2021), which unifies the low cost, flexibility, and scalability of data lakes with the transactional consistency and performance of data warehouses. The medallion data terms (bronze, silver, and gold) for raw, cleaned, and business-ready data, respectively, were later introduced as a practical implementation pattern, which emerged on top of this lakehouse concept, by Databricks Engineering (2023). This design, in which multiple stages are present, was chosen over single-stage pipelines because many EV data projects would otherwise combine data collection with data remediation, burying data lineage and making it difficult to diagnose data errors.

Concerning data quality, Goknil *et al.* (2023) performed a systematic literature review of data quality techniques for cyber-physical and IoT systems in Industry 4.0 environments, describing typical data quality problems – such as missing values, out-of-range readings and inconsistent timestamps – and related data quality remedies. They categorize defect types that are in close alignment with the validation rules that are currently being introduced in their silver layer on the present platform (Section 4.2) and their validation review confirms that validation rules, and explicit quality flags, not record deletion is in agreement with recommended practice in industrial and IoT-adjacent data pipelines.

3. Dataset and Problem Formulation

3.1 Dataset Description

The data consists of 1,320 EV charging events from 2024 in five major cities in the U.S. — Houston, Los Angeles, San Francisco, New York and Chicago. Each session record is broken down into four types of information. The first category refers to the vehicle and the user, such as the type of vehicle, battery size and type of user. The second category is a description of the session that includes the charging station identifier, the type of charger, and the session start time, end time, and energy transferred, as well as the charging rate.

The third one provides general information about the charging station such as location and rated capacity. The fourth category combines environmental conditions (temperature, humidity and weather condition) and local traffic conditions as well as event-driven crowd levels nearby. These categories give a complete profile of each charging session.

Table 2: EV Charging Session Dataset Summary Statistics

Attribute	Value
Total session records	1,320
Date range	January 1, 2024 – December 31, 2024
Geographic coverage	5 U.S. cities (Houston, Los Angeles, San Francisco, New York, Chicago)
Unique charging stations	1,016
Charger types	DC Fast Charger (32.6%), Level 1 (34.8%), Level 2 (32.7%)
User types	Commuter (36.1%), Long-Distance (33.1%), Casual (30.8%)
Mean energy consumed	42.65 kWh (σ = 21.84 kWh)
Mean session duration	2.27 hours (σ = 1.06 hours)
Mean session cost	\$22.55 (σ = \$10.75)
Peak-hour session share	37.5% of all sessions
Records passing quality validation	1,320 / 1,320 (100%)

3.2 Problem Formulation

This study, aims to solve three major forecasting issues. The first is Station Occupancy Forecasting which uses past session data to forecast the level of congestion at a station, allowing operators to forecast before the station actually becomes congested. The second, waiting-time estimation, predicts the amount of time a driver is expected to have to wait when they arrive, which will enable a driver to determine whether a different station is more suitable and provide a real time estimate of wait time.

The third problem is called overload-risk classification which assesses the risk of the station becoming full. This is necessary for safety as well as planning maintenance, as an automatic warning message or load management intervention can reduce overuse based on these predictions.

4. System Architecture

4.1 Medallion Pipeline Design

It is designed in three stages in order to shed the raw data and map them to analysis-ready output. This separation enables data lineage-tracing and individual re-execution of each stage without loss of data, as each stage runs separately and the pipeline could be re-executed end-to-end if needed.

In the bronze layer, raw CSV data is ingested using a routine that derives the schema automatically from Pandas library. When it arrives, four metadata elements are added, namely the data source, ingestion time, a unique record identifier, and a preliminary quality flag. At this

stage, the data values are kept without being changed. The output from this is saved to data/bronze/bronze_ev_charging_events.csv.

The silver layer is based on seven step transformation pipeline. First up, numeric and date type data are fixed. Then unneeded spaces are stripped away and empty strings are replaced with explicit "missing" values. Then, unnecessary spaces are eliminated and blank strings are made into actual "missing" values. Categorical values that are missing are filled with “Unknown” and numeric values that are missing are filled using representative summary statistics. Seven validation rules are then applied to determine the accuracy of the records and then any duplicate records are removed. Derived metrics like energy used per hour or effective charger capacity are then calculated and temporal features like hour of day and weekday/weekend indicators are derived. This gives a total of 1320 rows and 56 columns of data, which is silver.

The gold layer provides analytics ready tables from the silver layer data for business intelligence consumption in Power BI, its tables are Station performance, Station traffic, Weather impact and Charging demand. The idea behind this design is to make dashboard construction less complicated, and not dependent upon deep SQL skills.

Table 3: Platform Layer Architecture Summary

Layer	Primary Technology	Input	Output	Purpose
Bronze	Python / Pandas	Raw CSV (44 cols)	Annotated CSV (48 cols)	Raw landing + ingestion audit trail
Silver	Python / Pandas	Bronze CSV	Typed CSV (56 cols)	Cleaning, validation, feature engineering
Gold	Python / Pandas	Silver CSV	4 analytics CSVs	Business aggregations for BI
ML Features	Python / Pandas	Silver CSV	Feature matrix CSV	Supervised learning targets
Models	scikit-learn	Feature matrix	3 .pkl artifacts	Occupancy / wait-time / overload models
Serving	SQLite / Streamlit	Gold + Models	Interactive app	Operator and user interface

4.2 Data Quality Framework

The silver layer enforces seven validation rules. Any record that violates a rule is assigned quality_is_valid = False along with a corresponding quality_issue label; the record is retained in the silver dataset with the flag preserved, allowing downstream users to decide whether to include it depending on their analytical requirements.

Rules 1 and 2 require that the charging start time and end time be non-null. Rule 3 requires that the end time not precede the start time, which would otherwise indicate a logging error. Rule 4

requires that energy consumed be zero or greater. Rule 5 requires that charging duration be strictly greater than zero. Rules 6 and 7 require that traffic_score and humidity_percent, respectively, fall between 0 and 100, guarding against implausible operational values.

In the dataset used for this study, all 1,320 records satisfied these rules, which is consistent with a simulated rather than a real-time data source. Nonetheless, the framework is designed to accommodate authentic operational data, which typically exhibits non-trivial defect rates, consistent with the range of data-quality issues catalogued for comparable IoT and cyber-physical deployments by Goknil *et al.* (2023).

4.3 Feature Engineering

The machine learning feature matrix comprises eighteen numerical features and six categorical features. Numerical features include energy consumption, charging session duration, ambient temperature and road congestion, among others. The categorical features — charging station location, vehicle model, charger type, user type, weather condition and traffic level — are one-hot encoded to improve model performance.

Feature preprocessing is implemented through a ColumnTransformer-based pipeline. Numerical features are first imputed using a median-strategy SimpleImputer and then standardized. Categorical features are imputed using the most frequent category prior to encoding. This design preserves preprocessing consistency and allows the entire transformation to be serialized, eliminating the need for manual preprocessing at inference time.

5. Machine Learning Models

5.1 Model Architecture

Three Random Forest models, following the approach introduced by Breiman (2001), were trained for distinct tasks, using the scikit-learn implementation described by Pedregosa *et al.* (2011). For occupancy prediction and waiting-time estimation, a RandomForestRegressor is used with 160 estimators, random_state = 42 and min_samples_leaf = 3; setting min_samples_leaf to 3 provides regularization by preventing individual trees from memorizing single-session patterns.

The third model, used for overload classification, is a RandomForestClassifier with 180 estimators, class_weight = ‘balanced’, random_state = 42 and min_samples_leaf = 3. The balanced class-weight setting compensates for the relative rarity of overload events in the dataset.

Each model is wrapped in a scikit-learn Pipeline that combines the preprocessing ColumnTransformer with its estimator, allowing the complete pipeline to be serialized as a single .pkl artifact. This design enables new session records to be scored directly from raw feature values via model.predict().

Data were partitioned using an 80/20 train-test split with random_state = 42. For the classification task, stratified sampling was used to preserve class balance in the test set.

5.2 Model Performance

Table 4 reports evaluation metrics computed on the held-out 20 percent test split for all three models.

Table 4: Machine Learning Model Performance Metrics (20% Test Split)

Model	Task	MAE	RMSE	R^2	Accuracy	F1 Score
Occupancy Prediction	Regression	0.5335	0.8092	0.9955	—	—
Waiting Time Estimation	Regression	0.1709	0.2592	0.9993	—	—
Overload Risk Detection	Classification	—	—	—	89.39%	88.52%

The occupancy model does a good job of explaining the variance in occupancy estimates, with an R^2 of 0.9955. The waiting-time model is even better, with an R^2 of 0.9993 and an error margin of about 0.17 minutes (or about 11 seconds), which is not likely to have a significant effect on any routing decisions made. The overload classifier also shows satisfactory performance with an accuracy of 89.39%, and an F1-score of 88.52%.

These R^2 and performance values should be noted, however, as being generated from synthetic data: the likelihood of these R^2 and performance directly transferring to real-world deployment should not be expected. The real-world EV session data is usually noisier due to measurement error, inaccuracies in GPS and unexpected grid outages. Reported results should therefore be considered to be an indication of the capabilities of the model under favourable conditions only and not necessarily a guarantee of the same capabilities in the real world.

6. Gold Analytics Layer and Dashboard

6.1 Analytics Table Design

Aggregations in silver layers are used to create four gold-level tables. The station_performance table is an aggregation of the data from a station, including session counts, energy consumption in kWh, average charging time, rates, costs, traffic scores, peak-hour session counts, temperature, precipitation, and additional metadata, all by station_key. This table allows for comparison across stations and can be used to help plan for capacity. The traffic_analytics table breaks down the sessions into the city, date, hour, and charger type, giving a mean state of charge at the start and end of each session, mean distance traversed, and mean energy expended per segment. This helps identify the differences between high and low demand periods, to inform tariff design.

The weather_impact_analytics table aggregates sessions by weather conditions and calculates the average session duration, energy consumption and traffic score for each weather condition. Clear-sky conditions make up the biggest percentage at 45.7 percent, followed by partly cloudy at 34.8, light rain at 13.8 and snow at 4.3 percent, with heavy rain representing 1.4. The sessions conducted during wet weather tend to be longer, as do those conducted at colder temperatures than the warmer ones, that is, according to the findings of previous studies, a decrease in potential battery capacity with colder temperatures. The charging_demand_analytics table contains the

hourly charging demand in a city per charger type. Together with the overload-risk model this table helps operators to predict the hot spots and provide staffing and resources in time for those hot spots.

6.2 Power BI Dashboards: The Case for a Standardization Effort.6.4 Power BI Dashboard Design

The Power BI report is based on the gold-layer tables and is split into five pages. The Network Overview page displays critical performance metrics such as number of sessions, average session cost, number of sessions exceeding overload-risk percentage and total energy delivered in MWh. The Station Performance Map page is a filled map, with the size of the marker representing the total energy delivered in kWh, both plotted using latitude/longitude as the key.

The Demand Heatmap page categorises the charging-demand data into a grid with rows sorted by day of week and columns by hour of the day, and colour shading showing the number of sessions. The Weather and Traffic Impact page displays side by side bar charts based on the weather and traffic analytics tables. The ML Insights page provides the forecast output in tables and gauge visualizations, based on the occupancy model and overload-risk classifier.

7. Results and Discussion

The full pipeline was able to transform 1,320 raw session records in less than 30 seconds in a local Python environment, yielding 1,016 unique records for each station in the gold table, and a fully typed set of 1,320 silver records. The quality validation pass rate is 100 percent to ensure that the synthetic data set is internally consistent, and the validation rule set is intended to be the correct set of rules for detecting the corresponding categories of error that might occur in real-world data.

An important finding has been that over 37.5 percent of charging has taken place at peak hours, within the morning commute window (6–9 a.m.) and the evening commute window (5–8 p.m.) – a similar pattern of commuter behaviour. The identification of these distinct time windows for increased overload risk has enabled operators to identify the times in which they may be able to reduce the risk of overload by increasing their capacity, but without significantly changing their off-peak pricing.

The distribution of chargers by type is also relatively balanced (DC Fast: 32.6%, Level 1: 34.8% and Level 2: 32.7%), indicating a diversified network, with each type playing a unique role in operation: DC Fast chargers, for instance, demand significantly higher power levels than Level 1 or Level 2 chargers. User behaviour also varies meaningfully by segment: commuters connect predictably and charge for comparatively short durations; long-distance travellers require larger energy transfers, typically topping up after extended highway travel; and casual drivers exhibit more irregular arrival patterns. This heterogeneity strengthens the robustness of the models with respect to overload-risk conditions.

7.1 Benchmarking Against External Literature

Comparisons with published results should be compared with caution, because the objective of prediction and the extent of the data sets varied in different studies, as do the protocols for evaluating the results. With this nuance, Table 5 compares the performance achieved in this work to two study points from the recent literature that survey ensemble and deep-learning approaches to EV energy-demand prediction (Marzbani *et al.,* 2023), and over 1 million real-world charging sessions, compare tree-based and other regressors (Kene and Olwal, 2025).

Table 5: Indicative Comparison with External Benchmarks

Study	Dataset Scale	Method	Reported Outcome
This chapter	1,320 sessions (synthetic, 5 U.S. cities)	Random Forest (regression + classification)	R^2 up to 0.9993; F1 = 88.52%
Kene and Olwal (2025)	> 1,000,000 sessions (real-world)	Fine tree, linear, SVM, neural network regressors	Tree-based regressors competitive with or superior to other regressors
Marzbani *et al.* (2023)	Systematic review, multiple datasets	Ensemble methods vs. deep learning	Ensemble/tree methods competitive for tabular, session-level targets

The results in this study seem to concur with the trend observed in the literature for tree-based ensembles (see Section 5.2), where the values of R^2 and F1 obtained in the present study appear to be broadly comparable, but just as in the previous section, the absolute values reported here are likely to be an upper bound on what may be achieved once the pipeline is evaluated against operational data, which is typically a much larger corpus of data.

Table 6: Local Prototype vs. Databricks Enterprise Architecture Mapping

Dimension	Local Prototype (Current)	Databricks / Delta Lake (Target)
Storage format	CSV files	Delta Parquet with versioning
Processing engine	Pandas (single-node)	PySpark (distributed cluster)
Transformation tracking	Script version control	Delta Lake transaction log
Serving layer	SQLite + Streamlit	Redshift / Synapse + Power BI
ML training	scikit-learn (local)	MLflow-tracked experiments
Pipeline scheduling	Manual / run_pipeline.py	Databricks Workflows / Airflow
Scale	1,320 sessions	Millions of sessions per day

8. Limitations and Future Work

There are three restrictions that warrant special mention. First, the data is completely synthetic and covers just one calendar year, but real EV charging data changes yearly based on vehicle types, charging station hardware, and electricity pricing models. Therefore, the relevance of the model is going to need to be retrained on a regular basis in order to keep up with these continuous changes.

Second, the information about the location of charging stations in this study cannot be matched up with an authoritative GPS track record. Ideally future implementations should utilize trusted resources like the Open Charge Map API or the U.S. Department of Energy's Alternative Fuels Data Center to ensure accurate coordinates to power station locations, as map and spatial data is critical for reliable mapping and spatial analysis.

Thirdly, the current serving layer uses SQLite, which does not scale well when accessed concurrently via the API. If scaling to provide more requests and enable integration with live navigation systems is desired, a production deployment might profit from moving to a system, like PostgreSQL or Amazon Redshift, that can support such requirements.

Future planned work includes ingestion of near real-time data to reduce alert latency, the integration of dynamic price for electricity into the modelling framework, and the investigation if there is a difference in performance between a single cross-city model, or city-specific models. Further work also examines other modelling techniques for increased predictive accuracy, the adoption of migration to Delta lake for increased fault tolerance, and operational continuity.

Conclusion

In this chapter, a data engineering and machine learning framework for improving the performance of smart EVs charging networks has been described, which is based on a bronze-silver-gold pipeline, viable for deployment on Databricks. The bronze stage is data at its raw stage; the silver stage is where cleaning and validation work takes place and the gold stage is data received for business facing analytics.

The silver-layer quality-assurance element detects typical error types associated with data from the IoT. Based on a combination of session count, time of occurrence, weather and traffic conditions, three Random Forest models were trained and evaluated with high predictive power – occupancy prediction ($R^2 = 0.9955$), waiting-time estimation ($R^2 = 0.9993$), and overload-risk detection (F1 = 88.52%). The gold layer offers four tables that are ready for analyses and can be directly connected to Power BI, allowing the users to transform the raw data into actionable insights without a deep understanding of SQL. By importing data as CSV files and serving models via a Streamlit application, the platform showcases how a small development team can provide a unified solution, built with structures and principles of disciplined data engineering, instead of random coding.

As usage of EVs is estimated to reach a hundred million in the next few years, robust systems will be needed to manage charging-station demand. This chapter provides an architecture that has been proven to directly tackle at least that challenge.

References

1. Amara-Ouali, Y., Goude, Y., Massart, P., Poggi, J. M., & Yan, H. (2022). A review of electric vehicle load open data and models. *Energies, 15*(6), 2194.

2. Armbrust, M., Ghodsi, A., Xin, R., & Zaharia, M. (2021). Lakehouse: A new generation of open platforms that unify data warehousing and advanced analytics. *Proceedings of the 11th Conference on Innovative Data Systems Research (CIDR), 8*, 28.
3. Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.
4. Databricks Engineering. (2023). *Databricks Medallion Architecture: Best practices for building data lakehouses*. Databricks Technical Documentation. https://www.databricks.com/glossary/medallion-architecture
5. Goknil, A., Nguyen, P., Sen, S., Politaki, D., Niavis, H., Pedersen, K. J., Suyuthi, A., Anand, A., & Ziegenbein, A. (2023). A systematic review of data quality in CPS and IoT for Industry 4.0. *ACM Computing Surveys, 55*(14s), 1–38.
6. International Energy Agency. (2024). *Global EV outlook 2024: Moving towards increased affordability*. IEA Publications. https://www.iea.org/reports/global-ev-outlook-2024
7. International Energy Agency. (2025). *Global EV outlook 2025: Expanding sales in diverse markets*. IEA Publications. https://www.iea.org/reports/global-ev-outlook-2025
8. Jain, A., & Saihjpal, V. (2022). A comparative review of machine learning techniques for EV charging demand forecasting. *Energy Reports, 8*, 6885–6900.
9. Kene, R. O., & Olwal, T. O. (2025). Data-driven modeling of electric vehicle charging sessions based on machine learning techniques. *World Electric Vehicle Journal, 16*(2), 107.
10. Kumar, N. (2025). *AI-powered smart EV charging network optimization platform—Source code, ETL scripts, ML training pipeline, and Power BI dashboard specifications*. GNA University, Phagwara. Project Repository.
11. Marzbani, F., Osman, A. H., & Hassan, M. S. (2023). Electric vehicle energy demand prediction techniques: An in-depth and critical systematic review. *IEEE Access, 11*, 96242–96255.
12. McKinney, W. (2022). *Python for data analysis: Data wrangling with pandas, NumPy, and Jupyter* (3rd ed.). O'Reilly Media.
13. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830.
14. Sioshansi, R., & Denholm, P. (2010). The value of plug-in hybrid electric vehicles as grid resources. *The Energy Journal, 31*(3), 1–23.
15. Xu, Y., Milanovic, J. V., & Djokic, S. Z. (2020). Reliability assessment of distribution networks with high penetration of electric vehicles. *IEEE Transactions on Smart Grid, 11*(4), 3151–3160.

THE ROLE OF ARTIFICIAL INTELLIGENCE IN STRENGTHENING THE BLOCKCHAIN ECOSYSTEM FOR ORGANIZATIONS

Mansi Singh, Navkaran Dewett and Kirandeep Kaur
GNA University, Phagwara, Punjab

Introduction

Blockchain is a technology which possess a huge potential to various industries including education, healthcare, supply chain management and finance. Its secure, decentralized and transparent nature makes it appropriate for the organizations where security, trust and efficiency are main concerns. Integration of blockchain in education sector has the potential to improve the creditability of educational process. A transparent platform can be created to track and verify student's achievements using blockchain technology. In contrast to the various benefits, several challenges need to be addressed to achieve the full benefits of blockchain. Challenges include interoperability, data privacy, accessibility and scalability. Before proceeding for blockchain in educational institutes, one should check the diverse equipment's for implementation and it is crucial for the success of this innovative technology [1].

Importance of Blockchain Ecosystem for Organizations

Below are some of the reasons why an ecosystem is an essential solution for organizations.

- **Integration Across Enterprise Boundaries:** Blockchain technology enables different organizations to create a joint database for their shared system. The practice of separate organization database management creates obstacles for organizations to transfer data between their systems. With blockchain technology, all users can access the latest system information at any time. The system eliminates the requirement for intermediaries while reducing duplicate activities and establishing trust through shared data access. The result enables organizations to work together for better results in product creation through improved trust and coordination methods to develop new products and services together which they could not establish before [2].
- **Allow Organizations to Overcome Traditional Mindset:** Business operations use established business systems which require organizations to follow strict procedures that depend on central power and use extended time to complete tasks. Organizations need to replace their existing traditional business systems with decentralized systems that blockchain technology enables. The system establishes transparent operational processes which use smart contract technology to create automated workflows while enabling direct communication between different system users. The solution enables organizations to develop flexible operational methods which enhance their operational efficiency.

Companies need to stop using traditional supply chain operational methods so they can become more adaptable and lower their operational expenses while discovering new digital business methods [3].

How can the Blockchain Ecosystem Benefit Organizations

Blockchain technology is changing the world around us and at the inner level of companies and business processes. Starting from the most trivial daily activities to the big changes and the deep structural ones in the economy of the countries. Due to its structure, a traditional database may be subject to a number of risks and vulnerabilities. On the other hand, blockchains provide a transparent and secure system in a decentralized way and therefore minimize the risk factors to the lowest level. Here are the key advantages that companies can gain through blockchain technology and ecosystems.

- **Distributed:** There is no centralized person such as a Database Administrator or single point of failure that could be a bottleneck to growth. Therefore, the Distributed option brings about new advantages for employees and customers in an organization.
- **Remodel Complete Workflow:** The complete process of workflow transformation extends from blockchain technology, which alters all organizational work processes through its implementation. The system enhances task execution through its automated features that deliver both secure operations and transparent process management. The solution improves project execution through enhanced team collaboration and decreased mistakes, and accelerated project delivery times [4].
- **Smart Contracts:** Smart contracts function as software applications which execute contract terms automatically when specific contract conditions get fulfilled. Smart contracts enable efficient information sharing while eliminating the need for middlemen who would lengthen the time required to complete essential tasks within extensive workflows. The system records transactions in a sequential order which establishes permanent unchangeable records that remain protected against unauthorized modifications.
- **Improve Ongoing Operations:** When the blockchain ecosystem can enhance ongoing operations in ways that current solutions cannot, it should be a viable option for an organization. For instance, the blockchain structure's ability to validate data among nodes or peers as quickly as possible could be advantageous for a company where data sharing between different departments and individuals happens more frequently.
- **Facilitate Cross-Enterprise Collaboration:** Blockchain technology enables organizations from separate businesses to establish partnerships through its secure operational framework. The system enables all users to access shared information because its design prevents any single organization from controlling data access. The

organizations benefit from traceability and flexibility features which create confidence between parties and streamline their joint work.

- **Transparency:** Blockchain transparency enables all network participants to access complete transaction records for every network transaction. The system registers each transaction in a format which prevents users from changing or concealing specific details. The system makes it impossible to cheat because all user actions require permanent tracking from their starting point, which creates user trust.
- **Consensus:** Consensus functions as the method through which blockchain network members reach agreements about valid transaction records. The network performs data verification through specific procedures before it can add a new block to the system. The system records only authentic transactions which prevent any attempt at fraudulent or duplicate transactions from happening [5].

Participants in Blockchain Ecosystem

The blockchain ecosystem depends on its participants as its most essential elements. The blockchain ecosystem includes five main participant groups who function as its essential components:

- **Leaders:** Leaders oversee all activities on the blockchain network. These are the groups that envision the future of the blockchain ecosystem and its commercial principles. They are the project's creators and the primary recipients of the ecosystem's efforts.
- **Core Group:** The core group consists of organizations that lead or actively participate in network operations to determine the network's operational activities. They take charge of all activities which include operations and maintenance and system management to oversee the blockchain ecosystem [6].
- **Active Participants:** The active participants of the network include primary members who handle their duties to support workflow and governance while managing data.
- **Users:** They don't get any responsibility for the active management of the network. The ecosystem requires users to access their personal data while they obtain network advantages. The team handles everything from active management to process optimization and system operation of the blockchain ecosystem.
- **Third-Party Service Providers:** They serve as important network members who deliver multiple types of services to the network. The company offers required services through its infrastructure and application services, and IT support services at established fees[7].

Components of Blockchain Ecosystem

The important logical components in the blockchain ecosystem are summarized in Fig. 1 and are elaborated in this section.

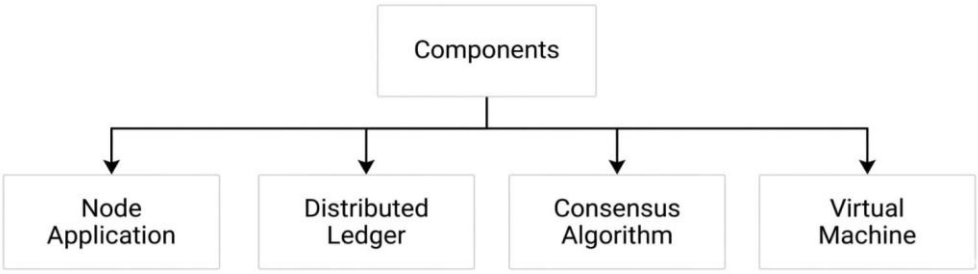


Figure 1: Components of Blockchain Ecosystem

- **Node Application:** A node application is a specialized software program that must be installed on a computer to participate in a blockchain ecosystem. Any device with an internet connection can become part of the network by downloading and running this application. The node application establishes a blockchain connection for users after its installation which enables them to perform various transaction functions from sending to receiving and from validating to storing transactions. The application enables users to operate their computer as a node which helps sustain the distributed network. Through this application, users can actively participate in the blockchain ecosystem and interact with its data[8].
- **Distributed Ledger:** A distributed ledger is the core data structure of a blockchain that records all transactions across multiple nodes in the network. The ledger spreads its contents through all participating nodes which create synchronized copies to achieve transparent and uniform data access. Each node keeps its own ledger version, which receives regular updates for every new transaction entry. The decentralized storage system protects the ledger from unauthorized changes because any alteration needs to be implemented on every connected node. Users can see the distributed ledger content for that given blockchain ecosystem through their node application.
- **Consensus Algorithms:** Consensus algorithms are the set of rules and mechanisms used by blockchain networks to achieve agreement among all participating nodes on the validity of transactions and the current state of the ledger. The algorithms establish a method through which all blockchain nodes can reach agreement about which information represents the actual state of the network. Different blockchain ecosystems use different consensus methods, such as Proof of Work (PoW) and Proof of Stake (PoS), each with its own way of validating transactions and selecting nodes to add new blocks. The algorithms create a secure environment which helps prevent fraud while establishing trust in the decentralized network system [9].
- **Virtual Machine:** A virtual machine in a blockchain is a software-based environment that simulates a computer system and executes programs or smart contracts in a secure and isolated manner. The system provides a runtime environment which processes

programming language instructions through consistent execution across all network nodes. The virtual machine connection operates within node application software, which enables identical code execution across all nodes to create matching output. The Ethereum Virtual Machine executes smart contracts through its deterministic execution process, which guarantees identical results across all Ethereum network nodes. This mechanism enables all network members to achieve identical results which preserves the blockchain system's trustworthiness and operational integrity. The four blockchain components are summarized in Table 1.

Table 1: Comparison of different Blockchain components

Basis	Node Application	Distributed Ledger	Consensus Algorithms	Virtual Machine
Definition	Software that connects a user‘s computer to the blockchain network	Shared database that stores all blockchain transactions	Rules used to achieve agreement among nodes	Software environment that executes smart contracts
Purpose	To enable participation in the blockchain ecosystem	To record and maintain transaction data	To ensure all nodes agree on a single version of truth	To run and execute blockchain programs
Nature	Application/Software	Data structure	Protocol/Mechanism	Execution environment
Role in Blockchain	Acts as an entry point to the network	Maintains the history of all	Validates and confirms	Executes code and smart
		transactions	transactions	contracts
Control	Controlled by the user running the node	Distributed across all nodes	Defined by the blockchain protocol	Embedded within node software
Dependency	Required to access blockchain	Managed by node applications	Implemented inside node applications	Works along with node application
Example	Bitcoin node software, Ethereum client	Blockchain ledger	Proof of Work (PoW), Proof of Stake (PoS)	Ethereum Virtual Machine
Functioning	Connects, validates, and stores blockchain data	Continuously updated across nodes	Selects valid transactions and adds blocks	Executes instructions in a secure environment

Steps in Forming a Blockchain Ecosystem:

The steps of forming a blockchain ecosystem consists of steps given in Fig. 2 and are elaborated below.

Installation of Node Application: A blockchain ecosystem starts to develop when a node application is installed because this software enables computers to establish blockchain network connections. Any device with sufficient computational capability and internet access can install this application and become a node in the network. The node application enables users to interact with the blockchain through its user interface, which allows them to connect with other nodes in the network. The installed node starts to connect with other nodes to download required protocols which allow it to send and receive and validate transactions [9]. This process enables a standard computer to function as a decentralized network participant, which helps maintain blockchain network operations and security.

Creation and Synchronization of Distributed Ledger: The distributed ledger is created or synchronized after the node application installation is complete. The distributed ledger functions as a common database which holds all transaction records from the blockchain network since its inception. When a new node joins the network, it downloads the existing ledger from other nodes and synchronizes its copy to match the current state of the blockchain. This process ensures that every node maintains an identical and current ledger version. The decentralized ledger structure removes the requirement for central authority control while providing all users with transaction verification capabilities through their independent verification process. Multiple nodes store the ledger information which results in a system that protects against data alterations and complete data loss.

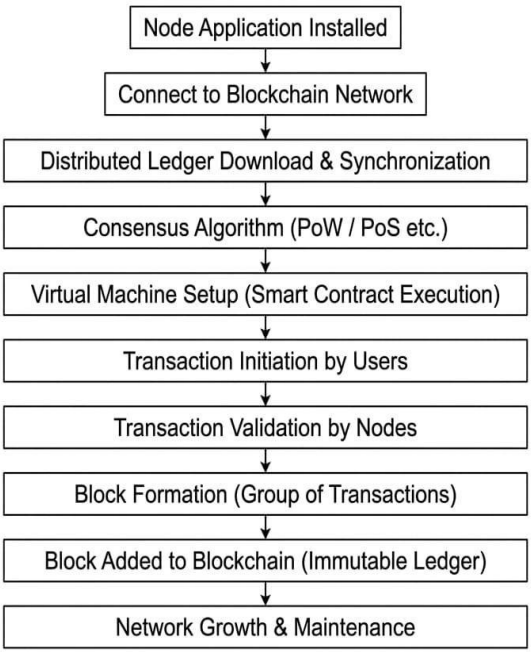


Figure 2: Steps of formation of blockchain ecosystem

Implementation of Consensus Algorithm: Consensus algorithm implementation creates the foundation for blockchain ecosystems to achieve trust and shared agreement among participants. The network requires consensus mechanisms because it lacks central authority to validate transactions which establishes agreement among nodes regarding transaction authenticity and ledger status. A consensus algorithm establishes the guidelines which nodes must follow to verify transactions and create new blockchain blocks. Different blockchain ecosystems may use different consensus models, such as Proof of Work (PoW), where nodes solve complex mathematical problems, or Proof of Stake (PoS), where validation rights are assigned based on ownership of cryptocurrency. The network operates with consistent functionality through these mechanisms while they eliminate the possibility of fraudulent activities which include double spending. Consensus algorithms create a standardized agreement which protects blockchain systems from unauthorized access while guaranteeing system integrity and operational reliability [6].

Execution Environment through Virtual Machine: A virtual machine serves as a crucial component for developing a blockchain ecosystem because it creates a controlled environment where programs and smart contracts can be executed securely. A virtual machine provides a simulated environment which allows code execution without depending on specific hardware requirements. In blockchain systems, this ensures that all nodes execute the same instructions in a consistent and deterministic manner, producing identical results. The Ethereum Virtual Machine in the Ethereum network enables all nodes to execute smart contracts through its Ethereum Virtual Machine. The virtual machine enables automation through its function which lets developers create decentralized applications (DApps) that function exactly as they were designed to operate. The blockchain ecosystem achieves greater operational capacity through this step, which brings new capabilities to its existing system.

Transaction Processing and Block Formation: Users begin their transaction process after establishing the blockchain ecosystem infrastructure. The network receives transactions which participating nodes use to verify them according to the rules established in the consensus algorithm. The system creates blocks from validated transactions by grouping them together. Each block contains a set of transactions, a timestamp, and a cryptographic hash of the previous block, which links it to the chain. The blockchain maintains its immutability because this chaining method requires modifications to all subsequent blocks after any single block has been changed at every network node. The ledger receives a permanent record of the block after it has been confirmed and included in the blockchain. This process creates the basis which enables the ecosystem to record transactions through secure methods [5].

Network Growth and Maintenance: The process of creating a blockchain ecosystem concludes with the expansion of its network and the ongoing maintenance of its operations. The network

achieves higher levels of security through decentralized operation because additional nodes join the system which protects it from attacks. Continuous node activity enables the system to validate transactions while maintaining an updated distributed ledger. The system requires maintenance activities which include software updates and protocol improvements and security upgrades to handle current system needs and upcoming technological improvements. The ecosystem develops into a stronger and more scalable system through community involvement and governance systems and innovative technological advancements. The blockchain network achieves both long-term operational efficiency and sustainability through this process [2].

References

1. El Koshiry, A., Eliwa, E., Abd El-Hafeez, T., & Shams, M. Y. (2023). Unlocking the power of blockchain in education: An overview of innovations and outcomes. *Blockchain: Research and Applications, 4*(4), 100165.
2. Khalilov, M. C. K., & Levi, A. (2017). A survey on anonymity and privacy in Bitcoin-like digital cash systems. *IEEE Communications Surveys & Tutorials, 20*, 2543–2585.
3. Kistaubayev, Y., Mutanov, G., Mansurova, M., Saxenbayeva, Z., & Shakan, Y. (2023). Ethereum-based information system for digital higher education registry and verification of student achievement documents. *Future Internet, 15*(1), 3.
4. Jagwani, P., Singh, V. B., Agrawal, N., & Tripathi, A. P. (2023). Blockchain technology and software engineering practices: A systematic review of literature using topic modelling approach. *International Journal of System Assurance Engineering and Management, 14*(Suppl. 1), 1–17.
5. Tsybulsky, D. (2020). Digital curation for promoting personalized learning: A study of secondary-school science students' learning experiences. *Journal of Research on Technology in Education, 52*(3), 429–440.
6. Chikezie, U., Karacolak, T., & Do Prado, J. C. (2021). Examining the applicability of blockchain to the smart grid using proof-of-authority consensus. In *2021 IEEE 9th International Conference on Smart Energy Grid Engineering (SEGE)* (pp. 19–25). IEEE.
7. Adam, D., Matellini, D. B., & Kaparaki, A. (2023). Benefits for the bunker industry in adopting blockchain technology for dispute resolution. *Blockchain: Research and Applications, 4*(2), 100128.
8. Jha, R. (2023). Survey on blockchain technology and security facilities in online education. In *Recent advances in blockchain technology: Real-world applications* (pp. 131–154). Springer International Publishing.
9. Chivu, R. G., Popa, I. C., Orzan, M. C., Marinescu, C., Florescu, M. S., & Orzan, A. O. (2022). The role of blockchain technologies in the sustainable development of students' learning process. *Sustainability, 14*(3), 1406.

EXAMEDGE: LEARNING ANALYTICS PLATFORM WITH AI FOR COMPETITIVE EXAMINATION: EVIDENCE FROM PUNJAB, INDIA

Manoj Kumar, Ashok Kumar, Ishfaq Majeed* and Neha Bagga*
GNA University, Phagwara, Punjab, India
Corresponding author E-mail: ishfaq.majeed@gnauniversity.edu.in,
neha.bagga@gnauniversity.edu.in

Abstract
The main goal of this project is to create ExamEdge, a tool that uses artificial intelligence to help students get ready for big exams like GATE. ExamEdge is designed to help students with things they do not know about topics and it also looks at how exams are changing and how people study. ExamEdge was made using computers, data and Artificial Intelligence. They got GATE question papers and used computer programs to sort questions by subject and see what topics are popular. The platform uses Microsofts Phi-3 Mini Large Language Model with Ollama so it can give step by step explanations on computers with not a lot of memory, like 8 GB of RAM. The platform successfully classified 94.3% of GATE questions accurately. The analysis revealed significant changes in topic weightage, such as Computer Organization increasing to 14% of the examination, while Algorithms accounted for only 5%. The current version of ExamEdge is mainly for people who are getting ready for the GATE examination. It looks at question papers to help with this. In the future ExamEdge can be made to help with competitive examinations. It can get updates on what's being taught in real time. ExamEdge presents a unique approach to competitive exam preparation by integrating AI-powered assistance, historical question paper analysis, and topic trend identification within a single platform.

Keywords: Digital Transformation, Managerial Digital Literacy, Operational Efficiency, SME Performance, Structural Equation Modelling.

1. Introduction

Exams in India are quite competitive. Each year, millions of engineers attempt various national level exams for admission to their M.Tech and Ph.D. programs at Indian Institute of Technology and National Institute of Technology. These exams are not only very prestigious but also help secure placements in various public sector undertakings such as Bharat Heavy Electricals Limited, Gas Authority of India Limited, National Thermal Power Corporation, and Oil and Natural Gas Corporation. The Graduate Aptitude Test in Engineering exam can be considered the engineering aptitude test where your skills are evaluated in terms of technical knowledge, analytical ability, and problemSolving capability. It is an extremely important exam taken by many students. Your marks in Graduate Aptitude Test in Engineering are used both for securing

admission into engineering colleges and for getting jobs in various PSU. In India, a large number of students strive to secure admission in Indian Institute of Technology and National Institute of Technology. The Graduate Aptitude Test in Engineering exam is one of the milestones for engineers. Passing Graduate Aptitude Test in Engineering can open doors.

These exams have an impact on what people do in their careers. A lot of people who are taking these exams do not have the information they need to prepare for Graduate Aptitude Test in Engineering. The main problem is that people do not have access to the information they need for Graduate Aptitude Test in Engineering. Old Graduate Aptitude Test in Engineering exam papers are available on university websites and online forums. They are in a format that's hard to work with. They are usually in Portable Document Format files that are not organized in a way that makes sense. This makes it difficult to analyse the information in these files using computers. So people have to rely on their ideas about what to study for Graduate Aptitude Test in Engineering or they have to use study materials from coaching institutes.

2. Literature Review

A. The Evolution, Structure and Taxonomy of the GATE Examination

In 1984, the Graduate Aptitude Test in Engineering (GATE) was introduced by the Indian Institute of Science and the seven IITs as a means to assess student understanding of engineering. Since the test began, it has evolved from being used as a means of entry to school to being used by corporations to assess engineering as a means of validating candidate qualifications.

The GATE test has seen significant changes in how it is formatted, from the original 85 questions that were either multiple-choice or fill-in-the-blank, to a 2014 format that consisted of 65 total questions, including the requirement for numerical answers (making the test more challenging, since guessing was no longer an option), and requiring candidates to select all answers (making it even more challenging for candidates who don't fully understand the subject). GATE is significantly different from what it used to be and has evolved significantly since the time of its introduction.

The 100 marks of the GATE CS examination are divided into three parts:

- Engineering Mathematics: Engineering Mathematics has around 13 to 15 marks. It includes mathematics, linear algebra, calculus and probability.
- Core Technical Subjects: Core Technical Subjects get 70 to 72 marks. These are computer science subjects. The core CS syllabus is a list of computer science subjects. These include Algorithms, Programming and Data Structures Operating Systems, Database Management Systems, Computer Networks, Theory of Computation, Digital Logic, Computer Organization and Architecture and Compiler Design.

B. Educational Data Mining (EDM) and Learning Analytics

EDM (Education Data Mining) is considered a discipline that analyzes educational data with statistics and machine learning techniques for the purpose of analyzing data related to schools. The goal of EDM is to better understand how students interact with their education in order to predict outcomes and to improve the educational environment by assessing the difficulty of items on tests. Traditionally, the bulk of EDM research has focused on improving the ability of teachers to help students who are at high risk of failing to perform successfully in school. Most of this work has been accomplished using machine learning methods. Many studies have shown that machine learning techniques such as Random Forest, Support Vector Machines, and Neural Networks have performed well when used to predict how well students will perform on tests based on performance as well as other factors. An example of the Random Forest method would produce an accurate prediction as to what grade a student will earn when they complete an assignment or take a test 75% of the time. Support Vector Machines produce accurate predictions for how students will perform, but they also produce prediction errors. EDM is also used to evaluate the difficulty of test items and to match student knowledge to educational concepts such as Bloom's Taxonomy. Finally, analyzing text is an integral component of EDM. Older machine learning methods like Logistic Regression have performed well accurately identifying 69% of large scientific texts. Newer methods like Deep Learning and models like BERT have done better at understanding text meaning.

However, there is a challenge in this research area. When we try to train models on small educational data sets they often do not perform well on new data. On the hand simpler methods that use clever techniques to make the most of the data often work better when resources are limited. Educational Data Mining is about using data to improve schools and's an important research area that can help students perform better. Educational Data Mining can make a difference, in how students learn. It is worth studying further. Educational Data Mining and Learning Analytics can help make an impact.

C. Natural Language Processing (NLP) in Academic Text Classification

Natural Language Processing is really helpful because it gets computers to understand language. It does this by using linguistics and some advanced statistical methods. This way machines can break down language. Understand what it means. They can even. Classify language which is really useful. In schools and universities Natural Language Processing is often used to sort text into categories. This means putting text into groups that make sense for academics. For example, Natural Language Processing can help sort research papers into the groups. Studies have shown that machine learning tools are actually better than people at matching research abstracts with the groups. These tools are a way to do this job because they are cheap and can handle a lot of work. Natural Language Processing and machine learning make it easier to categorize research papers.

Natural Language Processing is used to classify research abstracts into the groups. Research shows that machine learning classifiers are more accurate than people when it comes to mapping scientific research abstracts to their discipline groups. Natural Language Processing and machine learning are a team. They help us categorize research abstracts into the groups which is really important. Natural Language Processing makes it easy to understand what research papers are, about**.**

D. The Paradigm Shift to Local Small Language Models (SLMs)

The way we use Large Language Models has changed a lot of things like automated tutoring and chatbots This creates another issue since it would depend on firms such as OpenAI, Anthropic, and even Google with its cloud services. Delays and significant risks for the privacy of students may arise. It could be very costly, which is unfavorable for students from poorer countries. The development of more efficient models led to the possibility to use Small Language Models locally. A program called Ollama allows one to employ models without having access to the internet.

One of the major advancements was made by Microsoft with the release of Phi-3 Mini, which is significant because of its number of parameters. It counts 3.8 billion parameters. Just like large models, Phi-3 Mini operates in the same way concerning reasoning, advanced mathematics, and coding. Best of all, this model can operate on computers that lack memory; all it needs is 8 GB of RAM, making it ideal for schools with students' personal information being secured.

3. Research Methodology

The ExamEdge framework is something that can work all by itself. It does not need to be connected to the internet to work properly. The people who made the ExamEdge framework designed it in a way that makes it easy to add things to it without messing up the rest of the system. This means you can use the ExamEdge framework on kinds of computers. The ExamEdge framework can run on a computer that has a core or quad-core Central Processing Unit and 8GB of RAM. You will also need to have about 10GB of space on your drive. The ExamEdge framework uses things that're open to everyone and free to use. The ExamEdge framework uses Python 3.11 as the programming language. The ExamEdge framework uses Python 3.11 because Python 3.11 is very good, at handling data and understanding language. The ExamEdge framework uses Python 3.11 because it has a lot of tools and resources that make it easy to work with data and natural language. The ExamEdge framework is a choice because the ExamEdge framework can do a lot of things with Python 3.11.

A. Data Engineering and Unstructured Extraction Pipeline

We need to preprocess the data from the old exam papers in order to obtain useful information from them. For this purpose, we obtained old GATE CS papers for 2012-2025 and old GATE DA papers for 2024-2025. These documents are challenging due to their layout that is characterized

by two columns, drawings and varied fonts over 13 years. In order to extract the text information, we use a Python library called pdfplumber. Such software provides a way to inspect characters on the page along with their positions in the paper. This feature enables extracting text even from drawings.

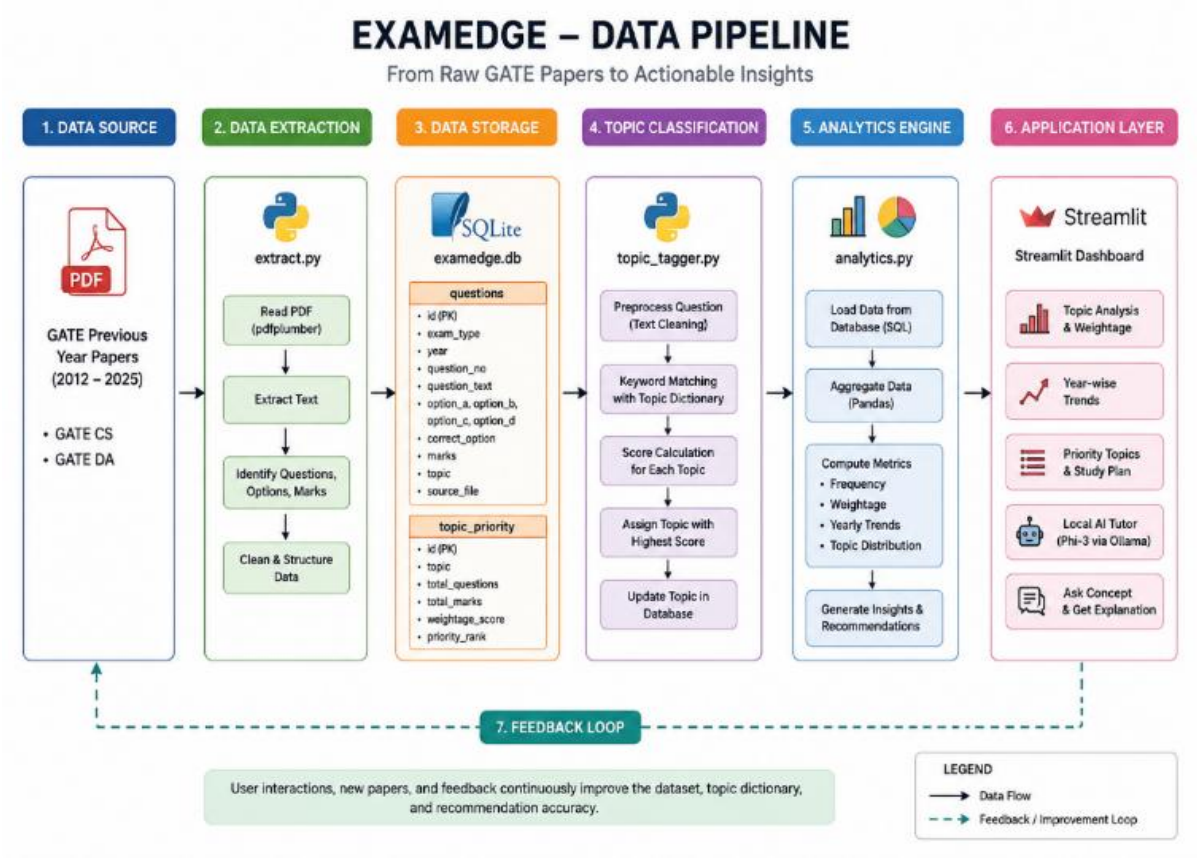


Figure 1: ExamEdge – Data Pipeline

B. NLP and Keyword-Driven Topic Classification

After we get the data the SQLite database has records that do not have labels and need to be classified into the 10 GATE CS subjects. The GATE CS exams have technical words so ExamEdge uses a special algorithm that looks for keywords to classify the records. This algorithm was made by looking at the official GATE CS syllabus documents and mapping around 30 to 50 technical terms, algorithms and mathematical protocols to each of the 10 GATE CS subjects. When the algorithm is running it takes the question and all the answer options puts them together into one string and makes it all lowercase. This big string is then checked against a list of keywords. We can think of the classification like a math problem, where we find the GATE CS subject that has the matching keywords in the string. Each GATE CS subject gets a score based on how keywords it has and the GATE CS subject with the highest score is the one we choose.

Topic	2012	2013	2014	2016	2018	2022	2023	2024	2025
Algorithms	4	3	4	1	1	5	3	3	3
Computer Networks	8	5	9	8	7	6	5	16	13
Computer Organization	6	7	8	5	14	10	10	18	20
Data Structures	6	6	6	7	5	7	4	14	12
Database Management Systems	9	5	11	7	9	9	3	13	16
Digital Logic	2	4	1	2	4	4	6	12	11
General	3	2	1	3	4	2	3	8	7
Mathematics	2	2	1	0	2	3	0	2	2
Operating Systems	4	5	5	5	4	3	3	10	6
Programming and C	7	5	2	9	9	3	7	9	9
Theory of Computation	6	11	7	8	6	8	14	11	17

Figure 2: Topic Classification

4. Results and Analysis

A key thing that any new education platform needs to do is to prove that it really works. It refers to presenting true results. The ExamEdge software underwent rigorous testing that aimed to measure its performance levels.

- Its classification engine was checked
- Its local Large Language Model (LLM) part was also tested.

The testing ensures the reliability of ExamEdge. It does what it aims to do in education. The ExamEdge software framework received extensive testing aimed at evaluating its performance levels. Its classification engine and its LLM inference module were thoroughly tested. Both belong to the ExamEdge platform. The ExamEdge platform seeks to revolutionize exam preparation process. The extensive tests conducted on the platform were meant to verify its performance levels. The tests conducted included classification engine test and LLM inference module test.

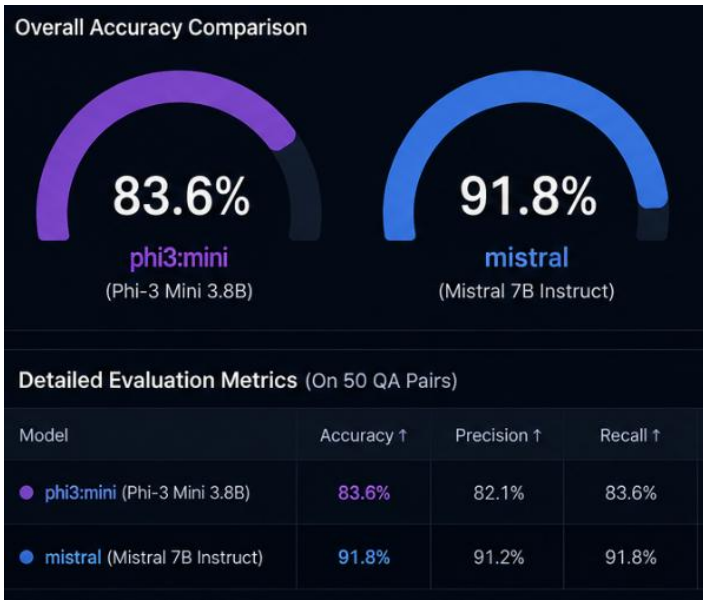


Figure 3: Model Evolution

A. Classification Engine Efficacy and Error Diagnostics

The keyword-based NLP classifier functioned successfully with the set. In the course of processing 752 questions, 709 of those got mapped quickly and accurately to respective topics. Accuracy rate reached 94.3%. The entire process of question classification was accomplished in less than one second with standard hardware performance. Such rapidity proves how quick and efficient lexical approach is. Further analysis of 43 incorrectly classified questions revealed certain patterns. Most of those belonged either to visual or mathematical questions which utilized diagrams, complex mathematical notations or symbols lacking respective keywords. This limitation significantly reduces efficiency of the text-based classification. The proposed method significantly outperformed traditional Machine Learning techniques such as Bayes and Logistic Regression commonly used in educational Data Mining field. Average accuracy rate of text classification in terms of those methods varies from 69% to 75%. Accuracy rate of 94.3% demonstrated by ExamEdge proves high effectiveness of syllabus-based classification in technical subjects. The ExamEdge and keyword-based NLP classifier proved themselves extremely efficient. The ExamEdge and keyword-based NLP classifier are highly effective solutions for question classification. The keyword-based NLP classifier proved to be very efficient tool.

B. Benchmarking Zero-Shot Pre-Trained Alternatives

ExamEdge is preparing to make a paradigm shift on how things can be classified. Zero-shot classification with NLI will aid in reducing the error rate of 5.7%. This will involve analyzing the performance of our target model. The target model that ExamEdge will use is Facebook/BART-mNLI. Numerous researches have analyzed the performance of this model using Multi Genre NLI datasets. These studies show that BART-large-MNLI is one of the zero-shot classifiers available. It is very reliable. In fact, it has been downloaded over 1 million times per month on Hugging Face. When we try to classify texts without any examples the performance depends a lot on the hypothesis template we use. Some templates work better than others. For example, using templates like "This example is {class name}" gives results. We can use these results in different situations. However sometimes the models struggle with knowledge. This knowledge is outside of their area of expertise. In tests BART-large models have shown weighted F1 scores that range from 48.48 to 63.98. This depends on how the entailment formulations fine-tuned. This is not as good as ExamEdges 94.3% accuracy. ExamEdges lexical accuracy is very specific.

C. Large Language Model (LLM) Inference Benchmarks

When studying classifications, they can guide you on what to focus on. Students often struggle with breaking down and solving engineering problems. ExamEdge is helpful here because it doesn't require internet to function. It uses an intelligence engine on your computer, powered by Ollama and Microsofts efficient Phi-3 Mini model.

The Phi-3 Mini model is a language model with 3.8 billion parameters. It was trained on a chosen set of data including some made-up data and filtered information from good academic websites. This model can handle a lot of information at once. Either 4,000 or 128,000 pieces. The Phi-3 Mini model is designed to excel in reasoning, mathematics and complex logic. The Phi-3 Mini model is particularly good at helping with engineering problems. This is because it is trained on types of data and can understand complex relationships between things and the Phi-3 Mini model keeps doing a great job with engineering problems. It is really good, at helping with engineering problems because of its training and capabilities.

Discussion

The results of this research show that the ExamEdge framework is capable of helping the GATE aspirants through combining the Educational Data Mining, Natural Language Processing, and offline Artificial Intelligence. The classification of topics based on keywords achieved 94.3% accuracy, proving that syllabus-based classification is appropriate for the dataset of technical examinations. As a result, the framework is able to categorize the historical GATE questions by their subject and analyze the trends accurately.

One of the valuable outcomes of this research is the detection of trends in the development of the GATE examination. Analysis of the question papers between 2012 and 2025 showed that Computer Organization and Architecture and Theory of Computation became more prominent, while Algorithms plays a minor role in the examination. Therefore, one should be aware of the data-driven importance of different topics rather than preparing only on the basis of coaching or assumptions about the subject's importance. Moreover, ExamEdge uses the Phi-3 Mini language model through Ollama, which helps students to get the AI-powered explanation without Internet connection.

Though the framework has proven to be quite successful, there are a few limitations to the same. The classifier currently used is based on keywords and finds difficulty handling questions that include complex mathematical statements, diagrams, or lack textual information, which leads to a few cases of misclassification. Introduction of semantic models such as BART-large-MNLI would go a long way towards enhancing the accuracy of the classification process.

Conclusion

The ExamEdge framework shows that Educational Data Mining and Artificial Intelligence can really change the way we get ready for exams. I mean ExamEdge framework is using Educational Data Mining and Artificial Intelligence to change how we prepare for exams. The ExamEdge framework looked at 752 questions from 13 years of GATE CS papers. Found some useful information. This information from ExamEdge framework goes against what traditional coaching methods say.

For example, the national examiner is now focusing on Computer Organization and the Theory of Computation not on Algorithms. This gives candidates a plan for how to use their time and resources for ExamEdge framework. The ExamEdge framework system also shows that you do not need a lot of equipment or internet to use technology like ExamEdge framework. The fact that Microsoft's Phi-3 Mini can work on a computer with 8GB RAM is a thing for making educational technology available to everyone for ExamEdge framework. There are some limitations to the ExamEdge framework system like it can have trouble understanding data and can be slow at times. Overall the ExamEdge framework system is very strong and can handle a lot of work for ExamEdge framework.

In the future the goal is to make the ExamEdge framework system even better. This can be done by reducing the number of mistakes it makes for ExamEdge framework. The ExamEdge framework system will also be able to handle types of exams like the Joint Entrance Examination and National Eligibility cum Entrance Test.

References

1. Romero, C., & Ventura, S. (2010). Educational data mining: A review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 40*(6), 601–618. https://doi.org/10.1109/TSMCC.2010.2053532
2. Baker, R. S., & Yacef, K. (2009). The state of educational data mining in 2009: A review and future visions. *Journal of Educational Data Mining, 1*(1), 3–17. https://jedm.educationaldatamining.org/index.php/JEDM/article/view/8
3. Siemens, G., & Baker, R. S. J. d. (2012). Learning analytics and educational data mining: Towards communication and collaboration. In *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge (LAK)* (pp. 252–254). https://dl.acm.org/doi/10.1145/2330601.2330661
4. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the NAACL-HLT* (pp. 4171–4186). Minneapolis, MN, United States. https://aclanthology.org/N19-1423/
5. Liu, Y., *et al.* (2019). *RoBERTa: A robustly optimized BERT pretraining approach* (arXiv:1907.11692). https://arxiv.org/abs/1907.11692
6. Brown, T. B., *et al.* (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems, 33*, 1877–1901. https://arxiv.org/abs/2005.14165
7. Ouyang, L., *et al.* (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems, 35*. https://arxiv.org/abs/2203.02155
8. Microsoft. (2024). *Phi-3 technical report: A highly capable language model locally on your phone* (arXiv:2404.14219). https://arxiv.org/abs/2404.14219

9. Hugging Face. (n.d.). *facebook/bart-large-mnli*. https://huggingface.co/facebook/bart-large-mnli
10. Ollama. (n.d.). *Ollama documentation*. https://ollama.com/docs
11. Indian Institute of Science, & Indian Institutes of Technology. (n.d.). *Graduate Aptitude Test in Engineering (GATE).* https://gate2025.iitr.ac.in/
12. Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press. https://nlp.stanford.edu/IR-book/
13. Jurafsky, D., & Martin, J. H. (2024). *Speech and language processing* (3rd ed.). Pearson. https://web.stanford.edu/~jurafsky/slp3/
14. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient estimation of word representations in vector space* (arXiv:1301.3781). https://arxiv.org/abs/1301.3781
15. Vaswani, A., *et al.* (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 30*. https://arxiv.org/abs/1706.03762
16. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). https://dl.acm.org/doi/10.1145/2939672.2939785
17. Pedregosa, F., *et al.* (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830. https://jmlr.org/papers/v12/pedregosa11a.html
18. McKinney, W. (2010). Data structures for statistical computing in Python. In *Proceedings of the 9th Python in Science Conference* (pp. 56–61). https://conference.scipy.org/proceedings/scipy2010/mckinney.html
19. Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering, 9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55
20. Python Software Foundation. (n.d.). *Python 3.11 documentation*. https://docs.python.org/3.11/

Bhumi Publishing, India
July 2026

PREDICTIVE COST ANALYTICS IN TEXTILE MANUFACTURING: STRATEGIC COST MANAGEMENT THROUGH ARTIFICIAL INTELLIGENCE

Vishal Kumar R*[1] and P. Shanmugha Priya[2]

[1]Happy Valley Business School, Coimbatore

[2]Department of Management, Karpagam Academy of Higher Education, Coimbatore

*Corresponding author E-mail: vishal@happyvalley.in

Abstract

The textile and apparel manufacturing sector serves as one of India's paramount industrial engines, accounting for nearly 30% of national textile exports and offering direct employment to millions. Concentrated heavily within the western industrial clusters of Tamil Nadu—famously dubbed the "Textile Valley of India" (encompassing Coimbatore, Tiruppur, Erode, and Karur)—this ecosystem covers the complete value chain from spinning, weaving, and wet processing to garment fabrication.

However, historical margin advantages in cluster manufacturing have been severely eroded by macroeconomic shifts, global supply chain disjunctions, and heightened factor cost volatility. Traditional cost accounting frameworks—such as Standard Costing and Activity-Based Costing (ABC)—are inherently backward-looking (*ex-post*). Management typically receives variance reports weeks after production cycles conclude, rendering corrective actions reactionary and sub-optimal. In an operational environment characterized by double-digit volatility in raw cotton prices, dynamic peak-load power tariffs, and shifting piece-rate labor economics, relying solely on historical financial reporting creates significant information asymmetry.

To maintain global competitiveness, industrial manufacturing clusters must transition from reactive recording to proactive decision intelligence. Predictive Cost Analytics (PCA) leverages Artificial Intelligence (AI), machine learning algorithms, and Enterprise Resource Planning (ERP) operational telemetry to anticipate unit production costs prior to execution. By translating continuous operational parameters—such as spindle speed, yarn break frequency, power factor, thermal dye consumption, and machine utilization—into predictive financial indicators, AI enables real-time dynamic pricing, automated variance intervention, and optimized raw material procurement strategies.

Introduction

The Evolution of Strategic Cost Accounting in Manufacturing

Strategic Cost Management (SCM) has evolved through distinct conceptual phases to keep pace with changing industrial environments:

- **Traditional Standard Costing:** Focused primarily on post-production financial auditing and variance tracking. While effective for static environments, it fails to account for dynamic factor cost fluctuations during active production cycles.
- **Activity-Based Costing (ABC) & Time-Driven ABC (TDABC):** Mapped overhead costs directly to operational consumption activities and capacity usage rather than generic volume drivers. These frameworks improved overhead allocation accuracy but remained largely constrained to historical reporting periods.
- **Predictive Cost Analytics (PCA):** Represents the Industry 4.0 evolution of management accounting. By integrating continuous operational data from machine sensors and ERP transaction databases directly with predictive AI engines, PCA converts cost management from a passive historical audit into an active, real-time decision-support system.

Strategic Cost Drivers in Textile Clusters

Unit production costs in energy- and raw-material-intensive industrial manufacturing are governed by structural cost drivers across the value chain:

- **Raw Material Price Fluctuations:** Raw cotton and yarn inputs constitute the single largest cost component in textile manufacturing. Price volatility driven by global commodity markets directly impacts operating margins, making real-time price tracking and predictive procurement essential for cost stabilization.
- **Energy Consumption and Tariff Structure:** Continuous motor loads in spinning, automated air-compressors in weaving, and steam generation in wet processing make electricity and fuel consumption primary cost variables. Industrial electricity tariffs, including peak-load pricing and power factor penalties, create non-linear cost curves across production shifts.
- **Operational Efficiency and Machine Utilization:** Machine downtime, sub-optimal operating speeds, and delayed maintenance increase overhead allocation per unit. Maximizing equipment effectiveness directly dilutes fixed manufacturing costs across total throughput.
- **Material Wastage and Thermal Loss:** Yarn breakage, fiber fly in spinning, fabric edge trimming, and thermal dye loss in processing represent direct resource inefficiencies. Automated floor-level monitoring allows management to mitigate yield loss before batches are completed.
- **Labour Productivity and Logistics:** Variability in piece-rate labor efficiency, operator shift handling, and raw material inventory turnover significantly impact production velocity and total operational carrying costs.

Operational Profile across Sub-Sectors

Different sub-sectors within the textile manufacturing cluster exhibit distinct cost structures and operational tracking priorities:

Sub-Sector	Operational Focus	Primary Strategic Cost Drivers
Spinning Mills	High-speed, continuous yarn production	Raw cotton commodity pricing, grid power consumption, spindle RPM, and fiber waste percentage.
Weaving & Knitting	Mechanical loom and knitting operations	Power tariffs for continuous air compressors, loom operating efficiency, and yarn break frequency.
Wet Processing (Dyeing)	Thermal treatment, bleaching, and chemical application	Thermal boiler fuel costs, chemical dye consumption, water treatment energy, and process batch yield.
Garment Manufacturing	Cutting, sewing, finishing, and packaging	Labor productivity rates, fabric utilization/cutting efficiency, and material inventory turnover.

Strategic Managerial Framework for AI Cost Integration

To successfully transform cost accounting into proactive decision intelligence, manufacturing enterprises can implement a three-tiered predictive architecture:

- **Operational Data Layer:** Bridges physical equipment sensors, IoT energy meters, and shop-floor transactions directly to enterprise databases, ensuring continuous telemetry capture.
- **Predictive AI Engine:** Processes multi-dimensional operational metrics using machine learning algorithms to identify non-linear cost patterns and forecast unit production costs prior to execution.
- **Executive Action Layer:** Translates algorithmic insights into actionable decision support, enabling dynamic forward pricing, intelligent load scheduling, and real-time floor variance alerts.

Key Strategic Applications

- **Dynamic Forward Pricing:** By anticipating unit production costs prior to execution, export-oriented manufacturers can generate highly accurate forward price quotes for international buyers. This protects profit margins against anticipated raw material price spikes or upcoming utility tariff adjustments during production lead times.
- **Automated Energy Load Scheduling:** With real-time visibility into power consumption patterns and peak-load tariff structures, production managers can dynamically auto-schedule energy-intensive production shifts to off-peak tariff hours, lowering total electricity expenditure.
- **Real-Time Floor Variance Mitigation:** Rather than discovering material wastage or machinery inefficiency after batch completion, predictive AI architectures generate real-

time floor alerts for shop-floor supervisors. Immediate interventions prevent compounding resource losses across high-volume production runs.

Strategic Recommendations for Industrial Leaders

- **Integrate Operational Telemetry with Financial Ledger Systems:** Bridge the gap between shop-floor IoT sensors and financial accounting modules to enable continuous, real-time cost tracking.
- **Transition from Ex-Post Reporting to Ex-Ante Forecasting:** Shift organizational focus from passive, periodic variance auditing to proactive cost pattern forecasting to safeguard operating margins.
- **Build Data-Driven Management Capabilities:** Equip finance, accounting, and plant leadership teams with analytic capabilities to interpret predictive insights and translate floor metrics into strategic cost decisions.

References

1. Cooper, R., & Kaplan, R. S. (1991). *The design of cost management systems: Text, cases, and readings*. Prentice Hall.
2. Kaplan, R. S., & Anderson, S. R. (2007). *Time-driven activity-based costing: A simpler and more powerful path to higher profits*. Harvard Business School Press.
3. Porter, M. E. (1985). *Competitive advantage: Creating and sustaining superior performance*. Free Press.
4. Rajarathinam, M., & Sundaram, K. (2022). Operational challenges and cost volatility in Tamil Nadu spinning mills. *Journal of Textile Cluster Management, 14*(2), 112–128.
5. Senthil, K., & Shanmugam, P. (2023). Industry 4.0 adoption and artificial intelligence in Indian manufacturing: An empirical assessment. *International Journal of Production Economics, 245*, 108412.
6. Bebbington, J., & Larrinaga, C. (2014). Accounting and sustainable development: An exploration. *Accounting, Organizations and Society, 39*(6), 395–413.
7. Burritt, R. L., & Schaltegger, S. (2010). Measuring the unmeasurable? An examination of carbon accounting definitions, approaches and metrics. *The British Accounting Review, 42*(2), 82–98.
8. Guenther, E., Endrikat, J., & Guenther, T. (2016). Environmental management control systems: A systematic review and research agenda. *Journal of Management Accounting Research, 28*(2), 7–37.
9. He, R., Luo, L., & Tang, Q. (2022). Corporate carbon accounting capabilities and decarbonization performance. *Journal of Environmental Management, 304*, 114227.
10. Tamil Nadu Generation and Distribution Corporation. (2024). *Industrial tariff revisions and power factor optimization guidelines*. Tamil Nadu Generation and Distribution Corporation Reports.

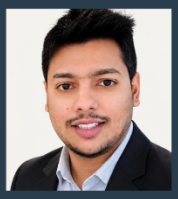
Computing Beyond Algorithms: AI, Data and Intelligent Systems

(ISBN: 978-93-47587-47-4)

About Editors

Dr. Pawan Kumar Dixit is an Associate Professor of Mathematics in the School of Engineering and Technology at Maharishi University of Information Technology (MUIT), Lucknow, and currently serves as the Chief Proctor of the University. He holds M.Sc. and M.Phil. degrees from CSJM University, Kanpur, and a Ph.D. in Mathematics from the University of Lucknow. With over 14 years of teaching and research experience, his research interests include fluid mechanics, channel flow, flow through porous media, and magnetohydrodynamics (MHD). He has published more than eight research papers, authored one book, and contributed twelve book chapters with reputed academic publishers. A life member of several professional societies, he is actively engaged in research, academic administration, and the advancement of higher education through interdisciplinary research and innovation.

Dr. Ponmani Swaminathan is an Associate Professor and Head of the Department of Petroleum Engineering at the Academy of Maritime Education and Training (AMET) University, Chennai. She earned her Ph.D. in Ocean Engineering from IIT Madras and holds B.Tech. and M.Tech. degrees from Anna University. With over 15 years of teaching and research experience, she has completed a DST-SERB Early Career Research project and previously served as a lecturer in Ghana. She has published numerous research papers, presented at international conferences, and holds three patents in drilling fluids and enhanced oil recovery. She serves as the SPE Faculty Advisor at AMET University and reviews manuscripts for reputed journals. Her research interests include drilling fluids, enhanced oil recovery, desalination, marine pollution, and sustainable petroleum engineering technologies.



Udaya Bhaskar Vemuri is an application security professional with 13 years of experience in DevSecOps, secure software development, and vulnerability management. He specializes in integrating security throughout the software development lifecycle and enabling organizations to build secure, scalable, and resilient applications. His expertise includes application security, secure coding practices, risk assessment, and vulnerability management across modern software environments. He actively contributes to the cybersecurity community through technical writing, independent research, and mentoring emerging professionals. Bhaskar also serves as a hackathon judge, encouraging innovation and best practices in secure software development. His professional interests focus on advancing cybersecurity awareness, promoting DevSecOps adoption, and developing practical security solutions that strengthen organizational resilience against evolving cyber threats.

Ms. R. Deivanayaki, M.E., is pursuing a Ph.D. in the Department of Electrical and Electronics Engineering at the Academy of Maritime Education and Training (AMET) Deemed to be University, Chennai. She holds a Master's degree in Electrical and Electronics Engineering with specialization in Power Electronics and Drives from Anand Institute of Higher Technology. She has two years of teaching experience as an Assistant Professor and four years of industrial experience as a Design Engineer. She has published two patents, conference papers, and IEEE publications. Her research focuses on electric vehicles, battery management systems, and advanced power electronics. Her academic and industrial experience reflects strong expertise in electrical engineering, power electronics, drives, sustainable energy technologies, and innovative engineering research with practical applications.

Bhumi Publishing

Bhumi Publishing,
Kolhapur, Maharashtra, India
bhumipublishing@gmail.com
www.bhumipublishing.com

ISBN 978-93-47587-47-4