Membrane Technology
ISSN (online): 1873-4049

Research on the Development of Computer Application Software based on Artificial Intelligence Technology

ZongkuiFu[1,„a*]

[1]*InformationEngineeringDepartmentofWeihaiOceanVocationalCollegeWeihai 264300, Shandong, China*

[a]*Email: jimoniuzai@163.com*

ABSTRACT

This paper explores the application of artificial intelligence technologies to software development processes. The research explores how machine learning techniques enhance software design, automated coding aids programmer productivity, and AI-enabled testing improves quality assurance. Case studies in the healthcare, finance, retail, and enterprise sectors provide examples of AI's transformative impact on applications in each of those sectors. The research identifies significant technical issues, including data quality issues, computing power limits, and difficulties in interfacing with legacy systems. The article also discusses ethical issues regarding privacy, comprehending decisions, and compliance with regulations. In the future, emerging AI technologies such as explainable AI, reinforcement learning, and hybrid combined systems are likely to transform software development. This review consolidates what is known so far and offers recommendations for organizations using AI development practices. It calls for a cautious balance between AI automation and human skills to enhance benefits and reduce dangers.

Keywords: Artificial Intelligence, Software Engineering, Machine Learning, Automated Coding, Quality Assurance, Predictive Analytics, Case Studies, Ethical Considerations

I. INTRODUCTION

The utilization of Artificial Intelligence (AI) in software development is a substantial change in how programs are designed, built, and maintained. AI has been the major contribution in revolutionizing the process of software development, with new ways of operating at different stages of the software development process (Upadhyaya, 2022). Artificial intelligence tools are used for automating difficult and repetitive tasks. A few of these tasks involve generating code, debugging, and optimizing deployment methodologies. Automation not only improves productivity but also reduces errors. This leads to better quality software and faster development processes.

Zhang (2024) highlights that software engineering is changing significantly with the introduction of AI. Traditional methods are failing to meet the rising demand for good quality software in shorter periods of time. This change is affecting many aspects as increasingly more requirements analysis, design development, coding, testing, and maintenance are being accomplished with the assistance of AI tools. AI technologies like machine learning (ML), natural language processing (NLP), and computer vision have transformed these stages (Upadhyaya, 2022). For example, GitHub Copilot and similar tools show how AI can speed up coding by automatically generating boilerplate code and allowing developers not to be slowed down by routine activities, thereby sparing them from mistakes (Salerno, 2024).

Table 1: Time Savings with AI Tools

Tool	Function	Time Saved	Description
GitHub Copilot	Code Generation	40%	Automates boilerplate code, speeding up development and reducing manual coding efforts.
Automated Testing	Test Execution	50%	Saves time by automating repetitive testing tasks, allowing for more comprehensive testing with less manual intervention.
DeepCode	Bug Detection	30%	Enhances productivity by identifying bugs more quickly than traditional methods, reducing the time spent on manual debugging.

Artificial intelligence software applications have been steadily growing to be an important market in the recent past. The research done by IDC for the year 2019 estimates that the spending on cognitive as well as AI systems which was approximately $24.0 billion in the year 2018 will grow up to $77.6 billion by 2022 (Bharadiya et al., 2023). This is further supported by Grand View Research that projected the global AI industry is set to grow from $136.6 billion in 2022 to $1811.8 billion by 2030 at a compounded annual growth rate of 38.1% (Bharadiya et al., 2023).

Figure 1: The Virtuous Cycle of AI (Bharadiya et al., 2023)

The purpose of this study is to provide an extensive review of computer application software based on the AI technology, discussing how the AI technology is improving the engineering process of the software. This ranges from the fundamentals of AI in software context, approaches towards AI and software integration, success stories, technical issues, and concerns, and prospects of AI.

Figure 2: Applications of AI in Business (Bharadiya et al., 2023)

Based on Güneşdoğdu (2024), AI in software development has progressed from simple tools that support the developers to sophisticated systems that perform complex and independent tasks. The given trajectory appears to be very much cooperative in nature where AI enhances creativity and decisions of human beings in order to bring out the best results in software development. This research aims at making a contribution to this discussion by shedding some light on how AI technologies should be adopted in software development life cycle to get the most out of it.

II.THEORETICAL FOUNDATION

Artificial Intelligence (AI) has undergone several developments that have influenced its use in software development. Dhar (2024) states that AI has developed through four main stages: Expert Systems, Machine Learning, Deep Learning, and General Intelligence. The Expert Systems stage lasted from the mid-1960s to the late 1980s and was characterized by the presentation of human knowledge through "IF/THEN" rules, semantic networks, or structurized object representations. Machine Learning started in the late 1980s and 1990s. It progressed from setting rules manually to learning automatically from data by using optimization methods based on a loss function (Dhar, 2024). Deep Learning uses complex neural network structures and solved the problem of feature engineering. It allows machines to directly process raw data like images, language, and sound (Alenezi&Akour, 2025).

Table 2: Evolution of Software Engineering Practices (Alenezi&Akour, 2025)

Era	Methodologies	Key Features	Introduction/Adoption Period	References
Traditional	Waterfall model	Linear, sequential approach to software development	1970s	Royce, 1970
Iterative	Agile, Scrum	Flexible, iterative approach emphasizing collaboration	1990s	Beck et al., 2001
Modern	DevOps, continuous integration	Automated testing, continuous deployment practices	2010s	Humble and Farley, 2010
AI-Driven	AI-powered tools, machine learning	Automated coding, adaptive software maintenance	2020s	Smith et al., 2023, Qian et al., 2024

The current AI paradigm, characterized by General Intelligence, represents a fundamental shift from traditional approaches. As Nielsen (2025) indicates, we are witnessing a transition from command-based interactions to an "intent-based outcome specification" paradigm where users specify desired outcomes rather than precise commands. This aligns with the evolution described by Dhar (2024), where AI has transitioned from an application specific to one task to a general-purpose technology configurable for various applications. This transformation allows developers to leverage AI capabilities across multiple domains, including code generation, testing, and maintenance, without requiring explicit task-specific training (Alenezi&Akour, 2025).

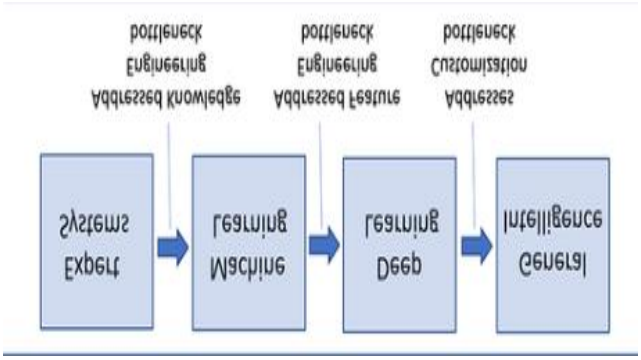

Figure 3: The History of Artificial Intelligence (Dhar, 2024)

As can be seen from Figure 13, the evolution of AI has systematically solved significant bottlenecks in every stage of its development. The evolution towards General Intelligence from Expert Systems illustrates how each paradigm has built on the previous one, surmounting specific weaknesses along the way and ultimately giving rise to a more robust and generalized AI technology. This evolutionary trajectory illustrates the ongoing enhancement of AI capabilities to make it ever more suitable for software development tasks.

Key AI breakthroughs specific to software development are machine learning, deep learning, natural language processing, and neural networks. Alenezi and Akour (2025) point to the exceptional developer productivity and accuracy gains attained with the likes of GitHub Copilot and AlphaCode via automated code generation. Conversely, machine learning algorithms, formulated on the basis of past experience, facilitate bug detection early on. Rashid and Kausik (2024) also describe that AI technology is applicable to all sectors, including software development, where techniques like machine learning and deep learning enhance performance and automate intricate processes.

Table 3: The Paradigm Shifts in AI (Dhar, 2024)

	Data	Exemplar	Scope	Curation
Expert Systems	Human	Rules	Follows	High
Machine Learning	+ Databases	Rules/networks	+ Discovers relationships	Medium
Deep Learning	+ Sensory	Deep neural networks	+ Senses relationships	Low
General Intelligence	+ Everything	Pre-trained deep neural networks	+ Understands the world	Minimal

Current paradigms in AI-based software engineering include explainable AI (XAI), which addresses the challenge of understanding and interpreting decisions made by AI models. Marar (2024) explains that XAI aims to provide insights into how AI algorithms reach specific conclusions, making them more interpretable for developers, stakeholders, and end-users. Another emerging paradigm is the integration of AI with DevOps practices, creating a more efficient software development lifecycle through AI-driven analytics and monitoring tools that enhance continuous integration and delivery processes (Marar, 2024). Furthermore, reinforcement learning and unsupervised learning are increasingly being applied to software engineering, creating more autonomous and adaptive systems that learn from the development process and adapt to changing requirements (Marar, 2024).

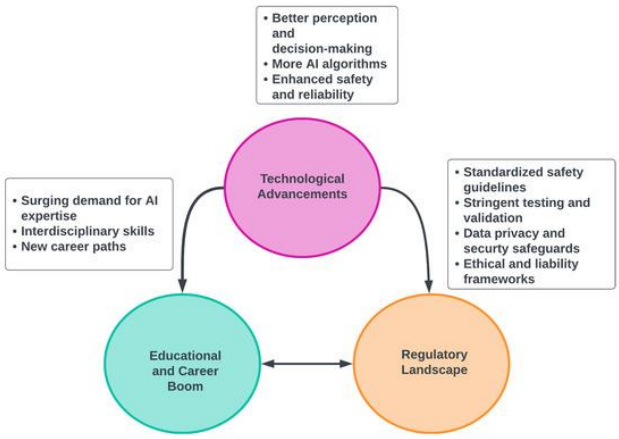

Figure 4: Benefits of AI in Autonomous Vehicles (Garikapati&Shetiya, 2024)

Figure 4 illustrates the interconnected ecosystem of AI advancement in autonomous vehicle technology. The diagram showcases how technological advancements in AI for autonomous vehicles are supported by both an educational and career boom as well as an evolving regulatory landscape. Each component features specific developments that collectively support the growth and implementation of AI in vehicle automation, representing a model that parallels similar integration patterns in software engineering.

III. AI-ENABLED SOFTWARE DEVELOPMENT METHODOLOGIES

Machine Learning Approaches in Software Design

Machine learning (ML) approaches have revolutionized software design by automating complex tasks and enhancing decision-making capabilities. Zhang and Tsai (2003) highlight that ML algorithms offer a viable alternative to existing approaches for many software engineering issues, particularly in domains with insufficient knowledge for humans to develop effective algorithms. According to Sarker (2021), supervised learning algorithms, such as decision trees, logistic regression, and support vector machines, are widely employed in software design to predict high-risk components and identify design patterns.

Table 4: Application Patterns of ML Methods (Zhang & Tsai, 2003)

Activity type	SE task	ML method
Prediction	Software quality (high-risk, or fault-prone component identification)	GP, NN, CBR, DT, CL, ILP
Property/model discovery	Program invariants, Identifying objects in programs, Process models	ILP, NN, EBL
Transformation	Transform serial programs to parallel ones, Improve software modularity	GP, CBR + NN, GA
Generation and synthesis	Test cases/data, Software agents, Design repair knowledge	ILP, GA, CBR + EBL

Machine learning approaches in software engineering help in the detection of problems and design pattern detection. ML-based approaches, as stated by Asaad and Avksentieva (2023), can identify signs of poor system design or coding sooner during the development process. Machine learning algorithms like logistic regression, random forest, and neural

networks have been shown to work well in the detection of design patterns like adapter, strategy, singleton, and factory method. In addition, ML approaches assist in selecting reusable software components, wherein decision trees and neural networks excel at identifying the correct objects for software design patterns (Asaad &Avksentieva, 2023).

Automated Programming and Coding Assistance

AI Automated coding is a revolution in writing software since it enhances the development of software. Campbell (2020) notes this transition in coding from the traditional Fortran type coding automation to the new coding automation with neural networks capable of developing whole codes from what humans desire. AI programming assistants in use today utilize advanced neural networks trained on big code datasets to generate, transform, and optimize code.

Emerging platforms like DeepCoder, RobustFill, and SketchAdapt illustrate the evolution of automated coding. These platforms, as per Campbell (2020), create program code from examples or sketches that the user inputs through inductive program synthesis and search-based techniques. For example, SketchAdapt utilizes a hybrid of neural networks in addressing the overall structure and search-based techniques for the finer details, and this has registered good performance in the translation of math problems from English to code.

Asaad and Avksentieva (2023) suggest that the most well-liked models for code generation are ChatGPT, Codex, and AlphaCode. These models use different machine learning methods, mostly recurrent neural networks and transformer models, for code generation in numerous programming languages. Sergeyuk et al. (2024) state that 84.2% of developers now utilize AI coding assistants, among which the most used ones are ChatGPT (72.1%) and GitHub Copilot (37.9%). Their survey shows that how people use AI assistants differs depending on what they are doing. AI assistants are most helpful to developers when implementing new features (87.3%) and least helpful for debugging (65.5%).

Testing and Quality Assurance with AI

AI has transformed testing and quality assurance by enhancing test coverage, automating repetitive tasks, and improving defect detection capabilities. Ramchand et al. (2021) indicate that 64% of companies plan to implement AI for Software Quality Assurance (SQA) processes, making testing more efficient and accurate. The authors categorize AI approaches in testing into rule-based (using knowledge bases and predefined rules) and learning-based (leveraging machine learning).

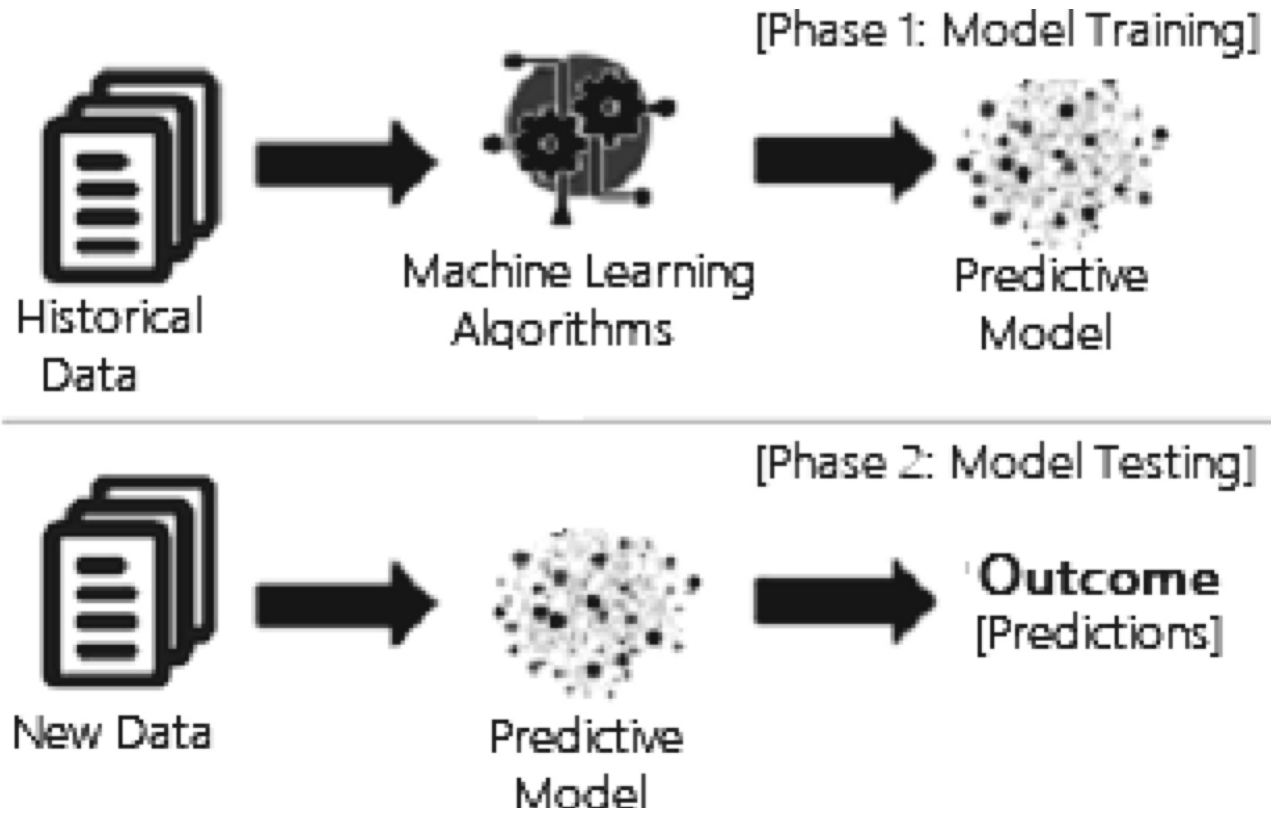

Figure 5: A general structure of a machine learning based predictive model(Sarker, 2021)

This diagram from Sarker (2021) illustrates the two-phase process of machine learning in predictive modeling. Phase 1 shows how historical data is processed through machine learning algorithms to create a predictive model, while Phase 2 demonstrates how new data is fed into this trained model to generate predictions or outcomes. This visualization effectively captures the fundamental workflow that underlies AI-enabled software development methodologies, particularly for automated testing and quality assurance applications.

AI-driven testing methods include the use of statistical software testing, performance testing, and automatic generation of test cases (Asaad &Avksentieva, 2023). According to Ramchand et al. (2021), AI has the ability to automate the setup, monitoring, and maintenance of automated testing tools to give precise outcomes. The above factors have seen the emergence of such positions as AI Testing Experts, AI QA Strategists, and QAOps.

According to Zhang and Tsai (2003), the machine learning techniques such as GA and ILP are very efficient in generating test cases. It can automatically create good test sets and this in turn aids in the identification of defects. Likewise, Asaad and Avksentieva (2023) opine that Q-learning, which is a form of reinforcement learning, is applied in ITS for performance testing. They also state that MINTest is used for software test automation.

The most commonly used area of AI in software maintenance is bug identification and prognosis. Asaad and Avksentieva (2023) indicate that CNN-based deep learning models are recommended for duplicate or similar bug report identification, while supervised learning algorithms such as logistic regression, Naive Bayes, and decision trees are applied to predict the occurrence of software bugs based on historical data. By using AI for automating these kinds of tasks, it greatly saves time in quality checks and renders software more reliable.

IV. EXAMPLES OF HOW AI IS USED

Sector-specific AI software solutions

Artificial Intelligence (AI) is very beneficial in various fields, and each field uses different AI tools to resolve problems specific to their field. In healthcare, AI has a significant role to play in identifying and diagnosing diseases at an early stage. Rahmaniar et al. (2023) present an example where AI and deep learning using convolutional neural networks (CNN) are utilized to assist physicians in analyzing scoliosis patients. This CNN model helps in finding the positions of spinal vertebrae in X-ray images and can automatically compute the Cobb angle with a 93.6% accuracy. This means that it is extremely reliable compared to measurements performed by doctors manually.

Table 5: AI in Industry (Rahmaniar et al., 2023)

Industry	Key AI Application	Brief Description
Healthcare	Diagnosing diseases using medical imaging	Use of neural networks to analyze medical images and detect anomalies.
Finance	Fraud detection and risk assessment	Machine learning models to identify unusual transactions and assess creditworthiness.
Agriculture	Predictive analytics for crop yield	AI algorithms predict crop yields based on environmental data and historical trends.
Retail	Personalized customer experiences and recommendations	Algorithms analyze customer data for personalized shopping experiences and product suggestions.
Energy	Renewable energy forecasting	Predictive models for forecasting energy production, especially from renewable sources like wind and solar.
Automotive	Autonomous vehicles	Complex AI systems interpret sensor data, make navigational decisions, and safely operate vehicles without human intervention.

In the financial industry, AI has revolutionized ways in which organizations identify cases of fraud and evaluate risks. In their article, Rahmaniar et al. (2023) establishes how deep learning models are trained on millions of transactions to recognize such patterns and anomalies more efficiently than humans. These can even alert of suspicious activities like when someone makes a transaction of higher value and has never transacted from a particular country before. Such AI systems have made it possible to achieve between 40% of the fraud transactions while at the same time decreasing the false positives.

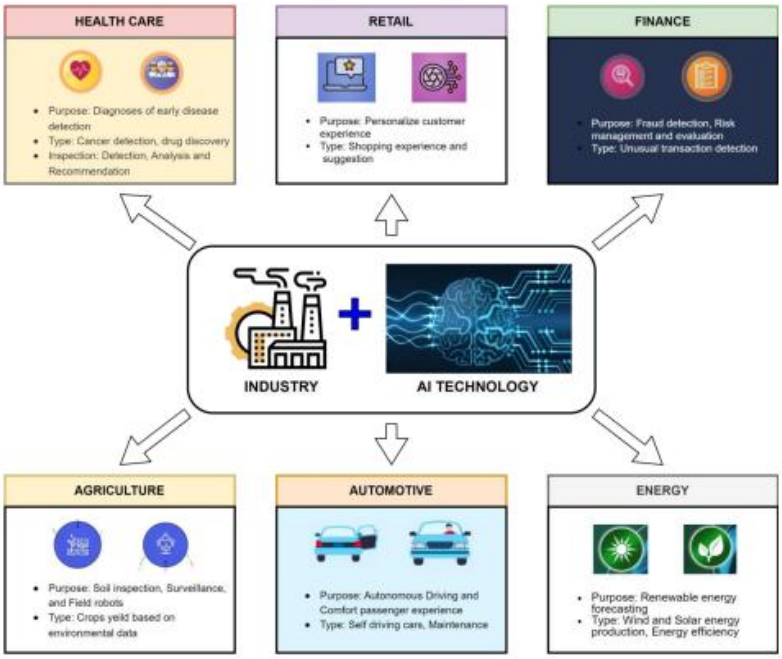

Figure 6: AI applications in industry (Rahmaniar et al., 2023)

The agricultural sector has also embraced AI to streamline various facets of farm management. Farms that utilize AI models have experienced yield increases of up to 20% due to timely interventions, irrigation optimization, and predictive pest control (Rahmaniar et al., 2023). Using advanced sensors and computer AI software, soil samples can be analyzed to see which plants grow most appropriately in different soil types and conditions. This can result in a 15% boost in crop yields and a 10% savings in fertilizer and upkeep.

Consumer-facing AI applications

AI products that customers use have revolutionized how businesses interact with their consumers, especially in retail and marketing. Haleem et al. (2022) note that AI is revolutionizing how brands and users interact, allowing marketers to be more customer-centric and fulfill their needs in the moment. Through AI, marketers can instantly decide what information to show customers and through what channel at the right moment, thanks to data collected and produced by its algorithms.

Figure 7: AI applications in industry (Rahmaniar et al., 2023)

In retail, online shops using AI personalization software have seen a 20% rise in conversion rates and a 15% rise in average order value (Rahmaniar et al., 2023). For creating real-time customized online shopping experiences, AI models learn from users' browsing behavior, purchase history, and click-through rates, which dynamically modify the online shopping environment. The personalization may include changes to the visual design and highlighting particular products.

AI-driven Augmented Reality (AR) has transformed shopping in brick-and-mortar stores. According to Rahmaniar et al. (2023), AR devices use AI to monitor a customer's purchase history and interactions with the store. Then, these devices show real-time product info and suggestions right in front of the customer. Those retailers who are offering personalized suggestions using AR and AI have seen sales in stores going up by 10-15% and customers returning by 20%.

Table 5: Enterprise Conversational AI: Market Analysis and Architectural Components (Mhaskey, 2024)

Component/Aspect	Key Features	Market Metrics	Leading Regions
Natural Language Understanding (NLU)	- Text-based Solutions - BFSI Sector Focus - Cloud Deployment Model	Market Size 2023: $10.7B Market Size 2028: $29.8B CAGR: 22.7%	1. North America 2. Europe 3. Asia Pacific
Dialog Management Systems	- State Tracking - Response Generation - Context Awareness - Machine Learning Integration	Market Size 2022: $8.24B Market Size 2023: $11.03B CAGR: 33.8%	1. Western Europe 2. North America 3. Asia Pacific
Integration Layer	- Security Features - Compliance Management - Cloud-based Solutions - Scalable Architecture	Market Size 2023: $11.03B Market Size 2027: $32.28B CAGR: 30.8%	Global Distribution
Implementation Focus	- Customer Support Services - 24/7 Availability - Omnichannel Deployment - Consistent Service Quality	Primary Adopters: - Large Enterprises - BFSI Sector - Financial Services	Cross-regional

Virtual try-ons and in-store style consultations are new ways in which consumers are advantaged by AI. Rahmaniar et al. (2023) show how AI and AR create virtual tools that allow customers to try on clothes, accessories, or makeup without being present in a store. AI recommends sizes, colors, and additional products according to what customers browse online and what they have bought before. Those retailers that provided virtual try-ons and AI style consultations experienced 20-40% fewer product returns and 3% higher sales of complementary products.

Enterprise AI software deployments

AI adoption in business software systems has significantly enhanced company performance, decision-making, and operational efficiency. Enterprise Resource Planning (ERP) systems have particularly been transformed by the integration of AI capabilities. Mhaskey (2024) informs that over 50% of organizations are planning to deploy AI capabilities within two years, with the AI in ERP global market predicted to grow at a CAGR of approximately 35% during the period from 2023 to 2030.

Machine learning helps ERP systems in predictive analysis to identify the abnormal data and in the process of optimization. Mhaskey (2024) defines machine learning as the analysis of data, recognition of patterns, optimization of processes in the ERP systems, and making them more effective and predictive. It helps organizations make their demand forecasts more accurate, hence improving the accuracy of MPS and MRP. There are postings of the transactions by the ERP and storing of sales history. Thus, the sales history data can be analyzed using the ML algorithms and sales pattern can be forecasted for better customer relationship management.

Table 6: AI Trust, Risk, and Security Management: Market Overview and Technical Challenges (Mhaskey, 2024)

Challenge Category	Key Components	Implementation Sectors	Market Insights
Data Privacy & Security	- Trust Management - Risk Assessment - Security Frameworks	1. Healthcare (Fastest Growth) 2. BFSI 3. Retail	Market Value 2022: $1.18B Projected 2030: $17.82B CAGR: 40.5%
Scalability	- Cloud-based Deployments - Security Measures - System Scalability	- Large Enterprises (Majority Share) - SMEs (Accelerated Adoption)	Regional Leaders: 1. North America 2. Europe 3. Asia-Pacific
Model Maintenance	- MLOps Practices - Data Quality - Model Monitoring - Data Labeling	1. Financial Services 2. Healthcare 3. Technology	Focus on Automated Pipelines and Validation
Infrastructure	- Automated Pipelines - Validation Processes - Edge AI Deployments - Human-in-the-loop	Critical Applications Sectors: - Financial Services - Healthcare	Resource Optimization for Edge Deployments
Resource Management	- Regulatory Compliance - Security Integration - Privacy Requirements	Sectors with Sensitive Data: - Healthcare - BFSI - Technology	Integrated Solutions Framework

NLP has brought significant changes in how individuals interact with business systems. As stated by Mhaskey (2024), NLP takes the ERP system beyond the interface layer and the users are able to talk with the system. The research funded by Sunway University and the Sustainable Business Research Cluster demonstrates how it is possible to use NLP to classify invoices according to the text descriptions in order to select the proper account from the Chart Of Accounts (COA). This coupled with Robotic Process Automation (RPA) can reduce it by up to 90%.

Enterprise systems have become more efficient using Robotic Process Automation (RPA). RPA bots are programmed to carry out repetitive tasks and rules-based tasks in ERP systems. Mhaskey (2024) states that some of the major uses of RPA bots in ERP systems include automating repetitive tasks, financial process management like invoice automation, data handling, improving customer service, and maintaining rules and regulations compliance. Companies that use AI in their ERP systems claim to have a 30% increase in user satisfaction and a 25% increase in productivity, owing to more enhanced personalized interfaces. In addition, approximately 40% of companies adopting AI-powered ERP solutions have applied predictive analysis in an effective manner to render demand planning, inventory management, and supply chain management more efficient. This has helped lower operational expenses on average by 20% owing to automation as well as better utilization of resources.

V. TECHNICAL CHALLENGES AND SOLUTIONS

Using AI for developing software is an enormous data-related hurdle. Zhang and Tsai (2003) mention that machine learning models need lots of high-quality data in order to work properly. However, Mhaskey (2024) mentions that the majority of companies face issues with data quality, which makes AI insights less reliable. This problem is especially acute in countries with few historical data or where data protection laws limit data availability, such as essential training data. Sarker (2021) states that data can be structured, semi-structured, or unstructured. All of these have preprocessing problems.

AI algorithms, especially deep learning algorithms, need lots of computing. Alenezi and Akour (2025) say that training sophisticated neural networks often needs special hardware like GPUs or TPUs, which might not be available to every development team. Campbell (2020) says that these computational demands can be very challenging, and can even slow down systems and reduce performance on large projects. Mhaskey (2024) points out that the maximization of resource utilization for edge deployments is a major AI deployment challenge, especially for resource-poor environments.

However, there are several problems associated with bringing AI capabilities into the existing software systems. Ramchand et al. (2021) pointed out that current software systems were not designed to incorporate AI and, thus, there are issues of integration . These integration complexities are compounded by what Mhaskey (2024) categorizes as ‘reactive’

nature of legacy systems that are centered on human activities and the past. As for the risks, Asaad and Avksentieva (2023) mention that the optimisation of AI systems does not take into account the background knowledge in a legacy code, which may lead to integration problems and have certain drawbacks and bottlenecks.

VI. ETHICAL AND PRACTICAL CONSIDERATIONS

Integrating AI into the software development process presents some issues on privacy and security. Mhaskey (2024) also stated that the AI trust, risk, and security management market will be USD 17.82 billion by 2030, making it evident that such problems should be solved. The data fed into the AI models can be of the personal nature and therefore it raises questions on how that data can be protected from breaches. According to Rahmaniar et al. (2023), it is apparent that the information processed and managed by such AI systems is highly sensitive and restricted to areas such as healthcare and finance, which makes it necessary to enhance the security of users' data.

Most AI algorithms are a "black box," and it is difficult to observe how they function and comprehend them. Mhaskey (2024) states that the explainable AI market will grow from USD 5.52 billion in 2022 to USD 43.34 billion by 2032, indicating how significant this issue is. Sergeyuk et al. (2024) discovered that merely 23% of developers believe that AI-generated code is secure, which is a concern regarding how transparent AI systems are. Dhar (2024) points out that today's AI systems have immense trust issues since we are not able to identify whether the machine's hidden agendas align with ours or not, possibly causing untoward problems in essential systems.

As more people utilize AI for software development, it is of very high importance that rules and regulations are adhered to. According to Mhaskey (2024), adhering to these rules is a very important facet of AI resource management, especially in industries that deal with sensitive information. Haleem et al. (2022) comments that AI has the capability of maintaining data confidentiality and ownership but that policymakers have to be very careful while bringing in regulations to emerging technologies. They have to find a middle path between protecting and innovating. Regulations like GDPR require stringent rules on the handling of data and how the decisions are reached by computers. The developers will have to implement such laws on their AI systems so as not to attract court challenges and lose consumer trust.

VII. FUTURE TRENDS AND RESEARCH DIRECTIONS

The field of AI software development is growing quickly, with many interesting new ideas in the pipeline. Campbell (2020) reports on hybrid systems like SketchAdapt that combine pattern matching and reasoning to translate sketches into code. Alenezi and Akour (2025) report on the development of explainable AI (XAI) to respond to transparency problems, making AI systems more interpretable for developers and stakeholders. Marar (2024) sees that reinforcement learning and unsupervised learning are major fields of research. They can make software development more automatic and responsive. These methods allow systems to learn from their environment and experience without being programmed.

The future of AI software integration is that human developers and intelligent systems will work more harmoniously together. Mhaskey (2024) writes that the AutoML market is growing exponentially because of the high demand for efficient ways of generating and using models. Campbell (2020) believes that AutoML technology will soon be used for hybrid systems. This will allow for an entire system where simple user sketches can be automatically translated into complex applications with AI elements. Nielsen (2025) speaks about a major transition from command-based interfaces to intent-based outcome specification. In this case, developers will define the results they want instead of particular commands, which will change how software is developed.

There are significant research directions ahead to transcend current limitations in AI-driven software development. Zhang and Tsai (2003) see the necessity for more effective approaches to successfully encapsulate and convey domain knowledge in machine learning systems. Sarker (2021) indicates directions in developing more robust algorithms that can function well with limited or noisy training data. Sergeyuk et al. (2024) provide several areas for improvement. They include resolving the lack of project-size context in current AI assistants and creating models with improved context comprehension. There is also a tremendous opportunity to research ethical AI frameworks that make AI fair, transparent, and accountable. This research resolves issues posed by Dhar (2024) with respect to the alignment issue and possible dangers that come with increasingly autonomous AI systems.

VIII. CONCLUSION AND RECOMMENDATIONS

AI has transformed software development in various ways and enhanced it from start to finish. It improves design by better forecasting what will occur, changes coding through advanced generation systems, and enhances testing by finding more places and bugs. Although examples in various fields suggest that AI can be employed to address difficult problems, organizations should be cautious about issues related to data quality, processing capacity, and handling legacy systems. Firms must prioritize areas like test generation and documentation where developers most desire AI assistance, invest in robust data governance practices, and adopt hybrid solutions that leverage human capabilities with AI capabilities rather than trying for full automation. Future research must focus on developing context-aware AI systems, having standardized evaluation systems for AI code, improving explainability and transparency, addressing evolving software requirements, and adhering to interdisciplinary strategies that ensure AI systems benefit humans without violating ethical aspects while being under appropriate human oversight.

References

[1] Alenezi, M., & Akour, M. (2025). AI-Driven Innovations in Software Engineering: A Review of Current Practices and Future Directions. Applied Sciences, 15(3), 1344. https://doi.org/10.3390/app15031344

[2] Asaad, J., & Avksentieva, E. (2023). Review of ways to apply machine learning methods in software engineering. E3S Web of Conferences, 449, 07018–07018. https://doi.org/10.1051/e3sconf/202344907018

[3] Bharadiya, J., Thomas, R. K., & Ahmed, F. (2023, June 6). Rise of Artificial Intelligence in Business and Industry. ResearchGate; Sciencedomain International. https://www.researchgate.net/publication/371307024_Rise_of_Artificial_Intelligence_in_Business_and_Industry

[4] Campbell, M. (2020). Automated Coding: The Quest to Develop Programs That Write Programs. Computer, 53(2), 80–82. https://doi.org/10.1109/MC.2019.2957958

[5] Dhar, V. (2024). The Paradigm Shifts in Artificial Intelligence. Communications of the ACM, 67(11), 50-59. https://doi.org/10.1145/3664804

[6] Garikapati, D., & Shetiya, S. S. (2024). Autonomous Vehicles: Evolution of Artificial Intelligence and the Current Industry Landscape. Big Data and Cognitive Computing, 8(4), 42. https://doi.org/10.3390/bdcc8040042

[7] Garikapati, D., & Shetiya, S. S. (2024). Autonomous Vehicles: Evolution of Artificial Intelligence and the Current Industry Landscape. Big Data and Cognitive Computing, 8(4), 42. https://doi.org/10.3390/bdcc8040042

[8] Güneşdoğdu, E. (2024, November 11). The Evolution of AI in Software Development: From Assistance to Full Automation. Medium. https://medium.com/@ensargnsdogdu/the-evolution-of-ai-in-software-development-from-assistance-to-full-automation-322b323121d6

[9] Haleem, A., Javaid, M., Qadri, M. A., Singh, R. P., & Suman, R. (2022). Artificial intelligence (AI) applications for marketing: A literature-based study. International Journal of Intelligent Networks, 3(3), 119–132. https://doi.org/10.1016/j.ijin.2022.08.005

[10] Marar, H. W. (2024). Advancements in software engineering using AI. Computer Software and Media Applications, 6(1), 3906–3906. https://doi.org/10.24294/csma.v6i1.3906

[11] Mhaskey, S. V. (2024). Integration of Artificial Intelligence (AI) in Enterprise Resource Planning (ERP) Systems: Opportunities, Challenges, and Implications. International Journal of Computer Engineering in Research Trends, 11(12). https://doi.org/10.22362/ijcert/2024/v11/i12/v11i1201

[12] Nielsen, J. (2025). AI: First New UI Paradigm in 60 Years. Nielsen Norman Group. https://www.nngroup.com/articles/ai-paradigm/

[13] Rahmaniar, W., Maarif, A., Haq, Q. M. ul, & Iskandar, M. E. (2023, August 23). AI in Industry: Real-World Applications and Case Studies. Www.techrxiv.org. https://doi.org/10.36227/techrxiv.23993565.v1

[14] Ramchand, S., Shaikh, S., & Alam, I. (2021). Role of Artificial Intelligence in Software Quality Assurance. 125–136. https://doi.org/10.1007/978-3-030-82196-8_10

[15] Rashid, A. B., & Kausik, A. K. (2024). AI Revolutionizing Industries Worldwide: a Comprehensive Overview of Its Diverse Applications. Hybrid Advances, 7(100277), 100277–100277. https://doi.org/10.1016/j.hybadv.2024.100277

[16] Salerno, P. (2024, November 15). The Impact of AI and Automation on Software Development: A Deep Dive. IEEE Chicago Section. https://ieeechicago.org/the-impact-of-ai-and-automation-on-software-development-a-deep-dive/

[17] Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. SN Computer Science, 2(3), 1–21. Springer. https://doi.org/10.1007/s42979-021-00592-x

[18] Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. SN Computer Science, 2(3), 1–21. https://doi.org/10.1007/s42979-021-00592-x

[19] Sergeyuk, A., Golubev, Y., Bryksin, T., & Ahmed, I. (2024). Using Ai-Based Coding Assistants in Practice: State of Affairs, Perceptions, and Ways Forward. ArXiv (Cornell University). https://doi.org/10.2139/ssrn.4900362

[20] Upadhyaya, N. (2022, August 15). The Role of Artificial Intelligence in Software Development: A Literature Review. https://doi.org/10.13140/RG.2.2.12291.92965

[21] Zhang, D., & Tsai, J. J. P. (2003). Machine Learning and Software Engineering. Software Quality Journal, 11(2), 87–119. https://doi.org/10.1023/a:1023760326768

[22] Zhang, Q. (2024). The Role of Artificial Intelligence in Modern Software Engineering. Applied and Computational Engineering, 97(1), 18–23. https://doi.org/10.54254/2755-2721/97/20241339