ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

NEXT-GEN AI DECISION SUPPORT IN SOFTWARE ENGINEERING: LEVERAGING COMPLEXITY WITH QUANTUM COMPUTING, DEVOPS AUTOMATION, AND RESPONSIBLE MACHINE LEARNING

Simran

Dept. of Computer Science and Engineering Chandigarh University Mohali, Punjab, India

Sonu Kumar

Dept. of Computer Science and Engineering, Chandigarh University, Mohali, Punjab, India

Er Priyanka Devi

Dept. of Computer Science and Engineering Chandigarh University Mohali, Punjab, India

ABSTRACT—

Utilizing next-generation AI decision support systems driven by quantum computing, DevOps automation, and responsible machine learning is the way of the future as software engineering struggles with increasing complexity. Current computational boundaries are broken by quantum computing, which makes it possible to solve complex algorithms and large amounts of data quickly. DevOps automation integrates AI to remove bottlenecks and increase efficiency while orchestrating smooth cooperation. Fundamentally, responsible machine learning acts as the moral compass, guaranteeing that decisions made by AI are open, accountable, and consistent with social norms. When these three ideas come together, software development and management undergo a radical change. They handle the complexities of contemporary software while improving speed and dependability by streamlining decision- making. This study presents a framework for Next-Generation AI Decision Support in Software Engineering, resulting in an AI- driven "software engineer" capable of managing the difficulties of modern development. This intelligent system uses quantum computing to solve complicated software design difficulties with its improved processing capabilities. Integrating DevOps automation to speed up continuous integration, delivery, and deployment ensures agile, real-time responsiveness to changing needs. The incorporation of responsible machine learning concepts, which address concerns of bias and interpretability while maintaining privacy, enables ethical and transparent decision-making. Through case studies and performance evaluations, this AI-driven software engineer has proven to optimize resources, reduce development timelines, and improve software quality, laying the groundwork for future autonomous engineering systems that support complex decision-making and operational efficacy.

***Index Terms*—Ethical AI, Quantum Computing, DevOps Au- tomation, Decision Support Systems, Autonomous engineering systems, Predictive Analytics, Automated Workflows**

I. INTRODUCTION

The field of software engineering faces significant ob- stacles in an environment characterized by rapid technical advancement and growing complexity. Modern expectations create a strain on conventional methodologies, as software systems must not only be resilient and efficient but also fluidly adapt to the changing needs of users and the marketplace. New approaches that improve decision-making and streamline development workflows are therefore desperately needed by **Goyal, D. [1].** An important factor in this change is the emergence of next generation artificial intelligence (AI) deci- sion support systems. These technologies unleash tremendous potential to analyze large datasets and solve complicated issues at previously unheard-of rates by utilizing the amaz- ing powers of quantum computing. Software engineers will be able to tackle difficulties that were previously thought to be insurmountable thanks to the revolutionary potential of quantum computing to redefine computational bounds. This work analyzes the intersection of these transforma- tive technologies—quantum computing, DevOps automation, and responsible machine learning—within the world of soft- ware engineering by **Marar, H. W. (2024)[11]**. By examining their synergistic effects on decision support systems, we hope to illuminate a way toward more ethically sound, intelligent, and efficient software development techniques that are equipped to meet the problems of the digital age.Software development and IT operations can be cleverly enhanced by integrating artificial intelligence (AI) [1] and machine learning (ML) [2] into DevOps procedures, or AIOps. The complexity, speed, and security requirements of modern software development make AIOps indispensable [3]. AI-Driven DevOps, which blends DevOps with AI methods, use machine learning to increase the productivity and continuity of software development. This tactic is essential for enhancing the Agile methodology throughout the software development lifecycle.

It is becoming increasingly challenging to oversee large projects, maintain high standards of quality, and meet accelerated delivery schedules due to the rapid advancement of software engineering by **Mulder, J. (2020) [4]** Traditional decision-support systems are sometimes unable to provide actionable information in real-time and cannot meet the diverse demands of modern development settings.

This study suggests a solution to these problems: Generation AI Decision Support in Software Engineering. The foundation of this strategy is the creation of an AI-driven "software engineer"—an autonomous, intelligent system designed to handle the intricate tasks of software design, development, and deployment. This system incorporates three important technology advancements: quantum computing, DevOps automation, and responsible machine learning. Complex design and optimization problems that were previously believed to be impossible can now be solved thanks to quantum computing's increased processing capacity by **Asil, M.[12].** Through the facilitation of seamless integration, continuous delivery, and efficient deployment, DevOps automation facilitates agile responsiveness to real-time demands and resource fluctuations. Finally, by guaranteeing that the AI-driven software developer operates with transparency, equity, and data privacy, responsible machine learning principles solve significant ethical challenges and enhance confidence. Through this integrated approach, the framework aims to improve software quality, expedite development timelines, and enable more concise decision- making. by **Tatineni,S. (2024) [7].** By examining use cases and performance outcomes to demonstrate the feasibility and potential impact of an AI-driven software engineer, this study establishes the foundation for more autonomous and successful software engineering methodologies.This research attempts to provide a framework for Next-Generation AI Decision Support in Software Engineering in order to produce an AI-driven "software engineer." This ingenious solution streamlines processes like resource allocation, design, and deployment by handling complex software development tasks independently by **Raghavendran, R. (2024)[18].** The study's objective is to leverage the promise of quantum computing, DevOps automation, and responsible machine learning to enhance software quality, reduce development times, and boost decision-making accuracy. The ultimate goal is to establish the foundation for autonomous, ethical, and efficient software engineering decision support that satisfies current and future industry standards.

II. Ease of Use

Achieving ease of use is crucial for user engagement and successful deployment in software engineering and next-generation AI decision support systems. The following points outline the essential components that provide an easy-to-understand and user-friendly counter.

A. Intuitive Interfaces:

By designing user interfaces that are easy to understand and intuitive, you may lower the learning curve for new users. Use visually appealing and clear layouts to increase understanding and accessibility..

B. User-Centric Design:

Prioritize the needs and preferences of end users by adopting a user-centered design approach throughout the development process. Make features easily navigable and available to a many different parties, including as engineers and project managers.

C. Comprehensive Documentation:

To assist users in making efficient use of the technology, provide comprehensive, understandable documentation and tutorials. Provide users with resources for support and contextual help so they can resolve problems and maximize functionality..

D. Streamlined Workflows:

Use DevOps automation to speed up cooperation procedures and lower obstacles to project completion. Promote rapid feedback and integration cycles so that teams may swiftly make the required corrections.

E. Transparency in AI Decision-Making:

Simplify teamwork processes and reduce project completion barriers by implementing DevOps automation. Encourage rapid cycles of integration and feedback so that teams can make necessary adjustments quickly and efficiently.

F. Responsive Feedback Mechanisms:

Incorporate real-time feedback features that guide users through tasks and inform them of system status. Let individuals offer feedback on usability so that ongoing efforts to improve can be directed.

G. Ethical Considerations:

Incorporate moral machine learning principles to promote accountability, fairness, and transparency in AI decisions. Promote a collaborative environment where AI insights and human judgment work together to generate better outcomes. By emphasizing these elements, next-generation AI-based decision support systems can significantly increase user engagement, speed up software engineering processes, and ultimately lead to more ethical and practically sound decisions.

III. **LITERATURE REVIEW**

The intersection of responsible machine learning, DevOps automation, quantum computing, and artificial intelligence (AI) has been the subject of much research in recent years. Each of these domains contributes in a unique way to the enhancement of software engineering decision support systems. Recent research has investigated the use of AI approaches to enhance a number of software development processes, such as intelligent code generation, automated testing, and predictive analytics for project management. Research by **Amani et al. (2022)** illustrates how past project data may be analyzed by machine learning algorithms to predict possible dangers and optimize resource allocation. The groundwork for incorporating cutting-edge AI capabilities into decision-making frameworks is laid by this seminal work. The ability of quantum computing to solve intricate computational issues that are beyond the capabilities of traditional computers is well known. New developments, such as those presented by **Arute et al. (2019)**, emphasize the superiority of quantum mechanics in resolving particular optimization issues. This In software engineering, new technologies provide new ways to handle complicated algorithms and process large datasets, especially in areas like performance analysis and code optimization. Through the promotion of continuous integration and delivery (CI/CD), the DevOps movement has significantly changed software development techniques. Research by **Kim et al. (2021)** highlights how development teams can collaborate more effectively, ship software faster, and produce higher-quality software by using automation tools. DevOps automation facilitates the successful integration of AI-driven decision support systems by optimizing workflows. The ethical ramifications of machine learning have drawn more attention as our reliance on AI grows. Studies by **Doshi-Velez and Kim (2017)** promote frameworks for AI systems that give fairness, accountability, and transparency first priority. To guarantee that AI technologies are in line with society values and reduce biases, these principles must be incorporated into decision support systems by **Rantanen, J. (2024) [6].** In conclusion, there is a great chance to advance software engineering decision support systems as a result of these domains coming together. Researchers and practitioners can create novel solutions that tackle the challenges of contemporary software development while advancing moral behavior by utilizing the advantages of artificial intelligence (AI), quantum computing, DevOps automation, and responsible machine learning by **Chaccour et al(2024)[3].** It is anticipated that next-generation AI decision support systems in software engineering will experience a radical transformation when quantum computing, DevOps automation, and responsible machine learning combine. As software projects become more complicated, AI-driven decision support systems will be crucial for navigating challenging development environments, optimizing workflows, and ensuring ethical outcomes by **George et alc(2023).[4]** Quantum computing has the potential to drastically change the role of AI in software engineering. Our ability to tackle difficult computational problems like large- scale optimization, cryptography, and deep learning model training will be greatly enhanced by the development of quantum technology. Compared to AI decision support systems with conventional capabilities, those with quantum capabilities will be able to process massive datasets, run simulations, and reach conclusions far more swiftly by **Kumble, G. P. (2020)[7].** This acceleration will alter how resources are distributed, projects are managed, and algorithms function, enabling software engineers to solve problems that are currently believed to be unsolvable. As DevOps automation continues to advance, AI-driven insights will be incorporated into every phase of the development lifecycle. In the future, there will be more intelligent automation solutions that can manage testing, deployments, and upgrades independently using real-time data and predictive analytics from AI decision systems by **Marar, H.**

W. (2024)[11]. This will allow for faster, more reliable software delivery, reduced downtime, and more flexible adaptability to user demands. AI-enhanced DevOps allows software engineers to iterate rapidly in a continuous feedback loop, minimizing errors and ensuring high-quality software delivery. Responsible machine learning will increasingly influence the ethical underpinnings of AI systems in software engineering. It will be crucial to ensure accountability, transparency, and equity as AI becomes more autonomous in making decisions that impact software development and business outcomes by **Jensen, A. (2024)[17].** Future AI decision support systems will include more robust security features to preserve data privacy, reduce biases, and comply with evolving regulatory mandates. Decision support systems will undoubtedly maximize performance while abiding by moral principles and societal norms if responsible AI techniques are used. Together, quantum computing, DevOps automation, and responsible machine learning in AI decision support systems will completely transform software development, testing, and deployment procedures in the future. Engineers will be better equipped to handle more complex issues, make wiser choices, and maintain moral principles throughout the development process if they have more authority. All industries will see innovation sparked by this next-generation approach, which will transform how software satisfies user needs and corporate objectives. The development of AI decision support systems into essential partners in the production of intelligent, high-performing, and ethically sound software solutions will be made possible by the advancement of these technologies.

IV. **RELATED WORK**

Responsible machine learning (RML) is essential to the ethical, open, and fair operation of AI decision support systems in software engineering. by **Karlovs-Karlovskis,U. (2024).[19].** As software engineers increasingly rely on AI for critical decision-making, the ethical implications of their choices are becoming increasingly significant. RML emphasizes the necessity of ethical frameworks in the creation and use of AI algorithms, allowing engineers to create efficient models free from biases that might lead to unfair outcomes. Transparency is highly valued in RML; users should understand the procedures AI models follow to produce their output. This will enable them to gradually alter their models and make the required corrections.by **Raghavendran, R. (2024)[17]**. Encouraging multidisciplinary cooperation between domain experts, ethicists, and software engineers also makes it easier to fully handle ethical issues. Transparency, compliance, ethical decision-making, cooperation, and ongoing progress are all given top priority in responsible machine learning. Quantum-enhanced machine learning algorithms can handle and analyze data more effectively, resulting in predictive models with improved accuracy and performance by **Kumble, G. P. (2020)[13].** This capability allows software engineers to make quick, precise, data-driven decisions, which significantly accelerates the development process.Managing the complexity of modern software engineering, such as integrating with DevOps processes, is another area in which quantum computing has promise by **Tatineni, S. (2024)[12].** Through data processing optimization and model training time reduction, quantum computing contributes to the acceleration of continuous integration and delivery (CI/CD) pipelines. Faster feedback loops and more adaptable responses to project requirements are made possible by this.

V. Quantum-DevOps ML Integration Framework for Autonomous Software Engineering (QDMASE)

A The Quantum-DevOps ML Integration Framework for Autonomous Software Engineering (QDMASE) is a novel approach designed to manage the growing complexity of software development in an era of rapid technological advancement. As the demands on software systems rise, traditional methods often fall short in managing complex operations. Quantum computing, DevOps automation, and responsible machine learning are the three key components of QDMASE, which combine to provide an intelligent, self- governing software engineer capable of optimizing decision-making and increasing operational efficacy. The quantum computing layer, the central part of QDMASE, performs complex computations at previously unheard-of speeds by utilizing the unique characteristics of quantum physics.This layer enables the AI-driven software engineer to perform complex tasks, such as data analysis, optimization problems, and software behavior simulation, that are challenging for traditional computing approaches. In addition to significantly reducing the time required for critical computations and decision-making, the development of quantum algorithms has made it possible to handle large datasets and difficult software design challenges more efficiently.A complementing element of the quantum layer is the DevOps automation layer, which integrates continuous integration, delivery, and deployment (CI/CD) methodologies into the software engineering lifecycle. This layer ensures smooth collaboration between the operations and development teams by automating repetitive processes. By utilizing strategies such as orchestration and containerization, QDMASE facilitates rapid iterations, immediate feedback, and adaptable responses to changing needs. Automating these processes not only boosts output but also lowers human error, resulting in a more reliable and stable software development environment.The responsible machine learning layer, the third component, emphasizes ethical AI practices by putting policies in place that ensure transparency, interpretability, and equity in decision-making. In order to establish confidence in the AI-driven software developer, this layer uses explainable AI (XAI) methodologies to correct any biases and guarantee adherence to data protection regulations. This layer ensures smooth collaboration between the operations and development teams by automating repetitive processes. Using technologyQDMASE enhances decision-making quality, complies with social and moral standards, and encourages accountability in automated processes by integrating ethical machine learning techniques.Together, these components form a logical framework that empowers the AI-driven software engineer to independently handle difficult software development tasks, make the most use of available resources, and improve the overall caliber of software. As a result, the Quantum-DevOps ML Integration Framework for Autonomous Software Engineering positions itself as a crucial tool for companies seeking to leverage advanced technology to assist in making smarter decisions.

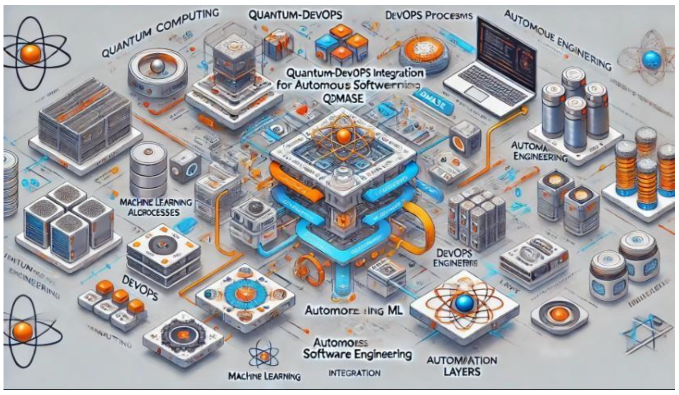


Fig. 1. ***Quantum-DevOps ML Integration Framework for Autonomous Software Engineering (QDMASE)***

Production and operational effectiveness in the dynamic field of software engineering

A. *Quantum Computing Layer*

1. *Purpose: The Quantum Computing Layer solves complex issues that are difficult for classical systems by utilizing the special powers of quantum mechanics. This layer seeks to address issues with computing efficiency and speed in software engineering jobs by utilizing quantum principles.*
2. *Functionality: This layer enables the execution of more complex data analysis techniques, simulations of program behavior, and faster optimization algorithms. Quantum algorithms such as Grover's and Shor's can speed up and improve the accuracy of critical tasks including resource allocation, project scheduling, and risk assessment. The ability to examine massive amounts of data simultaneously enables more sophisticated modeling and predictive analysis, which are necessary for making informed decisions in real time.*
3. *Impact: The incorporation of quantum computing into the QD-MASE framework significantly reduces the time and resources needed for critical software engineering processes. This capacity enhances agility and reactivity to changing project objectives while enabling teams to innovate and swiftly adapt to new challenges in the software development lifecycle.*

B. *DevOps Automation Layer*

4. *PurposeBy combining development and operations, the DevOps Automation Layer seeks to optimize the software development lifecycle. By breaking down silos between teams, this layer enhances communication and teamwork, leading to more efficient procedures.*
5. *Functionality: By combining development and operations, the DevOps Automation Layer seeks to optimize the software development lifecycle. By breaking down silos between teams, this layer enhances communication and teamwork, leading to more efficient procedures.*
6. *Impact: By automating these processes, the DevOps Automation Layer significantly reduces the likelihood of errors and boosts overall efficiency. It also facilitates quicker responses to evolving requirements, which ultimately leads to software releases of greater caliber. Development cycles become more predictable and aligned with business objectives through the use of automation.*

C. *Machine Learning Layer*

7. *Purpose: The Machine Learning Layer incorporates responsible machine learning concepts to ensure that the AI-driven decisions made inside the framework are transparent, ethical, and reliable. This layer aims to lessen the risks associated with biased or opaque AI systems.*
8. *Functionality: It uses algorithms that prioritize interoperability, minimize bias, and protect data privacy. To boost the system's trustworthiness, strategies like explainable AI (XAI) and model validation are used. By ensuring that decisions can be thoroughly understood and scrutinized, this layer promotes accountability in AI systems.*
9. *Impact: By using ethical machine learning techniques, QDMASE addresses significant ethical concerns in AI decision-making. This focus ensures that the independent software developer operates ethically, producing fair outcomes and fostering stakeholder trust. As a result, the framework is better able to comply with legal standards and societal norms.*

D. *Integration and Feedback Mechanism*

1. *Purpose: Communication, data interchange, and continuous learning are made easier by the Integration and Feedback Mechanism, which guarantees smooth interaction between the three layers of QD-MASE. This process is essential to preserving the framework's efficiency and coherence.*
2. *Functionality: It consists of feedback loops and ongoing monitoring that collect deployment outcomes, user interactions, and performance data. To improve algorithms, streamline procedures, and adjust to new demands, this data is examined. A closed-loop approach allows the framework to change in response to user feedback and actual performance.*
3. *Impact: By improving the system's learning capabilities, the feedback mechanism helps the AI-driven software developer make better decisions over time. This flexibility is necessary to keep up with evolving technology and shifting contexts, guaranteeing that the framework will continue to be applicable and efficient in producing high-caliber software solutions.*

VI. BUILDING AI SOFTWARE SYSTEMS THROUGH THE QDMASE FRAMEWORK

The way systems are created, maintained, and improved has been completely transformed by the incorporation of artificial intelligence (AI) into software engineering. Leading this change is the Quantum-DevOps ML Integration Framework for Autonomous Software Engineering (QD- MASE), which provides a thorough method for creating intelligent software systems. QDMASE optimizes and streamlines the software development lifecycle by combining responsible machine learning, DevOps automation, and quantum computing in a novel way. The quantum computing layer at the heart of QDMASE uses the unmatched processing power of quantum algorithms to solve challenging issues that conventional computing finds difficult to handle.Developers can swiftly make well-informed judgments because to this layer's facilitation of sophisticated data analysis, speedy optimization, and software behavior modeling. For example, the time and computational resources needed can be greatly decreased by performing tasks like resource allocation, project scheduling, and risk assessment more effectively.A key component of QDMASE is the De- vOps automation layer, which enhances the quantum capabilities. This layer promotes cooperation between the development and operations teams by automating continuous integration, delivery, and deployment (CI/CD) procedures. By utilizing techniques and tools such as orchestration, containerization, and Infrastructure as Code (IaC), QDMASE reduces human error and improves operational efficiency.

Rapid iterations, real-time feedback, and quick reactions to changing project requirements are made possible by this efficient process, which eventually results in software releases that are both faster and of greater quality.The responsible machine learning layer is another essential part of QDMASE that makes sure AI-driven judgments are trustworthy, transparent, and moral. This layer tackles important ethical issues related to automated systems by implementing strategies from explainable AI (XAI). It gives interpretability and data protection first priority, allowing developers to produce AI models that are not only efficient but also compliant with legal and social norms. By emphasizing ethical behavior, AI applications gain credibility and become more useful and accepted in practical situations. Furthermore, smooth communication between the three layers is guaranteed by QDMASE's integration and feedback mechanism.

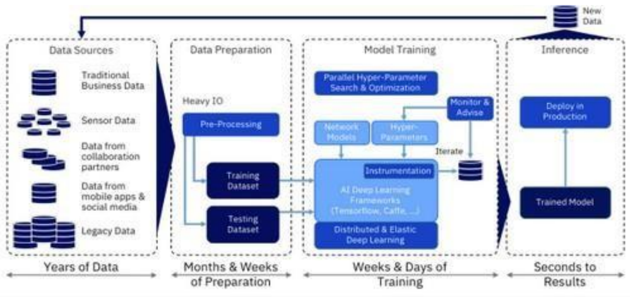
AI systems may learn and adapt over time thanks to continuous monitoring and feedback loops, which make it easier to gather performance data and user interactions. Decision-making skills are improved and software systems are guaranteed to adapt to changing user needs thanks to this iterative learning process.Using the QDMASE framework to build AI software systems is a comprehensive method of contemporary software engineering. QDMASE enables developers to design intelligent, flexible, and morally sound software solutions that can prosper in a technologically complex world by utilizing the power of quantum computing, DevOps automation, and responsible machine learning.

A. AI Algorithm Architecture

With efficient data analysis, AI-based algorithms may carry out computations, resolve issues, and produce useful results. An AI algorithm architecture gives a general idea of how an algorithm would function on its own and produce outcomes. The AI system is designed to identify whether an animal is a horse, crocodile, reptile, dog, or elephant. More sophisticated Based on the input given, AI algorithms are able to recognize spoken or written language and respond. AI Algorithm Architecture is the organized framework that defines how artificial intelligence algorithms are designed, implemented, and integrated into systems. Typically included layers are data input, preprocessing, model training, inference, and output. By including a range of algorithms, such as reinforcement learning, supervised learning, and unsupervised learning, the architecture provides flexibility in addressing a number of problems. Additionally, it includes components for feature extraction, model evaluation, and optimization techniques. This systematic approach, which ensures scalability, maintainability, and efficiency, enables the creation of robust AI systems that can adapt to changing demands and complexity in real-world scenarios.

Fig. 2. AI architecture for governing and scaling AI efforts

The development of strong AI systems that can adjust to shifting needs and complexity in real-world situations is made



possible by this methodical methodology, which guarantees scalability, maintainability, and efficiency.

B. Machine Learning (ML) Architecture

A subfield of artificial intelligence called machine learning (ML) uses mathematical algorithms to create predictive business models. Value business data can be actively analyzed by ML-based algorithms, which can then forecast particular business situations. An overview of process analysis, learning, and validation is given by an ML-based app architecture. A organized framework known as machine learning architecture (ML architecture) describes the elements and procedures needed to develop, train, and implement machine learning models. Data collection, preprocessing, feature engineering, model training, evaluation, deployment, and monitoring are some of the layers that are usually included. Through feature optimization, the architecture enhances model performance, guarantees effective data handling, and evaluates the efficacy of trained models using a range of indicators.Additionally, it facilitates the integration of models into production environments, allowing for continuous performance monitoring and real-time forecasting, which ultimately results in the creation of dependable and scalable AI solutions.

C. Deep Learning Architecture

Deep Complex algorithms are used in deep learning-based business applications to generate accurate projections. Compared to machine learning, deep learning-based business applications are more effective at analyzing large amounts of data. Each layer that makes up a deep learning application processes and analyzes data in a unique way. The deep learning architecture describes each layer, overall organization, and application behavior.Intricate patterns in data can be modeled using a structured neural network framework known as the "deep learning architecture." It typically consists of multiple layers, including input, hidden, and output layers. Because each hidden layer has several neurons that use activation functions to alter inputs, the network is able to learn hierarchical representations of data. It typically consists of multiple layers, including input, hidden, and output layers. The many neurons in each hidden layer that alter inputs using activation functions enable the network to learn hierarchical representations of data. By employing techniques like dropout, batch normalization, and transfer learning, these architectures enhance performance and pave the way for powerful applications in a range of domains, including speech, vision, and text.

D. Bots Architecture

These days, bots are used in many different industries to automate certain tasks and boost overall business productivity. These bot-based solutions employ basic algorithm logic to more efficiently execute simple, complex, and repetitive business operations. Advances in technology have made it possible for AI-based bots to mimic human behavior and decision-making abilities. The bots' architecture provides a clear picture of how the software modifications and the system as a whole will function and interact."Bot architecture" refers to the hierarchical structure that outlines how bots operate, interact, and integrate into other systems. Typically, it consists of many crucial components: An interface for user interaction, a backend logic layer that manages requests and performs activities, and natural language processing (NLP) for understanding and interpreting user input. It also has APIs for communicating with databases and other services to make data retrieval and storage easier. Bots may also include a Monitoring Layer to monitor user interaction and performance, as well as a Machine Learning component to enhance responses over time, in order to guarantee dependability and ongoing progress.

BUILDING A MACHINE LEARNING-BASED AI SOFTWARE SYSTEM

Creating a machine learning-based AI software system from data collection to deployment calls for a systematic approach that combines several elements. The first stage is to determine the needs and goals of the system. This include identifying the specific problem that the system is intended to solve, like as classification, regression, or clustering, and establishing measurable goals like speed, accuracy, and scalability. Next, data collection is crucial. Relevant data can be obtained from a range of sources, including databases, APIs, and publicly accessible datasets. Ensuring compliance with data privacy regulations is essential during this stage. Data preparation is done after the data is collected. This include correcting incorrect data, removing duplicates, and adding missing values.It can also be necessary to standardize or normalize the data and convert categorical variables into numerical representations.Feature engineering is carried out following preprocessing.

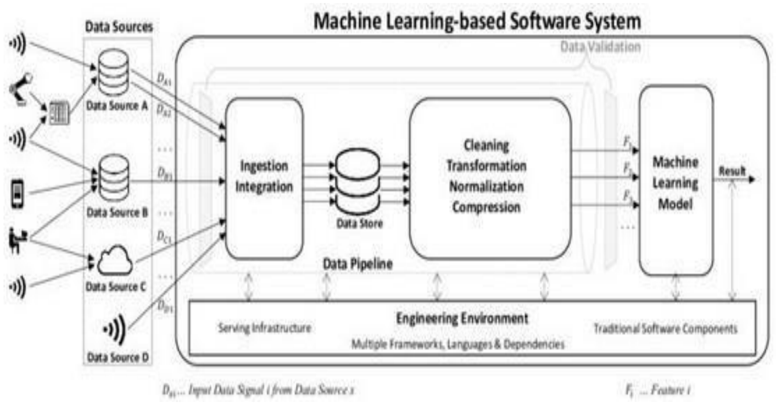


Fig. 3. Machine Learning-based Software System

Using techniques like polynomial expansion or aggregation, additional features are created in this step, along with the most relevant attributes that improve the model's performance. Effective feature engineering enhances the model's learning capacity.When the dataset is ready, the following step is to choose a model. Choosing the appropriate machine learning algorithms is crucial, depending on the nature of the issue. Neural networks, support vector machines, and decision trees are just a few of the many techniques that can be researched. Trying out various configurations and algorithms aids in determining which model performs the best.The model training step starts after an algorithm is selected. The dataset is often separated into training, validation, and test sets to ensure a comprehensive evaluation. Using techniques like polynomial expansion or aggregation, additional features are created in this step, along with the most relevant attributes that improve the model's performance. Effective feature engineering enhances the model's learning capacity.When the dataset is ready, the following step is to choose a model. Choosing the appropriate machine learning algorithms is crucial, depending on the nature of the issue. Neural networks, support vector machines, and decision trees are just a few of the many techniques that can be researched. The best model can be found by experimenting with different setups and algorithms.Once an algorithm has been chosen, the model training phase begins. The dataset is often separated into training, validation, and test sets to ensure a comprehensive evaluation.Model updates and enhancements are made possible by the collection of performance data through the establishment of a feedback loop.Machine learning-based AI software development is a multi-step process that requires careful planning and execution. By following a methodical methodology, developers can create dependable AI systems that effectively manage difficult issues and adapt to changing requirements.

VII. **PAIR MODELING ENHANCES SOFTWARE DEVELOPMENT EFFICIENCY:**

promoting cooperation, improving code quality, facilitating knowledge exchange, and adhering to Agile standards, pair modeling through approaches like pair programming boosts software development effectiveness. Pair programming's benefits as software engineering increasingly incorporates cutting-edge technologies like AI-driven tools and quantum computing This is made increasingly clearer when it comes to creating robust and efficient development environments.

Real-Time Code Review:

Pair programming, in which two developers work together, significantly reduces the likelihood of coding errors by facilitating error detection and immediate feedback.

A. Knowledge Sharing:

Enhanced Attention Focus and discipline are maintained when working in pairs. The collaborative setting lowers the possibility of distractions and motivates developers to remain involved.

B. Diverse Perspectives:

Combining many points of view results in more thorough problem-solving. Every developer contributes distinct perspectives that raise the solution's overall caliber.

C. Reduced Technical Debt:

With constant oversight, pair programming generates code that is clearer and simpler to maintain. This proactive approach eventually lowers technological debt.

D. Enhanced Collaboration:

Pair modeling promotes a culture of collaboration where communication is crucial. Developers who discuss strategy and solutions have better team dynamics. Flexibility in Response to Staff Changes When team members exchange information, projects become less dependent on individual developers. This resiliency is crucial to the project's success.

E. Higher Job Satisfaction:

By making coding more enjoyable and less alienating, pair programming's collaborative nature can boost job satisfaction.

F. Efficiency Gains:

Research suggests that pair programming can boost productivity by reducing the amount of time spent on debugging and rework due to the higher initial code quality, despite the seeming paradox.

G. Agile Alignment:

Agile methodologies and pair modeling facilitate ideas like flexibility and change-responsiveness, which are essential for modern software development. More experienced partners help junior developers by sharing their knowledge and experience. This enhances team performance and expedites the onboarding procedure.

VIII. **FUTURE OF NEXT-GEN AI DECISION SUPPORT IN** SOFTWARE ENGINEERING

It is anticipated that the future of next-generation AI decision support systems in software engineering will be completely transformed by the convergence of responsible machine learning, DevOps automation, and quantum computing. Quantum computing has the potential to revolutionize the application of AI to software engineering. As quantum technologies advance, they will greatly enhance complex computational problems such as deep learning model training, cryptography, and large-scale optimization. Quantum-enabled AI decision support systems will be able to process massive datasets, perform simulations, and come to conclusions far more swiftly than conventional systems.

IX. CONCLUSION

AI decision support systems are positioned as crucial to the future of software engineering by utilizing the complexity of quantum computing, DevOps automation, and responsible machine learning. These technologies offer an unparalleled chance to enhance decision-making, modify workflows, and boost productivity as development environments become more complicated. Because quantum computing can handle large datasets and solve difficult issues considerably faster than traditional systems, software developers now have effective tools for real-time data analysis, predictive model construction, and resource allocation. This would allow for faster decision-making and the resolution of problems that were previously thought to be unsolvable, greatly increasing the effectiveness of software systems and project management. A feedback loop that encourages teamwork and boosts innovation is produced when AI models are smoothly incorporated into operational contexts, eventually producing software that is more dependable and superior. The complexity of contemporary software development has been greatly addressed by the Quantum-DevOps ML Integration Framework for Autonomous Software Engineering (QDMASE). Combining the efficient procedures of DevOps automation, the moral precepts of responsible machine learning, and the revolutionary possibilities of quantum computing,

QDMASE builds a strong ecosystem for a self-sufficient software engineer powered by AI. This framework guarantees that the development process is flexible, effective, and morally sound in addition to improving decision-making skills. The unmatched processing power made possible by the integration of quantum computing can be used to solve complex software engineering challenges that are beyond the scope of conventional approaches. The QDMASE framework promotes itself as an essential tool for companies looking to leverage cutting-edge technologies for better results as the needs of software engineering continue to change. By promoting moral behavior, increasing operational effectiveness, and fostering creativity, QDMASE establishes the foundation for the future of autonomous software engineering. This approach advances the larger objective of creating reliable AI systems that can successfully negotiate the difficulties of a quickly evolving technological landscape and opens the door for more intelligent and sensitive development processes. In the end, QDMASE is a cutting- edge strategy that gives software engineering the instruments and techniques it need to prosper in a setting that is becoming more complex.

REFERENCES

1. Goyal, D. AI-Driven DevOps for Agile Excellence with Machine Learning.
2. Iosup, A., Kuipers, F., Varbanescu, A. L., Grosso, P., Trivedi, A., Rellermeyer, J., ... Regazzoni, F. (2022). Future Computer Systems and Networking Research in the Netherlands: A Manifesto.arXiv preprint arXiv:2206.03259.
3. Chaccour, C., Karapantelakis, A., Murphy, T., Dohler, M. (2024). Telecom's artificial general intelligence (agi)

vision: Beyond the genai frontier.IEEE Network.

4. George, B., Eric, R. (2023). Synergizing DevOps, Cloud Computing, and AI for Next-Gen RPA Solutions.International Journal of Advanced Engineering Technologies and Innovations,1(03), 51-65.
5. Bauskar, S. R., Reddy, M. S., Sarisa, M., KONKIMALLA, S.The Future of Cloud Computing Al-Driven Deep Learning and Neural Network Innovations. JEC PUBLICATION.
6. Rantanen, J. (2024). AI-enhanced web development.
7. Kumble, G. P. (2020).Practical Artificial Intelligence and Blockchain: A guide to converging blockchain and AI to build smart applications for new economies. Packt Publishing Ltd.
8. Mulder, J. (2020).Multi-Cloud Architecture and Governance: Leverage Azure, AWS, GCP, and VMware vSphere to build effective multi-cloud solutions. Packt Publishing Ltd.
9. Vaithiyanathan, R. K. ATOMIC-Book SPECTROSCOPY.
10. Srivastava, S. Utilizing AI systems to automate DevOps processes within the field of Software Engineering.
11. Marar, H. W. (2024). Advancements in software engineering using AI.Computer Software and Media Applications,6(1), 3906.
12. Tatineni, S. (2024).Integrating Artificial Intelligence with DevOps: Ad- vanced Techniques, Predictive Analytics, and Automation for Real-Time Optimization and Security in Modern Software Development. Libertatem Media Private Limited.
13. Asil, M. The Fusion of AI and DevOps: Transforming Software Devel- opment and Operations.
14. Liang, P., Wu, Y., Xu, Z., Xiao, S., Yuan, J. (2024). Enhancing Security in DevOps by Integrating Artificial Intelligence and Machine Learning.Journal of Theory and Practice of Engineering Science,4(02), 31-37.
15. Tatineni, S., Chakilam, N. V. (2024). Integrating Artificial Intelli- gence with DevOps for Intelligent Infrastructure Management: Opti- mizing Resource Allocation and Performance in Cloud-Native Applica- tions.Journal of Bioinformatics and Artificial Intelligence,4(1), 109-142.
16. Eramo, R., Said, B., Oriol, M., Bruneliere, H., Morales, S. (2024). An architecture for model-based and intelligent automation in De- vOps.Journal of Systems and Software,217, 112180.
17. Jensen, A. (2024). AI-Driven DevOps: Enhancing Automation with Machine Learning in AWS.Integrated Journal of Science and Technol- ogy,1(2).
18. Raghavendran, R. (2024). MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE IN DEVOPS: APPLICATIONS FOR PREDICTIVE ANALYTICS, ANOMALY DETECTION, AND AUTOMATED INCI- DENT.Journal ID,9471, 1297.
19. Karlovs-Karlovskis, U. (2024). Generative Artificial Intelligence Use in Optimising Software Engineering Process: A Systematic Literature Review.Applied Computer Systems,29(1), 68-77.
20. Zarour, M., Akour, M., Alenezi, M. (2024). Enhancing DevOps En- gineering Education Through System-Based Learning Approach.Open Education Studies,6(1), 20240012.
21. Karlovs-Karlovskis, U. (2024). Generative Artificial Intelligence Use in Optimising Software Engineering Process: A Systematic Literature Review.Applied Computer Systems,29(1), 68-77.