Computer Science and Information Technologies
Vol. 7, No. 3, November 2026, pp. 404~418
ISSN: 2722-3221, DOI: 10.11591/csit.v7i3.p404-418

A comparative review of modern large language model paradigms: GPT-4, BERT, Gemini, and DeepSeek

Kavish Sanghvi[1], Aparna S. Sharma[2], Surbhi Hooda[3]
[1]Department of Software Engineering, Stevens Institute of Technology, Hoboken, United States of America
[2]Daniel J. Epstein Department of Industrial and Systems Engineering, University of Southern California, Los Angeles, United States of America
[3]Department of Business, University of the Potomac, Washington, D.C., United States of America

Article Info

Article history:

Received Jun 12, 2025
Revised Jun 26, 2026
Accepted Jul 13, 2026

Keywords:

BERT
DeepSeek
Gemini
GPT-4
Large language models

ABSTRACT

This review provides comparative analysis of GPT-4, BERT (bidirectional encoder representations from transformers), Gemini, and DeepSeek large language models (LLM), focusing architectures, training methodologies, and real-world applications. The primary research question is: How do these models differ in design, strengths, limitations, and potential areas for enhancement? By addressing this question, the study aims to provide insights into the trade-offs and future directions for optimizing LLM performance and deployment. The analysis reveals GPT-4 excels in natural language generation and complex reasoning, supporting up to 128K tokens with moderate latency and higher costs making it effective for conversational artificial intelligence (AI). BERT excels bidirectional contextual understanding with smaller computational overhead and broad open-source adoption, effective for text classification. Gemini demonstrates superior multimodal integration processing text, image, audio, and code with context lengths up to 1M tokens, enabling cross-domain adaptability. DeepSeek excels in specialized domains like finance and programming, is optimized for efficiency and supports extended context windows exceeding 200K tokens. However, all models face challenges related to computational cost, hallucinations, and ethical concerns, necessitating further improvements. Despite advancements, LLMs continue to grapple with issues such as data bias, model interpretability, and responsible AI deployment. Future research should focus on hybrid model approaches, domain-specific fine-tuning, and transparency to mitigate risks while maximizing the transformative potential of LLMs in real-world applications.

Corresponding Author:

Kavish Sanghvi
Department of Software Engineering, Stevens Institute of Technology
1 Castle Point Ter, Hoboken 07030, New Jersey, United States of America
Email: sanghvi_kavish@yahoo.in

1. INTRODUCTION

Despite the rapid advancement of large language models (LLMs), selecting the most suitable model for a given task remains a complex challenge due to varied architectural designs, training data regimes, cost structures, and modality support. Existing studies often provide individual performance reports or benchmarks, but a comprehensive and practical comparison, especially covering recent models like GPT-4, Gemini, and DeepSeek, is lacking. This paper addresses this gap by evaluating leading LLMs across core dimensions such as architecture, domain specialization, multimodal capabilities, and real-world usability.

The novelty of this work lies in its synthesis of technical characteristics with deployment suitability, making it a practical reference for researchers and industry practitioners alike [1]–[5].

One of the biggest advances in artificial intelligence (AI) is LLMs, particularly natural language processing (NLP). Being able to construct human-like text in a variety of use cases and contexts has proven to be invaluable to industries such as finance, healthcare, and education. LLMs rely largely on neural network architecture, deep learning, and large datasets. Language models predict words in a sentence and, after seeing enough data, can suggest the most useful and context-related response. NLP LLMs require a lot of computing resources. Transformer architecture is a novel neural architecture and learning framework introduced by Vaswani *et al*. [6] making it easier to scale these models. LLMs also use a mechanism called self-attention, which allows the capture of complex relationships between components of a text, allowing them to resolve multiple ambiguities.

Among the most prominent LLMs in contemporary AI research and application are GPT-4, BERT (bidirectional encoder representations from transformers), Gemini, and DeepSeek. Each of these models embodies unique architectural innovations and training methodologies, tailored to specific tasks and performance objectives. GPT-4, an autoregressive model, builds upon its predecessors by employing a decoder-only transformer architecture optimized for long-context understanding and generative text production. It excels in tasks requiring creativity, problem-solving, and nuanced language generation [7]. In contrast, BERT adopts a bidirectional transformer encoder, which processes text in both directions simultaneously. This bidirectional processing enhances its contextual understanding, making BERT particularly effective for applications such as sentiment analysis, question answering, and entity recognition [8].

Gemini represents an evolution in LLMs by incorporating multimodal capabilities, enabling the integration of text, visual, and auditory data. This cross-modal reasoning ability facilitates more comprehensive AI applications in areas such as interactive education and medical diagnostics. Gemini's architecture extends traditional transformers to accommodate multimodal embeddings, drawing inspiration from models like contrastive language–image pre-training (CLIP) [9], which successfully merge visual and textual representations. On the other hand, DeepSeek, though less publicly documented, is designed for domain-specific applications. By incorporating targeted training strategies and architectural optimizations, DeepSeek achieves high efficiency in specialized tasks such as financial modeling and code analysis. Its domain-adaptive design aligns with principles of transfer learning, where fine-tuning enhances model performance on particular applications [10].

The increasing reliance on LLMs raises crucial questions about their underlying architectures, training methodologies, and real-world effectiveness. The primary research question guiding this comparative review is: How do GPT-4, BERT, Gemini, and DeepSeek differ in their structural design, learning paradigms, and application-specific performance, and what enhancements can be implemented to address their respective limitations? By systematically examining these models, this paper aims to elucidate the trade-offs inherent in each design, providing insights into their optimal use cases and future development trajectories. This review proceeds with a detailed exploration of key aspects that define LLM performance and applicability. Firstly, it delves into the technical underpinnings of each model, including their architectural frameworks, self-attention mechanisms, embedding strategies, and tokenization methods. Secondly, it analyzes their training methodologies, such as the nature of their datasets, pretraining objectives, fine-tuning approaches, and the role of reinforcement learning in optimizing performance. Thirdly, the review evaluates the practical applications of these models, considering their strengths and limitations in various domains. GPT-4's generative prowess, BERT's deep contextual understanding, Gemini's multimodal integration, and DeepSeek's domain-specific adaptations are critically compared to highlight their relative advantages.

Furthermore, this paper investigates the challenges associated with deploying LLMs, including computational costs, ethical considerations, bias mitigation, and interpretability. The computational demands of training large-scale transformer models necessitate significant hardware resources, raising concerns about accessibility and environmental impact [11]. Ethical considerations, such as bias in training data and potential misuse, remain pivotal challenges that demand robust mitigation strategies. Additionally, enhancing the interpretability of LLM outputs is crucial for fostering trust and reliability in AI-driven decision-making processes.

This comparative review aims to provide a comprehensive analysis of GPT-4, BERT, Gemini, and DeepSeek, delineating their architectural differences, training methodologies, and real-world applications. By addressing the central research question, this study seeks to inform researchers, developers, and industry practitioners about the strengths, limitations, and potential enhancements of these LLMs. The insights gained from this study not only contribute to the ongoing advancement of NLP technologies but also pave the way for future innovations in large-scale AI systems. Through continuous refinement and responsible deployment, LLMs can be harnessed to drive transformative progress across diverse sectors, enhancing the efficacy and

ethical considerations of AI applications in the modern world. Table 1 provides a high-level comparative overview of GPT-4, BERT, Gemini, and DeepSeek across five key parameters: architecture type, token limit, multimodal support, cost-efficiency, and accessibility. This summary table enables rapid identification of fundamental differences in model design and deployment characteristics.

Table 1. Comprehensive comparison between LLM models across key parameters

Model	Architecture	Token limit	Multimodal	Cost-efficiency	Access
GPT-4	Decoder-only	128K	Partial (GPT-4o)	High cost	Closed API
BERT	Bidirectional encoder	512	No	High efficiency	Open-source
Gemini	Multimodal Transformer	1M+	Yes	Moderate	Closed API
DeepSeek	Hybrid encoder-decoder	200K+	No	High efficiency	Open-source

Modern LLM development has converged toward four dominant architectural paradigms reflecting distinct optimization priorities: generative reasoning, contextual understanding, multimodal integration, and domain-efficient specialization. The autoregressive paradigm, exemplified by GPT-series models, prioritizes generative reasoning and conversational intelligence through decoder-only transformers [7], [8]. The encoder paradigm originating from BERT focuses on bidirectional contextual representation and efficient semantic understanding [9]. Multimodal architectures such as Gemini extend transformers to integrate heterogeneous modalities, including vision and audio [10], [11]. Domain-efficient LLMs such as DeepSeek emphasize specialized reasoning and reduced inference cost for technical domains [12], [13].

2. METHOD

This comparative review employs a systematic literature review methodology to analyze and synthesize the state-of-the-art in LLMs. The methodological framework was designed to ensure comprehensive coverage, objective evaluation, and reproducible findings across four prominent LLMs: GPT-4, BERT, Gemini, and DeepSeek.

2.1. Review framework and research questions

The review was structured around a central research question: how do GPT-4, BERT, Gemini, and DeepSeek differ in their structural design, learning paradigms, and application-specific performance, and what enhancements can be implemented to address their respective limitations? To address this question comprehensively, the following sub-questions were formulated:

- Architectural comparison: what are the fundamental architectural differences among the four models, including their transformer configurations, attention mechanisms, and tokenization strategies?
- Training methodology: how do training objectives, dataset compositions, and fine-tuning approaches differ across the models?
- Performance evaluation: what are the relative strengths and limitations of each model across various performance metrics and benchmark datasets?
- Application suitability: how do these models perform in domain-specific applications, including software engineering, finance, healthcare, and education?
- Deployment considerations: what are the trade-offs in terms of computational cost, latency, accessibility, and ethical considerations?

2.2. Large language models architecture

2.2.1. The transformer framework

At the core of modern LLMs lies the transformer architecture, which has fundamentally redefined how machines process sequential data. Introduced by Vaswani *et al*. [6], the transformer dispenses with traditional recurrent neural networks (RNNs) and instead employs multi-head self-attention mechanisms to capture dependencies across long sequences. Key elements of the transformer framework include:

- Self-attention mechanisms: self-attention allows models to dynamically weigh the relevance of each token in the input sequence. This mechanism, operating over multiple "heads," enables the model to capture different aspects of semantic and syntactic relationships simultaneously. The ability to consider all positions in the sequence concurrently represents a major departure from sequential processing paradigms [6].
- Positional encodings: given that transformers do not inherently encode positional information, positional encodings are added to the input embeddings [6]. These encodings provide the model with information

about the relative or absolute positions of tokens in the sequence, ensuring that the sequential nature of language is not lost.

- Feed-forward networks: each layer of the transformer architecture contains a fully connected feed-forward neural network working on the outputs generated by the attention process [6]. These networks help to capture complex non-linear transformations and improve the model's representational power.
- Layer normalization and residual connections: to ensure the stability of deep architectures during training, transformers incorporate layer normalization [14] and residual connections [15]. Such architectural properties help in maintaining the gradient flow and make it easy to train very deep models without suffering from vanishing and exploding gradient problems.

2.2.2. Embedding techniques

Embedding techniques are critical in transforming discrete tokens into continuous vector representations that capture semantic meaning. Previous approaches used static embeddings such as Word2Vec [12] and GloVe [13], leading to constant representations irrespective of the context. In modern LLMs, however, dynamic embeddings are utilized, allowing for adaptation in accordance with the lexical context. The evolution of embedding techniques includes:

- Static embeddings: early approaches such as Word2Vec [12] and GloVe [13] represent words as fixed vectors. These models capture semantic relationships but fail to account for polysemy and contextual nuance.
- Contextual embeddings: with the advent of transformers [6], embeddings are generated in a context-dependent manner. Models like BERT [8] and GPT-4 [1] update the representation of each token based on its context, thereby capturing subtle shifts in meaning.
- Subword tokenization: methods like byte pair encoding (BPE) [16] and SentencePiece [17] allow models to break down uncommon or complex words into subwords, making them easier to manage. This strategy does not only reduce the vocabulary size but makes the model better at dealing with out-of-vocabulary words.

2.2.3. Training datasets and pre-training objectives

The effectiveness of an LLM is largely determined by the quality and diversity of its training data, as well as the objectives used during pre-training. Common strategies include: i) training datasets: modern LLMs are trained on vast datasets comprising web pages, books, academic articles, and specialized texts [7]. Such diverse datasets ensure broad language coverage and the ability to generalize across domains; and ii) pre-training objectives:

- Autoregressive modeling: models like GPT-4 use an autoregressive objective, learning to predict the next token in a sequence. It develops powerful generative abilities in the model to construct contextually relevant and consistent text [7].
- Masked language modeling (MLM): BERT utilizes MLM, where the model predicts missing tokens that are masked in a sequence given the surrounding context. This method relies on a bidirectional approach, improving the model's comprehension of the structure of language and contextual meaning [8].
- Cross-modal objectives: in multimodal models like Gemini, training is extended to include stimuli beyond text, for example, images and sounds. Cross-modal objectives make certain that the model is taught to associate and incorporate different data modalities [2], [18].

2.3. LLMs capability comparison

This section compares representative LLMs, including GPT-4, Gemini, BERT, and DeepSeek, in terms of performance and capabilities, cost and latency, accessibility and openness, and multimodality and use-case alignment.

- Performance and capabilities: GPT-4 demonstrates superior generative reasoning and long-context handling [1], [4], while BERT is more effective in classification and semantic similarity tasks [8], [19], [20]. Gemini excels in multimodal inference [2], [21], and DeepSeek performs strongly in domain-specific benchmarks such as financial and code generation datasets [3], [22].
- Cost and latency: cost and inference speed play a critical role in deployment. GPT-4 variants (e.g., GPT-4.1) offer high performance but come with higher per-token application programming interface (API) costs (~$30-60 per million tokens) and moderate latency (~0.9s initial response time) [1], [4]. Gemini's top-tier reasoning and long-context capabilities incurs higher compute cost and struggles with low-latency edge deployment; its heavy multimodal architecture makes it resource-intensive [2], [21]. DeepSeek, being optimized for domain-specific reasoning, offers a cost-effective alternative [3], [22].
- Accessibility and openness: while GPT-4 and Gemini are currently closed-source with API-only access [1], [2], [21], BERT and DeepSeek provide open-source variants that encourage research and

customization [8], [3], [22]. DeepSeek's API access is notably cheaper and faster for inference in specific domains like finance or software engineering.

– Multimodality and use-case alignment: Gemini's ability to process and reason over text, images, and audio provides significant leverage for educational, assistive, and creative tools [2], [18], [21]. GPT-4, through plugins and image input (in GPT-4o), expands the text-centric paradigm into limited multimodal territory [1], [5]. DeepSeek is tuned primarily for technical domains like mathematics and programming, but lacks broader multimodal input support [3], [22].

2.4. GPT-4

GPT-4 represents the large-scale autoregressive generative paradigm in modern LLM research. Built on a decoder-only transformer architecture, GPT-4 demonstrates strong reasoning, long-context coherence, and conversational intelligence across diverse tasks [1]. The model's design focuses on optimizing inference speed and reducing issues such as hallucinations, a phenomenon where the model generates text that appears factual but is actually incorrect or fabricated, while preserving its creative language generation capabilities [7], [23], [24]. Its multimodal extension GPT-4 integrates visual and audio inputs while maintaining generative performance. Although newer models have emerged, GPT-4 remains a canonical baseline for large-scale generative LLM evaluation and deployment studies. GPT-4 uses dynamic, context-aware embeddings to capture the nuances of language. Its subword tokenization method allows the model to handle rare or compound words efficiently, ensuring that even infrequent terms are represented adequately in the generated text [9].

Pre-training for GPT-4 involves an extensive, heterogeneous dataset that includes web pages, literature, scientific texts, and other publicly available corpora. The autoregressive objective predicting the next token based on previous context forms the backbone of its training, enabling the model to excel in text generation tasks. Over successive versions, improvements have been made in scaling the model size and refining training techniques to enhance both reasoning and factual accuracy [1], [25].

From GPT-1 to GPT-4, each iteration has introduced refinements in model architecture, parameter scaling, and training protocols. These improvements have resulted in enhanced performance on complex reasoning tasks, longer context retention, and a reduction in common pitfalls such as factual hallucination. Recent studies have also highlighted the importance of fine-tuning on domain-specific data to mitigate biases and improve reliability in specialized applications [7], [26].

2.5. BERT

BERT's architecture is defined by its bidirectional encoder since it processes the entire sequence at once instead of sequentially. This makes it possible for BERT to understand the contextual nuances of the relationship between preceding and subsequent tokens, which makes BERT an effective tool in understanding language. BERT has proven its usefulness in NLP in tasks like sentiment analysis and question answering [8], [27], [28].

BERT employs WordPiece tokenization to break text into subword units. This approach ensures that even rare words or out-of-vocabulary terms are represented effectively through a combination of subwords. Contextual embeddings are generated dynamically, allowing BERT to capture the semantic nuances of words in varied contexts [29].

The training tasks for BERT include MLM and next sentence prediction (NSP). MLM involves masking a certain number of tokens in a sentence and then predicting them from the context surrounding the sentence. The training corpus includes diverse sources such as books, Wikipedia, and web texts, ensuring that BERT is well-grounded in general language use [8], [30]. Since its release, BERT has seen numerous adaptations and derivatives. Variants such as RoBERTa, DistilBERT, and domain-specific models (e.g., BioBERT) have emerged to optimize performance for different tasks, ranging from computational efficiency to specialized applications in biomedical text analysis [27].

2.6. Gemini

Gemini represents an innovative leap by integrating multimodal capabilities into the transformer framework. Unlike traditional LLMs that operate solely on textual data, Gemini incorporates specialized modules for processing visual and auditory inputs alongside text. This cross-modal approach leverages both convolutional techniques (as seen in vision transformers) and spectrogram-based methods for audio, allowing Gemini to handle a diverse range of data types concurrently [2], [18], [31]. Gemini 1.5 further extends context length to million-token scales, enabling long-document multimodal reasoning and large-context processing.

For text, Gemini employs similar contextual embedding techniques as found in BERT and GPT-4. For images and audio, modality-specific embeddings are generated using networks that are optimized for

each data type. These embeddings are then aligned through cross-attention mechanisms, enabling integrated reasoning over heterogeneous inputs [32].

The training regime for Gemini involves curated datasets that span multiple modalities. Annotated image collections and audio files complement textual data, ensuring that the model learns robust cross-modal representations. Pre-training objectives are extended to include tasks that require the model to correlate visual or auditory cues with text, such as generating descriptive captions for images or providing sentiment analysis from audiovisual inputs [9]. Gemini is still evolving, with iterative enhancements aimed at reducing modality-specific biases and improving cross-modal coherence. Future iterations are expected to further optimize the fusion of different data types, potentially integrating additional modalities (e.g., sensor data) to widen its applicability [33].

2.7. DeepSeek

DeepSeek exemplifies the emerging domain-efficient LLM paradigm, emphasizing specialized reasoning and reduced computational cost for technical domains such as mathematics and programming [22]. DeepSeek-R1 introduces reinforcement-learning-based reasoning optimization, achieving strong performance in code and mathematical benchmarks while maintaining efficiency advantages. While it is built on a transformer-based architecture, its design includes modifications tailored to specialized tasks such as code analysis and financial modeling. These adaptations focus on reducing computational overhead and enhancing performance in narrowly defined application areas [23]–[25].

DeepSeek employs a hybrid embedding strategy that combines general-purpose contextual embeddings with domain-specific tokenization schemes. For example, in code analysis, its tokenization may incorporate programming language syntax to preserve structural information, while in financial applications, specialized vocabulary is handled with customized subword units [34]. The model is pre-trained on carefully curated datasets that are reflective of its target domains. For code-related tasks, repositories of open-source code and technical documentation are used. In financial modeling, structured financial reports and market data form the core training corpus. This focused approach allows DeepSeek to excel in specific areas while maintaining a general level of language understanding [35], [36]. Although less publicized, DeepSeek has undergone iterative refinements that prioritize efficiency and domain accuracy. Later versions integrate more refined tokenization strategies and leverage continuous feedback from real-world applications to improve task-specific performance [37].

3. RESULTS AND DISCUSSION

3.1. Comparative discussion of LLMs models

The architecture of LLMs is the key to their efficiency and utility in various areas. This section is a comprehensive analysis of the architectures of GPT-4, BERT, Gemini, and DeepSeek, employing a systematic approach to review relevant research, categorize findings, and highlight key strengths and limitations. Table 2 presents a detailed comparative analysis of model frameworks, including architecture type, training data composition, cost/latency characteristics, use-case fit, multimodal support, and API/access models. This table facilitates systematic comparison of architectural and deployment features across the four models.

Table 2. Comparative analysis of LLM models frameworks

Feature	GPT-4	BERT	Gemini	DeepSeek
Architecture	Decoder-only	Encoder-only	Multimodal transformer	Hybrid transformer
Training data	Web, code, books	Wikipedia+books	Multimodal web-scale	Code and financial data
Cost/latency	High	Low	Moderate	Low
Use-case fit	General-purpose	Classification	Multimodal AI	Code, finance
Multimodal support	Limited (GPT-4o)	No	Full	No
API/access	Closed	Open	Closed	Open-source

3.2. Comparative analysis of LLM models

Table 3 provides a comprehensive structured comparison across 17 key parameters, including developer, model type, context length, attention mechanism, tokenization, pre-training objectives, training data, fine-tuning approaches, scalability, performance characteristics, security considerations, flexibility, code generation capability, inference speed, accessibility, strengths, weaknesses, and best use cases. This detailed comparison serves as a reference guide for model selection and deployment decisions.

Figure 1 presents a radar chart visualization comparing the capabilities of GPT-4, BERT, Gemini, and DeepSeek across multiple performance dimensions, including reasoning capability, contextual understanding, multimodal integration, domain specialization, and deployment flexibility. The radar chart

provides a holistic visual representation of the relative strengths and weaknesses of each model, enabling rapid comparison of their capability profiles.

Table 3. Structured detailed comparison between LLM models across key parameters

Feature	GPT-4	BERT	Gemini	DeepSeek
Developer	OpenAI	Google	Google DeepMind	DeepSeek AI
Model type	Decoder-only transformer	Encoder-only transformer	Multimodal transformer	Domain-specific transformer
Context length	128K tokens	512 tokens	1M tokens	200K+ tokens
Multimodal	Yes (text, vision)	No	Yes (text, vision, audio, and code)	No
Attention mechanism	Autoregressive self-attention	Bidirectional self-attention	Cross-modality attention	Optimized self-attention
Tokenization	BPE	WordPiece	Modality-Specific	Hybrid (contextual and domain-specific)
Pre-training objective	Next token prediction	MLM, NSP	Cross-modality learning	Task-specific learning
Training data	Internet-scale dataset, diverse	Wikipedia, BooksCorpus	Multimodal, web, and code	Large-scale web corpus
Fine-tuning	Supported via OpenAI APIs	Pretrained, fine-tuned models	Supported via Google AI	Fine-tuned for reasoning
Scalability	High, but resource-intensive	Moderate, optimized for NLP tasks	High, but expensive training costs	High for domain-specific applications
Performance	Excellent for generative AI	Best for semantic understanding	Strong multimodal capabilities	Best in specialized domains
Security and maintainability	Moderate, vulnerable to hallucinations	High, robust bidirectional training	Moderate, increased complexity	High, task-optimized and efficient
Flexibility	High, diverse text-based applications	Moderate, limited to NLP	Very high, text, image, and audio support	Low, specialized tasks
Code generation	Strong (via GPT-4 Turbo)	Not designed for code generation	Strong (supports multiple languages)	Good, optimized for coding
Inference speed	Fast (with optimizations)	Faster (lighter model)	Optimized for speed and scale	Fast, optimized for large contexts
Accessibility	API via OpenAI (paid)	Open-source (BERT-base, large)	Google Cloud AI (paid)	Limited public access
Strengths	Strong general AI, deep reasoning, coding expertise	Contextual embeddings, strong NLP applications	High multimodal capabilities, long context understanding	Optimized for long context and deep reasoning
Weaknesses	Expensive, potential hallucinations	Not generative, limited context window	Requires high resources, limited public access	Still emerging, less known
Best use cases	Chatbots, coding, content generation, reasoning	NLP tasks (sentiment analysis, Q&A, classification)	Multimodal tasks, AI research, general knowledge	AI reasoning, coding assistance

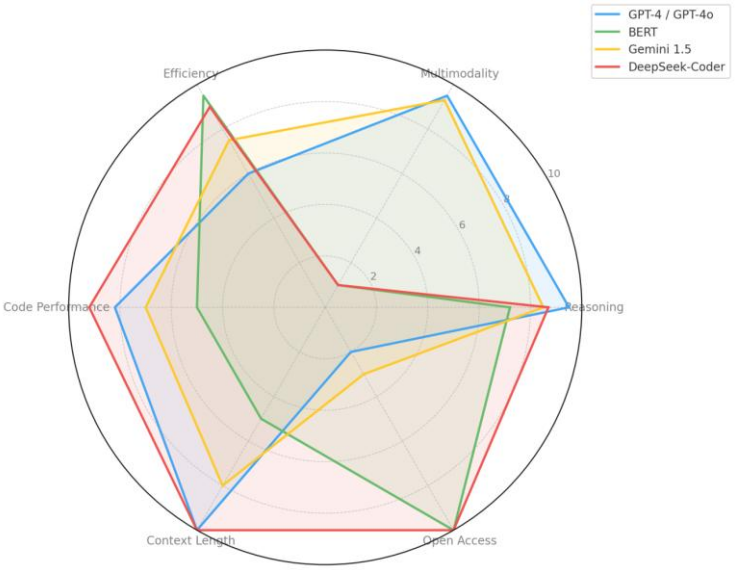

Figure 1. LLM models capability comparative radar chart

3.3. Benchmarks and reference leaderboards

Table 4 summarizes benchmark performance data for GPT-4/BERT/Gemini/DeepSeek across major evaluation frameworks including, HELM Benchmark, GLUE Leaderboard, SQuAD, Hugging Face Open LLM Leaderboard, and internal evaluations. The table includes specific benchmark scores, ranking positions, and appropriate citations to enable verification of reported performance metrics. Figure 2 illustrates the comparative benchmark performance of GPT-4, BERT, Gemini, DeepSeek, LLaMA 3, and Mixtral/Mistral across three major evaluation frameworks: HELM Benchmark, Hugging Face Open LLM Leaderboard, and Arena-Hard Leaderboard. The bar chart visualization enables direct comparison of model performance across standardized evaluation metrics, highlighting relative strengths and identifying performance gaps.

Table 4. Benchmarks and reference leaderboard summary

Model	Benchmark performance	Sources
GPT-4/GPT-4o	Top-ranked in HELM Benchmark (holistic evaluation), excels in reasoning, summarization, MMLU, and BBH.	Stanford HELM [4], OpenAI Technical Report [1]
BERT (base/large)	Competitive on GLUE Benchmark and SQuAD benchmarks for classification, QA, and NER.	GLUE Leaderboard [19], SQuAD [20]
Gemini 1.5	Not yet public in HELM Benchmark or Hugging Face, but shows strong performance on multimodal tasks in internal DeepMind evaluations.	Google DeepMind [21]
DeepSeek-Coder	Top 3 in Hugging Face Open LLM Leaderboard for code and reasoning; excels in long-context tasks (200K tokens).	HF Open LLM Leaderboard [5], DeepSeek AI GitHub [22]
LLaMA 3 (8B/70B)	Strong across open leaderboards, especially in multilingual and reasoning evaluations.	Hugging Face Arena [5], Meta AI [38]
Mixtral/Mistral	Mixtral-MoE achieves top results in Open LLM Arena and Arena-Hard; Mistral 7B performs well on cost-efficient setups.	Hugging Face Arena [5], Mistral AI [39]

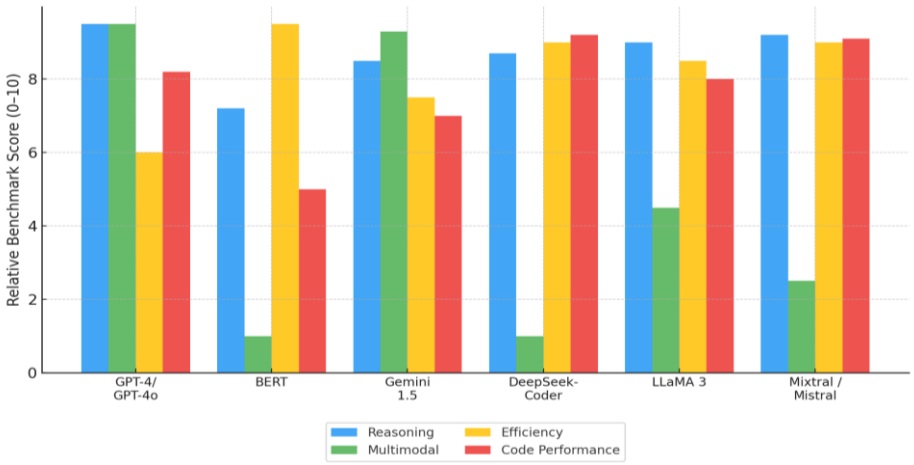

Figure 2. LLM models comparative benchmark ratings (HELM, HF, and Arena)

3.4. Applications across domains

The transformative potential of LLMs such as GPT-4 [7], BERT [8], Gemini [2], and DeepSeek [3] is evidenced by their rapidly expanding role across diverse domains. In contemporary research and practice, these models are revolutionizing industries by automating complex tasks, enhancing decision-making processes, and enabling innovative approaches to data analysis. Their integration into domains such as software engineering [40], finance [41], healthcare [42], and education [43] not only underscores their versatility but also highlights emerging patterns in model deployment, domain-specific tuning, and the convergence of multimodal data processing [44]. This section provides an in-depth analysis of key applications, strengths, limitations, and future directions, drawing on the latest literature and empirical findings, revealing several key themes: the drive toward automation [43], the need for explainability and trust [45], and the importance of domain-specific fine-tuning [37]. Below, we review applications in four critical sectors.

3.5. Software engineering

3.5.1. Use cases and integration

In the realm of software engineering, LLMs have begun to revolutionize traditional workflows. One primary use is code generation and debugging. GPT-4, for instance, is increasingly deployed to generate

boilerplate code, offer syntax suggestions, and even assist in real-time debugging. This automation not only speeds up the development cycle but also provides immediate feedback that can help developers identify and correct errors early on [6], [45], [46].

Another promising application is in documentation and code analysis. Models like BERT [8] and DeepSeek leverage their bidirectional context understanding to parse complex codebases, generate meaningful documentation, and pinpoint potential vulnerabilities. This capability is especially valuable in maintaining consistency and ensuring that critical details are not overlooked during rapid development cycles [6], [47]. Moreover, the integration of LLMs with development environments further exemplifies their utility. Tools such as GitHub Copilot [35], powered by GPT-4, provide contextual coding assistance directly within integrated development environments (IDEs) [35]. Gemini's multimodal abilities also extend to generating UI mockups and visually annotated code reviews, thereby enriching the development process with both textual and visual feedback [44].

3.5.2. Strengths and limitations

The integration of LLMs in software engineering offers several advantages. Automated code generation and debugging enhance productivity, reducing the cognitive load on developers and allowing them to focus on higher-level design and problem-solving [40]. Rapid prototyping facilitated by these models accelerates the agile development process, while automated documentation promotes uniform coding practices across teams.

However, challenges remain. A notable limitation is the propensity for hallucinations and errors that LLMs can generate code that, while syntactically correct, may contain logical or security flaws [46]. The risk of introducing vulnerabilities through automated suggestions necessitates robust validation and human oversight [6]. Additionally, dependency on external LLM services raises concerns about latency, cost, and data privacy, which must be mitigated by improved integration protocols and local fine-tuning strategies [44].

3.5.3. Future enhancements

Looking ahead, several avenues can further enhance the application of LLMs in software engineering:

- Domain-specific fine-tuning: training on curated code repositories and integrating secure coding practices could significantly enhance accuracy [40].
- Hybrid approaches: combining LLM outputs with traditional static analysis tools and formal verification methods promises to improve reliability [46], [47].
- Adaptive learning systems: incorporating continuous feedback loops from developers could help iteratively refine the models and reduce error rates [6].

3.6. Finance

3.6.1. Use cases and integration

In finance, LLMs are increasingly leveraged to automate and enhance the analytical processes critical to decision-making. One of the most compelling applications is automated report generation. For example, GPT-4 is capable of synthesizing vast amounts of financial data into coherent, comprehensive reports, facilitating rapid decision-making in fast-paced markets [41].

Additionally, risk assessment and sentiment analysis are significantly improved through models like BERT. Its contextual sensitivity allows for the nuanced analysis of market sentiment derived from news articles, social media, and financial reports. This capability aids in identifying emerging trends and potential risks, making it an invaluable tool for traders and risk managers [41].

LLMs also play a key role in algorithmic trading. By analyzing historical financial data and identifying patterns and anomalies, these models support the creation of predictive trading algorithms [48]. Gemini's multimodal data processing further enhances these analyses by incorporating visual information from charts and graphs alongside textual data, leading to more comprehensive market insights [44]. Furthermore, specialized financial applications such as portfolio management and risk analysis are being explored with DeepSeek, which can be fine-tuned to address the unique challenges of financial data analysis [41].

3.6.2. Strengths and limitations

The principal strengths of LLM applications in finance include the speed and efficiency with which large datasets can be processed, enabling real-time decision-making. The incorporation of unstructured data provides deeper analytical insights, and the automation of report generation and sentiment analysis significantly reduces operational costs [41]. Nevertheless, there are significant challenges:

- Data sensitivity: financial decisions based on LLM outputs require exceptionally high data quality, as noisy or biased inputs may lead to erroneous assessments [49].
- Lack of explainability: the "black box" nature of many LLMs complicates regulatory compliance and raises concerns among stakeholders regarding the transparency of decision-making processes [50].
- Regulatory constraints: given the strict regulatory environment of finance, any error or misinterpretation can have costly implications [51].

3.6.3. Future enhancements

To overcome these challenges, the following enhancements are recommended:

- Hybrid financial models: integrating LLMs with traditional quantitative models and econometric techniques could produce more robust predictions [52].
- Explainability frameworks: developing methods to elucidate LLM decision-making processes will be crucial in building trust with regulators and stakeholders [53].
- Real-time data integration: the incorporation of live data feeds can further boost the predictive accuracy and responsiveness of financial models [54].

3.7. Healthcare

3.7.1. Use cases and integration

In healthcare, the deployment of LLMs such as GPT-4, BERT, Gemini, and DeepSeek is transforming clinical and administrative workflows. One of the most impactful applications is clinical decision support. GPT-4 can assist clinicians by summarizing patient histories, suggesting preliminary diagnoses, and recommending treatment plans, thereby streamlining the decision-making process and potentially improving patient outcomes [55]. Another key application is the automated generation of medical reports. These models can produce discharge summaries, progress notes, and other clinical documentation, reducing the administrative burden on healthcare providers and allowing them to focus more on patient care [56].

LLMs are also pivotal in the realm of personalized medicine. By analyzing individual patient data, these models can offer tailored treatment recommendations and risk assessments. Gemini's ability to integrate multimodal data, such as radiological images with textual patient records, enhances diagnostic accuracy, while DeepSeek's capacity for parsing complex medical literature supports drug interaction analysis and evidence-based decision-making [57].

3.7.2. Strengths and limitations

The primary advantage of LLMs in healthcare is the dramatic improvement in efficiency, automating routine documentation and data analysis tasks, which frees up valuable clinician time [58], [59]. Enhanced patient communication through virtual assistants and improved resource optimization further underscores the potential of these models [60]. However, the healthcare domain presents distinct challenges:

- Risk of misinformation: inaccurate or hallucinated outputs can have serious implications for patient safety [61].
- Data privacy and security: the handling of sensitive patient data necessitates rigorous compliance with privacy regulations and robust cybersecurity measures [62].
- Ethical and legal concerns: issues of liability, consent, and potential algorithmic bias require careful management to ensure that AI-driven decisions do not adversely impact patient care [63].

3.7.3. Future enhancements

Future improvements in healthcare applications of LLMs could focus on:

- Multimodal data fusion: integrating diverse data sources, such as, genetic, laboratory, and imaging can lead to more holistic and accurate patient assessments [64].
- Regulatory-compliant models: developing healthcare-specific LLMs that strictly adhere to regulatory standards will be essential for safe and effective deployment [65].
- Interpretability and transparency: enhancing the explainability of model outputs will foster greater trust among clinicians and improve collaborative decision-making [66].
- Continuous clinical validation: ongoing real-world testing and iterative updates will be critical to ensure that LLM recommendations remain accurate and relevant [67].

3.8. Education

3.8.1. Use cases and integration

LLMs are also poised to revolutionize the education sector by enabling personalized learning and automating administrative tasks. GPT-4, for example, is used to create personalized learning content and provide real-time tutoring, adapting instructional materials to individual student needs [68]. This

individualized approach is complemented by the use of automated grading systems powered by models like BERT, which assess student submissions and provide detailed, objective feedback [69].

In addition to tutoring and grading, LLMs support content generation by creating educational materials, quizzes, and interactive exercises. Gemini's multimodal capabilities allow for the integration of textual, visual, and auditory content, enhancing engagement and catering to diverse learning styles [44]. DeepSeek is being explored to develop adaptive learning platforms that adjust instructional strategies based on real-time performance data, offering a more dynamic and responsive educational experience [70].

3.8.2. Strengths and limitations

The educational applications of LLMs bring numerous benefits. Enhanced student engagement through personalized and interactive content has the potential to improve retention and learning outcomes [71]. The scalability of automated systems also promises to broaden access to high-quality educational resources, while streamlined grading and content generation reduce the administrative burden on educators [72]. Yet, challenges persist:

- Bias and inaccuracies: training data limitations may result in biased or inaccurate content, which can adversely affect learning outcomes [73].
- Overreliance on automation: excessive dependence on AI-generated feedback may reduce opportunities for critical thinking and independent problem solving [74].
- Ensuring pedagogical soundness: while LLMs can generate diverse educational materials, ensuring that such content meets rigorous pedagogical standards remains a key concern [75].

3.8.3. Future enhancements

Future developments in educational applications should consider:

- Interactive, multimodal learning environments: combining LLM capabilities with augmented reality and interactive platforms could create immersive, engaging learning experiences [76].
- Ethical and bias mitigation strategies: ongoing refinement of training datasets and robust bias mitigation techniques are essential to ensure fairness and accuracy [77].
- Collaborative teacher-AI tools: empowering educators with customizable AI tools can foster a more effective blend of human insight and machine efficiency [78].
- Longitudinal feedback mechanisms: systems that track student progress over time and dynamically adapt teaching strategies will further personalize education and support continuous improvement [79].

3.9. Choosing the right model

Table 5 provides actionable recommendations for model selection based on specific use case requirements. The table categorizes each model's optimal application domains, enabling practitioners to make informed decisions when selecting LLMs for particular tasks, including general-purpose AI agents, enterprise NLP, multimodal applications, and domain-specific deployments. Executive summary: i) GPT-4 excels in reasoning and generation, best for general-purpose AI agents; ii) BERT offers fast, accurate NLP for classification tasks; iii) Gemini is the most powerful for multimodal tasks (text, image, and audio); and iv) DeepSeek is domain-specific and cost-efficient, ideal for finance/code.

Table 5. Recommended use cases for LLM models

Model	Recommended for
GPT-4	General-purpose AI agents, customer support bots, legal reasoning, creative writing.
BERT	Enterprises focused on classification, NER, semantic search, and real-time QA systems. Efficient for on-device inference and edge deployment.
Gemini	Education, accessibility, multimodal assistants, and perception-rich tasks (vision/audio/code).
DeepSeek	Industry-grade applications in finance, software engineering, law, or where cost-efficiency and long-context reasoning are essential.

4. CONCLUSION

This review analyzed GPT-4, BERT, Gemini, and DeepSeek, highlighting their distinct architectural paradigms and trade-offs across generative reasoning, contextual understanding, multimodal integration, and domain-specific efficiency. While these models have demonstrated significant impact in software engineering, healthcare, finance, and education, they continue to face common challenges, including hallucinations, high computational costs, limited interpretability, and persistent biases. Although LLM capabilities are advancing rapidly through larger context windows, multimodal enhancements, and improved reasoning mechanisms, this study provides a foundational reference framework for understanding core

architectural characteristics and deployment considerations, while inviting future empirical extensions as newer model generations emerge. Beyond technical progress, addressing AI safety, regulatory governance, privacy, transparency, and widespread AI literacy gaps will be essential to ensuring responsible adoption; consequently, the future of LLMs will depend not only on advances in capability and efficiency through approaches such as hybrid architectures, LoRA/QLoRA-based fine-tuning, and open-source innovation, but also on the development of trustworthy, human-centered, and policy-aware AI systems.

FUNDING INFORMATION

The authors state no funding is involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Kavish Sanghvi	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Aparna S. Sharma		✓		✓	✓	✓				✓	✓			
Surbhi Hooda				✓	✓	✓				✓	✓			

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors state no conflict of interest.

DATA AVAILABILITY

No new data were generated or analyzed in this study. The findings presented in this review are derived from previously published research articles, benchmark reports, technical documentation, and publicly available resources cited throughout the manuscript. All supporting materials are available through the referenced sources listed in the References section. Therefore, data availability is not applicable to this review article.

REFERENCES

[1] OpenAI, "GPT-4 technical report," *arXiv*:2303.08774, 2023.
[2] Gemini Team, "Gemini: a family of highly capable multimodal models," *arXiv*:2312.11805, 2023.
[3] DeepSeek-AI, "DeepSeek-R1: incentivizing reasoning capability in LLMs via reinforcement learning," *arXiv*:2501.12948, 2025.
[4] Stanford Center for Research on Foundation Models (CRFM), "HELM benchmark: holistic evaluation of language models," *Stanford CRFM*, 2022.
[5] Hugging Face, "Open LLM leaderboard and arena-hard," *Hugging Face*. Accessed: Mar. 12, 2025. [Online]. Available: https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard
[6] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS 2017)*, 2017, pp. 5998–6008.
[7] T. B. Brown *et al.*, "Language models are few-shot learners," *arXiv*:2005.14165, 2020.
[8] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, pp. 4171–4186, 2019, doi: 10.18653/v1/N19-1423.
[9] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," in *Proceedings of the 38th International Conference on Machine Learning*, 2021, pp. 8748–8763.
[10] S. Ruder, "Neural transfer learning for natural language processing," Ph.D. dissertation, National University of Ireland, Galway, Ireland, 2019.
[11] D. Patterson, J. Gonzalez, and Q. Le, "Carbon emissions and large neural network training," *arXiv*:2104.10350, 2021.
[12] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2013, pp. 3111–3119.
[13] J. Pennington, R. Socher, and C. D. Manning, "GloVe: global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543, doi: 10.3115/v1/d14-1162.
[14] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv*:1607.06450, 2016.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
[16] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016), Volume 1: Long Papers*, 2016, vol. 3, pp. 1715–1725, doi: 10.18653/v1/p16-1162.
[17] T. Kudo and J. Richardson, "SentencePiece: a simple and language-independent subword tokenizer," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018, pp. 66–71.
[18] M. A. Manzoor, S. Albarri, Z. Xian, Z. Meng, P. Nakov, and S. Liang, "Multimodality representation learning: a survey on evolution, pretraining and its applications," *arXiv*:2302.00389, 2023.
[19] GLUE Benchmark, "GLUE leaderboard," *GLUE*. Accessed: Mar. 12, 2025. [Online]. Available: https://gluebenchmark.com/leaderboard
[20] SQuAD, "Stanford question answering dataset," *SQuAD*. Accessed: Mar. 12, 2025. [Online]. Available: https://rajpurkar.github.io/SQuAD-explorer/
[21] Google DeepMind, "Gemini 1.5 technical overview," *Google DeepMind*. Accessed: Mar. 12, 2025. 2024. [Online]. Available: https://deepmind.google/technologies/gemini/
[22] DeepSeek AI, "DeepSeek-Coder: a comprehensive code LLM," *GitHub*. Accessed: Mar. 12, 2025. 2024. [Online]. Available: https://github.com/deepseek-ai/DeepSeek-Coder
[23] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks*, vol. 61, pp. 85–117, Jan. 2015, doi: 10.1016/j.neunet.2014.09.003.
[24] M. Shoeybi, M. Patwary, R. Puri, P. LeGresley, J. Casper, and B. Catanzaro, "Megatron-LM: training multi-billion parameter language models using model parallelism," *arXiv*:1909.08053, 2019.
[25] H. Touvron, "LLaMA: open and efficient foundation language models," *arXiv*:2302.13971, 2023.
[26] J. W. Rae *et al.*, "Scaling language models: methods, analysis & insights from training gopher," *arXiv*:2112.11446, 2021.
[27] Y. Liu *et al.*, "RoBERTa: a robustly optimized BERT pretraining approach," *arXiv*:1907.11692, 2019.
[28] Y. Goldberg, "Neural network methods for natural language processing," *Computational Linguistics*, vol. 44, no. 1, pp. 194–195, 2018, doi: 10.1162/COLI_r_00312.
[29] Y. Wu *et al.*, "Google's neural machine translation system: bridging the gap between human and machine translation," *arXiv*:1609.08144, 2016.
[30] A. Rogers, O. Kovaleva, and A. Rumshisky, "A primer in BERTology: what we know about how BERT works," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 842–866, Dec. 2020, doi: 10.1162/tacl_a_00349.
[31] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: transformers for image recognition at scale," *arXiv*:2010.11929, 2020.
[32] A. J. Jaegle *et al.*, "Perceiver IO: A general architecture for structured inputs & outputs," *arXiv*:2107.14795, 2021.
[33] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, G. Gu, and M. Auli, "Data2Vec: a general framework for self-supervised learning in speech, vision, and NLP," *arXiv*:2202.03555, 2022.
[34] D. G. Liang *et al.*, "DeepSeek-Coder: when the large language model meets programming -- the rise of code intelligence," *arXiv*:2401.14196, 2024.
[35] M. Chen *et al.*, "Evaluating large language models trained on code," *arXiv*:2107.03374, 2021.
[36] J. Kaplan *et al.*, "Scaling laws for neural language models," *arXiv*:2001.08361, 2020.
[37] Y. Sun *et al.*, "ERNIE 2.0: a continual pre-training framework for language understanding," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2020)*, 2020, pp. 8968–8975, doi: 10.1609/aaai.v34i05.6428.
[38] Meta AI, "LLaMA 3 models," *Developer.meta*. Accessed: Mar. 12, 2025. 2024. [Online]. Available: https://ai.meta.com/llama/
[39] Mistral AI, "Mixtral and mistral models," *Mistral.ai*. Accessed: Mar. 12, 2025. 2023. [Online]. Available: https://mistral.ai/news/mixtral-of-experts/
[40] M. Allamanis, E. T. Barr, P. Devanbu, and C. Sutton, "A survey of machine learning for big code and naturalness," *ACM Computing Surveys (CSUR)*, vol. 51, no. 4, pp. 1–37, 2019, doi: 10.1145/3212695.
[41] Y. Yang, M. C. S. Uy, and A. Huang, "FinBERT: A pretrained language model for financial communications," *arXiv*:2006.08097, 2020.
[42] J. Lee *et al.*, "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, Feb. 2020, doi: 10.1093/bioinformatics/btz682.
[43] L. Chen, P. Chen, and Z. Lin, "Artificial intelligence in education: a review," *IEEE Access*, vol. 8, pp. 75264–75278, 2020, doi: 10.1109/ACCESS.2020.2988510.
[44] T. Baltrusaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: a survey and taxonomy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 423–443, Feb. 2019, doi: 10.1109/TPAMI.2018.2798607.
[45] J. Austin *et al.*, "Program synthesis with large language models," *arXiv*:2108.07732, 2021.
[46] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," *arXiv*:1904.09751, 2019.
[47] U. Alon, M. Zilberstein, O. Levy, and E. Yahav, "code2vec: learning distributed representations of code," in *Proceedings of the ACM on Programming Languages*, Jan. 2019, pp. 1–29, doi: 10.1145/3290353.
[48] H. Zhou *et al.*, "Informer: beyond efficient transformer for long sequence time-series forecasting," in *35th AAAI Conference on Artificial Intelligence (AAAI 2021)*, 2021, pp. 11106–11115, doi: 10.1609/aaai.v35i12.17325.
[49] T. Loughran and B. McDonald, "When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks," *The Journal of Finance*, vol. 66, no. 1, pp. 35–65, Feb. 2011, doi: 10.1111/j.1540-6261.2010.01625.x.
[50] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv*:1702.08608, 2017.
[51] I. Anagnostopoulos, "Fintech and regtech: Impact on regulators and banks," *Journal of Economics and Business*, vol. 100, pp. 7–25, Nov. 2018, doi: 10.1016/j.jeconbus.2018.07.003.
[52] S. Gu, B. Kelly, and D. Xiu, "Empirical asset pricing via machine learning," *The Review of Financial Studies*, vol. 33, no. 5, pp. 2223–2273, May 2020, doi: 10.1093/rfs/hhaa009.
[53] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should i trust you?' Explaining the predictions of any classifier," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1135–1144, doi: 10.1145/2939672.2939778.
[54] M. Xiao *et al.*, "Enhancing financial time-series forecasting with retrieval-augmented large language models," *arXiv*:2502.05878, 2025.
[55] A. Rajkomar, J. Dean, and I. Kohane, "Machine learning in medicine," *New England Journal of Medicine*, vol. 380, no. 14, pp. 1347–1358, Apr. 2019, doi: 10.1056/NEJMra1814259.

[56] B. Jing, P. Xie, and E. P. Xing, "On the automatic generation of medical imaging reports," in *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 2018, pp. 2577–2586, doi: 10.18653/v1/p18-1240.
[57] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, "Deep learning for healthcare: review, opportunities and challenges," *Briefings in Bioinformatics*, vol. 19, no. 6, pp. 1236–1246, Nov. 2018, doi: 10.1093/bib/bbx044.
[58] F. Jiang *et al.*, "Artificial intelligence in healthcare: past, present and future," *Stroke and Vascular Neurology*, vol. 2, no. 4, pp. 230–243, Dec. 2017, doi: 10.1136/svn-2017-000101.
[59] B. Shickel, P. J. Tighe, A. Bihorac, and P. Rashidi, "Deep EHR: a survey of recent advances in deep learning techniques for electronic health record (EHR) analysis," *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 5, pp. 1589–1604, Sep. 2018, doi: 10.1109/JBHI.2017.2767063.
[60] L. Laranjo *et al.*, "Conversational agents in healthcare: a systematic review," *Journal of the American Medical Informatics Association*, vol. 25, no. 9, pp. 1248–1258, Sep. 2018, doi: 10.1093/jamia/ocy072.
[61] J. Amann, A. Blasimme, E. Vayena, D. Frey, and V. I. Madai, "Explainability for artificial intelligence in healthcare: a multidisciplinary perspective," *BMC Medical Informatics and Decision Making*, vol. 20, no. 1, p. 310, Dec. 2020, doi: 10.1186/s12911-020-01332-6.
[62] N. Rieke *et al.*, "The future of digital health with federated learning," *npj Digital Medicine*, vol. 3, no. 1, p. 119, Sep. 2020, doi: 10.1038/s41746-020-00323-1.
[63] J. Morley *et al.*, "The ethics of AI in health care: a mapping review," *Social Science & Medicine*, vol. 260, p. 113172, Sep. 2020, doi: 10.1016/j.socscimed.2020.113172.
[64] A. Esteva *et al.*, "A guide to deep learning in healthcare," *Nature Medicine*, vol. 25, no. 1, pp. 24–29, Jan. 2019, doi: 10.1038/s41591-018-0316-z.
[65] J. Wiens *et al.*, "Do no harm: a roadmap for responsible machine learning for health care," *Nature Medicine*, vol. 25, no. 9, pp. 1337–1340, Sep. 2019, doi: 10.1038/s41591-019-0548-6.
[66] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad, "Intelligible models for healthcare: predicting pneumonia risk and hospital 30-day readmission," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 1721–1730, doi: 10.1145/2783258.2788613.
[67] C. J. Kelly, A. Karthikesalingam, M. Suleyman, G. Corrado, and D. King, "Key challenges for delivering clinical impact with artificial intelligence," *BMC Medicine*, vol. 17, no. 1, p. 195, Dec. 2019, doi: 10.1186/s12916-019-1426-2.
[68] R. Luckin, W. Holmes, and L. B. Forcier, *An argument for AI in education*. London, U.K.: Pearson Education, 2016.
[69] K. Taghipour and H. T. Ng, "A neural approach to automated essay scoring," *in Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016, pp. 1882–1891, doi: 10.18653/v1/d16-1193.
[70] J. F. Pane, E. D. Steiner, M. D. Baird, L. S. Hamilton, and J. D. Pane, *Informing progress: insights on personalized learning implementation and effects*, Research Report RR-2042-BMGF. Santa Monica, CA: RAND Corporation, Jul. 2017, doi: 10.7249/RR2042.
[71] J. A. Kulik and J. D. Fletcher, "Effectiveness of intelligent tutoring systems," *Review of Educational Research*, vol. 86, no. 1, pp. 42–78, Mar. 2016, doi: 10.3102/0034654315581420.
[72] M. D. Shermis, J. Burstein, Y.-W. Choi, and B. Hamner, "Contrasting state-of-the-art automated scoring of essays: results from the automated student assessment prize," *Journal of Technology, Learning, and Assessment*, vol. 11, no. 3, pp. 1–18, 2012.
[73] A. Caliskan, J. J. Bryson, and A. Narayanan, "Semantics derived automatically from language corpora contain human-like biases," *Science*, vol. 356, no. 6334, pp. 183–186, Apr. 2017, doi: 10.1126/science.aal4230.
[74] N. Selwyn, *Should robots replace teachers? AI and the future of education*. Cambridge, U.K.: Polity Press, 2019.
[75] W. Holmes, M. Bialik, and C. Fadel, *Artificial intelligence in education: promises and implications for teaching and learning*. Boston, MA: Center for Curriculum Redesign, 2019.
[76] H.-K. Wu, S. W.-Y. Lee, H.-Y. Chang, and J.-C. Liang, "Current status, opportunities and challenges of augmented reality in education," *Computers & Education*, vol. 62, pp. 41–49, Mar. 2013, doi: 10.1016/j.compedu.2012.10.024.
[77] S. Slade and P. Prinsloo, "Learning analytics," *American Behavioral Scientist*, vol. 57, no. 10, pp. 1510–1529, Oct. 2013, doi: 10.1177/0002764213479366.
[78] O. Zawacki-Richter, V. I. Marín, M. Bond, and F. Gouverneur, "Systematic review of research on artificial intelligence applications in higher education – where are the educators?," *International Journal of Educational Technology in Higher Education*, vol. 16, no. 1, p. 39, Dec. 2019, doi: 10.1186/s41239-019-0171-0.
[79] Z. Papamitsiou and A. A. Economides, "Learning analytics and educational data mining in practice: a systemic literature review of empirical evidence," *Educational Technology and Society*, vol. 17, no. 4, pp. 49–64, 2014.

BIOGRAPHIES OF AUTHORS

Kavish Sanghvi holds an M.S. in Software Engineering from Stevens Institute of Technology, Hoboken, USA, and a B.Tech. in Computer Engineering from NMIMS University, Mumbai, India. He has professional experience in software engineering and technology leadership, with expertise in artificial intelligence, generative AI, cloud computing, and full-stack application development. He has professional experience leading critical projects. He has authored and co-authored several peer-reviewed journal articles and conference papers in the fields of artificial intelligence, machine learning, image classification, social media analytics, and educational technology. His research interests include artificial intelligence, large language models, natural language processing, machine learning, computer vision, and software engineering. He can be contacted at email: sanghvi_kavish@yahoo.in.

Aparna S. Sharma holds an M.S. in Engineering Management from the University of Southern California, Los Angeles, USA, and a B.Tech. (Honors) in Mechatronics with a Minor in Artificial Intelligence and Machine Learning from NMIMS University, Mumbai, India. Her research interests span human-computer interaction, ethical AI, social media, and responsible AI, with a focus on accessibility, equity, and safety of AI systems. She has authored and co-authored peer-reviewed publications in journals including Elsevier's International Journal of Information Management: Data Insights (97+ citations) and Springer's Education and Information Technologies, as well as IEEE conference proceedings. She has won an IEEE best paper award for elderly care humanoids or robots. She can be contacted at email: sharmaaparnasunil@gmail.com.

Surbhi Hooda is a Doctor of Business Administration student at University of the Potomac, Washington DC, USA, and holds a M.S. in Information Systems from University of Maryland, College Park, USA, and a B.Tech. in Computer Science and Engineering from SRM Institute of Science and Technology, Chennai, India. She has professional experience in financial analysis, business intelligence, data analytics, and technology-enabled process automation. She has co-authored scholarly publications in the areas of information systems and emerging technologies. Her research and professional interests include artificial intelligence, responsible AI, business intelligence, and data-driven decision-making. Her ongoing doctoral research examines the role of Al-enabled tools in supporting business decision-making and professional judgment. She can be contacted at email: surbhihooda15@gmail.com.