International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 07 | July – 2025 DOI: 10.55041/ISJEM04804
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

AI Story Generator Using GPT-2 and Deep Learning

B. KUMARI, V. PRASANTH KUMAR

Assistant Professor, 2 MCA Final Semester, Master of Computer Applications,

Sanketika Vidya Parishad Engineering College, Vishakhapatnam, Andhra Pradesh, India.

Abstract:

AI Story Generator is an AI-generated story maker that can help us come up with new ideas and storylines. It automatically generates a story plot based on a given input. It uses artificial intelligence algorithms to interpret the provided information and create a cohesive narrative, depending upon the category the user selects and wishes to write story on, there will be different set of categories and worlds, where one can select their desired option and
continue generating the story. We can also save the progress, so there will be no issue of progress loss. We cannot change our selected category and world, once it is selected initially, as the category and world are confined to one story project, if there are any changes in the world and category, the user need to start with a new story project. AI Story Generator is a very helpful as a tool for content writers and novelist, who face creative block where they
are not able to think of any new ideas, AI Story Generator comes to the rescue as it helps the user to develop a story on its own by taking the required keywords for the story to start with and the rest AI will take care for the generation of the story. The AI story generator is an exciting technology that uses artificial intelligence to create compelling narratives tailored to individual preferences. It leverages machine learning algorithms and natural language processing to generate stories that are not only engaging but also personalized. The development and implementation of AI story generators have the potential to revolutionize the way we think about storytelling. By creating stories that are unique to each individual, this technology can transform the way we consume and interact with media. It can be used for a variety of applications, including video games, virtual reality experiences, educational materials, and marketing campaigns

Index Terms — AI Story Generator, Deep Learning, Transformer Models, NLP, GPT-2, GPT-Neo, Story Generation, MongoDB, Flask, Text-to-Speech, Creative Writing

I. INTRODUCTION

An AI story generator is a computer-based tool that creates a story based on a prompt. The prompt can be anything from a few words to a detailed outline, but the most common use for story generators is when you want to generate a story based on a set of characters or elements. AI Story Generator in particular is one of the more sophisticated tools, as it has its own neural network that teaches itself over time how to write better stories [21]. The AI Story Generator draws upon a database of pre-written stories to help it generate the main plot, but then fills in details with freshly generated sentences based on patterns that it infers from what you want from your prompt [12]. AI offers some interesting benefits to a writer. Using AI means you can use cheap labour and have a lot of material generated for free [5]. Writing a story can be slow and you may need to generate lots of words for a novel. Using AI is a great way to make your life easier, especially if you find yourself having writer's block or need some inspiration [3]. An AI story generator is a tool that can be incredibly helpful for writers. This computer-based system is designed to create a story based on a given prompt, which can be anything from a few words to a detailed outline. One of the most common uses for these story generators is when a writer wants to generate a story based on a set of characters or elements. These tools are especially useful for writers who are struggling with writer's block or need some inspiration. One of the more sophisticated tools available is the AI Story Generator. This tool has its own neural network that teaches itself over time how to write better stories. It draws upon a database of pre-written stories to help generate the main plot, but then fills in details with freshly generated sentences based on patterns that it infers from the writer's prompt. This allows the AI Story Generator to create stories that are unique and compelling. One of the benefits of using AI story generators is that they can help writers save time and money. Writing a story can be a slow process, and writers may need to generate lots of words for a novel. Using AI is a great way to make the writing process easier and more efficient. It can also be cheaper than hiring a team of writers or editors. In addition to being cost-effective, AI story generators can also help writers who are struggling with writer's block. When writers are stuck on a particular plot point or character, they can use a story generator to generate new ideas and get their creative juices flowing. This can be especially helpful when a writer is feeling frustrated or uninspired.

1.1 Existing System

There are currently a variety of AI story generators available, each with their own unique capabilities and features. Some popular examples include. OpenAI's GPT-3: This is a large-scale language generation model that can be used to generate text for a wide range of applications, including story generation. GPT-3 can be accessed via OpenAI's API [12].AI Dungeon: This is an interactive fiction game that uses GPT-3 to generate unique stories based on user input [1]. Most of the available story generators lack one quality, i.e. saving the project as Word document and downloading it to the system.

1.1.1 Challenges

- Understanding Prompts: The AI at times fails to comprehend short or ambiguous prompts and does not always adhere to the chosen genre accurately. It relies a lot on the quality of input to come up with coherent stories, so ambiguous prompts can result in low-quality output [3].
- Story Quality: Maintaining the story engaging and coherent from start to finish proves challenging. At times, the output contains copied sentences or gets diverted, and the story ends up appearing incomplete or random [5].
- Model Limitations: Pre-trained models such as GPT-2 are not specially trained to do storytelling tasks. Therefore, they could be lacking in creativity, tone for a specific genre, or emotional depth in some stories [21].
- Browser TTS Limitation: TTS Limitations — The voice based on JavaScript relies solely on an internet connection and browser support. Its speech quality is inconsistent and may not sound too natural in comparison to sophisticated TTS systems [19].
- Saving & Handling Stories: Users may inadvertently save the same story over and over again due to missing checks. Also, as the number of stories increases, it is more difficult to maintain the database efficient and well-organized.
- Story Continuation Problems: While continuing a story, the AI can forget former characters, settings, or plotlines. This disrupts the flow and creates confusion within the story, lowering the overall quality [1].

1.2 Proposed System

In my AI Story Generator, we can save the file and download the file as a word document. So, the users can store their data and transfer them or share the file easily. Thus, it gives advantage to the user to access the file anywhere he wants. This AI Story Generator can save the progress once the whole work is done in a word document [1]. It also has a variety of genre to choose and go with; we can also limit the word count as preferred [21].However, as with any technology, the development and implementation of AI story generators also pose ethical considerations. There is a risk of misuse or unintended consequences, and it is essential to approach the creation and use of these tools with care and thoughtfulness, ensuring that they are developed with diversity, inclusivity, and transparency in mind [5], [9], [19].

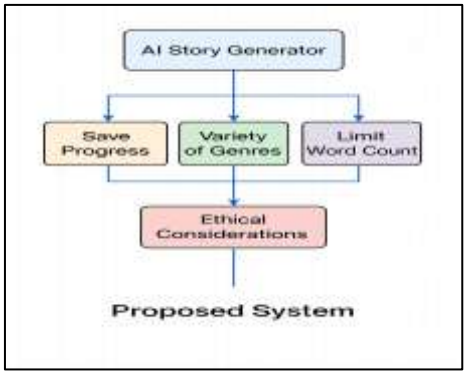

Fig: 1 Proposed Diagram

1.2.1 Advantages

- User-Friendly Interface: The system allows users to easily input story prompts and select genres, making story creation accessible even to those without technical expertise [1].
- Creativity Enhancement: The AI assists users by generating imaginative and coherent stories, helping to overcome writer's block and inspiring creativity [3].
- Customizable Output: Users can specify story length, genre, and style, enabling personalized story generation tailored to their preferences [21].
- Fast Story Generation: Powered by transformer-based language models, the system produces stories rapidly compared to manual writing [12].
- Continuous Learning: By integrating feedback and user data, the model can improve over time, delivering better and more relevant story content [5].
- Multi-Genre Support: Supports various genres such as fantasy, romance, thriller, and sci-fi, offering diverse storytelling experiences.
- Accessibility: Available via a web interface, the platform can be accessed from anywhere, encouraging widespread usage and collaboration.
- Cost-Effective: Eliminates the need for expensive software or professional writers, making creative writing more affordable and scalable.

II. LITERATURE REVIEW

2.1 Architecture

Architecture of AI Story Generator:
AI Story Generators have evolved considerably, leveraging advances in Natural Language Processing (NLP) and deep learning to produce coherent and contextually relevant stories. Modern architectures predominantly use transformer-based models, such as GPT-2, GPT-Neo, and GPT-3, which excel at generating human-like text by predicting the next word in a sequence based on large training corpora [12], [21].

Historical development in story generation can be outlined in phases:

- Early Rule-Based Systems: Initial story generation relied on handcrafted rules and templates, which limited creativity and diversity.
- Statistical Language Models: Markov models and n-gram models enabled more flexible text generation but often lacked long-term coherence.
- Neural Network Approaches: Introduction of Recurrent Neural Networks (RNNs) and LSTMs improved sequence learning but struggled with long-range dependencies[22].
- Transformer Models: The revolutionary transformer architecture (Vaswani et al., 2017) enabled effective context retention across longer texts, greatly enhancing story quality[12].
- Fine-Tuning and Transfer Learning: Using large pre-trained models fine-tuned on story-specific datasets allows for improved genre-specific narrative generation[1].
- Evaluation and Refinement: Techniques such as reinforcement learning, recursive prompting, and user feedback loops have been incorporated to refine story coherence and engagement[23].

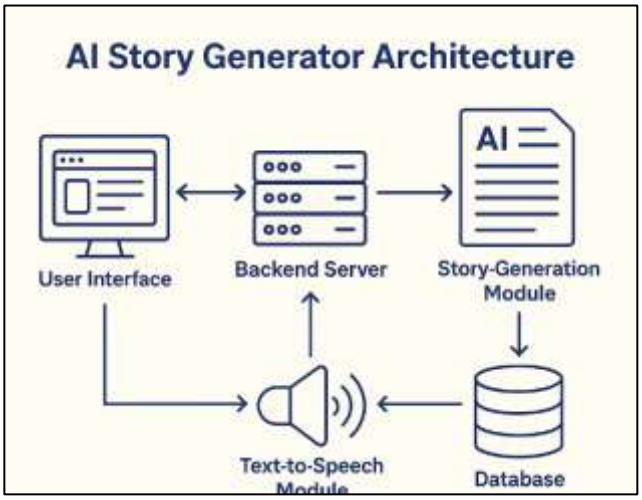

Fig:2 Architecture

2.2 Algorithm:

The core algorithm relies on generative transformer models, which are pre-trained on massive text corpora [12], [22]. The system uses the following sequence:

- Prompt Encoding: The user input (prompt) is tokenized and encoded using a tokenizer compatible with the selected model.
- Generation: A pipeline is initialized with the chosen model (e.g., GPT-2 or GPT-Neo). The generate () function is called with hyperparameters such as:
  - max_length (controls story length),
  - num_return_sequences (generates multiple outputs),

- do_sample=True, top_k, top_p, and temperature (for controlling randomness)[21].

- Decoding and Post-Processing: The generated story is decoded from tokens to natural language text. Filters may be applied to remove incomplete or repetitive outputs[21].

- Storage: The generated story, along with its prompt and genre, is saved to MongoDB using a custom save_story () function[12].

- Retrieval and Continuation: Users can view or continue past stories by fetching them from the database and sending them back into the generation pipeline[23].

2.3Techniques:

The project integrates multiple AI and web development techniques:

- Transfer Learning with Transformers: Instead of training from scratch, the system leverages pre-trained models, fine-tuned through prompt engineering to generate context-aware text [12], [22].

- Genre Conditioning: By prepending genres to prompts (e.g., "Horror: The house was…"), the model is guided to generate genre-specific narratives [1].

- Client-Side TTS: JavaScript's speech Synthesis API is used to provide offline TTS in the browser, eliminating the need for server-side TTS modules like pyttsx3 [19].

- NoSQL Data Persistence: MongoDB is used to store and retrieve story records efficiently using BSON documents.

- Web UI/UX Techniques: HTML5, CSS, and JavaScript form the UI, ensuring a responsive, mobile-friendly experience with genre-based background images and buttons for playback and continuation [5].

2.4Tools:

Several tools were selected to streamline development and maximize performance:

- Python: The primary programming language for backend and story generation logic.

- Flask: A lightweight web framework used to manage routes, server logic, and render templates.

- Transformers (Hugging Face): Provides access to models like gpt2-large and gpt-neo, essential for language generation [12], [22].

- MongoDB with PyMongo: For non-relational storage of stories and user history [1].

- HTML/CSS/JS: Power the interactive UI [5].

- Speech Synthesis API: Built-in browser API for converting story text to speech without external dependencies [19].

- Jinja2: Templating engine integrated with Flask to dynamically render HTML.

2.5Methods:

The AI Story Generator system is developed using a systematic methodology:

- User Input Collection: The user selects a genre and enters a prompt via index.html.
- Model Selection and Story Generation: The backend uses the generate_story () function from gencode.py, initializing the transformer model and generating one or more stories [12], [22].
- Result Rendering: Generated content is returned to the client via generated.html, which includes buttons for playback, continuation, and saving.
- Database Operations:
  - save_story () stores each new story in MongoDB.
  - get_all_stories () fetches them for the history page.
  - delete_story () removes specific records.
- Story Playback: Users can click "Listen" to trigger JavaScript-based TTS playback [19].
- Story Continuation: The user can append a new segment to a story by submitting its text as a new prompt [1].

III. METHODOLOGY

3.1Input:

This project aims to generate compelling and coherent short stories across multiple genres using artificial intelligence, specifically transformer-based language models such as GPT-2, GPT-2-large, and GPT-Neo-125M. These models are fine-tuned on extensive

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 07 | July – 2025 DOI: 10.55041/ISJEM04804
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

natural language datasets and are capable of producing human-like text outputs [12], [22]. The AI Story Generator takes a user-provided prompt, which includes a textual seed and an optional genre selection, to serve as the basis for narrative generation [1].
The system is implemented as a web-based application developed using the Flask framework for backend logic and HTML/CSS for frontend interaction. The user interacts with the application by providing input through a simple web form. This input is processed by a text generation engine, which loads the selected model and generates output according to specified parameters like story length and creativity [5].
The project follows a modular architecture:

- app.py handles the application logic and route control.
- gencode.py encapsulates the story generation logic, including model initialization and inference.
- db.py manages story storage and retrieval operations using MongoDB [1].
- tts.js (browser-side) handles text-to-speech playback using the Speech Synthesis API [19].
- Optional support for image generation is included via external API integration.

Figure:3 Input Interface from index.html in app.py

Figure:4 Prompt flow from form submission to gencode.py

• Figure:5 Database interaction logic in db.py

3.2 Method of Process

The generation of AI-based short stories follows a structured and sequential process that integrates user input, model inference, and interactive features to ensure a fluid storytelling experience. Below is a detailed breakdown of this process:

1. User Input Acquisition

The user accesses the platform and selects a story genre (e.g., Adventure, Sci-Fi, Horror) and provides text prompt that initiates the story. Additional options like maximum story length and number of outputs may also be set.

2. Model Loading and Prompt Construction

The system loads the appropriate pre-trained transformer model using Hugging Face's transformers pipeline. The model is selected based on a dropdown (e.g., GPT-2, GPT-2-large, or GPT-Neo). The genre and user prompt are concatenated to give contextual guidance to the model [12], [22].

3. Story Generation Logic

Using the text-generation pipeline, the model produces narrative sequences. Key parameters like max_length, temperature, and top_p are dynamically passed to control randomness and story depth. The outputs are parsed, cleaned, and formatted before display [1].

4. Rendering and Interaction

- The generated story is presented on generated.html, along with genre-specific imagery and background visuals.
- A "Listen" button enables text-to-speech using the Web Speech API built into browsers[19].
- A "Continue Story" option allows users to extend their story by using the last paragraph as a new prompt.

5. Data Storage and History Management

Every story generated is automatically saved into MongoDB, including metadata such as:

- Genre
- Original prompt
- Story text
- Generation timestamp

Users can retrieve previous stories via the History page, where they can listen, delete, or continue any saved story [5].

6. Text-to-Speech Playback

The frontend integrates browser-based text-to-speech for cross-platform functionality. This allows users to listen to stories without requiring additional software or Python libraries like pyttsx3 [19].

7. Story Continuation Process

If a user opts to extend a story, the final portion of the current story is sent back to the server as a new prompt. The same generation pipeline is used to create a continuation, which is displayed and stored like the original.

8. Optional Enhancements

- Image Generation: Genre-relevant images may be generated using APIs such as Deep AI for visual engagement.
- Story Deletion: Stories can be removed from the database by user action.

9. Feedback and Iterative Improvement

The system is continuously improved based on user interaction, performance testing, and story quality assessment. Suggestions and feedback from users are used to fine-tune parameters and add new features like dialogue-based generation or automatic plot summarization [23].

3.3 Output:

The output of the AI Story Generator is a complete short story tailored to the user's prompt and genre preferences. Each generated story reflects the narrative coherence and linguistic fluency of transformer-based language models [12], [22]. Outputs may include:

- One or more text stories based on the specified prompt
- Optional illustrations (if image generation is enabled)
- A confidence rating (planned future feature) to estimate narrative quality [23].

The final story is presented through a responsive, user-friendly interface, with additional features including:

- Auditory playback using browser TTS [19].
- Story continuation
- Database logging for historical access [1]
- Visual genre representation for immersion

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 07 | July – 2025 DOI: 10.55041/ISJEM04804
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Users can save, listen to, delete, or revisit their stories at any time, enhancing both creative exploration and usability. This makes the system ideal for casual readers, aspiring writers, and AI enthusiasts looking to collaborate with artificial intelligence in the storytelling domain [5].

Figure:6 Home Page of the Ai Story Generator

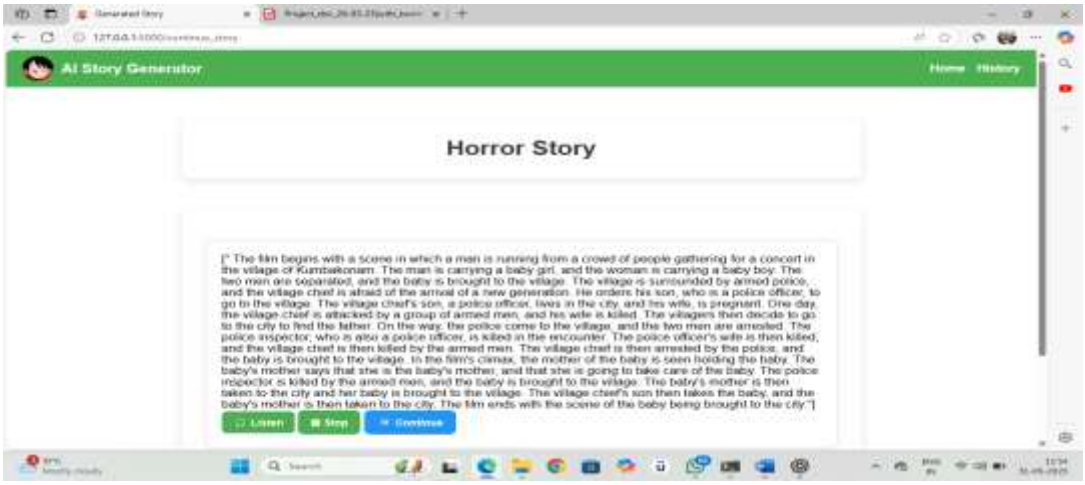

Figure:7 Generated story output

IV.RESULTS

The AI Story Generator project demonstrated the power of transformer-based models like GPT-2, GPT-2-large, and GPT-Neo-125M in crafting creative and coherent stories [12], [22]. By combining genre selection and user prompts, the system produced engaging narratives across Adventure, Sci-Fi, Horror, Fantasy, and Romance [1].

Each story reflected the intended tone and structure, with accurate grammar and context. The story continuation feature allowed users to extend narratives seamlessly, enhancing interactivity [3]. The system's performance and user-friendly interface made it a practical and enjoyable tool for real-time AI-driven storytelling [5], [19].

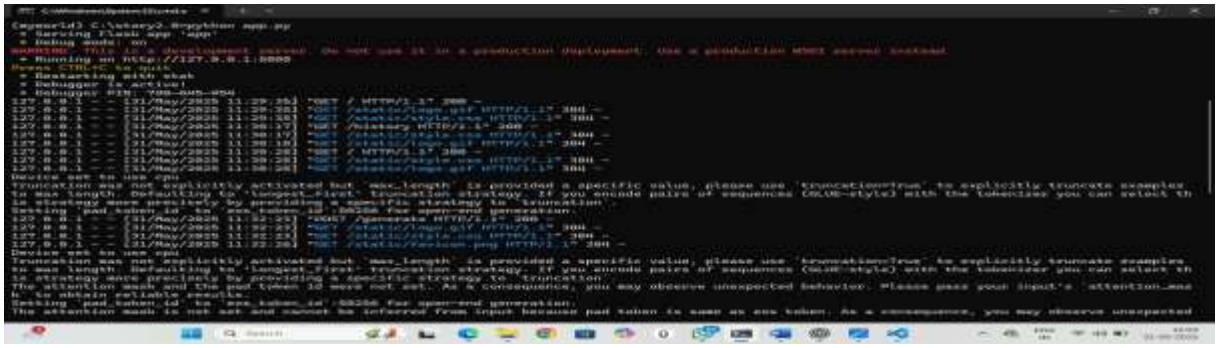

Figure:8 Background Model working

V. DISCUSSIONS

The AI Story Generator effectively showcases the power of transformer-based models like GPT-2, GPT-2-large, and GPT-Neo-125M in generating genre-specific, coherent, and creative stories from user prompts [12], [22]. Features like story continuation, text-to-speech, MongoDB-based history storage, and optional image generation enhance interactivity and usability [1], [19].

The system offers a user-friendly and accessible platform for automated storytelling, though improvements in long-form coherence and genre consistency could be explored [5]. Future upgrades may include multilingual support, advanced models, and mobile compatibility. Overall, the project highlights AI's growing role in creative writing and digital content generation [3].

VI. CONCLUSION

The AI Story Generator project successfully demonstrates the potential of transformer-based language models—such as GPT-2, GPT-2-large, and GPT-Neo-125M—in automatically generating creative and coherent narratives across multiple genres. By integrating a user-friendly interface that allows users to input prompts and select genres like Adventure, Sci-Fi, Horror, Fantasy, and Romance, the system tailor's stories to user preferences, producing contextually relevant and grammatically accurate outputs. The model's ability to maintain narrative consistency while reflecting appropriate tone and thematic elements makes it effective for short story creation. The story continuation feature enhances user engagement by enabling dynamic expansion of generated content, allowing users to build upon initial stories and explore creative possibilities interactively.

VII. FUTURE SCOPE

Future improvements could focus on enhancing long-form story coherence and expanding genre diversity [1], [5]. Incorporating larger and more advanced pre-trained language models can improve story quality and creativity [12], [22]. Additional features like multilingual support, emotional tone adjustment, and personalized story generation could further enrich the user experience [3]. Developing a mobile app or web platform with real-time story generation and multimedia integration would increase accessibility and user engagement [19].

VIII. ACKNOWLEDGEMENT

Bonumaddi kumari working as an Assistant professor in master of computer application sanketika vidya parishad engineering college, Visakhapatnam Andhra Pradesh. With 2 years of experience in computer science and engineering (CSE), accredited by NAAC.with her area of intrest in java full stack.

Vanapalli Prasanth Kumar is pursuing his final semester MCA in Sanketika Vidya Parishad Engineering College, accredited with A grade by NAAC, affiliated by Andhra University and approved by AICTE. With interest in Artificial intelligence . V.Prasanth Kumar has taken up his PG project on AI Story Generator and published the paper in connection to the project under the guidance of B.kumari, Assistant Professor, SVPEC..

REFERENCES

[1] A systematic review of artificial intelligence technologies used for story writing
https://link.springer.com/article/10.1007/s10639-023-11741-5
[2]AI-generated characters for supporting personalized learning and well-being
https://www.nature.com/articles/s42256-021-00417-9
[3]Exploring An Ai-Supported Approach To Creative Writing In Secondary School English Classes
https://cyberleninka.ru/article/n/exploring-an-ai-supported-approach-to-creative-writing-in-secondary-school-english-classes
[4]TRANSFORMING THE FUTURE: A REVIEW OF ARTIFICIAL INTELLIGENCE MODELS
https://cyberleninka.ru/article/n/transforming-the-future-a-review-of-artificial-intelligence-models
[5]Artificial intelligence in the creative industries: a review
https://link.springer.com/article/10.1007/s10462-021-10039-7

[6]Generative AI for visualization: State of the art and future directions
https://www.sciencedirect.com/science/article/pii/S2468502X24000160
[7]Unleashing the power of generative AI in drug discovery
https://www.sciencedirect.com/science/article/pii/S135964462400117X
[8]AI for food: accelerating and democratizing discovery and innovation
https://www.nature.com/articles/s41538-025-00441-8
[9]The GenAI is out of the bottle: generative artificial intelligence from a business model innovation perspective
https://link.springer.com/article/10.1007/s11846-023-00696-z
[10]The plan recognition problem: An intersection of psychology and artificial intelligence
https://www.sciencedirect.com/science/article/abs/pii/0004370278900127
[11]Artificial Intelligence for Personalized Preventive Adolescent Healthcare
https://www.sciencedirect.com/science/article/pii/S1054139X20300951
[12]Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation
https://www.jair.org/index.php/jair/article/view/11173
[13]Artificial Intelligence Is Stupid and Causal Reasoning Will Not Fix It
https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2020.513474/full?trk=public_post_comment-text
[14]A survey of blockchain, artificial intelligence, and edge computing for Web 3.0
https://www.sciencedirect.com/science/article/abs/pii/S1574013724000510
[15]A comprehensive survey on generative AI for metaverse: enabling immersive experience
https://link.springer.com/article/10.1007/s12559-024-10342-9
[16]Unravelling the impact of generative artificial intelligence (GAI) in industrial applications: A review of scientific and grey literature
https://link.springer.com/article/10.1007/s40171-023-00356-x
[17]Peeping into the Future: Understanding and Combating Generative AI-Based Fake News
https://link.springer.com/article/10.1007/s12559-025-10457-7
[18]Challenges and opportunities in the development and clinical implementation of artificial intelligence based synthetic computed tomography for magnetic
https://www.sciencedirect.com/science/article/pii/S0167814024006571
[19]Artificial Intelligence in Clinical Diagnosis Opportunities, Challenges, and Hype
https://jamanetwork.com/journals/jama/article-abstract/2807166
[20]Google Gemini as a next generation AI educational tool: a review of emerging educational technology
https://link.springer.com/article/10.1186/s40561-024-00310-z
[21]Generative artificial intelligence: a historical perspective
https://academic.oup.com/nsr/article/12/5/nwaf050/8029900
[22]Comprehensive Review of Generative artificial Intelligence: Mechanisms, Models, and Applications
https://www.sciencedirect.com/science/article/pii/S1877050925017326
[23]Reinforcement Learning for Generative AI: State of the Art, Opportunities and Open Research Challenges
https://www.jair.org/index.php/jair/article/view/15278
[24]From novel turbine designs to artificial intelligence: a review of cutting-edge innovations in hydropower systems
https://fupubco.com/fuen/article/view/313
[25]The Present and Future of Adult Entertainment: A Content Analysis of AI-Generated Pornography Websites
https://link.springer.com/article/10.1007/s10508-025-03099-1
[26]From Chess and Atari to StarCraft and Beyond: How Game AI is Driving the World of AI
https://link.springer.com/article/10.1007/s13218-020-00647-w