Hallucinations in Large Language Models (LLMs)

G. Pradeep Reddy[§], Y. V. Pavan Kumar[†], K. Purna Prakash[*]
[§]Kookmin University, 77 Jeongneung-ro, Sungbuk-Gu, Seoul 02703, REPUBLIC OF KOREA
[†]School of Electronics Engineering, VIT-AP University, Amaravati 522237, Andhra Pradesh, INDIA
[*]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, INDIA
gpradeepreddy@kookmin.ac.kr, pavankumar.yv@vitap.ac.in, kpurnaprakash@kluniversity.in

2024 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream) | 979-8-3503-5241-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/ESTREAM61684.2024.10542617

***Abstract*—The recent advancements in neural network architectures, particularly transformers, have played a crucial role in the rapid progress of Large Language Models (LLMs). LLMs are trained on many parameters. By training these parameters on vast amounts of text data, LLMs can learn to generate reactions to a wide variety of prompts. These models have enabled machines to generate new data (human-like), driving significant developments in Natural Language Processing (NLP). They have demonstrated remarkable capabilities in producing new content. Besides their impressive performance, LLMs occasionally generate hallucinatory responses that produce nonsensical or inaccurate information. In simple terms, hallucinations in LLMs happen when the model generates information that may sound believable but is actually wrong. It can make up details or go beyond what it has learned from the training data, resulting in inaccurate output. These hallucinatory responses appear to be authentic but lack grounding in reality. Such hallucinations can include fabrications such as facts, events, or statements that lack support from real-world data. Addressing this issue is important to enhance the reliability of AI-generated content. Hallucinations can be a significant challenge in critical applications such as healthcare, law, etc. In this view, this paper delves into the phenomenon of hallucinations in the context of LLMs. The objective is to understand the causes, explore the implications, and discuss potential strategies for mitigation.**

***Keywords*—hallucinations, large language models (LLMs), large language model operations (LLMOps), nonsensical, natural language processing (NLP), transformers.**

I. Introduction

Artificial intelligence (AI) has become a ubiquitous technology, with applications in a wide range of industries and fields [1], [2]. From speech and image recognition to autonomous vehicles, recommendation systems, smart homes, and virtual assistants, AI is transforming the way people live and work [3]-[5]. Current research in AI is concentrated more on developing new methods for making AI systems more explainable and fair. For example, XAI (Explainable AI) is a field of research that strives to develop methods to make AI systems more explainable [6]. Besides, the Internet of Things (IoT), is a network consisting of physical objects that are embedded with sensors, software, and connectivity to collect and exchange data [7], [8]. IoT is playing a key role in the advancement of AI by providing a vast amount of data that can be used to train and improve AI models. Large Language Models (LLMs) are a kind of AI that are trained on huge datasets [9], [10]. LLMs are advanced NLP (Natural Language Processing) models that use a deep neural network architecture designed to understand, generate, and process human language at a sophisticated level [11]. The earliest example of language model invention was ELIZA in 1967, developed by Prof. J. Weizenbaum at the artificial intelligence laboratory of MIT. A breakthrough came in 2017 with the transformer architecture, which made significant advancements in this research [12]. Furthermore, with the accessibility of computational power and large amounts of data, LLMs are performing well [13]. Since their inception, LLMs have grown with the advancements in machine learning techniques. Researchers are continuously exploring new architectures to further enhance their capabilities. The major developments in the growth of LLMs are shown in Fig. 1. Large Language Model Operations (LLMOps) focus on the challenges and best practices of deploying and managing LLMs in production environments. Key steps involved in the LLMOps are shown in Fig. 2.

The data used to train LLMs is massive and collected from various internet sources, including books, articles, websites, etc. [14]. The simplified representation of the entire process is shown in Fig. 3. LLMs can be used in a large range of industries and use cases including retail, healthcare, tech, etc. [15]. A comparison of popular LLMs is given in Table I.

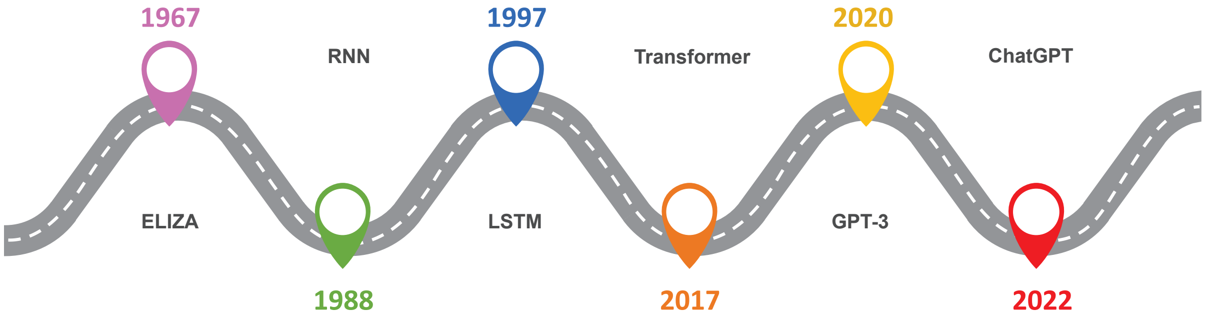

Fig. 1. Key developments in the evaluation of LLMs.

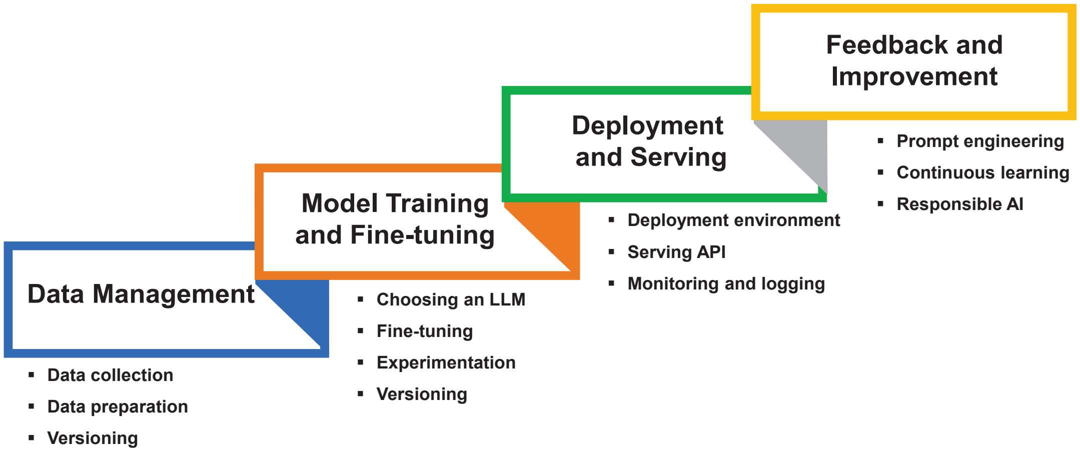

Fig. 2. LLMOps lifecycle.

TABLE I. COMPARISON OF VARIOUS LLMS

Model	Parameters (billion)	Parent Company
Generative Pre-trained Transformer 3 (GPT-3)	175	OpenAI
Large Language Model Meta AI (LLaMA)	65	Meta AI
Language Model for Dialogue Applications (LaMDA)	137	Google
Chinchilla	70	DeepMind
Megatron-Turing Natural Language Generation (MT-NLG)	530	Microsoft and NVIDIA
Gopher	280	DeepMind
Pathways Language Model (PaLM)	540	Google

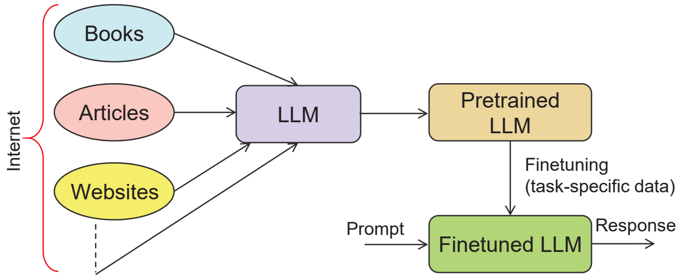

Fig. 3. The process of fine-tuning an LLM.

II. TRANSFORMER ARCHITECTURE

After the invention of the transformer architecture [12], it has evolved as the current advancement in the NLP [16]. Researchers are actively exploring the application of transformer architecture in diverse domains (e.g. vision transformers) [17]. Fig. 4 illustrates the high-level view of the transformer architecture. It includes two major parts: the decoder and the encoder. The encoder takes the input sequence and processes it to create a sequence of representations. The decoder takes these representations from the encoder and generates an output sequence.

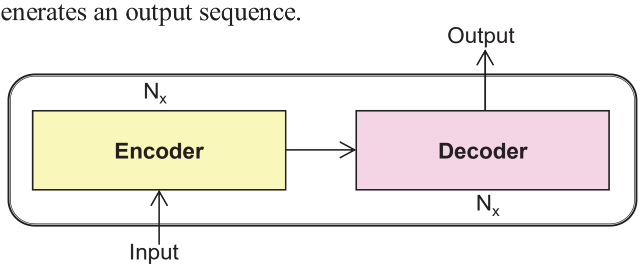

Fig. 4. Transformer encoder and decoder architecture.

These encoder and decoder layers are not used just once but rather stacked on top of each other, creating a deeper neural network. Each layer refines the representation of the input or output based on the previous layers. "NX" in Fig. 4 signifies that the transformer architecture utilizes N repetitions of the encoder-decoder structure.

The first step in preparing text for analysis is to tokenize the input sequence. The next step is input embedding, which converts the tokens into vectors. Positional encoding helps to capture the position of each token in the input sequence. The positional encoding precedes the input embedding prior to that they are fed into the encoder. Figure 5 provides a visual representation of all the steps involved in generating the positional encoding layer output. The output of the positional encoding layer is given as input to the encoder blocks parallelly. This is one of the advantages of the transformer network when compared to other neural networks because it can make use of current Graphics processing unit (GPU) parallel processing. This allows the GPU to process multiple input sequences at the same time.

Whereas in many other conventional neural networks, the input is given sequentially, which means that the GPU can only process one input sequence at a time. This can significantly slow down the training process, especially for large input sequences. Thus, parallel processing enables the transformers computationally efficient on modern GPU hardware. Each encoder block has one feed-forward layer and one multi-head attention layer. Self-attention is a mechanism that allows each word in the input sequence to attend to all other words in the same sequence. It captures the significance of each word in the context of the whole sequence.

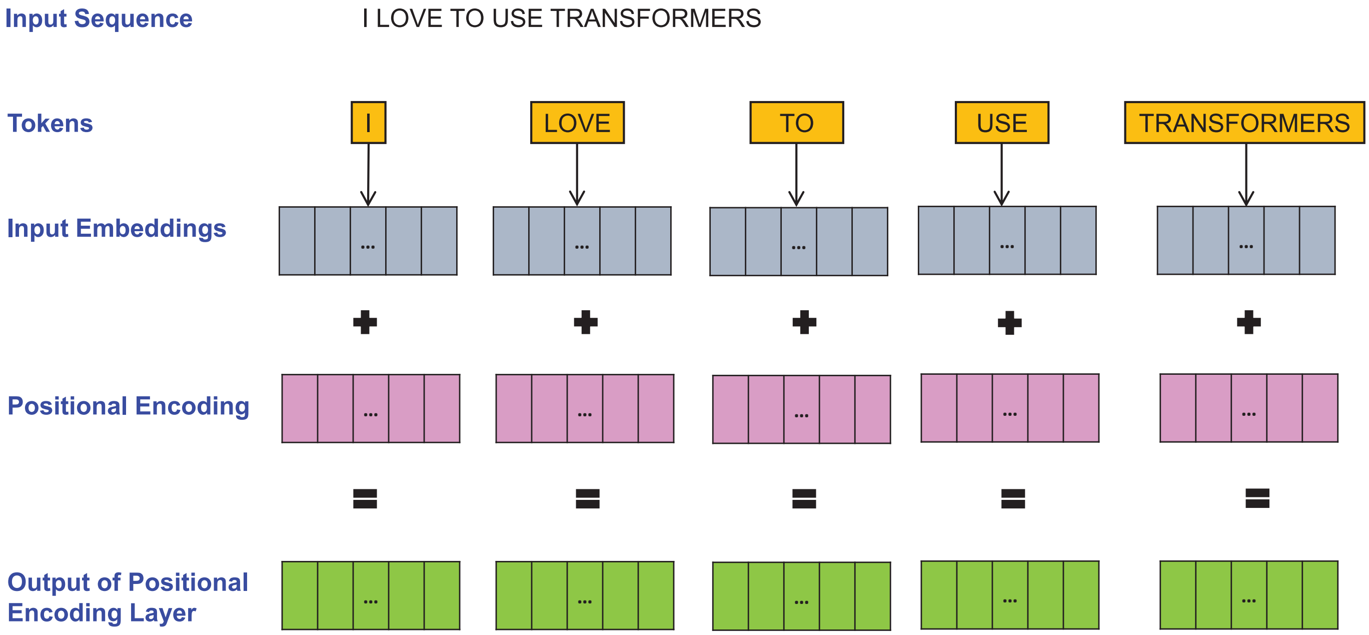

Fig. 5. Process for generating the output of the positional encoding layer.

The process involves comparing the similarity between each word's representation with all other words' representations to calculate attention scores (with the help of key (K), query (Q), and value (V)). Here, the mathematical operation "dot product" is used to compute the similarity between two vectors. Self-attention matrix is computed using (1) [12], where, d_k refers to the keys dimension. Multi-head attention permits the model to attend to multiple different representations of the sequence.

$$\text{Attention}(K,Q,V) = \text{softmax}\left(\frac{K^T Q}{\sqrt{d_k}}\right)V \qquad (1)$$

The add & norm operation is applied after each sub-layer in the transformer stack, including attention and feed-forward layers. The addition operation is used to add the output of the multi-head attention mechanism to the input sequence. Layer normalization is used to normalize the output of the addition operation. This process helps in creating a stable and effective model by reducing the vanishing gradient problem during training and improving the flow of information through the network. The feed-forward layer is a fully connected neural network that is given to each position separately and concurrently. It consists of two linear transformations with a ReLU activation in between. The feed-forward layer allows the model to capture complex patterns and dependencies between words in the input sequence, making it more powerful in handling various NLP tasks. The output of the encoder block is given as input to the decoder block. The decoder block consists of attention, feed-forward, and add & norm layers. Here one of the attention layers used is the masked multi-head attention layer. It is used to stop the decoder from attending to future words in the output sequence during training. The output of the decoder block is given to linear and softmax layers. These layers are responsible for generating the final predictions for the output sequence.

III. Causes of Hallucinations

Sometimes LLMs can generate hallucinatory responses that are incorrect and misleading [18]. As the outputs are produced in natural language, they may appear very convincing, even though they are wrong. Intrinsic and extrinsic hallucinations are two types of hallucinations [19]. Intrinsic hallucinations refer to hallucinations that are generated by manipulating the content in the input. These are like the direct contradiction between the input and the generated content. For example, if the input to LLM is "Ramu likes to play cricket", then the output will be "Ramu likes to play basketball". Whereas extrinsic hallucinations occur when the outputs cannot be validated as true or false based only on the input. For example, if the input to LLM is "The weather in Hyderabad is very good", then the output will be "The weather in Hyderabad is very good, and it has attracted 2 lakh tourists because of its beautiful weather conditions". It may or may not be true that it has attracted 2 lakh tourists, but it can't be verified from the input. This section discusses some of the major causes of hallucinations.

- ***Biases in Training Data:*** LLMs are typically trained on a large corpus of data, which might have imbalances or unfair representations of specific groups or topics, resulting in unintended consequences for the model's behavior and outputs. Some features might be linked to class labels falsely which leads models to learn misleading patterns and produce inaccurate predictions.
- ***Limited Context:*** LLMs do not understand the context the same as humans do. This leads to hallucinations where the model generates responses that do not closely match the input or fail to provide relevant information. This means sometimes they can produce a response that does not apply to the context.

- ***Ambiguous Prompts:*** An ambiguous prompt means it has several possible meanings or interpretations. When LLMs come across such prompts they depend on their prior knowledge or expectations to fill in gaps leading to hallucinations. If the prompt lacks clarity, the model generates unintended responses that are not as per the user.
- ***Adversarial Attacks:*** Adversarial attacks are used to manipulate the input data given to the model to generate misleading or incorrect responses. These attacks focus on the context to produce inaccurate responses. These attacks take advantage of the vulnerabilities in the model's behavior, revealing weaknesses and spotlighting areas where the model's understanding and reasoning capabilities are lacking.
- ***Overfitting:*** This occurs when the model learns the training data too well and is incapable of generalizing the new data. This can lead to the model making predictions that are accurate for the training data but are inaccurate for new data. It's essential to recognize the potential for overfitting to result in hallucinations in LLMs. When using LLMs, critical evaluation of the output is crucial, as it's important to be aware of the possibility that the LLM is generating a response that is inaccurate or nonsensical.
- ***Model Architecture:*** The model architecture includes the design and structure, such as the parameters or number of layers. Increased complexity in models with more layers or parameters can make them more prone to the generation of hallucinations. The architecture, complexity, covering design, and learning mechanisms, ominously influence behavior and proneness to produce hallucinatory responses.

IV. Implications of Hallucinations

The effects of hallucinations can be based on the frequency of occurrence, specific type, and underlying cause behind them [20]. This section discusses the general implications of the hallucinations as shown in Fig. 6.

A. Ethical Considerations

1) Spread of misinformation: When LLMs produce fabricated or incorrect information, people may consider this and share these details which results in web deceptive content. Bad actors take advantage of this vulnerability to impact public opinion or promote untruths, causing a serious impact on public discourse.

2) Legal and regulatory challenges: Generating false or misleading information by an LLM can lead to legal liability. In addition to that, there are numerous points to look at such as liability for harm caused by false information and intellectual property rights. Accountability and balancing innovation are very important to address the legal implications of hallucinations.

B. Social Impact

1) Erosion of trust: When LLMs produce misleading information, it can destroy people's trust in the information they receive. This leads to a lot of problems. Trust plays an important role in the wide acceptance of LLMs. If users perceive that LLM is inclined towards hallucinations, it will raise doubts about the reliability and dependability of its outputs. Such erosion of trust can obstruct the adoption of LLMs.

2) Influence on decision-making: Hallucinations can influence people's beliefs, which can then affect their decision-making. For example, if an LLM generates a response that supports a particular belief, it can make people more likely to believe that belief. This can lead to people making decisions that are consistent with that belief, even if the belief is not true.

C. Practical Consequences

1) Damaging reputation and brand image: Hallucinations can significantly harm the credibility and reputation of the companies or organizations that use LLMs. Inaccurate or misleading outputs from LLMs not only erode trust in the technology itself but also raise doubts about the competence and reliability of the entities implementing them.

2) Risks in critical applications: In applications where accuracy is very important, such as medical diagnoses, legal analysis, or autonomous systems, the risks associated with hallucinations are particularly severe. Hallucinatory information in these complex domains leads to improper diagnoses, legal issues, accidents, or other terrible consequences.

V. Mitigation Strategies

LLMs are moving into the deployment phase at scale and are used by a huge number of users. So, it's important to know various strategies which help to minimize the effect of hallucinations in LLMs. This section talks about some of the important mitigation strategies.

- ***Dataset Curation:*** Dataset curation is the process of selecting, cleaning, and organizing data for a certain objective. It is important to address biases and make sure the training data is balanced and representative. In the context of LLMs, dataset curation helps to reduce hallucinations by making sure the data used to train the model is reliable, relevant, and accurate.
- ***Improving Context Representation:*** Adding some more context, such as historical information or user-specific data can help the model to produce responses that are more contextually relevant and aligned with the user's intent. Prompt engineering can help refine input prompts. It provides clear instructions to the model. Users can guide the model by carefully crafting prompts that lead to generating more accurate and non-hallucinating responses.
- ***Human-in-the-loop Approaches:*** These approaches involve human feedback or interference in the training, testing, or deployment of LLMs [21]. Humans evaluate and validate the accuracy of the model's responses, which helps in reducing the probability of hallucinatory responses.

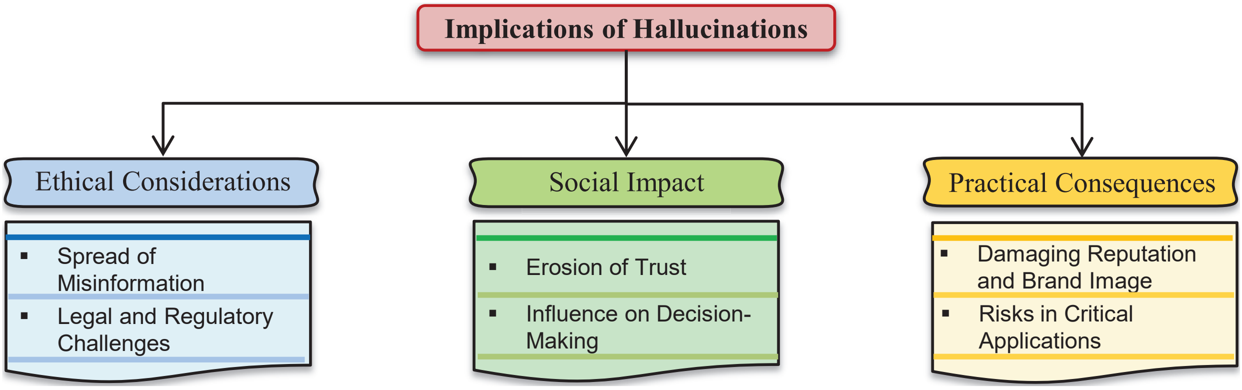
Implications of Hallucinations
Ethical Considerations
Spread of Misinformation
Legal and Regulatory Challenges
Social Impact
Erosion of Trust
Influence on Decision-Making
Practical Consequences
Damaging Reputation and Brand Image
Risks in Critical Applications

Fig. 6. Potential implications of hallucinations.

- ***Filtering Mechanisms:*** Filtering mechanisms are used for reducing hallucinations with LLMs both in terms of frequency and severity. By validating the generated outputs, these methods detect and mitigate hallucinations, thereby it flags or correct outputs that are not accurate. Factual verification and logic filters are two examples of filters that are helpful.
- ***Regularization Techniques:*** Regularization techniques are methods that prevent overfitting or memorization of training data by LLMs. Overfitting happens when a model learns the training data very closely, and as an outcome, it is unable to generalize the new data. Techniques such as dropout encourage the model to generalize rather than memorize the training data. This reduces the risk of the model generating erroneous output based on memorized information.

VI. Summary

The phenomenon of hallucinations in Large Language Models (LLMs) poses a substantial challenge that needs to be solved to ensure the reliability and accuracy of AI-generated content. While LLMs have proven remarkable capabilities in generating good responses, the occasional generation of nonsensical or inaccurate information can undermine their trustworthiness. This paper explored the causes and implications of these hallucinations in LLMs. Further, various mitigation strategies are also discussed. The collaboration between researchers, policymakers, and industry leaders is important to ensure that the developments align with societal values. With responsible deployment, LLMs can become powerful tools for advancing human progress across a wide range of applications.

References

[1] G. P. Reddy and Y. V. Pavan Kumar, “A Beginner’s Guide to Federated Learning,” in *2023 Intelligent Methods, Systems, and Applications (IMSA)*, Giza, Egypt: IEEE, Jul. 2023, pp. 557–562. doi: https://doi.org/10.1109/IMSA58542.2023.10217383.

[2] V. K. Velpula and L. D. Sharma, “Multi-stage glaucoma classification using pre-trained convolutional neural networks and voting-based classifier fusion,” *Front. Physiol.*, vol. 14, p. 1175881, Jun. 2023, doi: https://doi.org/10.3389/fphys.2023.1175881.

[3] K. Radha and M. Bansal, “Audio Augmentation for Non-Native Children’s Speech Recognition through Discriminative Learning,” *Entropy*, vol. 24, no. 10, p. 1490, Oct. 2022, doi: https://doi.org/10.3390/e24101490.

[4] P. P. Kasaraneni, Y. Venkata Pavan Kumar, G. L. K. Moganti, and R. Kannan, “Machine Learning-Based Ensemble Classifiers for Anomaly Handling in Smart Home Energy Consumption Data,” *Sensors*, vol. 22, no. 23, p. 9323, Nov. 2022, doi: https://doi.org/10.3390/s22239323.

[5] N. Chandrasekhar and S. Peddakrishna, “Enhancing Heart Disease Prediction Accuracy through Machine Learning Techniques and Optimization,” *Processes*, vol. 11, no. 4, p. 1210, Apr. 2023, doi: https://doi.org/10.3390/pr11041210.

[6] G. P. Reddy and Y. V. P. Kumar, “Explainable AI (XAI): Explained,” in *2023 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream)*, Vilnius, Lithuania: IEEE, Apr. 2023, pp. 1–6. doi: https://doi.org/10.1109/eStream59056.2023.10134984.

[7] G. Pradeep Reddy and Y. V. Pavan Kumar, “Internet of Things Based Communication Architecture for Switchport Security and Energy Management in Interoperable Smart Microgrids,” *Arab J Sci Eng*, vol. 48, no. 5, pp. 5809–5827, May 2023, doi: https://doi.org/10.1007/s13369-022-07056-1.

[8] G. Pradeep Reddy and Y. V. Pavan Kumar, “Retrofitted IoT Based Communication Network with Hot Standby Router Protocol and Advanced Features for Smart Buildings,” *International Journal of Renewable Energy Research*, vol. 11, no. 3, pp. 1354–1369, 2021, doi: https://doi.org/10.20508/ijrer.v11i3.12222.g8276.

[9] R. Bommasani, P. Liang, and T. Lee, “Holistic Evaluation of Language Models,” *Annals of the New York Academy of Sciences*, vol. 1525, no. 1, pp. 140–146, Jul. 2023, doi: https://doi.org/10.1111/nyas.15007.

[10] Y. Chang *et al.*, “A Survey on Evaluation of Large Language Models,” *ACM Trans. Intell. Syst. Technol.*, vol. 15, no. 3, pp. 1–45, Jun. 2024, doi: https://doi.org/10.1145/3641289.

[11] D. Bahdanau, K. Cho, and Y. Bengio, “Neural Machine Translation by Jointly Learning to Align and Translate,” 2014, doi: https://doi.org/10.48550/ARXIV.1409.0473.

[12] A. Vaswani *et al.*, “Attention Is All You Need,” 2017, doi: https://doi.org/10.48550/ARXIV.1706.03762.

[13] J. Hoffmann *et al.*, “Training compute-optimal large language models,” in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, in NIPS ’22. Red Hook, NY, USA: Curran Associates Inc., 2024.

[14] M. Shoeybi, M. Patwary, R. Puri, P. LeGresley, J. Casper, and B. Catanzaro, “Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism,” 2019, doi: https://doi.org/10.48550/ARXIV.1909.08053.

[15] Yao *et al.*, "Tree of Thoughts: Deliberate Problem Solving with Large Language Models," 2023, doi: https://doi.org/10.48550/ARXIV.2305.10601.

[16] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *AI Open*, vol. 3, pp. 111–132, 2022, doi: https://doi.org/10.1016/j.aiopen.2022.10.001.

[17] A. Dosovitskiy *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," 2020, doi: https://doi.org/10.48550/ARXIV.2010.11929.

[18] Z. Ji *et al.*, "Survey of Hallucination in Natural Language Generation," *ACM Comput. Surv.*, vol. 55, no. 12, pp. 1–38, Dec. 2023, doi: https://doi.org/10.1145/3571730.

[19] J. Maynez, S. Narayan, B. Bohnet, and R. McDonald, "On Faithfulness and Factuality in Abstractive Summarization," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online: Association for Computational Linguistics, 2020, pp. 1906–1919. doi: https://doi.org/10.18653/v1/2020.acl-main.173.

[20] T. B. Brown *et al.*, "Language Models are Few-Shot Learners," 2020, doi: https://doi.org/10.48550/ARXIV.2005.14165.

[21] L. Ouyang *et al.*, "Training language models to follow instructions with human feedback," 2022, doi: https://doi.org/10.48550/ARXIV.2203.02155.