International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-13 Issue-4, March 2024

A Case Study on the Diminishing Popularity of Encoder-Only Architectures in Machine Learning Models

Praveen Kumar Sridhar, Nitin Srinivasan, Adithyan Arun Kumar, Gowthamaraj Rajendran, Kishore Kumar Perumalsamy

Abstract: ***This paper examines the shift from encoder-only to decoder and encoder-decoder models in machine learning, highlighting the decline in popularity of encoder-only architectures. It explores the reasons behind this trend, including advancements in decoder models that offer superior generative capabilities, flexibility across various domains, and enhancements in unsupervised learning techniques. The study also discusses the role of prompting techniques in simplifying model architectures and enhancing model versatility. By analyzing the evolution, applications, and shifting preferences within the research community and industry, this paper aims to provide insights into the changing landscape of machine learning model architectures.***

Keywords: ***Machine Learning, Deep Learning, Encoder, Transformers, Decoder, Natural Language Processing, Generative Model, Model Evolution.***

I. INTRODUCTION

In the landscape of machine learning and natural language processing, encoder and decoder architectures have long been foundational components, enabling a diverse array of applications, from machine translation to text summarisation. The encoder model is designed to process input data and encode it into a dense, fixed-sized representation, which can then be utilized by a decoder to generate or predict output. This synergy between encoder and decoder models has been pivotal in achieving state-of-the-art results in numerous tasks [1]. However, recent years have seen a marked shift towards the adoption of decoder-only models, such as GPT (Generative Pre-trained Transformer),

Manuscript received on 28 February 2024 | Revised Manuscript received on 08 March 2024 | Manuscript Accepted on 15 March 2024 | Manuscript published on 30 March 2024.
*Correspondence Author(s)
Praveen Kumar Sridhar*, Department of Data Science, Northeastern University, San Jose, United States. E-mail: prasri.pk@gmail.com, ORCID ID: 0009-0001-6486-1614
Nitin Srinivasan, Department of Computer Science, University of Massachusetts Amherst, Sunnyvale, United States. E-mail: nitinsr1217@gmail.com, ORCID ID: 0009-0001-6472-4441
Adithyan Arun Kumar, Department of Information Security, Carnegie Mellon University, San Jose, United States. E-mail: adithyanhaxor@gmail.com, ORCID ID: 0000-0001-9790-2657
Gowthamaraj Rajendran, Department of Information Security, Carnegie Mellon University, San Jose, United States. E-mail: gowthamaraj360@gmail.com, ORCID ID: 0000-0001-9422-3907
Kishore Kumar Perumalsamy, Department of Computer Science, Carnegie Mellon University, San Jose, United States. E-mail: kishore7173@gmail.com, ORCID ID: 0009-0009-5661-7644

Which have demonstrated remarkable proficiency in generating coherent and contextually relevant text based solely on the input they receive, without the need for a separate encoding step [2]. The purpose of this paper is to investigate the factors contributing to the decline in popularity of encoder-only models in favour of decoder-only and encoder-decoder frameworks, especially in the context of advancements in machine learning and artificial intelligence. By examining the evolution of these models, their applications, and the shifting preferences within the research community and industry, this case study aims to provide insights into the changing landscape of model architectures. This exploration not only highlights the technological advancements driving these trends [3] but also sheds light on the implications for future research directions and practical applications in the field [4][5].

II. BACKGROUND

A. Evolution of Encoder Models

The evolution of encoder models has been pivotal in the advancement of artificial intelligence, particularly in the realms of natural language processing (NLP) and machine learning. Encoder models are fundamentally designed to ingest input data—whether text, audio, or image—and encode it into a dense, contextual representation. This encoding captures the essential features and nuances necessary for a wide range of computational tasks. The technical foundation of these models is built upon sophisticated algorithms and architectures, with recurrent neural networks (RNNs) initially playing a central role due to their ability to handle sequential data. The introduction of the transformer architecture, however, marked a significant evolution, employing self-attention mechanisms to process data in parallel, thereby enhancing the model's efficiency and ability to capture complex dependencies within the input [1]. This historical progression of encoder models is characterized by several key milestones, including the development of Long Short-Term Memory (LSTM) networks, which addressed the challenge of learning long-term dependencies in sequence data—a limitation of earlier RNNs. The publication of the transformer model in "Attention is All You Need" represented a paradigm shift, enabling more sophisticated encoder architectures that have since dominated the field.

Encoder models have found success in a myriad of applications, from machine translation, where they have dramatically improved the quality and efficiency of translations, to voice recognition systems and even in the analysis of biological data, demonstrating their versatility and the profound impact they have had across various domains [6].

B. The Peak of Encoder Models

The zenith of encoder models' influence and adoption can be traced back to a period shortly after the introduction of the transformer architecture, which heralded a new era of efficiency and capability in handling complex sequences of data. This period, spanning the late 2010s and early 2020s, saw encoder models, especially those based on transformer technology, becoming foundational elements in a wide array of NLP tasks, including machine translation, text summarization, and sentiment analysis. The unparalleled ability of these models to capture deep contextual relationships within data, without the constraints imposed by the sequential processing inherent to RNNs and LSTMs, contributed significantly to their peak popularity. Technologies like BERT (Bidirectional Encoder Representations from Transformers) and its variants, which utilized encoder architectures to understand the nuances of language in unprecedented depth, exemplify this pinnacle of achievement and widespread adoption in both academic research and practical applications [3].

Several key factors contributed to the widespread adoption and the peak of encoder models during this time. Firstly, the improvement in computational power and the availability of large-scale datasets enabled the training of more complex models, making the sophisticated capabilities of encoder models accessible and practical for real-world applications. Secondly, the versatility of encoder models demonstrated through their successful application across different domains beyond NLP, such as computer vision and bioinformatics, showcases their adaptability and utility in tackling a variety of computational challenges. Additionally, the open-source movement within the AI research community played a critical role, with organizations and researchers sharing pre-trained models and tools, thereby democratizing access to state-of-the-art technology and fueling further innovation and experimentation across the field [1].

C. Emerging Challenges and Limitations

As the landscape of artificial intelligence and natural language processing evolved, encoder models, despite their significant contributions, began facing emerging challenges and limitations, particularly when compared to decoder-only or encoder-decoder frameworks. The inherent limitations of encoder models become evident in their handling of complex tasks that require not just the understanding and encoding of input data but also the generation of new content or responses based on that understanding. While encoder models excel in creating rich, contextual representations of input data, they lack the intrinsic capability to generate output sequences, a task for which decoder or encoder-decoder architectures are specifically designed. This limitation is particularly pronounced in applications requiring creative content generation, such as story generation, or in tasks requiring precise, contextually relevant responses, such as in conversational AI systems [7].

Moreover, encoder models face specific challenges in processing sequential data and scalability. Despite advances, such as the introduction of the transformer architecture, which mitigated some of the sequential data processing issues inherent to RNNs and LSTMs, encoder models still struggle with scalability and computational efficiency. Training these models requires substantial computational resources, especially as the complexity of tasks and the size of datasets increase. Additionally, encoder models often struggle to handle long-range dependencies within data, a limitation that becomes increasingly problematic as the length of the input sequences increases. This issue is not just a matter of computational power but also of the architectural limitations in capturing and retaining information over long sequences, which can lead to a loss of context or relevance in the encoded representations [1].

Integration challenges with downstream tasks further exacerbate the limitations of encoder models. Unlike encoder-decoder models, which can seamlessly integrate the encoding and generation processes within a single framework, encoder models often require additional mechanisms or architectures to effectively utilise the encoded representations in generating outputs. This separation can lead to inefficiencies and a lack of coherence between the understanding of input data and the generation of outputs. Furthermore, encoder models traditionally do not incorporate a general or external knowledge source, making it difficult to update them with changing information, such as the current president of the United States, without retraining. This limitation highlights a significant challenge in maintaining the relevance and accuracy of models in dynamic, real-world scenarios where information can rapidly change [3].

III. SHIFT TOWARDS DECODER-ONLY MODELS

The shift towards decoder-only architectures in natural language processing (NLP) has been catalysed by significant advancements in these models, which have demonstrated remarkable proficiency in a wide array of tasks, particularly those involving language generation. Decoder-only architectures, exemplified by models such as GPT (Generative Pre-trained Transformer), harness the power of transformers to generate text by predicting the next word in a sequence based on all the previous words. This approach has not only simplified the model architecture by eliminating the need for a separate encoder but has also facilitated training on a broader range of generative tasks, making these models extremely versatile and powerful in applications that range from automated content creation to conversational AI systems [8].

A comparative analysis between decoder-only and encoder-only models highlights the intrinsic advantages of the former, especially in tasks that require generation and creativity.

Retrieval Number: 100.1/ijitee.D982713040324
DOI: 10.35940/ijitee.D9827.13040324
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Decoder-only models inherently excel at generating coherent and contextually relevant text sequences, outperforming encoder-only models in terms of fluency and versatility. This superiority is attributed to their architecture, which is optimized for sequential data generation, enabling them to capture and replicate the nuances of human language more effectively.

Furthermore, the ability of decoder-only models to perform unsupervised learning on vast corpora of text allows them to develop a deep understanding of language patterns and structures, which is pivotal for tasks requiring a high degree of linguistic sophistication [1].

Technical innovations have played a crucial role in favouring decoder models over their encoder-only counterparts. Breakthroughs in self-attention mechanisms, a core component of transformer architectures, have allowed decoder-only models to efficiently process and generate text by focusing on different parts of the input sequence as needed, without the constraints of sequential data processing. Efficiency gains have been further bolstered by sparse attention and other techniques such as adaptive attention span and locality-sensitive hashing, which reduce the computational load by focusing on the most relevant parts of the input data, thereby enabling the training of larger, more powerful models [9]. These innovations have not only enhanced the models' performance but have also made them more scalable and efficient in handling complex generative tasks.

The application domains driving the popularity of decoder models underscore their impact and the growing importance of natural language generation (NLG). With the advent of large language models (LLMs) like GPT-3, GPT-4, and LLaMA [10], the field has seen breakthroughs in a variety of creative and generative tasks, from writing coherent and contextually rich narratives to generating code from natural language descriptions. These models have transcended traditional NLP tasks, venturing into areas such as automated journalism, poetry generation, and even creating content for virtual reality environments. The expansion of decoder-only models into these domains reflects not only their technical capabilities but also their potential to redefine how we interact with and leverage technology for creative expression [4].

IV. ADVANTAGES OF DECODERS OVER ENCODERS

A. Enhanced Generative Capabilities

Decoder models are inherently better suited for generative tasks, such as text generation, due to their design, which emphasizes sequential processing. This architectural focus enables them to excel in generating coherent and contextually relevant content, outperforming encoder models in tasks that require the creation of new data from learned patterns. For example, in the realm of natural language processing, decoder-only models like GPT-3 have demonstrated an unparalleled ability to produce text that is not only grammatically correct but also contextually rich and varied, spanning a wide range of genres and styles. The success of these models in generating high-quality content across diverse domains underscores their superior generative capabilities [11].

B. Improved Handling of Sequential Data

Decoder models process data sequentially, which grants them a significant advantage in managing temporal dependencies and context. This capability is particularly vital in language modelling, where understanding the flow and relationship between words in a sentence determines the quality of the generated content. By treating the input as a sequence of tokens and predicting the next token based on the preceding context, decoder models can capture the nuances of language, including syntax and semantics, more effectively than encoder models. This sequential processing leads to a more accurate representation of language patterns, facilitating superior performance in tasks like text completion and conversation generation [1].

C. Flexibility in Task Adaptation

The versatility of decoder models in adapting to a wide array of tasks without substantial architectural modifications is another significant advantage. These models can be employed for translation, summarisation, and even creative writing, demonstrating their ability to understand and generate text across various contexts and formats. This flexibility stems from their generalist approach to processing language, where the same model can be fine-tuned for multiple applications, demonstrating a high degree of adaptability. Such versatility is not only a testament to the efficiency of decoder models but also highlights their potential to serve multiple purposes within the AI domain, reducing the need for specialized models for each task [4].

D. Efficiency in Learning Representations

Decoder models, particularly those utilising self-attention mechanisms, are adept at learning rich and nuanced representations of data. This ability is crucial for improved performance on downstream tasks, as it allows the model to capture and utilize complex patterns within the data. The self-attention mechanism, by focusing on different parts of the input sequence for each prediction, enables the model to consider the entire context of the input, leading to a deeper understanding of the data. This comprehensive approach to data representation is a key factor in the success of decoder models in a wide range of applications, from language understanding to generative tasks, showcasing their efficiency and effectiveness in learning from data [3].

E. Simplified Architectural Requirements

Decoder-only architectures are celebrated for their relative simplicity compared to models that utilize both encoders and decoders or encoder-only configurations. This simplicity not only streamlines the model development and training processes but also lowers the barrier to entry for new researchers and developers looking to innovate within the field of natural language processing (NLP). By focusing on the sequential processing of data, decoder-only models eliminate the need for complex alignment and transformation processes typically required in encoder-decoder architectures, facilitating more straightforward implementation and experimentation [1].

F. Scalability and Performance

Decoder models have been at the forefront of scalability and performance enhancements in NLP. Innovations in parallel processing techniques and the refinement of attention mechanisms, particularly self-attention, have enabled these models to process large volumes of data while maintaining efficiency, or even improving performance. Such advancements have made decoder-only models particularly adept at handling the demands of large-scale applications, from generating coherent, lengthy texts to understanding and translating languages at an unprecedented scale. The ability of decoder models to scale effectively while improving computational efficiency has been a key factor in their widespread adoption [4].

G. Enhanced Language Understanding

Large language models (LLMs) based on decoder architectures have set new benchmarks in understanding the subtleties of human language, including context, nuance, and even elements of common sense reasoning. By analyzing vast datasets, these models have developed a profound understanding of language patterns, idiomatic expressions, and cultural references, enabling them to generate responses that are not only relevant but often indistinguishable from those a human might produce. Examples of decoder models demonstrating this level of language understanding include GPT's ability to generate news articles, stories, and dialogue that reflect a sophisticated grasp of language and context [4].

H. Advancements in Unsupervised Learning

Decoder models have significantly advanced the field of unsupervised learning by leveraging large volumes of unlabeled data to learn patterns, generate predictions, and provide insights. These models, through their flexible architecture, are particularly suited to unsupervised learning tasks, where they can infer structure from unlabelled data, generate coherent text, and even perform functions like translation without direct examples of input-output pairs. The progress in unsupervised learning techniques, driven by decoder-only models, has opened new avenues for NLP applications, enabling more robust and versatile systems capable of learning from the vast amount of data available on the internet without extensive human annotation [3].

I. Impact of RAG with Decoder-only Models

Retrieval-Augmented Generation (RAG) represents a significant advancement in combining the retrieval of relevant information with the generative capabilities of decoder-only models, thereby enhancing the model's ability to generate responses that are not only coherent but also rich in context and informed by external data sources. This approach leverages a hybrid model architecture that integrates a retrieval component with a generative decoder model, such as GPT. By fetching pertinent information from a dataset or knowledge base before generating a response, RAG allows decoder models to produce outputs that are significantly more accurate and context-aware. This method effectively expands the knowledge base of decoder models beyond their training data, enabling them to answer questions and provide information that requires up-to-date or specialized knowledge, thereby addressing one of the traditional limitations of decoder-only models in handling queries that require external information [13].

J. Low-Rank Adaptation (LoRA)

Low-Rank Adaptation (LoRA) introduces a novel approach to enhancing the adaptability and efficiency of decoder models, particularly in the context of large language models like GPT and LLaMA. By applying low-rank matrices to adapt the pre-trained weights of a model, LoRA enables significant modifications to the model's behaviour without requiring extensive retraining or fine-tuning. This technique provides a mechanism to adjust the model's responses based on new tasks or data, improving its performance and versatility while maintaining the original model's integrity. LoRA's efficiency lies in its ability to make targeted adjustments to a model's weights, thereby enabling rapid adaptation to new tasks or changes in the data landscape with minimal computational overhead. This makes LoRA particularly valuable in scenarios where deploying fully retrained models is not feasible due to resource constraints, showcasing a practical path towards dynamic, adaptable NLP applications [14].

V. IMPACT OF PROMPT ENGINEERING ON THE ENCODER-DECODER DYNAMICS

A. Introduction to Prompting in Decoder Models

Prompting in the context of decoder models represents a paradigm shift in how we interact with machine learning systems, particularly in the field of natural language processing (NLP). Unlike traditional approaches that require explicit programming or task-specific fine-tuning, prompting guides model behaviour through natural language inputs that effectively "ask" the model to perform a task. This method leverages the pre-existing knowledge embedded within the model, enabling it to apply learned patterns to new, unseen tasks. By framing requests or functions in the form of prompts, users can leverage the model's generative capabilities without the need for direct task-specific programming, thereby streamlining the interaction process [4].

B. Decoder Models and Zero-Shot Learning

Large language models (LLMs), such as GPT-3, have been particularly adept at utilising prompting for zero-shot learning. In this scenario, the model performs tasks for which it has not been explicitly trained. This ability significantly reduces the dependency on encoder-based approaches, which are traditionally used for task-specific preprocessing. Through the strategic use of prompts, decoder models can understand and execute a broad range of tasks, from answering questions to generating content, without requiring additional training data or task-specific model adjustments. This approach not only demonstrates the flexibility of decoder models but also their potential to generalize across tasks, a significant advantage over more rigid, encoder-dependent architectures [4].

Retrieval Number: 100.1/ijitee.D982713040324
DOI: 10.35940/ijitee.D9827.13040324
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

C. Efficiency of Prompt Engineering

Prompt engineering has emerged as a critical skill set in the NLP domain, allowing users to "program" decoder models to perform various tasks effectively. The art and science of crafting the right prompts can significantly influence a model's performance, making prompt engineering a form of soft programming. This skill is particularly valuable in domains where traditional programming approaches are impractical or where the task complexity would require extensive data preprocessing and model fine-tuning. By efficiently utilizing prompts, practitioners can harness the power of advanced decoder models for applications ranging from language translation to creative content generation, demonstrating the versatility and adaptability of this approach [12].

D. Reduction in Model Complexity

The advent of prompt-based interaction with decoder models marks a significant step towards reducing the complexity inherent in NLP systems. Traditional NLP architectures often rely on specialized encoder modules designed to handle specific input types or tasks. However, the use of prompts eliminates the need for many of these specialized components, simplifying the overall model architecture. This simplification not only makes models more accessible to a broader range of users but also enhances their adaptability and efficiency. By leveraging the inherent knowledge and generative capabilities of decoder models, prompt-based approaches enable a more streamlined and flexible framework for NLP tasks, reducing both the technical barriers to entry and the computational resources required for state-of-the-art performance.

E. Enhanced Versatility and Adaptability

Decoder models, when coupled with sophisticated prompting strategies, have demonstrated unparalleled versatility and adaptability across a wide range of domains, significantly reducing the reliance on specialised encoder models. For instance, in natural language understanding (NLU) and generation (NLG) tasks, decoder models like GPT and LLaMA have been applied to create content, answer questions, and even generate code, all with minimal domain-specific adjustments. This adaptability extends to areas such as medical diagnosis prediction, where decoder models, through the use of carefully crafted prompts, can assimilate and apply knowledge from vast text corpora to make informed predictions or suggestions. These examples underscore the diminishing necessity for domain-specific encoder models, as decoder models, powered by effective prompting, exhibit the capacity to generalize across tasks, showcasing their broad applicability [4].

F. Prompting and Few-Shot Learning

Few-shot learning represents a significant stride in the evolution of machine learning, particularly within the context of decoder models. This approach, wherein the model is provided with a handful of examples to guide its understanding of a task, significantly enhances task performance without the need for extensive data preprocessing and feature extraction, which are typically associated with encoder models. Decoder models, such as GPT and LLaMA, have exemplified the efficacy of few-shot learning, adeptly handling tasks ranging from translation to complex problem-solving with only a few examples provided in the prompt. This capability not only showcases the models' efficiency in learning from limited data but also their potential to revolutionize the way we approach machine learning tasks, making them more accessible and less resource-intensive [12].

G. Challenges and Limitations

Despite the significant advantages presented by decoder models and prompting strategies, there are inherent limitations and challenges associated with them. One of the primary concerns is prompt ambiguity, where poorly designed prompts can lead to unpredictable or inaccurate model outputs. Additionally, the effectiveness of prompt-based approaches heavily relies on the expertise in prompt engineering, a skill that may not be readily available across all user segments. Furthermore, there are specific scenarios where encoder models may still hold an advantage, such as tasks that require intensive feature extraction from structured data. These challenges highlight the necessity for ongoing research and development to refine prompting techniques, improve model interpretability, and explore hybrid models that combine the strengths of both encoders and decoders to overcome these limitations [1].

VI. CONCLUSION

This paper has explored the significant transitions within the field of natural language processing (NLP), focusing on the shift from encoder models to the more versatile and efficient decoder and decoder-only models. We've detailed the evolution from the peak of encoder models, characterized by their widespread adoption due to their robustness in understanding complex patterns in data, to the emergence of decoder models which have showcased superior generative capabilities and flexibility across various domains. This transition is underscored by the advancements in self-attention mechanisms, scalability, and the integration of unsupervised learning techniques that decoder models offer. Moreover, the introduction and refinement of prompting techniques have further highlighted the adaptability and efficiency of decoder models, enabling them to perform a wide array of tasks with minimal task-specific tuning. Reflecting on the transition away from encoder models, it's evident that the NLP field is moving towards architectures that not only reduce complexity and computational demands but also enhance model versatility and performance on generative tasks. The ongoing evolution of encoder and decoder models suggests a future where the boundaries of NLP are continuously expanded, leveraging the strengths of both architectures to address the inherent limitations of each. Future research directions may focus on hybrid models that combine the precise understanding capabilities of encoder models with the generative flexibility of decoder models, thereby optimising for both efficiency and effectiveness across various tasks. Additionally, the development of more sophisticated prompting techniques and few-shot learning capabilities promises to democratise access to advanced NLP technologies further, enabling more intuitive and natural interactions between humans and machines.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

DECLARATION STATEMENT

Funding	No, I did not receive.
Conflicts of Interest	No conflicts of interest to the best of our knowledge.
Ethical Approval and Consent to Participate	No, the article does not require ethical approval or consent to participate, as it presents evidence that is not subject to interpretation.
Availability of Data and Material/ Data Access Statement	Not relevant.
Authors Contributions	All authors have equal participation in this article.

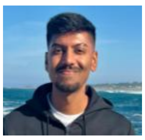
REFERENCES

1. Vaswani, A., et al. (2017). "Attention Is All You Need." Advances in Neural Information Processing Systems.
2. Radford, A., et al. (2018). "Improving Language Understanding by Generative Pre-Training." OpenAI Blog.
3. Devlin, J., et al. (2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." arXiv preprint arXiv:1810.04805.
4. Brown, T., et al. (2020). "Language Models are Few-Shot Learners." arXiv preprint arXiv:2005.14165.
5. Raffel, C., et al. (2019). "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer." Journal of Machine Learning Research.
6. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation https://doi.org/10.1162/neco.1997.9.8.1735
7. Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. arXiv:1409.3215
8. Radford, A., et al. (2018). Improving Language Understanding by Generative Pre-Training.
9. Kitaev, N., Kaiser, Ł., & Levskaya, A. (2020). Reformer: The Efficient Transformer. arXiv:2001.04451
10. Touvron, Hugo, et al. "LLaMA: Open and Efficient Foundation Language Models." ArXiv, 2023, /abs/2302.13971.
11. Radford, A., et al. (2019). Language Models are Unsupervised Multitask Learners
12. Liu, Pengfei, et al. "Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing." *ArXiv*, 2021, /abs/2107.13586.
13. Lewis, P., et al. (2020). "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.
14. Hu, H., et al. (2021). "LoRA: Low-Rank Adaptation of Large Language Models.

AUTHORS PROFILE

Adithyan Arun Kumar is a distinguished Product Security Engineer at Salesforce, California, with a rich background in Information Security from Carnegie Mellon University. Specialising in risk evaluation, threat modelling, penetration testing, and security automation, Adithyan has a notable history of automating threat models during his tenure at Salesforce and the TEEL Lab at Carnegie Mellon. His contributions to the field are recognised by major corporations such as Apple and Microsoft, where he is inducted into their Hall of Fame.

Gowthamaraj Rajendran is a Security Researcher at Splunk, California, with a rich background in Threat detection and Red Teaming. He also holds certificates such as OSWE, OSWA, OSCP, and CRTP. He has authored around 10+ CVEs to date.



Nitin Srinivasan is a Software Engineer III at Google in California, where he works on the ML Developer Infrastructure and Security team. He has made significant contributions to prominent ML Frameworks such as TensorFlow, enhancing build efficiency and expanding its accessibility and usability across various platforms. He holds a Master's degree in Computer Science from the University of Massachusetts Amherst, where his focus was on deep learning and natural language processing.

Praveen Kumar Sridhar is a highly skilled Machine Learning Engineer and Research Assistant with an extensive background in data science, natural language processing (NLP), and machine learning. Holding a Master's degree in Data Science from Northeastern University, he has contributed significantly to the fields of machine learning and NLP. His research and development efforts include the creation of an advanced moderation system for social media platforms, the development of NLP pipelines utilizing architectures such as Llama and GPT, and the creation of churn frameworks. These contributions demonstrate his capacity to innovate and push the boundaries of AI and machine learning technologies.

Kishore Kumar Perumalsamy works as a Software Engineer at Adobe Headquarters in San Jose, California, with a fervour for building secure, intelligent, and scalable software systems. He earned his Master's in Mobile and IoT Engineering from Carnegie Mellon University, serving as a Teaching Assistant for Software Architecture and a Research Assistant in the renowned TEEL Labs of the CS department.