

## The Ecosystem of Machine Learning Methods

Rahul C. Basole, PhD

Accenture AI

[rahul.basole@accenture.com](mailto:rahul.basole@accenture.com)

### Abstract

*Machine learning (ML) is a rapidly evolving field and plays an important role in today's data-driven business environment. Many digital innovations in domains as diverse as healthcare, banking, energy, and retail are powered and enabled by ML. Examples include search engines, recommendation systems, pattern recognition, computer vision, and natural language processing. A key element in ML innovation is the advancement of the underlying methods, which specify how machines should algorithmically process, derive patterns, and learn from data for a given decisioning task. The speed at which this is happening is exponential, with researchers leveraging and building upon existing building blocks as well as introducing entirely new methods. Given the speed, scale, and complexity, understanding this complex evolving ML method space can be challenging. What methods are core and peripheral to ML? Which methods span task areas? How are ML methods evolving? In this exploratory research paper, I address these questions by (1) framing the ML method space and (2) visualizing the evolving structure of the ML methods ecosystem. The results reveal several foundational ML building blocks, different coupling levels between ML areas, and variable speeds of evolution. The study also provides insights into how digital innovation evolves at an algorithmic level. I discuss the implications of the findings and describe opportunities for future ML ecosystem-focused research.*

### 1. Introduction

Whether you like it or not, we live in a highly algorithmic world [1, 2, 3]. Search engines find relevant information to our daily queries [4]. Virtual assistant technologies recognize our voice commands to serve up the latest content. Social media algorithms decide what content we see [5]. Facial recognition algorithms enable us to open our smartphones. Navigation systems

suggest optimal routes. Retail providers customize our shopping experience [6]. Medical diagnostic tools make personalized treatment suggestions [7]. Home automation tools dynamically adjust temperature and lighting in our living environments. And digital media companies make movie recommendations based on our viewing history and preferences [8]. The pervasiveness of algorithms in the business domain is equally staggering. Today's high-performing companies leverage algorithms to optimize their supply chains, predict system and infrastructure failures, identify fraudulent behavior, and innovate new products [9]. If you can think of a domain, it is quite likely that there is some algorithm behind it.

All of these seemingly pervasive digital experiences are powered and enabled by simple to highly sophisticated algorithms that have been made possible by massive advances in a combination of digital technologies, including big data, artificial intelligence, networks, and high-performance computing [10, 11]. At the core of these intelligent, algorithmic-centric digital offerings are machine learning (ML) methods that can process data, detect complex patterns, and continuously learn at staggering scale and speed [12]. The pace of innovation in ML appears to be exponential. Today there are nearly 50,000 ML-specific public GitHub repositories with thousands of commits daily<sup>1</sup>. Similarly, we can identify tens of thousands of ML-related research papers on the open-access archive arxiv.org. Driven by an open-source oriented developer ecosystem, ML researchers leverage and build upon existing building blocks as well as introduce entirely new methods on a continuous basis.

While there is a growing interest in ML in both research and practice [12, 13], it is the aforementioned scale, speed, and complexity that makes an understanding of the evolving ML method space challenging. What methods are core and peripheral to the ML field? How is this field evolving? Which methods span task areas? In this exploratory research

<sup>1</sup><https://github.com/topics/machine-learning>

paper, I address these questions by (1) framing the ML method space and (2) visualizing the evolving structure of the ML methods ecosystem following a data-driven approach. The results reveal complex networked structure, with several foundational ML building blocks, different coupling levels between ML areas, and variable speeds of evolution. The study provides particular insights into how innovation organizes and evolves at an algorithm-level. I discuss the implications of the findings and describe opportunities for future ML-focused digital innovation research.

## 2. Framing the ML Method Space

While ML has become a particular prominent topic in the last few years, the concept of machine learning has in fact been around for decades [14, 15]. Rooted in the belief that machines could be designed in ways that they could automatically improve through experience, artificial intelligence (AI) pioneers in the late 1950s developed methods that taught computers how to play games and get better (through a technique called alpha-beta pruning<sup>2</sup> that evolved into the minimax algorithm) [16], to recognize images (through a technique called the perceptron algorithm<sup>3</sup>, an early version of an artificial neural network) [17], or to recognize patterns through the “nearest neighbor rule”, a key predecessor to modern GPS mapping applications and recommendation systems [18]. Indeed, many of today’s ML applications, such as data mining, natural language processing, or facial recognition are based on this early pioneering work.

So what is ML? There appears to be no single definition. However, according to leading organizations and academic institutions, ML “is the science of getting computers to act without being explicitly programmed” (Stanford)<sup>4</sup>, “aims to produce machines that can learn from their experiences and make predictions based on those experiences and other data they have analyzed” (Georgia Tech)<sup>5</sup>, “uses statistics to find patterns in massive amounts of data” (MIT)<sup>6</sup>, and “is concerned with programs or systems that build predictive models from input data and uses the learned model to make useful” (Google)<sup>7</sup>. Taken together, several common aspects can be identified. First, ML is a method for data analysis that automates analytical model building. Second, these models can learn from data, identify and predict patterns and make decisions with minimal

human intervention.

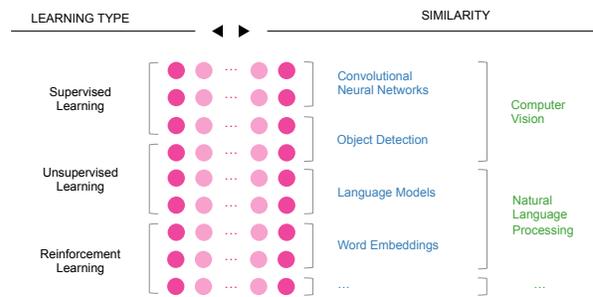


Figure 1: Framing the ML Method Space — Area (●); Categories (●); Methods (●)

With this perspective in mind, the question then becomes how to make sense and organize the seemingly exploding ML method space. Pending the disciplinary background of the reader, there are likely several ways to do that. Following a review of leading introductory textbooks and survey research [19, 20, 21, 22], I posit that there are two common lenses of framing ML methods (see Figure 1). One lens occurs around the **type of learning** that a ML method uses. Broadly considered, there are three key learning models<sup>8</sup>: supervised, unsupervised, and reinforcement. *Supervised learning* refers to the process of training a model using labeled input and output data (i.e., training data) to make accurate predictions about outputs based on new data [23]. Supervised learning is one of the most common learning types used in ML methods and found pervasively in many real-world applications. Recommendation systems by online marketplaces (e.g., Amazon) and content providers (e.g., Netflix, Spotify) are largely powered by supervised learning. The more the digital service is used, the better the system learns and predicts what a user may like. *Unsupervised learning*, on the other hand, does not use any labeled data to find patterns in a dataset. It typically uses grouping or clustering to achieve a desired goal. The more data is fed into the model, the more refined the output gets. Single-layer ML systems (those that only use a single step to process data) are not very efficient in dealing with unlabeled data; multi-layer systems on the other hand, and those typically used in deep learning, are more suited for this type of data. Real-world examples of unsupervised learning include audience segmentation, anomaly detection, and chatbots. *Reinforcement learning* also uses unlabeled data to reach a predefined goal. But in contrast to

<sup>8</sup>Some may argue that there are additional learning models, including semi-supervised, instance-based, ensemble, etc. I focused on three of the most common learning types.

<sup>2</sup>[https://en.wikipedia.org/wiki/Alpha-beta\\_pruning](https://en.wikipedia.org/wiki/Alpha-beta_pruning)

<sup>3</sup><https://en.wikipedia.org/wiki/Perceptron>

<sup>4</sup><http://mlclass.stanford.edu/>

<sup>5</sup><https://ml.gatech.edu/>

<sup>6</sup><https://bit.ly/2OncBNm>

<sup>7</sup><https://bit.ly/2CFqj1>

unsupervised learning, reinforcement learning uses a reward/penalty-based scoring system to direct the model [22]. Reinforcement learning is fairly new but has been shown to be quite powerful in teaching AI system play games (e.g., AlphaGo Zero<sup>9</sup>), support scientific experiments, and robotics.

The other common lens of framing ML methods is to group them in terms of their **similarity** (i.e., in terms of how they work). In contrast to the learning type delineation, this approach provides a more functional lens to organize the ML method landscape. While not perfect, as some methods could fit into multiple categories, I believe it offers a more intuitive approach. With a high-level differentiation, methods are organized into key ML areas (or subfields) they occur in (e.g., computer vision, natural language processing, audio, graphical models, etc.). Within each area, methods are then grouped by similar categories (e.g. regression algorithms, instance-based algorithms, decision trees, Bayesian algorithms, artificial neural network algorithms, etc.). A potential intermediate level may further differentiate them by the types of tasks it helps to address (e.g., object detection, question answering, link prediction, etc.). To facilitate exploration and sense-making, I opted to use a combined area-category approach to organize the ML method space.

### 3. Methodology

#### 3.1. Data

To the best of my knowledge, there is no single data source that tracks and classifies existing ML methods and their application areas. Arguably one of the most comprehensive sources for identifying ML methods is GitHub, an open-access code repository. While the purpose of the ML method may be noted in the description section or read.me files of the corresponding GitHub entry, the type of method or the underlying algorithm may not be fully described. A more comprehensive description of the ML method and its application areas is likely found in corresponding research papers. Many research teams post in-progress and completed ML research papers on arxiv.org. A promising new data repository that links these two data sources is Papers With Code<sup>10</sup>, which was recently acquired by Facebook AI. The mission of Papers With Code is “to create a free and open resource with ML papers, code and evaluation tables.” The repository currently has nearly 26,000 papers with code, 760

<sup>9</sup><https://deepmind.com/blog/article/alphago-zero-starting-scratch>

<sup>10</sup><https://paperswithcode.com/>

methods, and 1,626 tasks (July 13, 2020). Using an algorithmic- based and community-validated approach, method(s) and task(s) for each paper are identified and organized. I extracted all relevant data fields necessary to build the ML method ecosystem (i.e., nodes, links, and their attributes) using custom R scripts (code available upon request). All data is further cleaned and organized in relational tables.

#### 3.2. Network Construction

Given the nature of the data, I constructed three types of multi-partite networks. Network *ACM* is a tripartite method-hierarchy network, consisting of Area (*A*) nodes (e.g., Computer Vision), Category (*C*) nodes (e.g., Convolutional Neural Networks), and Method (*M*) nodes (e.g., ResNet). In this network, there are two types of directed edges: methods belonging to categories and categories belonging to areas. Since some methods can belong to one or more categories, I do not expect a clean tree network structure. Moreover, I posit that these category spanning method nodes will play a particularly important role.

Network *MM* is a unipartite method-component network, consisting of Method (*M*) nodes. In this network, a directed edge indicates that a particular method is a sub-component of another method, thus representing a form of functional dependency relationship.

#### 3.3. Metrics

For each network I compute a number of important graph theoretic metrics, including average degree (in, out, total), average weighted degree, and betweenness centrality [24]. To identify prominent subclusters in each network, I used Louvain’s modularity algorithm [25]. Since networks are time-varying, I compute each of the metrics over time on a yearly basis.

#### 3.4. Visualization

Following design considerations suggested by prior work [26, 27, 28], I considered several different visualization techniques to depict the ML methods ecosystem. Given the hierarchical nature of the *ACM* network, space-filling visualization techniques, such as treemaps, sunbursts, or icicles were potential options [29]. However, given that some methods may belong to multiple categories, these techniques would fail. Given that relationships and connectivity between ML ecosystem entities are of primary interest in this study, I opted for a traditional node-link representation.

I visualize the resulting networks using a

Area (●)	Categories (●)			Methods (●)			
	#	Examples	#	Examples	First	Latest	Avg. Age
General	50	Regularization, Activation Function, Normalization	318	relu, softmax, dropout	1943	2020	6.04
Computer Vision	39	Convolutional Neural Networks, Object Detection Models, Generative Models	430	1x1 convolution, resnet, faster-r-cnn, cyclegan	1980	2020	3.52
Natural Language Processing	15	Language Models, Transformers, Word Embeddings	93	gpt-3, bert, ELMo, GloVe	2003	2020	1.82
Reinforcement Learning	14	Policy Gradient Methods, Q-Learning Networks	57	ppo, dqn, ddpq	1984	2019	9.2
Audio	6	Generative Audio, Text-to-Speech Models, Speech Synthesis Blocks	15	wavenet, Tacotron 2, CBHG	1984	2019	4.20
Sequential	6	Recurrent Neural Networks, Sequence-to-Sequence Models, Temporal Convolutions	36	lstm, seq2seq	1997	2020	4.82
Graphs	2	Graph Models, Graph Embeddings	6	gcn, gat, node2vec	2013	2017	4.5

Table 1: ML Method Ecosystem Data Summary.

combination of several custom force-directed algorithms that optimize the overall inter-node repulsion and clustering in the network, including the proportional Yifan Hu layout [30] and the OpenORD layout [31]. I also apply a no-overlap heuristic to avoid node occlusion and improve readability [32]. I color-encode and size nodes by different existing and computed metrics. All visualizations are implemented in Gephi 0.9.2, an open-source software for visualizing and analyzing large network graphs [33].

## 4. Results

Table 1 provides a summary of the seven ML areas, categories, and methods, and identifies some illustrative examples. The *Computer Vision* area (430) has the most methods followed by the *General* (318) and *Natural Language Processing* (93) areas. The first methods in each area appeared quite early, except for *Graphs*, which is a more recent area (2017). Almost all ML areas had new methods introduced in the last two years (2019 and 2020). The average age of methods ranges from 1.82 years (in *Natural Language Processing*) to 9.2 years (in *Reinforcement Learning*).

Figure 2 shows the structure of the ML methods ecosystem. The ecosystem contains 895 nodes (7 areas, 127 categories, and 760 methods) and 1,285 hierarchical links. Nodes are colored by the three node types and labeled<sup>11</sup>; edges are colored by source node to emphasize the parent connection. The visualization reveals that the ML methods ecosystem consists primarily of a single, large giant component (subnetwork) and several much smaller disconnected clusters (e.g., *reinforcement learning* and *graph models*). This highlights two things: first, since the

<sup>11</sup>To reduce clutter, but provide full annotation, I chose small labels. When opened with a pdf-viewer and zoomed in, the labels can be read more clearly.

ML ecosystem is highly integrated, a few methods are category spanning and connect the *ACM* network. Second, methods in *reinforcement learning* and *graph models* are more autonomous from the remainder of the ML method ecosystem.

Figures 3a-c shows the evolution of the ML method ecosystem over time using a small multiple visualization approach. I split the overall timeframe - the occurrence of the first ML method until present - into four cumulative periods (pre-2005, 2006-2010, 2011-2015, 2016-2020) and annotate some of the key structural characteristics and changes. Several observations can be made. The first observation is that the pre-2005 period had already established a substantive number of ML methods, while likely not referred to as such, across almost all major ML areas (see Figure 3a). The primary focus of these methods were in the areas of *General*, *Computer Vision*, and *Reinforcement Learning*. The second observation is that there was very little structural change into the 2006-2010 timeframe (see Figure 3b). Modest additions were in *Generative Models*, *Heuristic Search Algorithms*, and *Normalization*. One explanation for the lack of development is that enabling technologies advances had not yet occurred and ML was not yet widely diffused. A significant increase in ML methods occurred between 2011-2016, with the emergence of the *Graph* area and growth focused in categories such as *Image Models*, *Convolutional Neural Networks*, *Object Detection Models*, *Recurrent Neural Networks*, and *Language Models* (see Figure 3c). One important finding is the emergence of methods in the *Image Model Blocks* category which connect the *Computer Vision* and *General* ML method areas. Finally, the most massive shift occurs during the most recent time period (2016-2020) (see Figure 3d). Significant structural growth occurs across almost all areas and categories, but in particular in *Convolutional Neural*

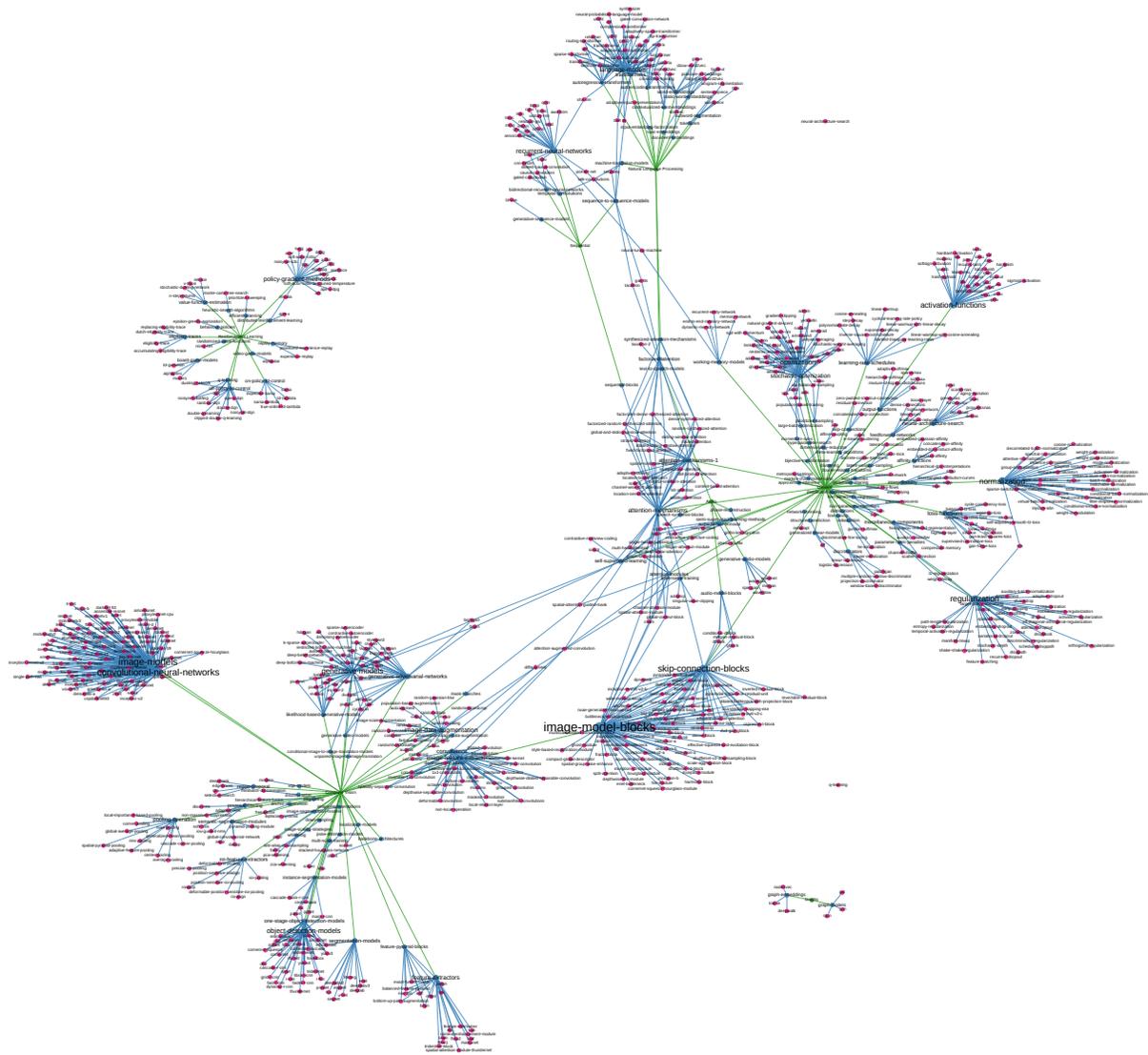
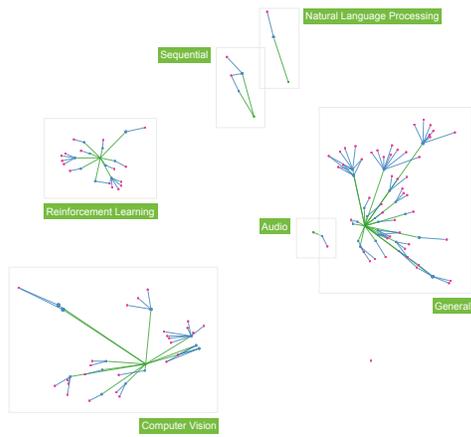


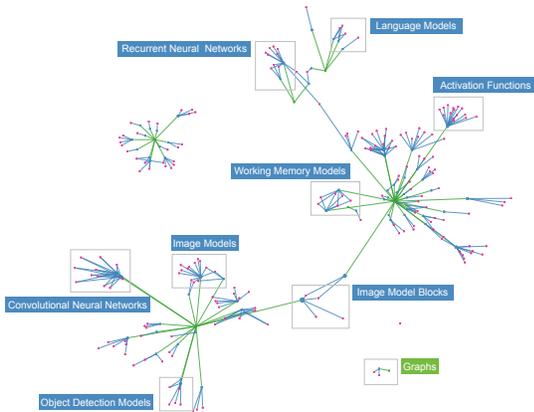
Figure 2: The ML Methods Ecosystem — Area (●); Categories (●); Methods (●).



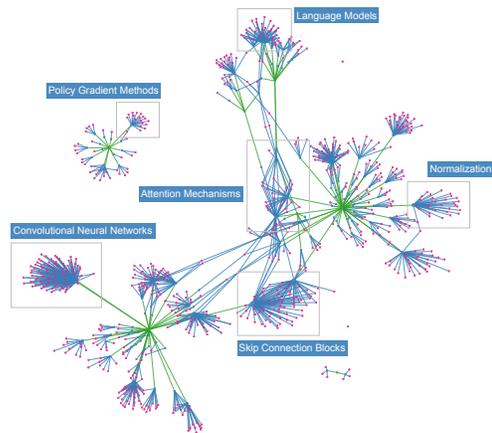
(a) pre-2005



(b) 2006-2010



(c) 2011-2015



(d) 2016-2020

Figure 3: Evolution of the ML Methods Ecosystem — Area (●); Categories (●); Methods (●).

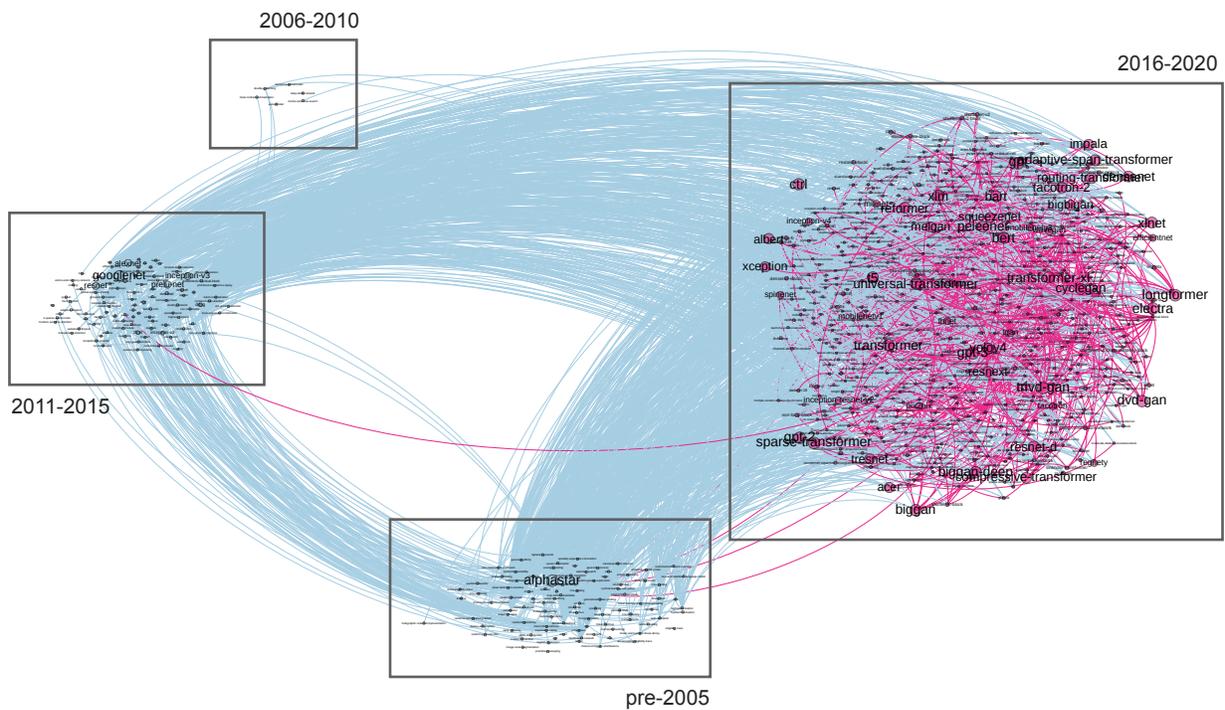


Figure 4: Method Interdependency Network — pre-2005-2015 Methods (●); 2016-2020 Methods (●).

*Networks, Language Models, Image Model Blocks, Skip Connection Blocks, Object Detection Models, Feature Extractors, and Normalization.* This growth is clearly a result of rapid advances in enabling technologies, the increased use of ML in a wide range of digital products and services, and the reusability of existing ML methods. This last point is more pronounced in the analysis that follows.

Figure 4 shows the method interdependency network. Since not all components have dependencies, I focus only on the main component of this network. The size of this network is 610 method nodes (80.2% of total method nodes). A few notes on how to interpret this custom visualization. First, since this is a unipartite network, all nodes are methods. I use a two-color encoding scheme to denote pre-2016 methods in blue (●) and more recently launched methods in fuchsia (●). I size nodes by weighted out-degree, indicating their level of influence on the network. I color-encode edges by the source. Second, the layout was generated using a period-clustered force-directed layout. In other words, nodes belonging to the same period are clustered together. I positioned the most recent period to the right (as would be typical in a timeline) and the three other periods to the left of it in a manner that reduces

edge crossings. I should also note that I utilized curved edges to visually highlight the directionality between methods. Specifically, I used a clockwise-directed edge to denote the dependent components of a method.

The visualization reveals several important findings. Following the directionality of the large band of edges, there is clearly a temporal sequence to ML methods development over the four periods. Many of the methods developed pre-2005 have been foundational to methods developed in 2011-2015 as well as 2016-2020. Similarly, we see that 2011-2015 methods are important components of more recent methods. On the other hand, methods developed in 2006-2010 only weakly influence their subsequent periods. Second, we also see some recursive influences of methods. For instance, methods developed in 2011-2015 and 2016-2020 influence methods developed in prior periods. The influence is likely on updated iterations of those methods. Third, when examining the interconnectedness within each time period cluster, it is quite evident to see that the highest density is in the most recent time period, suggesting a much greater co-influence of contemporaneous methods. When considering the size of the nodes, which indicates its importance in method development, we observe that some of the

most influential nodes are found in the most recent time period. Only few older methods are prominent. Examples include *alphastar*, *googlenet*, *inception-v3*, and *resnet*. Together these observations suggest that while methods build on top of each other, most recent methods are likely driving new method innovation.

## 5. Discussion

The exploratory analysis presented in this study provides several important insights into the organizing structure and evolution of the ML methods ecosystem, specifically, and digital (algorithm) innovation more broadly.

First, the visual analysis highlights that the ML methods ecosystem is not a pure hierarchical, but rather a complex networked system (as shown in Figure 2). While most methods fit into singular categories, there are quite a significant number of methods that span multiple categories as well as areas. Broadly considered, this finding suggests that the majority of ML methods are typically engineered to address area- and category-specific tasks. Methods that are boundary spanning highlight that there are tasks that are cross-disciplinary, for instance methods that sit at the intersection of *Audio* and *Natural Language Processing*. As ML application areas become more intertwined, and boundaries may blur, it can be reasonably argued that more cross-area methods may emerge.

Second, the evolutionary analysis of the ML methods ecosystem highlights that algorithm development has followed differing speeds both at a macro- and micro-network level, with periods of slow downs as well as significant acceleration. What explains this evolution? Existing theories in traditional

product innovation suggest that products and product categories evolve incrementally until a dominant design appears or the area is matured [34, 35, 36]. Similarly, industry life cycle theory suggests that industries go through multiple stages including development, growth, shakeout, maturity, and decline [34, 36, 37]. Technological hype cycles propose that technologies progress through successive stages that are pronounced by a peak, followed by disappointment, and later a recovery of expectations [38, 39]. Using these lenses, the ML method ecosystem appears to have gone through the first few phases, influenced by supply (e.g., technology maturity, competition, enabling infrastructures), demand (e.g., market need), and contextual factors (economic conditions, social, geography, policy, etc.).

Third, our temporal interdependency analysis reveals that there is a strong sequential development in ML methods (see Figure 4). Prior methods act as salient foundational components to more recent methods suggesting both incremental and building block development. The level of interdependency is most pronounced in the most recent period. Furthermore, some methods are structurally more influential than others. These findings strongly underline some of the core ideas of digital artifacts and innovation, in particular modularity, generativity, and platformification [40, 41, 42, 43].

While this study represents an important first step, many exciting future research questions for understanding the ML ecosystem remain. Broadly considered, potential extensions include investigations related to method usage, quality, performance, updates, developers, and development frameworks (see Table 2).

Lens of Inquiry	Potential Research Questions
Usage	How often has the method been used/cited? How popular is the method? For what types of tasks has the method been used? How often has the method been combined with others (ensembles)? What dataset has been used? How has the method been implemented? What licensing models are used?
Quality Performance	How has the method been rated? How often has the method been forked? How well documented is the method? How well does the method perform? How accurate are predictions by the method (classification accuracy, confusion matrix, log-loss, AUC, etc.)? How has performance improved overtime?
Updates	What practices are used to coordinate, test, release, and maintain ML code? How often are methods updated? Who updates the method? What is the reason for the update? What are the incremental changes?
Developer(s)	Who are lead developers of ML Methods? Is the lead developer a private organization or public company? How are the development communities organized? What is the geographic footprint of the developer base? What role do leading companies and organizations play in the design, development, and distribution of ML methods? How does ML method development differ to traditional open source development? What are the motivations/incentives for ML developers?
Development Framework(s)	What development frameworks are used to implement the method (e.g., PyTorch, TensorFlow, Keras)? How well-designed are the APIs of these frameworks? How pervasive are implementations between research and productions?
Competitive Dynamics	What is the relationship between open source ML development and competition? What strategies do organizations use?

Table 2: Open Research Opportunities

## 6. Concluding Remarks

Machine learning is unquestionably a rapidly evolving field, with new methods, contributors, and challenges emerging on a continuous basis. Yet, given the speed, scale, and complexity, an understanding of the ML method space can be challenging. In this exploratory study, I used a visual, data-driven approach to map the ML method landscape to determine both the structure and dynamics of this important AI domain. The investigation revealed numerous important insights into how the ML field is evolving, how new methods emerge, and what areas are growing or slowing down. The study is merely a first step towards a systematic understanding of ML innovation. Moreover, it is not without gaps and limitations. I highlight some of them in the discussion section and posit that each of them is an exciting future research direction. Our understanding of the ML space is still in its infancy and more work is needed to uncover the intricacies of this dynamic ecosystem. I hope that this study will provide a stimulating first step for the research community.

## References

- [1] A. Agrawal, J. Gans, and A. Goldfarb, *Prediction machines: the simple economics of artificial intelligence*. Harvard Business Press, 2018.
- [2] P. Domingos, *The master algorithm: How the quest for the ultimate learning machine will remake our world*. Basic Books, 2015.
- [3] C. Besteman and H. Gusterson, *Life by Algorithms: How Roboprocesses are Remaking Our World*. University of Chicago Press, 2019.
- [4] J. Battelle, *The search: How Google and its rivals rewrote the rules of business and transformed our culture*. Hachette UK, 2011.
- [5] N. Diakopoulos, *Automating the news: How algorithms are rewriting the media*. Harvard University Press, 2019.
- [6] J. Rossman, *Think Like Amazon: 50 1/2 Ideas to Become a Digital Leader*. McGraw Hill Professional, 2019.
- [7] E. Topol, *Deep medicine: how artificial intelligence can make healthcare human again*. Hachette UK, 2019.
- [8] C. A. Gomez-Uribe and N. Hunt, “The netflix recommender system: Algorithms, business value, and innovation,” *ACM Transactions on Management Information Systems (TMIS)*, vol. 6, no. 4, pp. 1–19, 2015.
- [9] M. Iansiti and K. Lakhani, *Competing in the Age of Artificial Intelligence*. Harvard Business Press, 2020.
- [10] A. McAfee and E. Brynjolfsson, *Machine, platform, crowd: Harnessing our digital future*. WW Norton & Company, 2017.
- [11] P. R. Daugherty and H. J. Wilson, *Human+ machine: reimagining work in the age of AI*. Harvard Business Press, 2018.
- [12] K. Awalegaonkar, R. Berkey, G. Douglass, and A. Reilly, “AI: Built to scale. From experimental to exponential,” 2019.
- [13] P. Choudhury, R. Allen, and M. G. Endres, “Machine learning for pattern discovery in management research,” *Strategic Management Journal*, vol. n/a, no. n/a.
- [14] A. Plasek, “On the cruelty of really writing a history of machine learning,” *IEEE Annals of the History of Computing*, vol. 38, no. 4, pp. 6–8, 2016.
- [15] M. Kubat, I. Bratko, and R. S. Michalski, “A review of machine learning methods,” *Machine Learning and Data Mining: Methods and Applications*, pp. 3–69, 1998.
- [16] G. C. Stockman, “A minimax algorithm better than alpha-beta?,” *Artificial Intelligence*, vol. 12, no. 2, pp. 179–196, 1979.
- [17] B. Widrow and M. A. Lehr, “30 years of adaptive neural networks: perceptron, madaline, and backpropagation,” *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1415–1442, 1990.
- [18] M. Pelillo, “Alhazen and the nearest neighbor rule,” *Pattern Recognition Letters*, vol. 38, pp. 34–37, 2014.
- [19] A. Ng, *Machine Learning Yearning*. Stanford Press, 2017.
- [20] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press.
- [21] L. P. Kaelbling, M. L. Littman, and A. W. Moore, “Reinforcement learning: A survey,” *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.
- [22] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
- [23] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, “Supervised machine learning: A review of classification techniques,” *Emerging Artificial Intelligence Applications in Computer Engineering*, vol. 160, no. 1, pp. 3–24, 2007.
- [24] S. Wasserman, K. Faust, et al., *Social network analysis: Methods and applications*, vol. 8. Cambridge University Press, 1994.
- [25] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and É. Lefebvre, “The louvain method for community detection in large networks,” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 10, p. P10008, 2011.
- [26] R. C. Basole, “Visualization of interfirm relations in a converging mobile ecosystem,” *Journal of Information Technology*, vol. 24, no. 2, pp. 144–159, 2009.
- [27] R. C. Basole, M. G. Russell, J. Huhtamäki, N. Rubens, K. Still, and H. Park, “Understanding business ecosystem dynamics: A data-driven approach,” *ACM Transactions on Management Information Systems (TMIS)*, vol. 6, no. 2, pp. 1–32, 2015.
- [28] B. R. Iyer and R. C. Basole, “Visualization to understand ecosystems,” *Communications of the ACM*, vol. 59, no. 11, pp. 27–30, 2016.
- [29] H.-J. Schulz, S. Hadlak, and H. Schumann, “The design space of implicit hierarchy visualization: A survey,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 4, pp. 393–411, 2010.
- [30] Y. Hu, “Efficient, high-quality force-directed graph drawing,” *Mathematica Journal*, vol. 10, no. 1, pp. 37–71, 2005.

- [31] S. Martin, W. M. Brown, R. Klavans, and K. W. Boyack, "Openord: an open-source toolbox for large graph layout," in *Visualization and Data Analysis 2011*, vol. 7868, p. 786806, International Society for Optics and Photonics, 2011.
- [32] C. Bennett, J. Ryall, L. Spalteholz, and A. Gooch, "The aesthetics of graph visualization.," *Computational Aesthetics*, vol. 2007, pp. 57–64, 2007.
- [33] M. Bastian, S. Heymann, M. Jacomy, *et al.*, "Gephi: An open source software for exploring and manipulating networks.," in *International Conference on Web and Social Media*, vol. 8, pp. 361–362, 2009.
- [34] J. M. Utterback and W. J. Abernathy, "A dynamic model of process and product innovation," *Omega*, vol. 3, no. 6, pp. 639–656, 1975.
- [35] W. J. Abernathy and J. M. Utterback, "Patterns of industrial innovation," *Technology Review*, vol. 80, no. 7, pp. 40–47, 1978.
- [36] S. Klepper, "Industry life cycles," *Industrial and Corporate Change*, vol. 6, no. 1, pp. 145–182, 1997.
- [37] E. M. Rogers, *Diffusion of innovations*. Simon and Schuster, 2010.
- [38] H. Van Lente, C. Spitters, and A. Peine, "Comparing technological hype cycles: Towards a theory," *Technological Forecasting and Social Change*, vol. 80, no. 8, pp. 1615–1628, 2013.
- [39] O. Dedehayir and M. Steinert, "The hype cycle model: A review and future directions," *Technological Forecasting and Social Change*, vol. 108, pp. 28–41, 2016.
- [40] Y. Yoo, O. Henfridsson, and K. Lyytinen, "Research commentary—the new organizing logic of digital innovation: an agenda for information systems research," *Information Systems Research*, vol. 21, no. 4, pp. 724–735, 2010.
- [41] Y. Yoo, R. J. Boland Jr, K. Lyytinen, and A. Majchrzak, "Organizing for innovation in the digitized world," *Organization Science*, vol. 23, no. 5, pp. 1398–1408, 2012.
- [42] J. Kallinikos, A. Aaltonen, and A. Marton, "The ambivalent ontology of digital artifacts," *MIS Quarterly*, pp. 357–370, 2013.
- [43] M. de Reuver, C. Sørensen, and R. C. Basole, "The digital platform: a research agenda," *Journal of Information Technology*, vol. 33, no. 2, pp. 124–135, 2018.