Proceedings of CONF-MLA 2026 Symposium: Intelligent Systems and Automation: AI Models, IoT, and Robotic Algorithms
DOI: 10.54254/2755-2721/2026.LD39061

A Review of Five Major Machine Learning Paradigms and Their Representative Algorithms

Zhuo Liu

University of British Columbia, Vancouver, Canada
zliu7163@gmail.com

Abstract. Machine learning methods use different forms of data and feedback, which can make their relationships difficult for new learners to understand. This paper draws on related research to compare supervised, unsupervised, semi-supervised, self-supervised, and reinforcement learning and their representative algorithms. Foundational studies are considered alongside selected research from 2022–2024. Self-supervised learning is treated separately for clarity, while its relationship to unsupervised learning is acknowledged. The comparison considers learning signals, objectives, applications, strengths, and limitations. The main finding is that the five paradigms differ in how training information guides learning, rather than only in whether labels exist. Supervised learning supports prediction with known targets, while unsupervised learning supports grouping and exploration without such targets. Semi-supervised learning uses limited labels together with unlabeled examples, but its results depend on the training examples and the reliability of generated labels. Self-supervised learning creates targets from data and supports later tasks, although reduced labeling does not necessarily mean reduced computation. Reinforcement learning addresses decisions whose effects continue over time but learning stability and interaction costs require attention. The reviewed methods also show that a system can combine several paradigms. These findings suggest that method selection should begin with the task, available feedback, and evaluation conditions. No common ranking across all five paradigms follows from the reviewed evidence.

Keywords: Supervised learning, Unsupervised learning, Semi-supervised learning, Self-supervised learning, Reinforcement learning

1. Introduction

Machine learning allows a computer system to build prediction rules from examples. For instance, support vector machines learn a boundary between classes, while random forests and XGBoost combine decision trees to make predictions [1-3]. Other methods serve different purposes. Clustering groups observations, and dimensionality reduction represents data with fewer variables to help researchers inspect patterns [4-6]. These differences matter because choosing an algorithm also involves choosing what the system should learn. A method that predicts a known category and a method that discovers groups do not necessarily solve the same problem.

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.LD39061

The selected algorithmic papers explain specific methods and evaluate them on particular tasks. Reading these studies separately can make it difficult to compare their goals and evaluation conditions. This review brings them together to provide a common basis for choosing between learning paradigms, without claiming that broader surveys are absent. It asks how five major machine learning paradigms differ in their learning signals, objectives, representative algorithms, strengths, limitations, and suitable applications.

This paper draws on related literature to compare supervised, unsupervised, semi-supervised, self-supervised, and reinforcement learning. It connects training information with objectives and practical trade-offs. Its purpose is to offer a clear starting point for readers with limited machine learning knowledge. Foundational studies establish the representative algorithms, while selected studies published in 2022–2024 update their comparisons and reliability concerns. The review examines principles and reported evidence, without attempting to identify the latest leading system in every category.

2. Classification basis and relationships among the five paradigms

A learning paradigm describes the general arrangement through which a model learns. An algorithm provides the steps used within that arrangement. For example, supervised learning describes learning from examples with known answers, while a random forest provides one way to do this [2]. Training adjusts the model to improve performance according to an objective, which specifies what counts as improvement.

The learning signal is the information that guides this adjustment. The categories differ by more than a yes-or-no choice about labels. Supervised learning uses supplied targets, while unsupervised learning searches for structure without externally supplied targets [1, 4-6]. Semi-supervised learning adds unlabeled examples to a limited labeled set [7]. Self-supervised learning constructs targets from the original data, such as hiding parts of text and predicting them [8]. Reinforcement learning uses rewards from actions, including rewards whose consequences extend into the future [9]. Unsupervised learning still requires a designed objective, so the absence of labels does not mean the absence of guidance.

These categories form a useful framework, but their boundaries overlap. Self-supervised learning can be classified as a branch of unsupervised representation learning because it does not require externally supplied labels. It can also be discussed separately to emphasize how it constructs training targets from data. This review follows the second convention to distinguish these objectives from clustering and visualization. The five categories are therefore an organizing choice, rather than a unique or mutually exclusive classification. Section 6 explains this distinction with reference to the reviewed studies. A self-supervised model can later use labeled examples for a specific task [8]. Deep learning refers to models with multiple neural-network layers, rather than a separate source of supervision. Such models appear in both representation learning and reinforcement learning [8, 10].

3. Supervised learning and representative algorithms

Supervised learning connects inputs with supplied target values. Classification predicts categories, while regression predicts numerical values. The support vector machine (SVM), random forest, and XGBoost provide different ways to learn this relationship. SVM seeks a decision boundary with a large margin, meaning separation from nearby training observations. Kernel functions allow nonlinear separation without explicitly constructing every transformed feature [1]. Random forest combines trees trained with randomness in observations and feature selection [2]. XGBoost adds

DOI: 10.54254/2755-2721/2026.LD39061

trees sequentially to improve a prediction objective, while penalizing unnecessary model complexity [3].

The main trade-off concerns how these methods control error. SVM depends on the chosen kernel and settings. A random forest reduces dependence on one tree by combining different trees, while XGBoost directs later trees toward improving the existing model [1-3]. Breiman reports greater resistance to noise than the boosting method examined in his study, but this does not establish that forests outperform every boosting system [2]. XGBoost contributes both model controls and computing improvements for large datasets [3]. Therefore, the studies support different strengths rather than a universal winner. More recent evidence keeps this comparison open. Holzmüller and colleagues show that RealMLP, an improved multilayer neural network, can compete with boosted trees on their tabular benchmarks [11]. They also improve default settings for XGBoost and other tree systems [11]. This supports comparing prediction quality and computing time under the same conditions.

These approaches are relevant when reliable target values are available. Random forests provide feature-importance measures, and XGBoost reports applications involving customer behavior and sales prediction [2, 3]. However, combining many trees makes a prediction harder to follow than one tree. This follows from their structures and matters when explanations are required. In practice, prediction quality should be compared with tuning effort and interpretability. Reliable labels require a separate check, since a prediction method does not verify the meaning of its targets.

4. Unsupervised learning and representative algorithms

Unsupervised learning explores structure without supplied target answers. K-means and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) produce groups, while t-SNE produces a low-dimensional display. In the alternating procedure associated with Lloyd's work, observations are assigned to nearby centers and the centers are then updated [4]. DBSCAN instead connects sufficiently dense regions and treats observations outside these regions as noise [5]. The t-distributed stochastic neighbor embedding method, or t-SNE, represents high-dimensional observations in a small number of dimensions while emphasizing neighborhood relationships [6].

An unlabeled task still needs assumptions. K-means requires a predetermined number of centers, and distance to centers favors compact groups. DBSCAN allows irregular shapes and does not require the number of groups beforehand, but the neighborhood radius and minimum density influence its result [4, 5]. A density choice suitable for one group may fail for another. Meanwhile, t-SNE addresses visual exploration rather than the same clustering objective. Its optimization can produce different displays, and preserving local relationships does not guarantee that every distance in the display reflects the original data [6]. A recent review by Ikotun and colleagues examines K-means variants and identifies initialization, cluster-number selection, and scalability as continuing concerns [12]. Improvements address different parts of the algorithm, so they should be checked against the data and the grouping objective. A newer variant does not remove the need to interpret the resulting groups.

For example, a business could explore customer groups, while a researcher could inspect representations with many features. These are illustrative uses, not additional experimental findings. Exploration can begin without supplied categories, but a visible group does not automatically represent a useful real category. This review recommends checking patterns under alternative settings and domain interpretation. A clustering score or an attractive visualization cannot answer a prediction question that the method was not designed to solve.

DOI: 10.54254/2755-2721/2026.LD39061

5. Semi-supervised learning and representative algorithms

Semi-supervised learning uses a limited labeled set together with additional unlabeled examples. Deep generative models, MixMatch, and FixMatch share this aim, but use the extra data differently [7, 13, 14]. Generative models learn a probabilistic account of how observations arise, using hidden variables to represent underlying factors [7]. MixMatch estimates labels from modified examples and combines labeled and unlabeled data. FixMatch also generates labels, but uses a simpler confidence-based procedure [13, 14].

MixMatch combines several operations. It changes inputs through data augmentation, sharpens estimated label distributions, and blends inputs and targets through MixUp [13]. FixMatch first predicts a label from a weakly changed image. When the prediction passes a confidence threshold, the model learns to predict that label from a strongly changed version [14]. Both encourage consistent predictions across related examples. In comparison, generative learning places more emphasis on modeling the observations themselves [7]. They agree on the value of unlabeled data but differ in how that information should guide training. FreeMatch develops this confidence-based approach by adjusting thresholds to the model's learning progress and the confidence of each class [15]. The study shows how a fixed high threshold can leave useful unlabeled examples unused early in training [15]. Its adaptive selection improves results on limited-label image benchmarks, while still relying on the model predictions [15].

MixMatch and FixMatch report improvements on image-classification benchmarks with limited labels [13, 14]. FixMatch also shows sensitivity to which examples receive labels when very few are available [14]. A confident estimated label is not necessarily correct, so confidence filtering does not prove label quality. Medical classification is a possible setting when annotation requires expertise, but these studies do not establish clinical reliability. Before applying a method to a new setting, researchers should check class coverage and the relationship between labeled and unlabeled data.

6. Self-supervised learning and representative algorithms

Self-supervised learning builds training tasks from the data themselves. Its usual purpose here is representation learning, which converts inputs into features useful for later tasks. Bidirectional Encoder Representations from Transformers (BERT) learns from text, while SimCLR and Momentum Contrast (MoCo) learn from images [8, 16, 17]. BERT predicts hidden text tokens, which are words or word pieces, using surrounding context. Its original training also includes predicting whether two text segments follow each other [8]. SimCLR and MoCo instead compare related and unrelated image views [16, 17]. The terminology also reflects overlapping classifications: the MoCo paper describes self-supervised learning as a form of unsupervised learning, while SimCLR emphasizes its self-supervised training objective [16, 17]. Separating this category here highlights its target construction without denying that relationship.

In contrastive learning, the model learns to associate two modified views of the same image and distinguish them from comparison examples. SimCLR uses examples within a training batch and a small additional network that transforms features for this comparison [16]. MoCo stores comparison features in a queue. It updates their encoder, the network producing those features, through a moving average so that the stored features remain relatively consistent [17]. Both learn useful visual features, but they organize the comparison data differently. SimCLR benefits from larger batches, while MoCo allows the comparison collection to extend beyond one batch [16, 17]. Masked autoencoders (MAE) provide another route for images. They hide image patches and reconstruct the missing pixels, using visible patches in the encoder and a lightweight reconstruction network [18].

DOI: 10.54254/2755-2721/2026.LD39061

This extends masked prediction beyond text. Compared with SimCLR and MoCo, MAE learns through reconstruction rather than comparing image views, showing different ways to construct training targets [18].

A model can learn before labels for the final task are available. BERT demonstrates adaptation to language tasks, and the visual studies evaluate learned features on later tasks [8, 16-18]. However, constructing training targets still requires design choices. SimCLR shows that image modifications and training duration affect results [16]. Less manual annotation therefore does not mean learning without human decisions or computing costs. The final task should be checked directly, a constructed training task is not sufficient evidence of reliable deployment.

7. Reinforcement learning and representative algorithms

Reinforcement learning concerns an agent that acts in an environment. A state describes its situation, an action changes that situation, and a reward evaluates an outcome. Return combines rewards over time. Q-learning estimates the expected return associated with taking an action and then continuing appropriately. It is model-free because it does not require a supplied model of the environment, and off-policy because it can learn an optimal policy while following exploratory actions [9]. A policy specifies the method of choosing actions.

Tabular Q-learning stores values for state-action pairs. Its convergence result depends on conditions including sufficient repeated sampling and suitable learning rates [9]. A deep Q-network (DQN) replaces the table with a neural network. It reuses stored experiences and maintains a separate target network to reduce instability during training [10]. Policy-gradient methods instead adjust a parameterized policy to increase expected return. Sutton and colleagues explain how learned value estimates can support this update and reduce the variability of gradient estimates [19]. Thus, value-based and policy-based methods differ in what they directly learn, although they can share information.

DQN demonstrates learning from game images, while policy-gradient theory supports learning action selection directly [10, 19]. These contributions address different parts of the decision problem and do not justify ranking all reinforcement-learning methods together. Tabular methods face storage limits as the number of situations grows, neural methods introduce approximation and stability concerns [9, 10]. Actions also influence later observations, making data collection part of learning. For a possible control application, interaction cost and safety should be evaluated before transferring a method from a simulated task. Success in games alone does not establish that real-world exploration will be acceptable. Recent evaluation guidance also emphasizes sensitivity to settings and implementation details [20]. Patterson and colleagues discuss performance variation, statistical testing, and bias in hyperparameter selection [20]. This supports examining results across repeated runs and reporting the evaluation conditions, rather than drawing conclusions from one favorable run.

8. Cross-category comparison, research gaps, and future directions

Across the five paradigms, training information determines both opportunities and costs. Supervised methods depend on supplied targets, clustering relies on assumptions about structure, semi-supervised and self-supervised methods create additional uses for unlabeled data [1-8, 13, 14, 16, 17]. Reinforcement learning connects information gathering with actions and rewards [9, 10, 19]. The comparison therefore suggests examining how guidance is obtained, rather than assuming that fewer labels always mean an easier learning problem.

DOI: 10.54254/2755-2721/2026.LD39061

Their outputs also require different evaluation. Prediction error measures agreement with targets, whereas clustering quality concerns grouping and representation quality concerns usefulness for later tasks [2, 5, 16]. Reinforcement learning examines return [19]. Because these quantities answer different questions, the reviewed results cannot support a shared numerical ranking. Future comparisons should hold the practical task and available resources constant where a comparison is meaningful.

The selected evidence identifies specific reliability concerns, including sensitivity to labeled examples, image transformations, and training stability [10, 14, 16]. It does not provide a common evaluation of demographic fairness, transferred data bias, or changes between training and deployment data. This is a gap in this review's evidence base, not proof that these are unstudied issues. Future work should test these conditions directly and report performance for relevant groups. Such checks would help distinguish good average results from reliable use across settings.

Combining paradigms offers another direction. Self-supervised pretraining followed by supervised adaptation is demonstrated in the reviewed studies [8, 16-18]. Recent work also shows that progress can come from improving default model settings, adjusting how generated labels are selected, and changing representation-learning targets [11, 15, 18]. These developments preserve the basic learning signals while improving how models use them. Combinations with semi-supervised or reinforcement learning should be tested under equal data and computing budgets. Reporting annotation effort, training time, memory, and repeated-run variation would make the trade-offs clearer [20].

Finally, this framework is selective. Active learning, transfer learning, online learning, and federated learning lie outside the detailed comparison. These approaches describe additional aspects of learning, so the five categories should be used as an educational framework rather than an exhaustive division of the field.

9. Conclusion

The five learning paradigms differ mainly in the source, form, and timing of the information that guides training. They also differ in the output they aim to produce. Supervised learning connects inputs with supplied targets. Unsupervised learning explores structure without these targets. Semi-supervised learning combines limited labels with additional observations, while self-supervised learning constructs targets from the observations themselves. Reinforcement learning uses rewards to improve decisions over time. This comparison explains why the availability of labels alone cannot describe every learning arrangement.

The representative algorithms show that each arrangement involves a trade-off. Supervised methods provide a direct route to prediction when suitable targets exist. Unsupervised methods support exploration, but their patterns require interpretation. Semi-supervised learning can make limited labels more useful, while self-supervised learning can prepare features for later tasks. However, both depend on how unlabeled information is used. Reinforcement learning addresses action sequences, but the cost of collecting experience and the stability of training remain important. The reviewed evidence therefore supports choosing a method according to its task and constraints. It does not establish one paradigm as the best choice across unrelated applications.

This review provides a unified framework linking learning signals, representative algorithms, and practical choices. Selected studies from 2022–2024 also show how these choices continue to develop. The review has two limitations. First, its representative algorithms cover only part of each paradigm. Second, the literature review is selective and therefore does not provide a comprehensive account of recent developments. Future reviews should include a wider range of algorithms and

DOI: 10.54254/2755-2721/2026.LD39061

studies under clear comparison criteria. Practical research should evaluate data efficiency, computing cost, interpretability, and reliability together. Combinations of learning paradigms also deserve attention, but their benefits should be tested against suitable alternatives under comparable conditions.

References

[1] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*, 273–297.

[2] Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

[3] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794).

[4] Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory, 28*(2), 129–137.

[5] Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining* (pp. 226–231).

[6] van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research, 9*, 2579–2605.

[7] Kingma, D. P., Rezende, D. J., Mohamed, S., & Welling, M. (2014). Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems (Volume 27)* (pp. 3581–3589).

[8] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT (Volume 1)* (pp. 4171–4186).

[9] Watkins, C. J. C. H., & Dayan, P. (1992). Q-learning. *Machine Learning, 8*, 279–292.

[10] Mnih, V., et al. (2015). Human-level control through deep reinforcement learning. *Nature, 518*, 529–533.

[11] Holzmüller, D., Grinsztajn, L., & Steinwart, I. (2024). Better by default: Strong pre-tuned MLPs and boosted trees on tabular data. In *Advances in Neural Information Processing Systems (Volume 37)* (pp. 26577–26658).

[12] Ikotun, A. M., Ezugwu, A. E., Abualigah, L., Abuhaija, B., & Heming, J. (2023). K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences, 622*, 178–210.

[13] Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., & Raffel, C. A. (2019). MixMatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems (Volume 32)* (pp. 5050–5060).

[14] Sohn, K., et al. (2020). FixMatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems (Volume 33)* (pp. 596–608).

[15] Wang, Y., et al. (2023). FreeMatch: Self-adaptive thresholding for semi-supervised learning. In *Proceedings of the International Conference on Learning Representations* (pp. 1–20, paper PDF pagination).

[16] Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning (Volume 119)* (pp. 1597–1607). PMLR.

[17] He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 9729–9738).

[18] He, K., Chen, X., Xie, S., Li, Y., Dollár, P., & Girshick, R. (2022). Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 16000–16009).

[19] Sutton, R. S., McAllester, D., Singh, S., & Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems (Volume 12)* (pp. 1057–1063).

[20] Patterson, A., Neumann, S., White, M., & White, A. (2024). Empirical design in reinforcement learning. *Journal of Machine Learning Research, 25*(318), 1–63.