

## Chapter 4

# Tools and frameworks for machine learning and deep learning: A review

Nitin Liladhar Rane <sup>1</sup>, Suraj Kumar Mallick <sup>2</sup>, Ömer Kaya <sup>3</sup>, Jayesh Rane <sup>4</sup>

<sup>1</sup> Vivekanand Education Society's College of Architecture (VESCOA), Mumbai 400074, India

<sup>2</sup> Shaheed Bhagat Singh College, University of Delhi, New Delhi 110017, India

<sup>3</sup> Engineering and Architecture Faculty, Erzurum Technical University, Erzurum 25050, Turkey

<sup>4</sup> Pillai HOC College of Engineering and Technology, Rasayani, India

[^1nitinrane33@gmail.com](mailto:nitinrane33@gmail.com)

**Abstract:** The fast progress of Artificial Intelligence (AI) has resulted in major advancements in tools and structures for machine learning (ML) and deep learning (DL), changing numerous industries. This study offers a thorough examination of the most recent tools and frameworks that aid in the creation, implementation, and expansion of ML and DL models. The capabilities of key frameworks like TensorFlow, PyTorch, and Keras are assessed in their ability to facilitate the construction of complex neural networks by researchers and practitioners. Furthermore, cutting-edge resources such as Hugging Face Transformers for simplifying natural language processing duties and NVIDIA's RAPIDS suite for speeding up data science processes are praised for their inventive impacts. The research delves into incorporating these tools with cloud services such as Google Cloud AI, Amazon SageMaker, and Microsoft Azure ML, highlighting how they help make high-performance computing resources more accessible. The discussion revolves around how automated machine learning (AutoML) frameworks like Google's AutoML and H2O.ai ease the process of model selection and hyperparameter tuning, making it more accessible to non-experts. In addition, the study explores the significance of MLOps tools such as MLflow and Kubeflow, which support the continuous integration and deployment of ML workflows, guaranteeing scalability and reproducibility. This research offers valuable insights into the current ML and DL frameworks landscape by examining the strengths and limitations of these tools, helping researchers and industry professionals choose the right tools for their AI projects.

**Keywords:** Artificial intelligence, Machine learning, Deep learning, Tools, Frameworks, Automated machine learning, Reinforcement learning.

---

**Citation:** Rane, N. L., Mallick, S. K., Kaya, O., & Rane, J. (2024). Tools and frameworks for machine learning and deep learning: A review. In *Applied Machine Learning and Deep Learning: Architectures and Techniques* (pp. 80-95). Deep Science Publishing. [https://doi.org/10.70593/978-81-981271-4-3\\_4](https://doi.org/10.70593/978-81-981271-4-3_4)

---

## 4.1 Introduction

The progress of machine learning (ML) and deep learning (DL) has transformed multiple industries, such as healthcare, finance, manufacturing, and entertainment (Nasir & Sassani, 2021; Sharifani & Amini, 2023; Ferreira et al., 2021). These technologies have proven to be highly effective in automating intricate tasks, revealing hidden patterns in extensive datasets, and offering predictive insights that influence decision-making procedures (Nasir & Sassani, 2021; Jiao et al., 2020; Chiche & Yitagesu, 2022; Taye, 2023). At the core of these progressions are a variety of tools and frameworks created to simplify the development, training, and launching of ML and DL models (Ferreira et al., 2021; Geetha & Thilagam, 2021; Kaluarachchi et al., 2021; Soori et al., 2023). These tools improve model development efficiency and also make advanced analytics accessible to a wider range of researchers and practitioners, allowing individuals with different levels of expertise to utilize ML capabilities (Serin et al., 2020; Fregoso-Aparicio et al., 2021; Xu et al., 2021; Hopkins et al., 2022). Recently, the ML and DL tool and framework ecosystem has grown considerably, providing a variety of choices to suit various needs and preferences (Dargan et al., 2020; Janiesch et al., 2021; Mijwil et al., 2023). A variety of resources, ranging from extensive libraries like TensorFlow and PyTorch to specialized platforms such as Keras and Scikit-learn, are available to meet various needs in the ML process. These tools offer necessary features like data preprocessing, creating model architecture, optimizing training, and deploying, simplifying the implementation of complex algorithms and achieving superior performance (Ferreira et al., 2021; Al-amri et al., 2021; Forootan et al., 2022; Vercio et al., 2020). Furthermore, by combining these tools with other technologies like cloud computing and hardware accelerators, their abilities have improved even more, making it easier to manage large datasets and complex model structures. Even with numerous tools to choose from, deciding on the right framework for a particular application is still a significant obstacle. The decision is influenced by different factors such as the type of issue, the amount and intricacy of the information, the necessary speed, and the user's skill level (Kaluarachchi et al., 2021; Avci et al., 2021; Moein et al., 2023; Jayatilake et al., 2021).

Machine learning (ML) and deep learning (DL) are both sub-branches of the field of artificial intelligence (AI), and although they share many similarities, they are distinguished by some key differences. These differences arise in various aspects such as complexity of algorithms, data requirements, need for computing power, and application areas.

**Machine Learning (ML):** It includes algorithms that learn from data and apply what they have learned on future data. Machine learning predicts and classifies using statistical

methods. ML includes a wide range of algorithms, such as linear regression, decision trees, k-nearest neighbors (KNN), support vector machines (SVM), etc.

Deep Learning (DL): It is a sub-branch of machine learning and performs the learning process using multi-layer artificial neural networks. DL is especially effective for learning large data sets and complex structures. Basically, neural networks and deep versions of these networks (e.g., convolutional neural networks, recurrent neural networks) are used. The main differences between ML and DL is discussed below in Table 4.1.

**Table 4.1** The main differences between ML and DL

Criterion	ML	DL
Definition and Scope	Learns from data and makes predictions using algorithms	Subset of ML using multilayer neural networks
Data	It can work with small data sets.	Requires large data sets.
Requirements		
Computing Power	Requires less computing power; It can run on CPU.	Requires high computing power; Works better with GPU/TPU.
Model Complexity	Simpler models; easy to understand and interpret	More complex models; difficult to interpret
Application Areas	Wide application areas such as finance, health, marketing	Successful in fields such as image recognition, NLP, speech recognition, autonomous driving
Feature Engineering	Requires manual feature engineering	Automatically extracts features from data
Education time	Generally faster training time	Usually longer training period
Interpretability	Easier to interpret and understand	Difficult to interpret

This study intends to give a thorough overview of the present status of ML and DL tools and frameworks, emphasizing their characteristics, advantages, and constraints. This study aims to provide valuable insights through a systematic evaluation of these tools, assisting researchers and practitioners in making informed decisions suited to their individual requirements.

Contributions of this research work are as follows:

- 1) A detailed comparison of the most widely used ML and DL tools and frameworks, emphasizing their unique features and suitability for different types of applications.

- 2) An analysis of the integration capabilities of these tools with other technologies, such as cloud platforms and hardware accelerators, and their impact on performance and scalability.
- 3) A discussion on the challenges and future directions in the development and utilization of ML and DL tools, providing insights into potential areas of improvement and innovation.

## 4.2 Methodology

The literature review started by thoroughly searching important academic databases such as IEEE Xplore, SpringerLink, ScienceDirect, and Google Scholar. Terms like "machine learning tools," "deep learning frameworks," "ML software," "DL libraries," and "AI development environments" were employed to find relevant literature. The search was not limited to a specific publication date in order to include both past and present advances in ML and DL tools. Specific inclusion and exclusion criteria were put in place to guarantee the relevance and quality of the literature. The selection included peer-reviewed articles, conference papers, and authoritative reviews that examined tools and frameworks utilized for machine learning and deep learning. Articles without actual data or thorough reviews, and sources not reviewed by peers were not considered. Furthermore, articles that concentrated only on theoretical elements without assessing practical tools were excluded. Data extraction was a thorough process in which pertinent information was gathered from every chosen article. This contained information regarding the tools and frameworks talked about, along with their characteristics, benefits, drawbacks, and uses in different ML and DL assignments (Geetha & Thilagam, 2021; Kaluarachchi et al., 2021; Soori et al., 2023; Serin et al., 2020; Meyer, 2021; Padarian, 2020; Pramod et al., 2021). Key indicators like performance benchmarks, user-friendliness, scalability, and community support were also recognized. The data that was taken out was sorted into thematic groups to make a structured analysis easier. Qualitative synthesis was used to analyze the extracted data. This included a process of examining the tools and frameworks by looking at the themes and metrics that were identified. An evaluation was conducted to emphasize the advantages and disadvantages of each tool and framework (Jiao et al., 2020; Chiche & Yitagesu, 2022; Taye, 2023; Soori et al., 2023; Serin et al., 2020; Meyer, 2021; Padarian, 2020; Pramod et al., 2021; Thakkar & Lohiya, 2021; Bal & Kayaalp, 2021). Identifying trends and patterns in the adoption and evolution of these tools offer insights into their development and utilization within the field.

Table 4.2 summarizes various tools and frameworks commonly used in the ML and DL fields. A description and area of focus for each tool or framework is noted. These tools and frameworks are important software and platforms that make it easier for researchers and practitioners to develop, train, and deploy models (Dietterich, 2000; Hall et al., 2009;

Nguyen et al., 2019). The information in the table is provided as a guide when choosing the most appropriate tools for ML and DL projects.

**Table 4.2** Tools and Frameworks for ML and DL

<b>Tool/Frame</b>	<b>Explanation</b>	<b>Focus Area</b>
Scikit-Learn	A Python-based ML library; It contains many algorithms such as regression, classification and clustering.	ML
TensorFlow	An open source ML and DL framework developed by Google; Provides GPU/TPU support for high-performance calculations.	ML and DL
Keras	A user-friendly, high-level API; It can run on TensorFlow, CNTK and Theano; Ideal for rapid prototyping.	DL
XGBoost	An ML library based on the gradient boosting algorithm; It is known for its high performance and fast training process.	ML
LightGBM	Gradient boosting framework developed by Microsoft; It offers a fast and scalable solution for large data sets.	ML
PyTorch	An open source DL framework developed by Facebook; It is popular for its dynamic calculation graphics and easy learning curve.	DL
MXNet	DL framework developed by Apache; known for its flexibility and efficiency; Supports symbolic and imperative programming modes.	DL
Caffe	An open source DL framework developed by the Berkeley Vision and Learning Center; It is especially widely used in image recognition applications.	DL
Microsoft Cognitive Toolkit (CNTK)	An open source DL framework developed by Microsoft; It can work on large data sets with high efficiency.	DL
Deeplearning4j	An open source Java-based DL framework; It provides integration with big data processing tools such as Hadoop and Spark.	DL
WEKA	A Java-based open source ML software; offers extensive data mining and analysis tools.	ML
H2O.ai	An open source ML platform; It offers a wide algorithm library and automatic ML support.	ML
RapidMiner	A software that offers data mining and analysis tools; Easy to use with visual workflows.	ML
Apache Mahout	ML library developed by Apache; It offers scalable ML algorithms on large data sets.	ML
Orange	Offers data analysis and ML tools with a visual programming interface; easy use.	ML

Theano		A Python library for deep learning models; Optimized for mathematical operations.	DL
Turi Create		ML framework developed by Apple; It is especially suitable for iOS and macOS applications.	ML
Google Colab		A free cloud-based notebook environment provided by Google; It allows running ML and DL models with GPU/TPU support.	ML and DL
Azure Machine Learning		Microsoft's cloud-based ML platform; Provides tools for model training, deployment, and management.	ML
Amazon SageMaker		Cloud-based ML platform provided by Amazon; simplifies model development and deployment processes.	ML
IBM Watson		AI platform developed by IBM; It offers extensive ML and DL tools and is suitable for business analytics and data science.	ML and DL
KNIME		Open source data analytics platform; It offers visual workflows and broad data integration support.	ML
DataRobot		Automated ML platform; It automates data preparation, model training and evaluation processes.	ML
MLflow		An open source ML lifecycle platform; Provides tools for experiment tracking, model management and deployment.	ML
DL4J		An open source Java-based DL library; Provides integration with big data and Hadoop.	DL

---

## 4.3 Results and discussions

### Emerging ML and DL tools

One major trend in ML and DL is the increasing popularity of AutoML platforms. AutoML platforms like Google's AutoML, H2O.ai, and DataRobot aim to streamline the entire process of applying ML to practical problems through automation. They make tasks such as feature engineering, model selection, and hyperparameter tuning easier, enabling non-experts to construct efficient ML models. AutoML tools are beneficial in making AI accessible, enabling businesses lacking in-depth data science knowledge to use ML for their requirements. TFX, a new tool in the ML and DL field, is becoming more prominent as an end-to-end platform for implementing ML pipelines in production. TFX offers a strong foundation for handling the complete ML process, starting from acquiring data to deploying models. It consists of features for validating data, analyzing models, and serving at scale, guaranteeing that ML models can be managed and updated effectively in production settings. TFX's connection with TensorFlow, a popular DL library, enhances its capabilities for creating and implementing intricate neural networks.

PyTorch, created by Facebook's AI Research lab, has been increasingly popular in the past few years as a favorite DL framework. Its flexible model development is popular among researchers and developers because of its dynamic computation graph. PyTorch Lightning extends the functionalities of PyTorch by offering a more advanced interface for handling training loops and distributed training. This assists in making the development of DL models more efficient, allowing for quicker experimentation and iteration. The increase in the use of transfer learning and pre-trained models has also risen due to the presence of powerful models such as BERT, GPT-4, and ResNet. Transfer learning enables models pre-trained on extensive datasets to be adjusted for particular tasks, effectively minimizing the data and computational resources needed for training. Resources like the Transformers library from Hugging Face offer convenient access to numerous pretrained models, making it simpler to incorporate them into a variety of applications like NLP, image recognition, and others.

Graph Neural Networks (GNNs) are an exciting advancement in the domains of ML and DL. GNNs are specifically created for processing data that has a graph structure, making them ideal for use in social network analysis, recommendation systems, and molecular biology. DGL (Deep Graph Library) and PyTorch Geometric provide extensive resources for designing and training GNNs, allowing researchers to investigate intricate data relationships more efficiently. Federated learning is becoming increasingly important for improving privacy and security in ML applications. This method enables training models on various separate devices or servers without the need to share raw data. Google's TensorFlow Federated and PySyft, a library for secure and private ML in PyTorch, are key players in this field. Federated learning is especially important for healthcare and finance, where maintaining data privacy is crucial. Clarifying AI tools are growing in significance as ML and DL models are utilized in crucial decision-making procedures. XAI strives to ensure AI models' decision-making process is clear and comprehensible to people. LIME and SHAP offer understanding on how models generate predictions, aiding in establishing confidence and responsibility in AI systems.

In the field of deploying models, MLOps tools are changing how ML models are handled in production. MLflow, Kubeflow, and Amazon SageMaker provide extensive solutions for monitoring experiments, organizing code, and scaling model deployments. These tools make it easier to constantly integrate and deploy ML models, allowing for efficient updates and maintenance in the long run. Reinforcement learning (RL) is a field where the emergence of new tools is having a substantial influence. RL is about teaching agents to make choices by reinforcing preferred behaviors, and it has been applied in various fields such as robotics and game playing. OpenAI Gym and RLlib libraries offer tools for creating and evaluating RL algorithms, allowing researchers to address intricate decision-

making challenges. The merging of ML and DL tools with cloud services has also emerged as a significant trend. Top cloud providers such as AWS, Google Cloud, and Microsoft Azure provide a variety of ML services, including ready-to-use models and personalized training platforms. These services offer flexible structure and resources for handling data processing, training models, and deploying them, facilitating the implementation of ML solutions for businesses without the need for a large initial investment in hardware.

Furthermore, edge AI is picking up speed, with tools created to implement ML models on edge devices like smartphones, IoT devices, and autonomous systems. TensorFlow Lite and PyTorch Mobile are frameworks that allow lightweight ML models to be used on devices with limited resources. Edge AI enables immediate data processing and decision-making, essential for tasks such as self-driving cars, industrial automation, and smart home gadgets. Generating artificial data is becoming an important resource for training ML models in situations where real data is limited or confidential. Software options such as Nvidia's GauGAN and Unity's Perception provide the ability to create realistic artificial data for a range of uses, including computer vision and autonomous technologies. Synthetic data can enhance model performance and address data limitations by supplying varied and controlled datasets for training purposes. The rise of quantum machine learning (QML) tools is expanding the limits of what can be achieved with ML and DL. Quantum computing can solve specific problems at a much faster rate compared to classical computers. Frameworks like IBM's Qiskit and Google's Cirq allow for the creation and evaluation of QML algorithms, giving us a sneak peek into the upcoming advancements in computing and AI. Ethical AI tools are gaining significance as worries over bias and fairness in AI systems escalate. Products such as IBM's AI Fairness 360 and Google's What-If Tool offer platforms for examining and reducing bias in ML models. These tools are necessary to ensure that AI systems are created and implemented in a fair, transparent, and accountable way.

### **Emerging ML frameworks**

TensorFlow, created by Google Brain, has been an important part of the ML field for a long time. The launch of TensorFlow 2.0 represented a notable change towards simplicity and incorporation. One of the most significant updates is the close integration with Keras, which is a Python-based high-level neural networks API. This incorporation streamlines the process of constructing and training models, enabling developers to concentrate on creating intricate neural networks instead of handling detailed computations. TensorFlow 2.0's eager execution mode allows for easier debugging and a coding style that is similar to traditional programming languages. PyTorch, created by Facebook's AI Research lab, has become a formidable contender against TensorFlow. Its dynamic computational graph



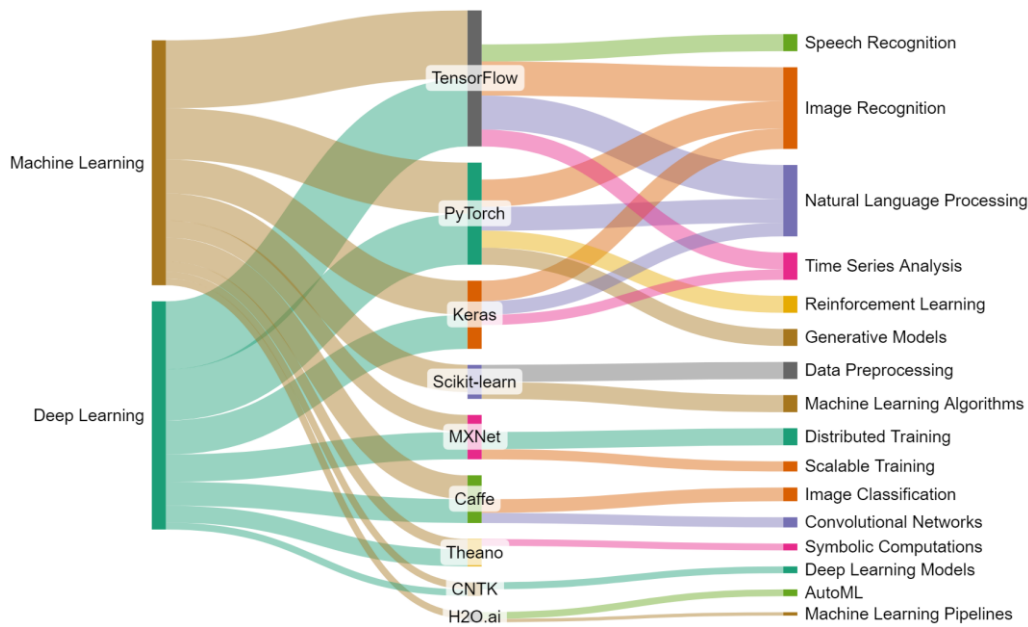
is especially valued for the increased flexibility it provides during model development. PyTorch Lightning, an lightweight PyTorch enclosure, has improved the framework's ease of use. It simplifies a lot of the redundant code needed for training models, allowing researchers and developers to create code that is more concise and easier to read. PyTorch Lightning also provides distributed training, automatic checkpointing, and smooth integration with modern hardware accelerators such as GPUs and TPUs, making it a robust tool for extensive ML projects.

Apache MXNet, backed by Amazon Web Services (AWS), is also a major contributor to the ML field. MXNet is recognized for its effectiveness and ability to scale, especially in situations where performance and resource allocation are crucial. The creation of Gluon, a high-level interface for MXNet, has simplified the process of defining, training, and deploying DL models. Gluon merges the adaptability of a dynamic computational graph with the speed advantages of a static graph, enabling developers to transition smoothly between training and inference modes. This combination of methods enables MXNet and Gluon to be well-suited for both research and production settings. Google's JAX, which converts numerical Python code into efficient ML models, is becoming increasingly popular. The main advantage of JAX is its ability to automatically differentiate and compile code Just-In-Time (JIT), enhancing the performance of ML algorithms on CPUs and GPUs. It utilizes XLA (Accelerated Linear Algebra) for creating efficient lower-level code, resulting in quicker training times and better utilization of hardware resources. The functional programming approach of JAX promotes cleaner and more modular code, which is advantageous for complex ML applications.

The Hugging Face Transformers library has transformed the field of natural language processing (NLP). This library offers pre-made models and simple interfaces for various NLP tasks such as text classification, translation, and summarization. The Hugging Face library utilizes advanced models such as BERT, GPT, and T5, enabling developers to easily harness strong NLP capabilities. Hugging Face provides tools for adapting these models to specific datasets, making it suitable for various uses in NLP research and business applications. The ONNX format has become a widely accepted standard for sharing models across various frameworks, enabling seamless interoperability. Microsoft's ONNX Runtime is a tool for making inferences that is compatible with models trained in different frameworks such as TensorFlow, PyTorch, and MXNet. ONNX Runtime enhances model efficiency for faster inference across various hardware platforms such as CPUs, GPUs, and specialized accelerators like FPGAs. This framework is especially valuable in production settings where model deployment effectiveness and scalability are crucial.

MLflow, created by Databricks, is an open-source platform that deals with the challenges of handling the ML lifecycle. It offers features for monitoring experiments, tracking model versions, and deploying, facilitating collaboration and project maintenance for data scientists and ML engineers. MLflow's flexible and scalable architecture can be integrated with a variety of ML frameworks and cloud services, enabling different workflows. MLflow improves efficiency and guarantees that ML experiments can be replicated and tracked by simplifying the entire ML process. TFX is a platform created by Google that enables the deployment of ML pipelines for production from start to finish. It expands TensorFlow's capabilities to handle all aspects of the ML process, including data intake, verification, model development, assessment, and implementation. TFX consists of elements such as TensorFlow Data Validation (TFDV) for assessing data quality, TensorFlow Transform (TFT) for creating features, and TensorFlow Model Analysis (TFMA) for evaluating models. These parts are created to function smoothly in unison, allowing for strong and expandable ML pipelines capable of managing extensive data processing and model training duties.

Ray is a framework for distributed computing that makes it easier to create scalable and high-performance ML applications. It offers a versatile and user-friendly API for concurrent and distributed programming, allowing the effective completion of intricate tasks on numerous nodes. Ray Tune, an automated hyperparameter tuning tool developed on Ray framework, streamlines the search for optimal hyperparameters in ML models. It offers different search algorithms and early stopping mechanisms, making it a strong tool for enhancing model performance. Ray's ability to grow and adapt makes it perfect for big ML projects and research trials. FastAI is a high-level library created on PyTorch to make DL easier to use and applicable. It offers simple APIs for creating and training models, as well as a wide range of pre-trained models and techniques for data augmentation. FastAI focuses on applying practical aspects and simplifying DL to make it accessible to a wider audience, allowing more users to harness its capabilities. The library's emphasis on transfer learning, along with its thorough documentation and tutorials, has attracted a wide range of users in the ML community, including both novices and experts. H2O.ai offers tools for creating and implementing ML models on an open-source platform. It is recognized for its ability to handle large datasets efficiently and effectively, especially in terms of scalability and performance. AutoML by H2O.ai automates model selection, hyperparameter tuning, and feature engineering, allowing users to create effective models efficiently with minimal manual input. This automation is especially important in business settings where speed-to-market and model effectiveness are crucial. H2O.ai's compatibility with well-known data science tools and cloud platforms enhances its usefulness and adaptability even more.



**Fig. 4.1** Sankey diagram of emerging ML and DL frameworks

The Sankey diagram (Fig. 4.1) depicts the interconnected relationship among various ML and DL (DL) frameworks and their wide range of uses. At the top level, the diagram separates general ML frameworks from those specifically designed for DL, emphasizing the intersection and expertise within these areas. Popular tools such as TensorFlow, PyTorch, and Keras are highlighted due to their common use and flexibility in ML and DL applications. TensorFlow is becoming an important tool used in various applications, including image recognition, natural language processing (NLP), time series analysis, and speech recognition. PyTorch is widely used in image recognition, NLP, reinforcement learning, and generative models, highlighting its popularity in cutting-edge DL research and development. Keras is mostly utilized for image recognition, natural language processing (NLP), and time series analysis due to its easy-to-use interface, making it a popular option for quick prototyping and testing. Even though Scikit-learn is mainly focused on ML, it plays a crucial role in preparing data and applying ML techniques, connecting traditional ML with DL. MXNet is known for its strength in distributed training and Caffe is recognized for its expertise in convolutional networks, both meeting the demands of scalability and high performance. The older framework Theano is still important for performing symbolic computations, which are crucial for building new DL models. CNTK, Microsoft's Cognitive Toolkit, and H2O.ai, recognized for AutoML and ML pipelines, demonstrate unique yet effective tools within the ML environment. This in-depth diagram outlines the most common frameworks and their specific uses, giving

an overview of how different tools are utilized in a range of AI tasks. This visualization helps to comprehend how each framework specializes and adapts, showcasing their impact on advancing ML and DL technologies.

### **Emerging DL frameworks**

Google has developed TensorFlow 2.0, which is considered one of the prominent new frameworks. TensorFlow 2.0 has made the model building process easier by using its user-friendly high-level Keras API, which is now more accessible to all, from beginners to experts. Eager execution is enabled by default, providing increased interactivity and versatility in constructing models. TensorFlow 2.0 also smoothly incorporates TensorFlow Extended (TFX) for complete ML pipelines, and TensorFlow Lite for model deployment on mobile and edge devices. These improvements make TensorFlow 2.0 a flexible platform appropriate for various uses, spanning from research to deployment. PyTorch, created by Facebook's AI Research lab, has become extremely popular because of its dynamic computation graph, providing more flexibility and ease of use than static graph frameworks. PyTorch's focus on simplicity and transparency in its design approach has garnered popularity with both researchers and developers. The introduction of PyTorch 1.9 brought forth new functionalities like distributed training abilities, built-in assistance for mixed-precision training, and enhanced performance optimizations. PyTorch's collection of libraries, including TorchVision, TorchText, and TorchAudio, offer pre-trained models and tools for different types of data, expanding its usefulness.

Another significant framework is JAX, created by Google. JAX stands out by its capacity to convert numerical functions into extremely optimized code that can run on CPUs, GPUs, and TPUs. It utilizes function transformations like automatic differentiation, vectorization, and parallelization to facilitate extremely efficient computations. JAX's interface is created to work seamlessly with NumPy, which simplifies its adoption and integration into current operations for users. JAX's increasing popularity in the research community is due to its performance and flexibility, making it ideal for extensive DL experiments and simulations. Hugging Face's Transformers library has become a primary framework for natural language processing (NLP). This library offers pre-trained models for various NLP tasks like text classification, translation, and question answering. The Transformers library provides support for various DL frameworks like TensorFlow and PyTorch, allowing smooth compatibility between them. A major advantage of this library is its capacity to utilize transfer learning, allowing users to customize pre-trained models for particular tasks using small datasets. This ability has greatly reduced the obstacles for creating advanced NLP applications.

ONNX, also known as Open Neural Network Exchange, is a significant framework that has become increasingly popular within the DL sector. ONNX offers a format for DL models that is open-source, enabling smooth transfer of models across various frameworks. This compatibility is essential for implementing models in various settings, including cloud services, mobile devices, and edge computing platforms. ONNX Runtime, the engine used for making inferences with ONNX models, is designed for high performance and compatible with multiple hardware accelerators, making it an efficient solution for deploying DL models in real-world scenarios. Huawei's MindSpore is a recently developed DL framework that caters to cloud and edge environments. MindSpore's goal is to offer a comprehensive framework that assists all stages of the ML process, from preparing data to deploying models. It includes an automatic differentiation engine and an effective distributed training system. MindSpore also facilitates collaborative development and model sharing with its ModelZoo, where a range of pre-trained models are available. The performance of the device is further improved through its combination with Huawei's Ascend AI processors, making it an attractive option for creating and using AI applications.

MXNet, backed by the Apache Software Foundation, has also made considerable progress in the field of DL. MXNet is recognized for its ability to scale effectively and efficiently, especially when it comes to distributed training. It offers a versatile programming model that accommodates both imperative and symbolic programming styles. The Gluon API in MXNet makes it easier to construct and train models with its simple and user-friendly interface. MXNet's versatility in being able to operate on various platforms such as cloud services and edge devices enables it to cater to a diverse set of application needs. The rise of TinyML frameworks has also been a noticeable development in recent years. TinyML is centered on implementing ML models on devices that have limits on their resources, like microcontrollers and IoT devices. TensorFlow Lite Micro is an iteration of TensorFlow Lite made particularly for microcontrollers. It allows the implementation of small DL models on devices that have restricted memory and processing capabilities. In the same way, Edge Impulse offers a system for creating and implementing ML models on edge devices, with a focus on user-friendly features and compatibility with different hardware platforms.

Aside from these frameworks, numerous other new tools and libraries are expanding the capabilities of DL. Constructed using JAX as a foundation, Flax offers a versatile and adaptable structure for constructing neural networks. It provides advanced abstractions for creating intricate models while retaining JAX's versatility. Fastai, developed using PyTorch as its foundation, seeks to enhance the accessibility of DL through the provision of advanced APIs and pre-existing models. It focuses on real-world applications and quick

prototyping, serving as a useful tool for both professionals and academics. DL frameworks are also developing to tackle ethical and responsible AI practices. An instance would be the Toolkit for Responsible AI in TensorFlow, which offers tools and guidelines for creating unbiased and equitable models. It consists of functions for evaluating models, fairness measures, and interpretability, assisting developers in creating AI systems that are more transparent and responsible. In the same way, PyTorch has released tools like Captum for interpreting model predictions and Fairseq for creating fair models. The growing complexity and scale of modern AI applications have fueled rapid progress in DL frameworks. As the need for advanced models increases, frameworks are developing to offer improved performance, scalability, and usability. The inclusion of cutting-edge functionalities like distributed training, mixed-precision training, and hardware acceleration has enabled the tackling of larger and more intricate challenges.

#### **4.4 Conclusions**

The development of tools and frameworks for ML and DL has greatly changed the field of artificial intelligence, allowing researchers and practitioners to create more advanced and effective models. One of the most influential progressions is the ongoing growth and extensive use of open-source frameworks like TensorFlow, PyTorch, and Keras. These frameworks have not just made ML and DL techniques more accessible to everyone, but have also created a collaborative atmosphere where new ideas are easily exchanged and put into practice. TensorFlow, created by Google, has proven to be a flexible platform able to manage a range of ML jobs, from basic linear models to intricate neural networks. The complete ecosystem it offers, which includes TensorFlow Lite and TensorFlow Extended (TFX), gives full support for implementing ML models on different platforms, from edge devices to cloud systems. Many researchers and developers prefer PyTorch due to its dynamic computation graph and user-friendly design, making it a popular choice for cutting-edge DL research. Its adaptability and user-friendly interface make it well-suited for prototyping and experimentation. Besides these basic structures, the emergence of specific tools such as Hugging Face's Transformers library has transformed the field of natural language processing (NLP). This library offers pre-trained models and fine-tuning features that significantly lessen the time and computational resources needed to achieve top performance in NLP tasks. Additionally, improvements in autoML platforms like Google's AutoML and Microsoft's Azure ML are making it easier for non-experts to get involved in ML by automating the creation and improvement of ML models. The noteworthy aspect is the incorporation of ML and DL frameworks with other upcoming technologies like Internet of Things (IoT) and edge computing. TensorFlow Lite and PyTorch Mobile are making it easier to use small models on devices with limited resources, allowing for quick data processing and decision-making at the edge. As these

tools progress, they are expected to speed up innovation and adoption in different industries, ultimately influencing the future of intelligent systems and applications.

## References

- Al-amri, R., Murugesan, R. K., Man, M., Abdulateef, A. F., Al-Sharafi, M. A., & Alkahtani, A. A. (2021). A review of machine learning and deep learning techniques for anomaly detection in IoT data. *Applied Sciences*, 11(12), 5320.
- Avci, O., Abdeljaber, O., Kiranyaz, S., Hussein, M., Gabbouj, M., & Inman, D. J. (2021). A review of vibration-based damage detection in civil structures: From traditional methods to Machine Learning and Deep Learning applications. *Mechanical systems and signal processing*, 147, 107077.
- Bal, F., & Kayaalp, F. (2021). Review of machine learning and deep learning models in agriculture. *International Advanced Researches and Engineering Journal*, 5(2), 309-323.
- Chiche, A., & Yitagesu, B. (2022). Part of speech tagging: a systematic review of deep learning and machine learning approaches. *Journal of Big Data*, 9(1), 10.
- Dargan, S., Kumar, M., Ayyagari, M. R., & Kumar, G. (2020). A survey of deep learning and its applications: a new paradigm to machine learning. *Archives of Computational Methods in Engineering*, 27, 1071-1092.
- Dietterich, T. G. (2000). Ensemble methods in machine learning. In *International workshop on multiple classifier systems* (pp. 1-15). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Ferreira, L., Pilastrri, A., Martins, C. M., Pires, P. M., & Cortez, P. (2021). A comparison of AutoML tools for machine learning, deep learning and XGBoost. In *2021 international joint conference on neural networks (IJCNN)* (pp. 1-8). IEEE.
- Forootan, M. M., Larki, I., Zahedi, R., & Ahmadi, A. (2022). Machine learning and deep learning in energy systems: A review. *Sustainability*, 14(8), 4832.
- Fregoso-Aparicio, L., Noguez, J., Montesinos, L., & García-García, J. A. (2021). Machine learning and deep learning predictive models for type 2 diabetes: a systematic review. *Diabetology & metabolic syndrome*, 13(1), 148.
- Geetha, R., & Thilagam, T. (2021). A review on the effectiveness of machine learning and deep learning algorithms for cyber security. *Archives of Computational Methods in Engineering*, 28(4), 2861-2879.
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1), 10-18.
- Hopkins, E. (2022). Machine learning tools, algorithms, and techniques. *Journal of Self-Governance and Management Economics*, 10(1), 43-55.
- Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets*, 31(3), 685-695.
- Jayatilake, S. M. D. A. C., & Ganegoda, G. U. (2021). Involvement of machine learning tools in healthcare decision making. *Journal of healthcare engineering*, 2021(1), 6679512.

- Jiao, Z., Hu, P., Xu, H., & Wang, Q. (2020). Machine learning and deep learning in chemical health and safety: a systematic review of techniques and applications. *ACS Chemical Health & Safety*, 27(6), 316-334.
- Kaluarachchi, T., Reis, A., & Nanayakkara, S. (2021). A review of recent deep learning approaches in human-centered machine learning. *Sensors*, 21(7), 2514.
- Meyer, J. G. (2021). Deep learning neural network tools for proteomics. *Cell Reports Methods*, 1(2).
- Mijwil, M., Salem, I. E., & Ismaeel, M. M. (2023). The significance of machine learning and deep learning techniques in cybersecurity: A comprehensive review. *Iraqi Journal For Computer Science and Mathematics*, 4(1), 87-101.
- Moein, M. M., Saradar, A., Rahmati, K., Mousavinejad, S. H. G., Bristow, J., Aramali, V., & Karakouzian, M. (2023). Predictive models for concrete properties using machine learning and deep learning approaches: A review. *Journal of Building Engineering*, 63, 105444.
- Nasir, V., & Sassani, F. (2021). A review on deep learning in machining and tool monitoring: Methods, opportunities, and challenges. *The International Journal of Advanced Manufacturing Technology*, 115(9), 2683-2709.
- Nguyen, G., Dlugolinsky, S., Bobák, M., Tran, V., López García, Á., Heredia, I., ... & Hluchý, L. (2019). Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey. *Artificial Intelligence Review*, 52, 77-124.
- Padarian, J., Minasny, B., & McBratney, A. B. (2020). Machine learning and soil sciences: A review aided by machine learning tools. *Soil*, 6(1), 35-52.
- Pramod, A., Naicker, H. S., & Tyagi, A. K. (2021). Machine learning and deep learning: Open issues and future research directions for the next 10 years. *Computational analysis and deep learning for medical care: Principles, methods, and applications*, 463-490.
- Serin, G., Sener, B., Ozbayoglu, A. M., & Unver, H. O. (2020). Review of tool condition monitoring in machining and opportunities for deep learning. *The International Journal of Advanced Manufacturing Technology*, 109(3), 953-974.
- Sharifani, K., & Amini, M. (2023). Machine learning and deep learning: A review of methods and applications. *World Information Technology and Engineering Journal*, 10(07), 3897-3904.
- Soori, M., Arezoo, B., & Dastres, R. (2023). Artificial intelligence, machine learning and deep learning in advanced robotics, a review. *Cognitive Robotics*.
- Taye, M. M. (2023). Understanding of machine learning with deep learning: architectures, workflow, applications and future directions. *Computers*, 12(5), 91.
- Thakkar, A., & Lohiya, R. (2021). A review on machine learning and deep learning perspectives of IDS for IoT: recent updates, security issues, and challenges. *Archives of Computational Methods in Engineering*, 28(4), 3211-3243.
- Vercio, L. L., Amador, K., Bannister, J. J., Crites, S., Gutierrez, A., MacDonald, M. E., ... & Forkert, N. D. (2020). Supervised machine learning tools: a tutorial for clinicians. *Journal of Neural Engineering*, 17(6), 062001.
- Xu, Y., Zhou, Y., Sekula, P., & Ding, L. (2021). Machine learning in construction: From shallow to deep learning. *Developments in the built environment*, 6, 100045.