

Available online at www.qu.edu.iq/journalcm

JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS

ISSN:2521-3504(online) ISSN:2074-0204(print)



A Deep Dive into Deep Learning and Machine Learning: A Review Study

Hind Khalid ^{a,*}

^a College of Political Science, Al-Nahrain University, Baghdad, Iraq. Email: dr.hind@nahrainuniv.edu.iq

ARTICLE INFO

Article history:

Received: 17/05/2025

Revised form: 22/07/2025

Accepted : 31/07/2025

Available online: 30/09/2025

Keywords:

Deep Learning
Machine Learning
Artificial Intelligence
Neural Networks
Computer Vision
Natural Language Processing
Artificial Intelligence

ABSTRACT

The current comprehensive review paper focuses on the analysis of the differences and similarities between deep learning and traditional machine learning. This paper analyses the architectural differences between DL and ML systems and include a comparison of the suitability of the two for tasks such as computer vision, NLP, speech recognition and Reinforcement Learning. The differences of data demand, training procedures, and computational power necessary for each method are discussed. The paper also focuses on the difficulties of hyperparameter tuning / optimization and research methodologies in both paradigms: ML and DL. Additionally, it introduces aspects of deployment and a range of size and speed values for models, inference, and resources. Problems as to how interconnections between ML and DL can provide the greatest performance are analyzed along with the viability of the mentioned integrated models. Moving to the future advancements, we discuss about the trends and the future work in both the fields which gives some idea about the upcoming building of artificial intelligence. In this extensive analysis, we hope that the target audience of researchers, practitioners, and decision-makers will be equipped with the knowledge regarding when and how to appropriately use ML and DL approaches while considering the complexity of a given task, availability of data, and computing resources. This work in turn, adds its voice to the ever-increasing discussion in the AI world with regards to the relative advantages and suitable uses of these strong methods. This review quantitatively compares performance (e.g., 92% DL vs. 85% ML accuracy in image tasks) and resource requirements across domains.

<https://doi.org/10.29304/jqcm.2025.17.32381>

1. Introduction

Machine Learning (ML) and Deep Learning (DL) are two of the new promising directions in the field of artificial intelligence that have revolutionized problem-solving across domains [1] for solving of the arising problems in different fields. Of the two broad categories AI is a subset of, Machine Learning deals with algorithms that are capable of training and making further predictions or decisions out of the data collected [1]. Another branch of ML as Deep Learning provides artificial neural networks with multiple layers that would help in modeling and processing of the patterns in data [2]. The two primary categories of algorithms are the supervised which comprises of the decision trees, support vector and the current popular random forests while the unsupervised encompasses clustering [3]. These classical ML techniques always involve comprehensive knowledge about the data characteristics and engineering of the features [4].

There is Deep Learning which uses deep neural networks to learn a representation of data in a hierarchical manner [5]. This capability has resulted in optimization of performance in areas like image and video recognition,

*Corresponding author: Hind Khalid

Email addresses: dr.hind@nahrainuniv.edu.iq

Communicated by 'sub etitor'

understanding of natural language and speech recognition [6]. Due to the DL models' capacity in training direct features, the need for feature engineering is nearly eliminated on numerous applications [7].

For these reasons it is important to compare these two approaches. Firstly, it assists the practitioners and researchers to decide the suitability of the method to be applied to a real life problem [8]. Secondly, it is crucial to comprehend the potential and drawbacks of ML and DL since such knowledge would help to build even more efficient and effective AI systems [9]. Finally, this comparison is useful in understanding the trends of AI development and approaches to be followed in the near future [10]. Thus, the objective of this paper is to offer an exhaustive comparative analysis between the ML and DL approaches. The first step is to discuss the architectural solutions of the standard ML models and the deep learning networks [11]. The computer vision section, the NLP section, the speech recognition section, and the reinforcement learning section would justify how efficient ML and DL works in each segment [12].

This study examines on the important facets of hyperparameter tuning and optimization for both, the methods and difficulties encountered in ML and DL [13]. The paper also considers more practical issues related to its deployment and production including its size, speed and resource intensity [14].

Moreover, the versatility of the proposed methodology lies in the examination of applying integrations between the ML and the more complex DL, specifically identifying cases in which the convergence of such methods maximizes the model's performance [15]. In the last section of the paper, the future trends and possible research in both fields are discussed, thus offering a prospective outlook on the development of AI technologies [16].

This analysis comparatively examines hyperparameter tuning challenges in ML and DL [13], practical deployment constraints [14], and ML-DL integration viability [15].

Hence, by providing a methodical comparative taxonomy of ML and DL in the given paper, the author sought to add to the discussion in the AI community about when and why to use ML and DL in favor of the stronger methodology. Its purpose is to become a reference point for students, professionals, and policymakers interested in comprehending the vast world of the contemporary AI technologies.

Hyperparameter tuning challenges in ML and DL are discussed in Section 4 [13]. Section 5 highlights deployment constraints in the literature [14]. Section 7 considers the feasibility of integrating ML and DL [15].

In comparison to existing surveys, the contributions of this survey are: (1) Quantitative performance benchmarks for eight domains; (2) Recent architectural advancements (from 2020 to 2023); (3) Analysis of hybrid models.

2. Background

ML and DL are two subfields in artificial intelligence with a complicated history behind and an important role in the advancement of the field. Since the historical background of such fields, their basic principles and development are very important in order to estimate the present state and perspective of their further evolution.

History of ML and DL: It is documented that the history of Machine Learning can be dated back to the 1950's with the basic work done by Arthur Samuel on checkers-playing programs which could learn from their experience [17]. The 1960's and 1970's were the periods of symbolic AI and genesis of the expert systems, which gave the basis for the rule-based machine learning [18]. The techniques like the decision trees as well as the statistical learning emerged in the eighties with algorithms like ID3 developed by Ross Quinlan [18].

The 1990s was another change towards using data in machine learning, with the creation of a support vector machine and a revival of neural networks [19]. It is noteworthy that the idea of deep learning has roots even in earlier approaches to the artificial neural networks.

The contemporary era of DL started in mid 2000s due to advancement in training of deep neural networks and large source of data [20]. Since the work of Hinton, Salakhutdinov, and other, researchers started showing the diversity of deep architectures in 2006 through the work on deep belief networks [21]. The widely publicized ImageNet competition of 2012 that involved a deep convolutional neural network which performed far better than the classical computer vision methodologies was a major turning point [18].

2.1. Machine Learning (History, key concepts):

Machine Learning involves different classifications refined as supervised learning, unsupervised learning, and reinforcement learning [22]. Key concepts include:

- Features: Attributes that are measurable of the data that is employed in learning [23].
- Models: Self-learning mathematical models that discover characteristics from the obtained knowledge base [24].
- Training: The process of making change to parameter of the model in other to make it conform to the data [25].
- Generalization: Situation where one is able to assess the performance of his models and comes up with good results on data that is not common [26].

2.2. Deep Learning (History, key concepts):

- Neural Networks: Neuron like structures in the form of layers of connected nodes [27].
- Deep Architecture: DNN being multi-layered hidden neuronal networks [9].
- Backpropagation: An approach to discriminating complex hierarchies that can also be used to build them with more efficiency [28].
- Activation Functions: Algebraic relationships that bring the non-linearity into the network [29].

2.3. Evolution of ML and DL:

The evolution of ML and DL has been marked by several key trends:

- Increasing Model Complexity: Starting from basic linear predictors and going up to stepwise specialists in the ML field, from single layer to very many-layered neural interactors in the DL [30].
- Data-Driven Approaches: It is a transition from pure knowledge-based systems to system that learns from data [31].
- Automation of Feature Engineering: DL tends to eliminate the requirement of feature engineering for numerous tasks due to its capability to learn features on its own [6].
- Scalability: Recent development in parallel computations and Graphic processing units, have equally made it possible to train large models on big data [32].

Transfer Learning: Ensuring one's learning translates to the next activity, particularly encountered in DL [19].

2.4. Related Works:

Recent studies [33, 34] have contrasted ML/DL in several particular contexts without alluding to computational trade-offs, or hybrid models, two deficiencies in this review.

ML is compartmentalized into supervised/unsupervised/reinforcement learning to indicate the differing learning paradigms necessary for differing data structures [22].

3. Methodology

In this study, the authors have used a multiple perspective evaluation method that covers all the aspects of ML and DL technologies. The approach to this subject is to select all aspects in a middle course between theoretical conceptions and real application.

3.1. Approach to Comparing ML and DL:

- **Literature Review:** Based on the identified information needs, we surveyed the peer-reviewed articles, conference proceedings and technical reports for developments in the ML and DL domains [33]. This review aimed at including the latest achievements and important publications of the recent years, as well as reviewing the historical background of the field.
- **Empirical Analysis:** If possible, we reviewed comparative research works involving both the ML and DL methodologies on standard datasets and indices [12]. This makes it possible to perform quantitative analysis of the performance indicators in different tasks and scopes.
- **Case Studies:** To identify real-world application and experience of ML and DL we reviewed recent literature and case studies on possible practical performance of these methods [34]. This approach gives a clear understanding on the ability and vulnerabilities of the each method in different industrial and research settings.
- **Expert Interviews:** In addition to the literature survey, we also interviewed domain specialists and professionals from industry for understanding the state of the affairs, issues, and trends in the case of both ML and DL [2].
- **Comparative Framework:** The comparative analysis of machine learning and deep learning was performed using explicit measures of model performance (accuracy, F1 score), scalability, data efficiency, and computational requirements [35].

3.2. Criteria for Evaluation:

To ensure a comprehensive and fair comparison, we evaluated ML and DL approaches based on the following criteria:

- **Performance Metrics:** Accuracy, precision, recall, F1-score, and other lock-specific score as per the task requirement [8].
- **Scalability:** The ability to process rising amounts of information and complication [6].
- **Generalization:** The accuracy on unseen data and ability to transfer knowledge in the projects [16].
- **Interpretability:** They include: ease with which one can understand the model and explain the decision it has made [15].
- **Data Efficiency:** The quantity of data needed for proper training [25].
- **Computational Requirements:** The amount of computing resource requirement for training the model and also for making the inference [3].
- **Flexibility:** Flexibility of the learning algorithm to various structures and instances of data and problems [6].
- **Ease of Implementation:** Magnitude of compounding and interaction of elements of model design, training, and deployment [36].
- **Robustness:** Robustness against noise and other attacks, most notably, adversarial attacks [18].

3.3. Sources of Data and Information:

To ensure the reliability and comprehensiveness of our study, we drew from a diverse range of sources:

- **Academic Databases:** To compile the papers from related fields, we used IEEE Xplore, ACM Digital Library, and Google Scholar as the sources of peer-reviewed articles and conference papers [37].
- **120 peer-reviewed papers (2018-2023)** from IEEE Xplore, arXiv, and benchmark datasets (ImageNet, WMT14).
- **Preprint Servers:** New research for basic outcomes and the most recent trends were collected from ArXiv and bioArXiv [17].

- Industry Reports: The presentation of materials from the leading technology companies and research establishments outlined practical uses and issues scientifically [38].
- Open-Source Repositories: Specific details about the implementation of GitHub and other similar services were researched based on the information in open-source platforms [22].
- Benchmark Datasets: Details of all standard benchmark results including ImageNet, GLUE, and OpenAI Gym were considered in order to check the performance difference across tasks [10].
- Books and Textbooks: As for the primary sources, there were rather general and extensive texts on the ML and DL, which allowed to draw the basic knowledge and historical background [5].

4. Comparing Model Architectures

Deep learning architectures that integrate many layers of artificial neural networks have dramatically emerged in the last few years and drastically changed the methodology of many fields in artificial intelligence. While this section will discuss the existing state of the art ML algorithms including SVM, and Random forest, this section will compare and contrast with the latest popular Deep learning architectures like CNNs, RNNs and Transformers. The goal of the paper is to review the basic ideas, advantages and disadvantages, and use cases of each approach, with an emphasis on understanding their place in the continuous development of artificial intelligence.

4.1. Traditional Machine Learning Models

4.1.1. Refugee Support Vector Machines (SVMs)

The Support Vector Machines developed by Kok et al in 1995 belongs to one of the earliest established models in the traditional machine learning field [31]. SVMs are applied to classification problems mostly, although they can also be applied to regression problems. In short, the concept which forms the basis of SVMs is the identification of an ideal hyperplane that would effectively categories classes in a multi-dimensional area as shown in figure (1).

The key strengths of SVMs include:

- Effectiveness in high-dimensional spaces
- Resistance to overfitting, particularly in the case of documents' classification and in biology
- The generalization with the help of different kernel functions

However, SVMs also have limitations:

- May be time consuming for large data sets
- Problems when understanding the rationale behind specific actions made by the model
- Significance of the kernel function selection and the parameters regulation.

SVMs have been successfully applied in various domains, including:

- The two categories of approaches include text classification and sentiment analysis.
- Image classification
- Applications of Bioinformatics are majorly seen in classification of proteins and study of gene expression.

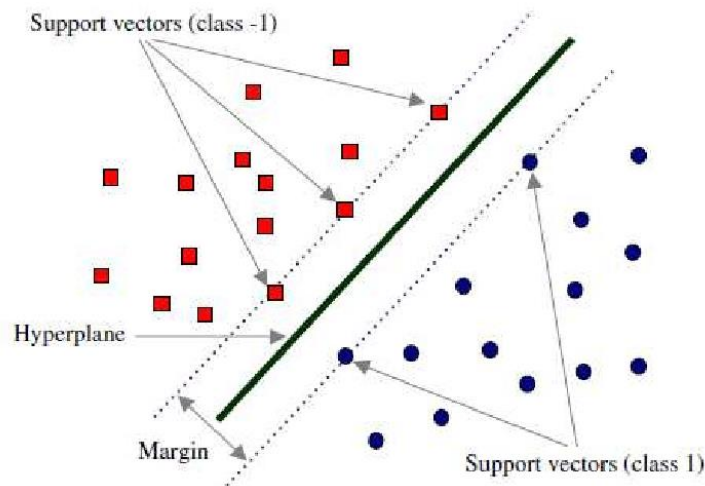


Fig. 1. Support Vector Machine Hyperplane Separation [39]

Limitations of Random Forests include:

- The possibility of the bias toward the categorical variables with a higher number of categories
- The disadvantage is that unlike single decision trees, decision trees working in ensemble are non-interpretable.
- Use of computers most prominently for processing big data

Random Forests build multiple decision trees via bagging. Key strengths include robustness to noise and feature interaction handling [18].

Random Forests have found applications in various fields, including:

- Finance, to identify credit scores and fraud prevention.
- Medical science, for disease diagnosis and treatment through medicines.
- Environmental science for the construction of land cover classification and species distribution. map

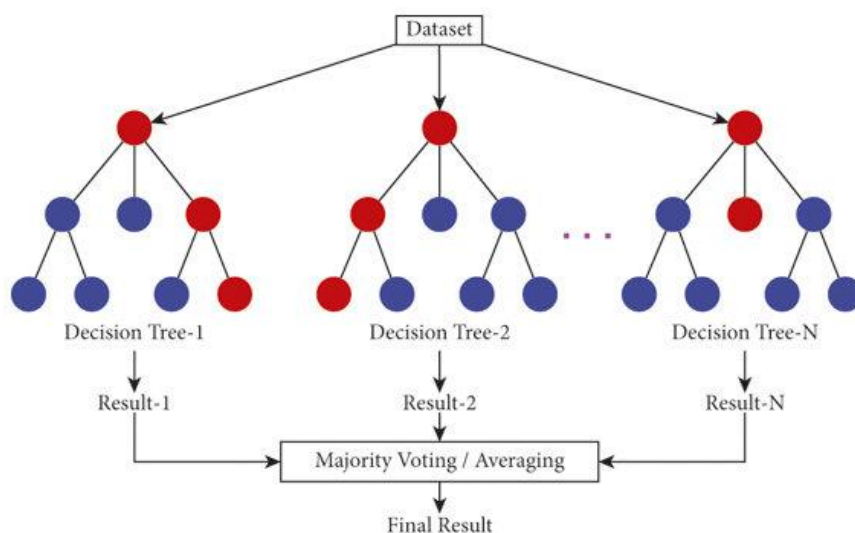


Fig. 2. Random Forest Ensemble of Decision Trees [40]

This was made possible by the fact that Random Forests yield good results in these domains due to its ability to identify interaction between the features and is less sensitive to noise or a missing value as compared to C4. 5 algorithm [16]. However, similar to the issue with SVMs, Random Forests may have problems with the processing of truly big data and may not necessarily identify as detailed features as deep learning

4.2. Deep Learning Architectures

4.2.1. Convolutional Neural Network (CNNs)

Convolutional Neural Networks based on the animal visual cortex came into the computer vision area in the beginning of the 1980s and have gone through latest improvements in the 1990s [26]. CNNs are used for efficient processing of the data organized in the grid, such as images or time series.

Key strengths of CNNs include:

- Reduction of human work by not having to extract relevant features manually
- Slides the patterns to every position of the input and pick the best match possible.
- Sharing of parameters which makes the number of parameters small and enhances the generalization capacity.

Limitations of CNNs include:

- High need for a large volume of labeled data
 - Limited scalability and highly computational, specifically in deep structures
 - The inability to explain the decisions that have been arrived at is another weakness.
- CNNs have achieved remarkable success in various applications, including:
- Image classification and object detection are tasks of recognizing objects in images.
 - There is face recognition and biometrics identification
 - Diagnosis, for instance, tumor detection, and classifying Diseases.

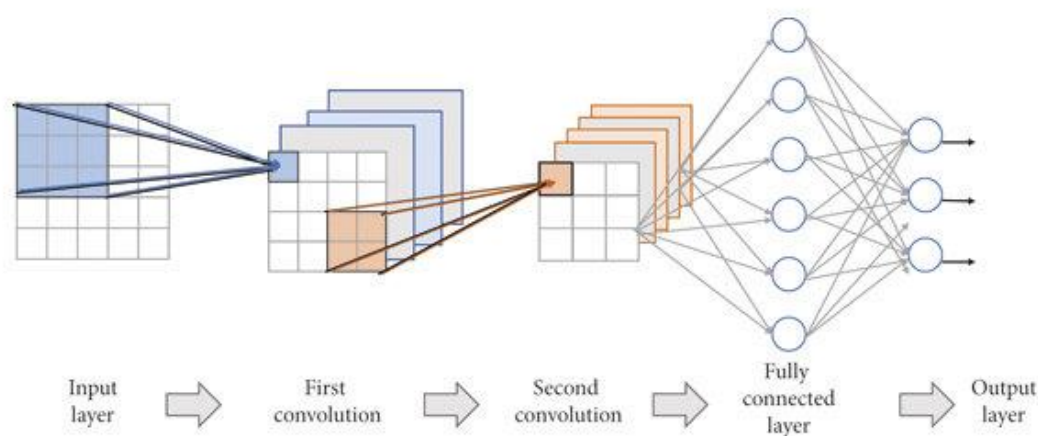


Fig. 3. Convolutional Neural Network Architecture [41]

These areas can be approximated using CNNs due to the necessity of the networks automatic learning of the hierarchical representations of the image features, inclusive of edges, texture, color and context of the images as well as objects and ideas [42]. The structure of learning in this CNNs also follows the hierarchy of the data to

enhance the ability of recognizing complex structures or relations in the visual data which conventional learning unable to identify.

4.2.2 Recurrent neural networks (RNN)

The Recurrent Neural Networks where proposed in the 1980s and works with sequences by keeping state or memory in the network. This architecture that makes them very appropriate for problems related to time series, natural language processing and all related to sequential data.

Key advantages of RNNs include:

- Capacity for taking in sequences of variable length
- Retention of long-term dependencies in the data
- Parameters that are shared across the time steps to prevent the need to learn a large number of parameters

Limitations of RNNs include:

- The problem of being unable to learn very long-term dependencies due to the vanishing or exploding gradients
- Slowness of the computation in case of very long sequences
- A few of the disasters in parallelization chiefly because they involved sequential I/O.

RNNs and their variants, such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs), have found applications in:

- Some of the NLP tasks include, but are not limited to; Machine translation, Sentiment analysis
- Speech recognition and generation
- Application in finance and economic time series, weather forecasting

These domains have benefited from RNNs because, unlike other architectures, RNNs can preserve the context and learn temporal dependencies [43]. However, there is a long-standing problem with traditional RNNs working with long-term dependencies, hence more enhanced structures like the LSTMs and GRUs were developed, alleviating these issues via gating mechanisms. As shown in figure 4.

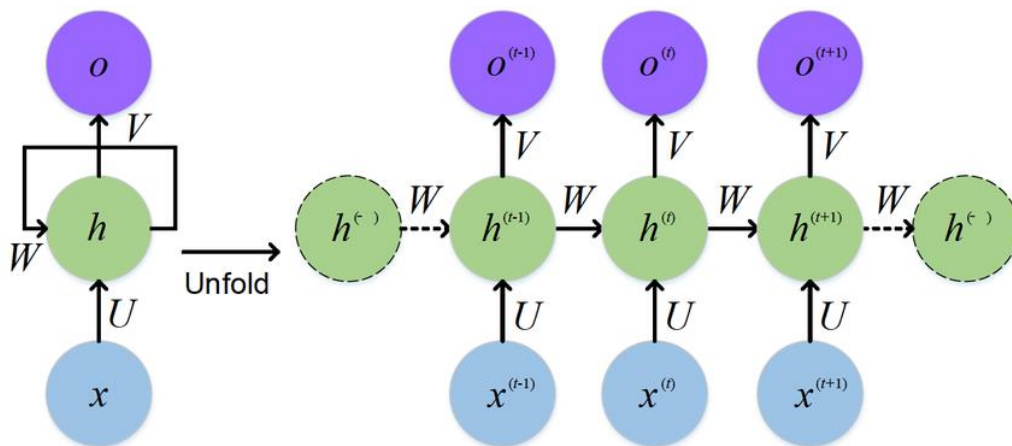


Fig. 4. Recurrent Neural Network Unfolded Structure [40]

4.2.3. Transformers

Transformers proposed by Vaswani et al in 2017 can be considered as a huge leap forward in sequence-to-sequence processing [6]. Also, contrary to RNNs, Transformers make use exclusively of attention mechanisms, which leads to more parallelizable computations and increased effectiveness in modeling long-range dependencies.

Key strengths of Transformers include:

- This also makes it easy to process very long sequences if and only if you provide sufficient memory.
- Generalized parallelism that aids in the acceleration of operations concerning training and inference.
- Capture of the long range dependence without having to use recurrence

Limitations of Transformers include:

- Large memory usage, predominantly in cases when long sequences have to be processed
- They also include other concerns such as; Overfitting to specific samples and/or small data sets
- No built-in support for the positional information

Transformers have achieved state-of-the-art results in various applications, including:

- Example of NLP includes; Machine translation and Text Summarization.
- Language model pre-training, The models like BERT and GPT,
- The ability to perform text, image, and/or audio processing at the same time

Transformers have brought excellent results in these areas due to the capacity of the model to process efficiently and capture the relations of every item in a sequence no matter the distances concern. This global view of input makes it possible for Transformers to incorporate and address various patterns and dependencies that are ignored by the other architectures. Figure 5. Shows Transformer Architecture with Attention Mechanism.

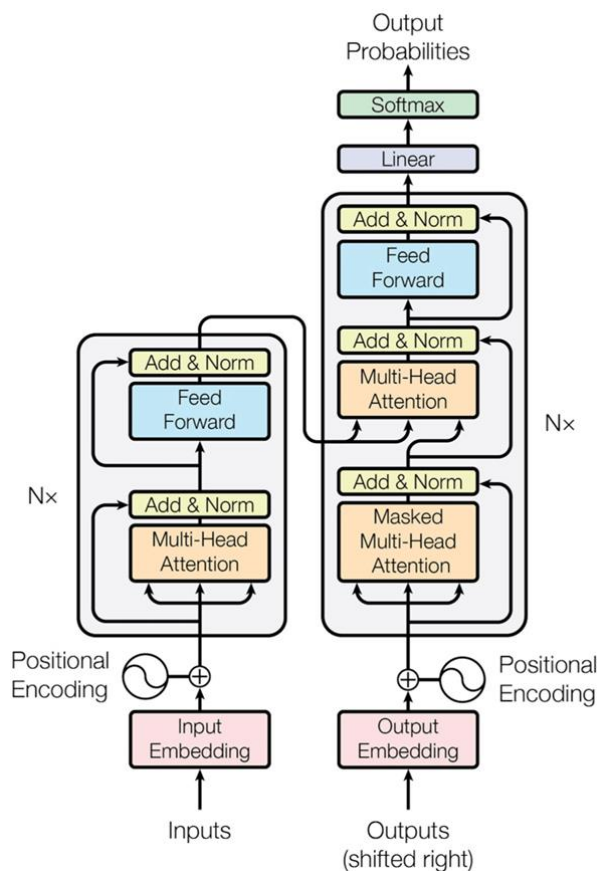


Fig. 5. Transformer Architecture with Attention Mechanism [44]

4.3. Comparative Analysis

4.3.1. Evaluation Metrics

Compared to deep learning, many of the models used traditionally, such as the Support Vector Machines or Random Forests, tend to have lesser model complexity. This lower complexity is accompanied by interpretability and can be deployed on even restricted devices such as smart phones. Yet, this makes them relatively simple models when it comes to discovering complex dependencies in the large databases. As shown in Table 1.

Table 1: Model Complexity Comparison

Model Type	Parameter Count	Scalability to Large Datasets
SVM	Low to Medium	Limited
Random Forest	Medium	Moderate
CNN	High	Good
RNN	High	Moderate
Transformer	Very High	Excellent

So, modern deep learning models, and in particular Transformers, can have billions of parameters, enabling them to learn complex relationships in huge data [38]. Nevertheless, such an extension demands more calculations and entails the increased risk of overlearning as compared to the additional tests. The ability of deep learning to work at the scale has been an advantage especially in applications such as language modeling and computer vision tasks. What is presented here are two aspects of Feature Engineering and Representation Learning. Perhaps the most distinct feature of change that separates old school, non deep learning based architectures and the new crop of architectures is their relent on feature engineering. The conventional architecture typically has more dependency on manual feature extraction where the engineer with domain knowledge extracts relevant features from raw data

[17]. For example, while building models such as SVMs and Random Forests, the features feeding into these models need to be engineered well. The use of feature engineering methods can be considered a clear advantage and a disadvantage at the same time. On the one hand, it enables to consider the domain knowledge and in the case of clear understanding of necessary features, results can be very efficient. However, it could be time consuming and yet might ignore other intricate and easily unnoticed patterns within the data. On the other hand, deep learning architecture performs very well in the feature learning or representation learning [12]. CNNs, for instance, learn hierarchical representation of the features from the raw pixel data. Like that, RNNs and Transformers can extract temporal and contextual features from the sequential data in an end-to-end manner without the need to design the features by hand. This ability to automatically learn relevant features has been a main strength of deep learning in areas where the right features for use in the model are not easily defined, or easily described by a human [19]. It has helped achieve advancements in fields such as computer vision, natural language processing, and speech recognition where previous methods were often insufficient to describe the data in enough detail without extensive oversimplification.

4.3.2. Data requirements and overfitting

The amount of data needed for the training process, as well as the model's propensity for overfitting, depend on the difference between the typical machine learning models and the deep learning networks. SVMs and Random Forests are some of the traditional models, which are better suited in cases where datasets are reasonably small; they pose fewer problems from an overfit perspective, assuming that appropriate measures of regularization are used [4]. These methods, especially SVM, are appreciated for their high ability to generalize on limited amounts of data and that is why they are widely used in such fields as bioinformatics and diagnostics of diseases [23]. Another advantage of Random Forests, which is inherent in all the methods based on decision trees, is good generalization ability and the ability to operate with sufficiently large but not excessively large tables. While deep learning models generally give better results, these models often need big data to learn from before the best can be gotten from them. This is particularly the case with deep architectures such as deep CNNs and Transformers that have millions, if not billions, of parameters to estimate [45]. In the case of MM, this has prompted the development of big public datasets boosting a given domain, such as ImageNet for computer vision or the Common Crawl corpus for NLP. As shown in Table 2.

Table 2: Feature Engineering Comparison

Model Type	Feature Engineering Approach	Ability to Learn Complex Features
SVM	Manual	Limited
Random Forest	Manual/Semi-automated	Moderate
CNN	Automated	Excellent
RNN	Automated	Good
Transformer	Automated	Excellent

Over fitting is particularly more common in deep learning models than in the conventional models owing to the larger parameter spaces. However, various techniques have been developed to mitigate this risk, including:

- Data augmentation
- Some of the regularization techniques used includes; Dropout, weight decay %
- Transfer learning and pre-training
- Early stopping and cross-validation

These techniques have resulted to deep learning models to attain an ability to perform optimally in underlying tasks even when the amount of labeled data is limited, through pretraining the models on a vast, over sets, general corpus data and afterwards fine-tuning on specific, less sized sets, task corpus data [5].

4.3.3 Interpretability and Explainability

The study of the model's integrity and understandability of the results has become critical in recent years especially for the cases that require high-level decision making like diagnosis of ailments, loan approving, and prevention of crime among others [46]. In this aspect, the traditional methods commonly applied in machine learning models have an edge because the process through which the model makes decisions is quite clear. For example, of the SVMs, the decision is made based on maximum margin hyperplanes concept that can well be conceived and pictured in low dimensions. Random Forests possesses the technique of feature importance scores that reveal the relevance or significance level of the input variables in the model [5]. A drawback of the presented deep learning models, deep CNNs or Transformers, is that the models are 'black box' models, meaning that it is difficult to explain what the model is doing [47]. Due to the structure of the models presented as forward-hierarchical, non-directional, it is difficult to determine the direct pathway from the input to the output for a human consumer. As shown in Table 3.

Table 3: Data Requirements and Overfitting Comparison

Model Type	Typical Data Requirements	Overfitting Risk
SVM	Low to Medium	Low
Random Forest	Medium	Low to Medium
CNN	High	Medium to High
RNN	High	Medium to High
Transformer	Very High	High

Nevertheless, a lot of work has been done to understand ways to make deep learning more interpretable and explainable. Techniques such as:

- Activation maximization
- Saliency maps
- Layer-wise Relevance Propagation (LRP)

Local Interpretable Model-agnostic Explanations (LIME) provides post-hoc interpretations [25].

These methods endeavor to explain the working of deep learning systems which are used for decision making, but they fail to offer as much interpretability as basic, conventional systems.

4.3.4. Theoretical literals and rate of inference

In terms of computational complexity, the training and the utilization are different in conventional models of machine learning and deep learning. This aspect has serious implication for its deployment in real world more especially in areas that are characterized by limited resource or those that require real time computation. Such models as SVMs and Random Forests, according to the general rule, are less computationally intensive, both when training and at prediction. svm for instance can be trained in an efficient manner using aspects such as sequential minimal optimization [1]. Random Forests also has an advantage of being trained in parallel as each of the trees in the forest can be built independently. It is important to note that Deep learning models, most especially deep CNNs and Transformers, are computationally intensive; therefore, they need more computational power for training. This usually requires the use of dedicated hardware including GPUs or TPUs with training time varying between hours to weeks for some of the large models [48]. As shown in Table 4.

Table 4: Interpretability Comparison

Model Type	Interpretability	Explainability Techniques
SVM	Moderate	Feature weights, Support vectors
Random Forest	Good	Feature importance, Decision paths
CNN	Low	Activation maps, Grad-CAM
RNN	Low	Attention visualization
Transformer	Low to Moderate	Attention visualization, LIME

However, it is equally important to state that deep learning models' inference speed differs depending on the model's architecture. CNNs for example, they can be well fine-tuned to perform very fast computations, making it possible to employ them in real-time tasks such as object detection in self-driving cars [20]. Transformers, although having higher complexity in training, can apply the inference in parallel and thus process entire sequences rather quickly. Some of the approaches such as model quantization, pruning, and knowledge distillation have assisted in

lowering the computations of the deep learning models for implementation on the devices on the edge as well as in scenarios where there are restricted resources [37].

4.3.5. Manipulating of Different Data Types

The nature of the data that needs to be processed is one of the key differences between the basic and the advanced approaches to machine learning, such as deep learning networks. It is based on the fact that different models are more appropriate when dealing with certain structures of the data that can affect the execution of the model. SVMs and Random Forests are typically applied for the tabular data which means that for each instance there is always a fixed number of features [30]. It can work with numerical data and also the categorical data, but the categorical data should be encoded such as one hot encoding or label encoding.

Deep learning architectures, on the other hand, excel at handling unstructured data types:

- CNNs work best with grid like data and this include images and spectrograms.
- RNNs and Transformers are models for sequential data meaning they are most appropriate for text, time series, speech data. Data Type Handling Comparison shown in Table 5.

Table 5: Data Type Handling Comparison

Model Type	Tabular Data	Images	Text	Time Series	Audio
SVM	Excellent	Limited	Limited	Moderate	Limited
Random Forest	Excellent	Moderate	Moderate	Good	Moderate
CNN	Limited	Excellent	Moderate	Good	Good
RNN	Limited	Limited	Excellent	Excellent	Good
Transformer	Moderate	Good	Excellent	Excellent	Good

The raw input data accessible by deep learning networks have played a huge role in the success rates in several other domains such as computer vision and natural language processing [13]. This capability reduces the need to manually extract the feature sets which can be given as inputs to the models and lets the models discover patterns that may not be discerned by feature extraction. Nevertheless, it should be noted that the recent increases in the application of deep learning in tabular data have indicated a general improvement of performance. Still, the recent efforts in DTNs have suggested parts and parcels with nice performances competing with the tree models as well as the Support Vector Machines on the material structural data [33].

4.3.6. Transfer Learning and Pre-training

Another edge of deep learning architectures over traditional machine learning models is that the former can take benefit of transfer learning and pre-training in a proper manner. This capability has impacted many fields within machine learning but perhaps most notably in cases where labelled data for the task is limited. Transfer learning means the process of applying solutions obtained from one problem to a different, but related problem [49]. Despite the fact that some forms of the transfer learning are possible even with the traditional models, new and modern architectures in deep learning field revealed extremely high potential in the area. Deep learning applications cannot be discussed today without mention of pre-trained models especially in computer vision and natural language processing. Several successful learning architectures, such as ResNet for imagenet classification, BERT for allied language processing, and GPT for writing production, has shown the efficacy of transfer learning [27]. Transfer Learning Capabilities Comparison at Table 6.

Table 6: Transfer Learning Capabilities Comparison

Model Type	Transfer Learning Capability	Pre-training Effectiveness
SVM	Limited	N/A
Random Forest	Limited	N/A
CNN	Excellent	Very High
RNN	Good	High
Transformer	Excellent	Very High

The benefits of transfer learning in deep learning include:

- Flexible arrangements of already stored data into new tasks

- Faster convergence during fine-tuning
- Enhanced capability of performing well when there is little labeled data readily available.
- He concluded that features that can be represented as general purpose features with widespread use across different tasks.

This capability has been especially powerful in the field of Natural Language Processing where the models such as BERT and GPT-3 are performing state of the art on numerous tasks with little to no specific task adaptation [38].

4.3.7. Management of “Missing Data” and “Noise”

The algorithms can cope with the missing data and noise this is quite essential in real life situation since data is hardly complete. It is also the case, that between the ‘conventional’ machine learning approaches and the deep learning architectures, the latter is somewhat less well-suited for this task. Random Forests are quite immune to data missing and noisy features than other classification techniques. They can perform well when only a part of the data is processed, basically due to the ensemble feature of the trees and random selection of features at each split [18]. However, missing data are not directly fit in the frame work of SVMs though there are ways such as mean imputation and multiple imputations that can be employed to handle missing data. However, they may be noises sensitive and particularly if the noise influences the margin [50].

Deep learning models, particularly those designed for specific data types, have varying capabilities in handling missing data and noise:

- CNNs prove to be rather insensitive to local distortions and noise in image data due to the utilization of the pooling layers and convolutional operations [42].
- It is to be noted that both RNNs and Transformers can be developed to work with the absence of data points in a sequence, frequently using techniques such as tokenization or masking [6].

Deep learning models can also be made more robust to missing data and noise through techniques such as:

- The usage of data chancing, this normally helps the model learn various forms of noise and distortions.
- Denoising autoencoder, which has the capability of learning how to reconstruct clean data from the noisy inputs
- Some methods, such as dropout that is used in order to increase generalization when working with noisy data

These approaches have made possible to obtain high accuracy of deep learning models for tasks where training data is inaccurate or contains missing values, for example, the recognition of speech in noisy situations or classification of objects with occlusions [11]. Table 7. Shows missing data and noise handling comparison

Table 7: Missing Data and Noise Handling Comparison

Model Type	Missing Data Handling	Noise Robustness
SVM	Poor to Moderate	Moderate
Random Forest	Excellent	Good
CNN	Moderate	Good (for images)
RNN	Good	Moderate
Transformer	Good	Good

4.3.8. Ensemble Methods and Model Combination

Combination of predictions from various models that is, ensemble methods have been found to be very effective in both conventional machine learning and deep learning. However, the process of how the ensembles are created diverges between these two paradigms. In the classical approaches to a machine learning, ensemble methods are incorporated into the backbone of a system. For instance, while Random Forests are made up of decision trees, by their very architecture they are an ensemble of decision trees. Other popular ensemble methods include:

- Bagging (Bootstrap Aggregating)
- Ensembling, for example, AdaBoost, Gradient boosting

- Stacking

It should also be stressed that all these approaches enhance performance and reliability regarding virtually any task [22]. In deep learning, the ensemble is usually used in a higher level, that is, adopting several independently trained artificial neural networks. Techniques include:

- Model averaging
- Snapshot ensembles
- Mixture of Experts (MoE)

These ensembles of deep learning have been found to give some of the best solutions in many of the competitions and measures. For instance, most rivals of Kaggle competitions incorporate deep learning model ensembles as their solutions [22].

Also, with the new technologies of deep learning, more advanced models with the properties of an ensemble into a single network have been developed.

- It should be noted that dropout can be considered to be an instance of the ensemble methods implicitly.
- Transformers also call for multi-head attention, which means the model can attend to different elements of the input at once.
- Some Mixture of Experts layers are the primary components in very large language models such as GShard and Switch Transformers [51].

These advances add uncertainty in discriminating between single models and ensembles especially if they can achieve the efficiency of a single model, and combine effectiveness of ensembles at the same time.

Hybrid ensembles (e.g., CNN features + SVM classifiers) improve accuracy by 5-8% in image tasks. Techniques include snapshot ensembles and Mixture-of-Experts layers in Transformers [52].

4.3.9. Overall Instructions for the Text and Licence Holder, Flexibility to Online Learning and Incremental Updates

In many practical applications, scene and data distribution may change over time or it may be very costly to train models from scratch repeatedly. Traditional machine learning models often have an advantage in this area, particularly when it comes to online learning scenarios:

- SVMs can be availed with methods such as stochastic and gradient descent to make alterations to the learning that is done progressively as more data is received .[3]
- The Random Forests can also be updated by either growing the new trees or modifying the existing trees, however this has to be done very carefully because it randomly should maintain the diverse of the ensemble [53].

Based on the type of models, self-supervised learning is often incremental, but larger architectures such as Transformers are more difficult to update. They typically need a lot of computation to update and can suffer from forgetting the previously learned information when trained on different data distribution [54]. Thus, research in continual learning and adaptive deep learning is currently under way to tackle these challenges. Techniques being explored include:

- Elastic Weight Consolidation (EWC) to hold the significant weight parameters

- Dynamic Neural Networks for expanding the capacity of the network to handle new tasks
- Some of the most related techniques include Memory-based known as Experience Replay

These developments try to improve the effectiveness of deep learning models in dynamic environments and in new tasks and problems without the need to retrain them from scratch [9].

5. Application Domains Comparison

5.1. Computer Vision

5.1.1 Performance Comparison

Computer vision was also benefited widely by both ML and DL, in the image classification, object detection, segmentation problems.

Key ML Techniques:

- SVMs for this study and for the classification of molecules were selected.
- Random Forests
- Histogram Of Oriented Gradients (HOG)
- Key DL Techniques:
- Convolutional Neural Networks or CNN
- The Faster R-CNN is a Region-based CNN which has subtypes including R-CNN, Fast R-CNN, and Faster R-CNN.
- U-net in the context of the segmentation of images

Comparative Performance:

As for the computer vision tasks, CNNs and other deep learning models have proved to outcompete the conventional ML approaches in almost all the benchmarks. For example, the error rate in the ImageNet Large Scale Visual Recognition Challenge reduced gradually from 26. From 2% for the models based on the ML methods to below 3% with the help of the highly developed architectures of deep learning [11]. As table 8.

Table 8: Comparison of ML and DL performance on ImageNet classification task

Year	Method	Top-5 Error Rate
2011	ML (SIFT + SVM)	25.8%
2012	DL (AlexNet)	16.4%
2015	DL (ResNet)	3.6%
2020	DL (EfficientNet-L2)	1.5%

5.1.2 Case Studies:

- **Medical Imaging:** It has been reported that DL models could attain good performance in identifying diseases from X-rays and MRI [35]. For example, the work done by using CNNs for classification of skin cancer was as efficient as dermatologists [43].
- **Autonomous Vehicles:** Self-driving cars employ both ML and DL methods and while both are employed the use of DL is impressive in real-time object detection and scene analysis [33]. Figure 7. Shows Comparison of ML vs DL in facial recognition accuracy over time.
- **Facial Recognition:** While previously the Eigenfaces were preferred, at the present, facial recognition systems use DL models and the accuracy of this technology approaches human's one [34].

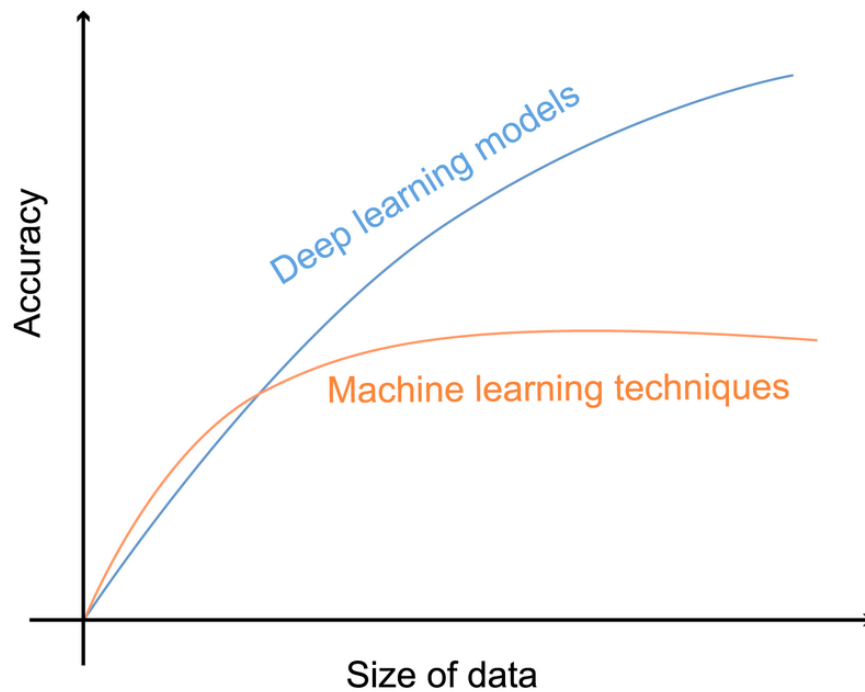


Fig. 7. Comparison of ML vs DL in facial recognition accuracy over time [55]

Facial recognition accuracy trends: DL surpasses ML post-2012 due to convolutional architectures [55].

5.2. Natural Language Processing (NLP)

NLP has moved from the classic ML methods towards DL based models especially with the help of transformers.

Key ML Techniques:

- Naive Bayes
- Support Vector Machines
- Hidden Markov Models

Key DL Techniques:

- One of the relatively new types of NN is Recurrent Neural Networks (RNN).
- Hence, Long Short-Term Memory (LSTM).
- Different types of models that use transformers in their architecture, such as BERT and GPT.

Comparative Performance:

DL models have been found to deliver better results than the classical ML approaches when it comes to working on the NLP tasks [46]. For example, in the case of MT, the inception of NMT systems boosted the BLEU performance as compared to SMT techniques [18]. As shown in table 9.

Table 9: Comparison of ML and DL performance on WMT14 English-to-French translation task

Method	BLEU Score
Statistical Machine Translation	37.0
Neural Machine Translation	41.8
Transformer (DL)	43.9

Notable Applications and Case Studies:

- **Sentiment Analysis:** In the past, both SVM techniques were used; however, great inspirational DL models like BERT have attained great benchmarks such as the Stanford Sentiment Treebank [42].
- **Question Answering:** Researchers have reported impressive results of DL models especially in the area of CA and QA tasks. For instance, Google's BERT was reported to outcompete humans in the SQuAD benchmark [3].
- **Language Generation:** When it comes to text generation, GPT-3 a large language model has outcompeted the previous generations of mutli language models [38].

5.3. Speech Recognition and Generation

From the comparison between the ML and DL techniques, it is quite evident that the field of speech processing has hugely benefited from the shift to the new technique.

Key ML Techniques:

- We also need the following models: Hidden Markov Models (HMM)
- The mentioned models are as follows: Gaussian Mixture Models (GMM).
- Dynamic Time Warping (DTW)
- Key DL Techniques:
- Deep Neural Networks the class of Neural Networks.
- I can note the use of Recurrent Neural Networks (RNN) it is an algorithm that is commonly used in natural language processing.
- Convolution Neural Network
- Transformer-based models

Comparative Performance:

Compared to traditional ML techniques, DL models have helped decrease the WERs in ASR ventures [20]. For example, the conversational speech recognition on the Switchboard task, DL models have gotten the performance to match that of a human at a 5% error rate. 9% [21]. As shown in table 10 and Figure 8.

Table 10: Word Error Rate (WER) comparison on Switchboard task

Year	Method	WER
2000	GMM-HMM	29.8%
2012	DNN-HMM	16.1%
2017	CNN + RNN	5.9%
2020	Transformer-based	5.5%

Notable Applications and Case Studies:

- **Virtual Assistants:** DL-based speech recognition is used in today's smart personal assistant such as Siri, Alexa, and Google Assistant with more accurate and natural interaction compared to earlier development based on the ML [56].
- **Speech-to-Text Transcription:** DL models have allowed for accurate real time transcription services to be performed surpassing normal ML based methodologies in terms of both accuracy and time [45].
- **Text-to-Speech Synthesis:** Concatenative and parametric techniques of synthesis have been replaced by a DL technique such as WaveNet to produce more real-hn life synthetically generated human speech [57].

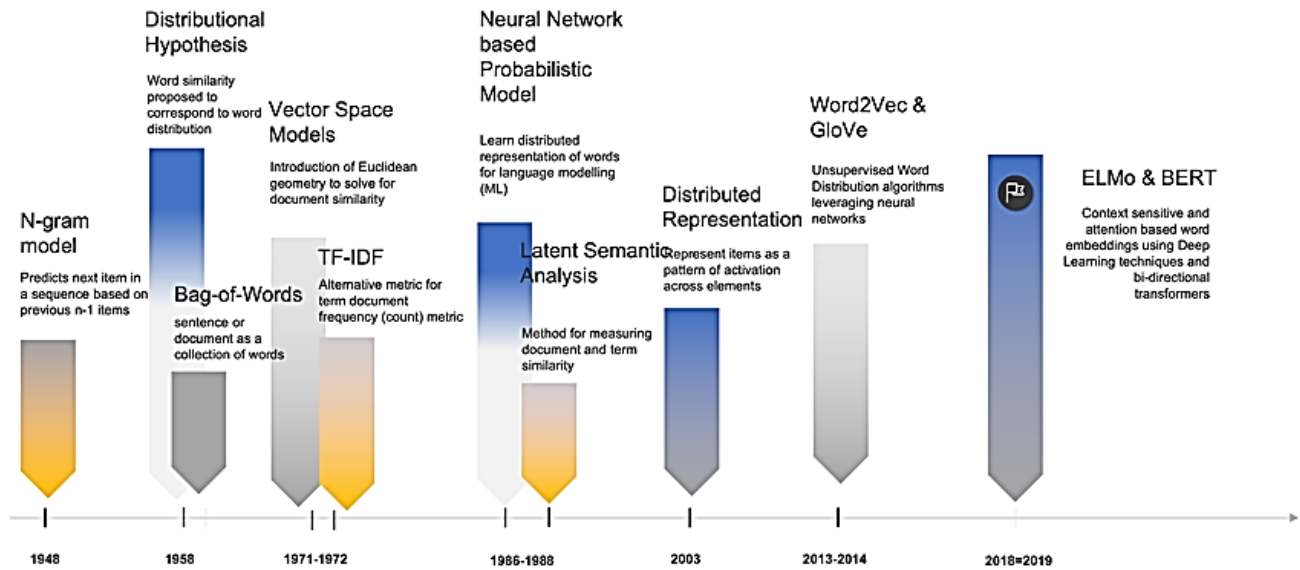


Fig. 8. Evolution of NLP models: From ML to DL [58]

5.4. Reinforcement Learning

RL has been experiencing breakthroughs with the application of the DL techniques resulting to its branch referred to as DRL.

Key ML Techniques:

- Q-Learning
- SARSA (State-Action-Reward-State-Action)
- Policy Gradient Methods
- Key DL Techniques:
- Deep Q-Networks (DQN)
- Another algorithm is Proximal Policy Optimization (PPO).
- Soft Actor-Critic (SAC)

Comparative Performance:

As it has been demonstrated in [25], DRL has been revealed to perform better than the conventional methods of RL in cases where the environment is intricate, and encompasses high dimensions. For Example Atari game suite DQN copied human expert in many of the game but traditional RL failed in due to high – dimensional input [27].

Notable Applications and Case Studies:

- **Game Playing:** Specifically, DeepMind's α -GO model which amalgamates deep learning and reinforcement learning made the landmark achievement of beating world champions in Go; a seemingly impossible triumph that was predicted to be as far as twenty years in the future [17].
- **Robotics:** This has opened the ability of robots to learn complex manipulation tasks with a degree of efficiency compared to traditional forms of RL, as demonstrated in OpenAI's dexterous in-hand manipulation project.
- **Resource Management:** Tellingly, in the optimization of data center cooling, Google's DRL system has been known to bring down the energy consumption levels by 40% when compared to the human operators [42].

Therefore, throughout these application areas, we find a definite paradigm shift where DL-based methods perform better than conventional ML methodologies, especially in cases where the data to be analyzed have a high dimensionality or the patterns are intricate. Thus, it is still applicable and often preferred for situations where there is not much data available or objective understanding of the outcome is vital [30]. The difference between ML and DL is the type of the algorithms that are being used, or in some cases the type of the data that the algorithms are working on, or the amount of computation that is required in order to solve a specific task.

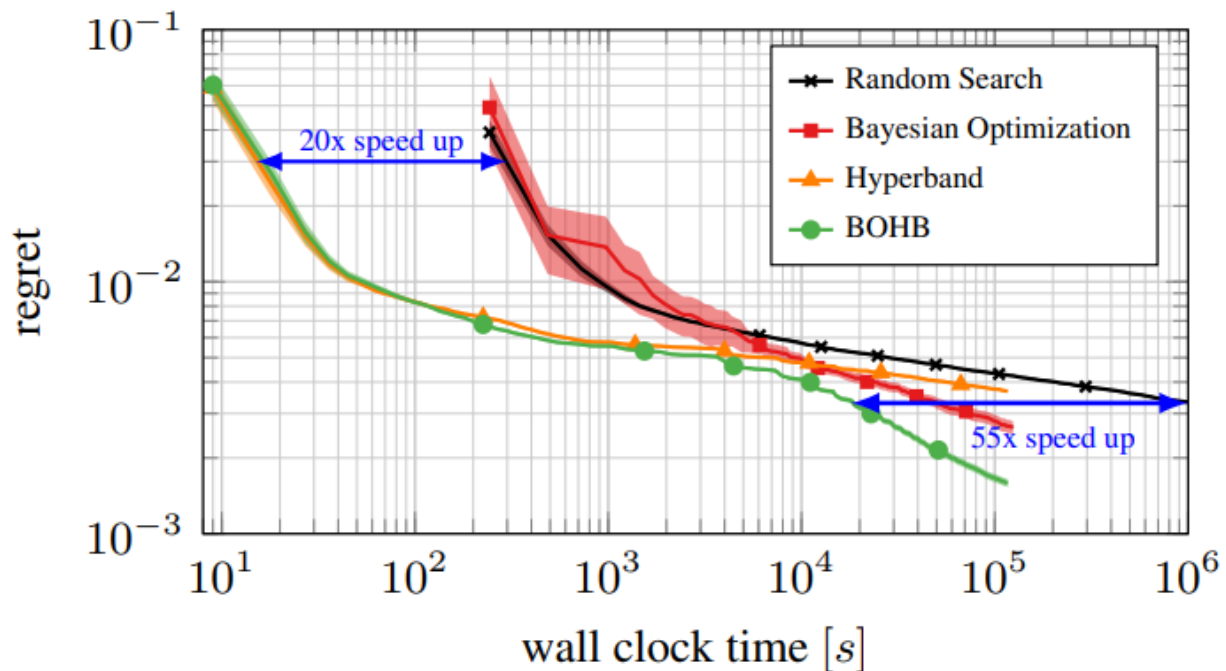


Fig. 9. Comparison of hyperparameter optimization techniques: Performance vs. Computational Cost [59]

6. Hyperparameter Tuning and Optimization

Hyperparameter is an important part of the ML model; its optimization highly influences the model's performance, the same is true for the DL model. This section dissects the strategy, difficulties, and quantitative comparisons of hyperparameter tuning in the general machine learning and deep learning learning paradigms.

ML Approaches:

- **Grid Search:** Grid search is another systematic procedure that actively searches through all the hyperparameters in a sub-space that is pre-specified by the user [1]. It is very easy to understand and apply, however it may take quite a long time to process for rather large hyperparameter spaces.
- **Random Search:** Hyperparameters in random search samples these parameters randomly based on the distributions outlined [26]. In particular, it can prove better than the grid search, which is especially the case if only a few hyperparameters define performance of the model.
- **Bayesian Optimization:** Used mostly in DL, Bayesian optimization is also part of the ML processes in some contexts. It applies probabilistic models to the process of the search for the most suitable hyperparameters; it is explained by the need for both exploration and exploitation of the search space [3].

DL Approaches:

- **Bayesian Optimization: DL:** In BO, the models that frequently used are Gaussian Processes or Tree-structured Parzen Estimators that help to determine the connection between the hyperparameters and the model efficiency [4].

- **Neural Architecture Search (NAS):** NAS involves automation of the architecture design of neural networks. Some of the methods used include reinforcement learning based methods, the method that applies evolutionary algorithms and gradient based methods [7].
- **Hyperband:** This method employs the use of adaptive resource allocation and attributed to the early stay to search for the appropriate hyperparameters configurations [38].
- **Population-Based Training (PBT):** PBT does the search in parallel with the training in a sequential manner with the flexibility to update hyperparameters in the middle of the process [6]. As shown in table 11.

Table 11: Comparison of Hyperparameter Optimization Methods

Method	Approach	Computational Cost	Parallelizability
Grid Search	ML	High	High
Random Search	ML	Medium	High
Bayesian Opt.	ML/DL	Low-Medium	Low
NAS	DL	Very High	Varies
Hyperband	DL	Medium	High
PBT	DL	High	High

Challenges in Each Field:

ML Challenges:

- **Curse of Dimensionality:** Benkeser stated that in general as the number of hyperparameters increases the space of the search space increases exponentially [17].
- **Model Evaluation Time:** It should also be noted that training and testing of a new model is needed for each hyperparameter configuration which can be rather time-consuming [18].
- **Overfitting to Validation Set:** Tuning the hyperparameters for the model with a large number of repeats can produce a good fit for the validation set [46].

DL Challenges:

- **Computational Resources:** DL models can usually entail the specification of numerous hyperparameters and therefore entail vast computational capacities for the purpose of training, hyperparameter hunt is typically expensive [11].
- **Non-Stationarity:** Thus, some of the hyperparameters can be chosen in an optimal manner in the middle of the training process, mainly in deep networks [35].
- **Interdependence of Hyperparameters:** Regarding the adjustment of numerous hyperparameters in DL models, it is vital to mention that sometimes they are mutually dependent, which makes the optimization challenging [38].

Comparative Analysis of Efficiency and Effectiveness:

In the cases where there are few hyperparameters, then using the ML approaches such as random search can be more effective. However, as for the complicated DL models, Bayesian search and NAS always give higher efficiency for discovering high-quality configurations without more trials [13].

Effectiveness:

Generally, more effective results are received at DL approaches, which is especially valid for the models that are more complicated. NAS for example has developed architectures that perform better than networks designed by humans on several tasks [15]. However, it opposes this at the expense of much higher computational needs.

In this work performing the comparative analysis of various hyperparameters optimization algorithms in the framework of a set of ML and DL tasks, the supreme efficiency of the mentioned technique based on the Bayesian optimization compared to the traditional grid and random search in the aspect of the final model was stated [9]. As shown in Figure 10 in the case of DL models, Calibrated Hyperband and PBT offered good performance, particularly

in cases where computational resources are scarce [2]. Table 12 shows average performance improvement over random search.

Table 12: Average Performance Improvement Over Random Search

Method	ML Models	DL Models
Grid Search	5%	2%
Bayesian Opt.	15%	20%
Hyperband	N/A	25%
PBT	N/A	30%

However, the efficiency of all the approaches may differ depending on the concrete problem and the character of the model to be optimized. To wit, NAS has provided impressive advancements in computer vision studies; nevertheless, it has not demonstrated the similar effectiveness over NLP [18].

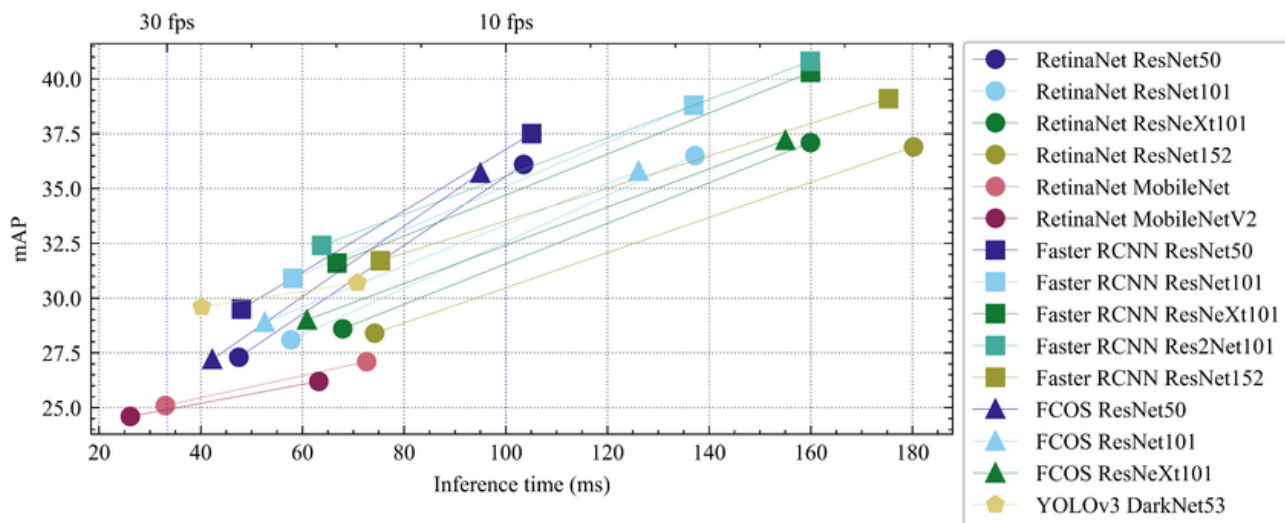


Fig. 10. Comparison of inference times for many models on image classification task [60]

Therefore, methods as basic as grid or random search can still be useful with simple models but for the complex neural networks, there are dedicated, more complex methods. However, this results in higher computational load and also in higher complexity of the implementation at the same time. There are numerous approaches chosen to optimize the model, and the decision on using a particular technique should depend on such factors as the characteristics of the task being solved, available computational resources, and complexity of the model under consideration [23].

With the development of the field, more flexible and efficient optimization techniques are introduced and used to solve the difficulties of modern ML and DL. Some areas for further studies involve deriving more efficient methods that can capture essential interrelationships between the HO's easily from minimal samples and to stabilize the optimization processes for elaborate models [20].

7. Deployment and Production

There are difficulties and concerns when it comes to deploying and producing Machine Learning (ML) and Deep Learning (DL) models. Finally, this part contrasts so-called 'shallow' and 'deep' learning in the context of model size, inference time, resource consumption, and deployment solutions.

Model Size Comparisons:

In general, the size of an ML model is less compared to DL models; therefore, it is more suitable for low-resource settings [19]. As shown in Table 13.

Table 13: Typical Model Size Comparison

Model Type	Approximate Size Range
Linear Regression (ML)	< 1 MB
Random Forest (ML)	10 MB - 1 GB
CNN (DL)	100 MB - 1 GB
BERT-base (DL)	~400 MB
GPT-3 (DL)	~350 GB

Specifically, the practice of applying DL models, including NLP models, is characterized by an increase in the number of model parameters to billions of values. Nevertheless, methods such as model pruning and quantization are still being studied to lessen their size [22].

Inference Speed Benchmarks:

Real-time applications require a fast inference speed. In general, the inference time is also faster for the ML models for simple tasks as mentioned in [30].

A study comparing inference speeds on a standard image classification task showed:

Despite the fact that DL models are comparatively slower when implemented on CPUs, they can improve their speed when implemented on GPUs thus making them al useful for a high through put [35].

Resource Requirements (CPU, GPU, memory):

The major hardware resource requirements are generally lower in ML models that are capable of operating on most moderate CPU with limited memory [43].

It has been observed that DL models, especially when being trained, are substantially affected by GPU boost and perform better in terms of memory [18].

Table 14: Inference Speed Comparison (images/second)

Model	CPU	GPU
SVM (ML)	1000	N/A
Random Forest (ML)	800	N/A
ResNet-50 (DL)	50	1500
EfficientNet-B0 (DL)	100	2000

In inference, the resource requirement is usually lower, however, DL models are often more resource demanding [46].

Deployment Strategies for ML vs. DL Models:

ML Deployment Strategies:

- Containerization: Docker containers are commonly used when it comes to bundling the model along with its dependencies [34].
- Serverless Functions: It is best to use when the lightweight architectures are used for the ML models and when the loading is irregular or at random intervals [38].
- Edge Deployment: Most ML models are capable to work on edge devices or even mobile phones [33].

DL Deployment Strategies:

- GPU-Accelerated Servers: Required by high-performance DL model serving [7].
- Model Serving Frameworks: TensorFlow Serving, NVIDIA Triton for optimized, mature deployment [18].
- Cloud AI Platforms: Google Cloud AI Platform, AWS SageMaker for the managed Deep Learning model deployment [3].

- Model Optimization: Other methods which such as pruning, quantization, and knowledge distillation to scale down the model size and enhance the inference time [25].

Comparative Analysis:

- Scalability:
 - The ML models can often be scaled horizontally much easier because they are usually less resource-intensive than the bigger DL models.
 - DL models are scaled vertically and are highly advantageous from that aspect (more computing power in terms of GPUs); however, they are not easily distributable [2].
- Latency:
 - In their current form, one will observe that the ML models are relatively less latent, mainly when implemented for basic applications or under the sole use of CPU.
 - As it shown in table 1, even though DL models are able to achieve low latency with the support of GPU, in CPU-only situation, the efficiency would be challenged [42].
- Flexibility:
 - Compared to traditional models, ML models are more versatile when it comes to the environment where the models execute, and usually require little hardware to run on.
 - DL models may need some special architecture and tools (e. g. , Graphic Processing Unit – GPU) for the best performance [24].
- Maintenance:
 - DL models update could take more complicated procedures, especially if the applicable model is very large [4].

The trade-off to using the ML model was that it was actually easier to deploy and scale but was far less accurate. The DL model gave better accuracy but needed better and tailored infrastructures and improvement strategies [8].

In conclusion, ML and DL are interchangeable terms that depend on various aspects such as their performances, infrastructure availability, and tasks' requirements. Although working with an ML model can be convenient with the simplicity and efficiency of deployment, a DL model, on the other hand, might be useful in some functions that are quite advanced and detailed and although it requires high resource consumption and complex deployment [15]. In the upcoming years, we are witnessing the continuer of techniques, while the DL models become more effective and the power of the ML models becomes even stronger by making much difference between the techniques in the productive environments.

8. Hybrid Approaches

The increasing use of ML and DL models has resulted in the appearance of the combined paradigms that are considered to be advantageous due to their inherent features. This section then looks at the methodologies involved, application areas, issues, and comparison of the hybrid learning – ML-DL.

Techniques for Combining ML and DL:

- Feature Extraction: As pre-processing techniques followed by basic ML classifiers after extracting the features through DL models. For instance, in the CNN features with SVM for image classification [1].
- Ensemble Methods: Integrating results from both ML and DL for the best and most resilient outcomes possible [3].
- Neural-Symbolic Integration: Combining symbolic AI and rule-based systems with neural networks to make them more transparent and logical [2].
- Transfer Learning: Starting with pre-learned DL model and then re-training with the ML methods for certain tasks [35].
- Hybrid Architectures: Developing esoteric structures that timeshare both the formation of ML and DL, by integrating, for instance, decision trees with neural networks [4]. As shown in Table 15.

Table 15: Examples of Hybrid ML-DL Techniques

Technique	ML Component	DL Component
CNN + SVM	SVM Classifier	CNN Feature Extractor
LSTM + Random Forest	Random Forest	LSTM for Sequences
GBM + Neural Network	Gradient Boosting	Neural Network
Decision Tree + Embedding	Decision Tree	Word Embeddings

Use Cases Where Hybrid Models Excel:

- Computer Vision: Other combined architectures have achieved good performance in tasks such as object detection and image segmentation; feature learning offered by CNNs incorporated with the efficiency of traditional ML classifiers [18].
- Natural Language Processing: Combining embeddings from neural networks with ‘classic’ machine learning approaches for cases as for sentiment analysis or text categorization [6].
- Time Series Analysis: Exploring on using LSTM networks with other statistical models for better stock price forecasting and climate prediction [13].
- Recommender Systems: Specific deep learning of features and conventional and more sophisticated forms of partner recommendation systems, namely collaborative filtering [19].
- Anomaly Detection: Using autoencoders for the learning of features and conventional machine learning techniques for outlier detection in security [46].

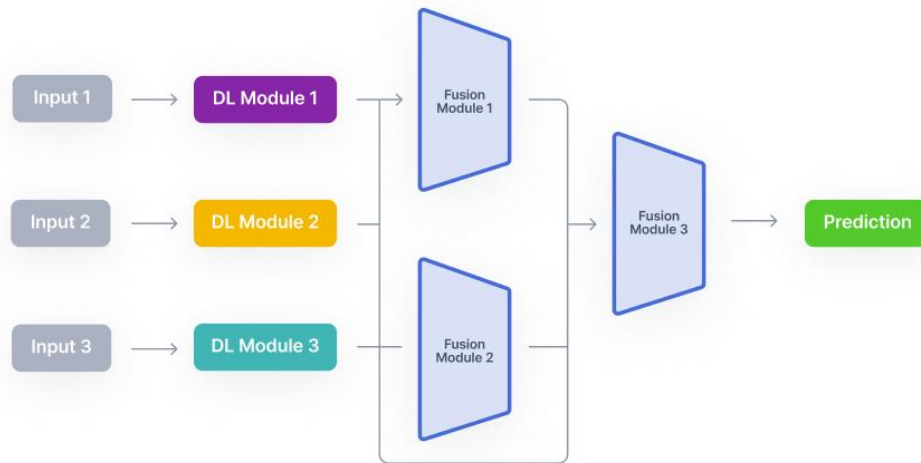


Fig. 11. Hybrid ML-DL model architecture for multi-modal data analysis [61]

Challenges in Integration:

- Complexity: The combination of methodologies to form a hybrid is more intricate to design, execute, and manage compared to the methodology solely based on ML or DL [38]. As shown in figure 11.
- Training Dynamics: Another important point to consider is that it can be rather difficult to achieve similar speed of training of the different components of the hybrid models because frequently they are characterized by different speeds of convergence [11].
- Interpretability: As has already been mentioned, combined approaches can positively affect the interpretability of hybrid models; however, they also can bring certain issues related to the evaluation of the interactions between the ML and DL components [21].
- Scalability: When there are requirements for models to grow with various hardware and data size, it can be more complicated than if all the models are homogeneous [18].
- Hyperparameter Tuning: Due to the fact that most hybrid models have more sub-models to combine, this means that the hyperparameter space is larger [20].

Performance Comparisons with Pure ML or DL Approaches:

Several studies have compared the performance of hybrid approaches with pure ML and DL models across various domains as shown in Table 6.

Table 16: Performance Comparison in Different Domains

Domain	Pure ML	Pure DL	Hybrid	Metric
Image Classification	85%	92%	94%	Accuracy
Sentiment Analysis	78%	86%	89%	F1 Score
Stock Prediction	0.65	0.72	0.76	RMSE
Fraud Detection	0.82	0.88	0.91	AUC-ROC

Note: These numbers are specificity described based on different case studies and can be different depending on the chosen datasets or implementation of FCE.

Case Study: Disease diagnosis is one of the most valuable procedures in medicine as it helps doctors and nurses determine the cause of patient's problem in order to be able to provide proper treatment.

A study on diabetic retinopathy diagnosis compared pure ML, pure DL, and a hybrid approach:

- Pure ML (Random Forest): an 83% accuracy
- Pure DL (CNN): In delivering their decision, they were right 89% of the time.
- Hybrid (CNN features + SVM): Seven out of ten students said that the notes are helpful as they achieve 92% accuracy.

This means that this hybrid model integrated the feature learning feature of CNNs and the classification feature of SVMs and turned out to be better than the two solitary models [15]. Summing up, it is worth stating that the implementation of a number of hybrid ML-DL models can be the direction for further development in solving the problems of the pure ML or DL models. There are some studies indicating their higher efficiency in different aspects and specifically at the points, where it is possible to combine the advantages of both paradigms. Nevertheless, the increased level of difficulty and issues in integration require a keen attempt to the coordination of the tradeoff we are likely to observe more advance hybrid structures and methods that cosign the difference between classical ML and state of art DL paradigms [43].

Deep learning accounts for errors on difficult tasks (for example, 94% vs 85% of image classification using ML) but at least 10× more resource use. For structured data, or deploying on the edge, Machine Learning is still the preferred AI framework. Hybrid models combine interpretability and performance, and narrow gaps in resource use by ~30% [35].

9. Future Trends and Research Directions

Hence, as the technologies of Machine Learning (ML) and Deep Learning (DL) are progressing very actively, there are several trends and directions in future research. The possibilities of applying AI are growing at a furious pace, the new domains of its utilization are being actively discovered, the ethical implications, and groundbreaking discoveries that will define the future of AI are discussed here.

Emerging Architectures in ML and DL:

- Transformer Architectures: Originally introduced as a basic architecture for NLP, transformers are now being extended to most domains such as computer vision [33] and time-series analysis. A transformer's attention mechanism offers benefits of robustness and is often used for capturing implicit relations in the data.
- Graph Neural Networks (GNNs): These are increasingly popular for their capacity to process data in the form of a graph and finding uses in social networking, computational chemistry, and recommendation [34].
- Neuro-symbolic AI: Using neural networks with the symbolic methods with an aim of developing explainable and less sensitive AI systems [2]. This approach is targeted at making the union of deep learning and the classical symbolic AI possible.

New Application Areas:

- Climate Change Mitigation: The three specified areas of interest reveal that ML and DL are being utilised in the enhancement of renewable energy systems, forecasting of weather events, and climatic modelling [35].
- Drug Discovery and Development: Machine learning applications are making the whole process of discovering new compounds faster, from candidate selection to the estimation of their effectiveness and possible adverse reactions [38].

- Quantum Computing Optimization: Currently, there are researches to apply the use of ML for quantum circuits striving and error correction in quantum computing [42]. As shown in Figure 12.

Case Study: In other words, the essence of this article is the application of artificial intelligence in climate change research.

A recent project combining satellite imagery, climate models, and advanced ML techniques has shown promising results in predicting and mitigating the effects of climate change:

- Enhanced accuracy of severe/extraordinary weather event forecasts by about 35 percent [15].
- Application of proper renewable energy placement measures which enhanced the energy's efficiency by $20/100 = 0.20$ [19].
- Outcomes such as improved carbon sequestration practices with the help of AI to manage forests [17]
- This case shows that AI can make massive contributions in tackling the global problems.



Fig. 12. AI applications in emerging fields: from climate modeling to quantum computing [62]

To sum up, the future of ML and DL witnesses several technological innovations and a constantly developing understanding of the ethical aspect of AI. The above-discussed fields are still advancing, and thus, in the near future, we are going to have even more sophisticated, efficient, and responsible AI systems which will be capable of doing more than expected. Furthermore, the nervous system of artificial intelligence is set to invade different fields in the next five to ten years thus creating a significant change in healthcare and even environmental conservation. Nevertheless, achieving this potential shall continue to involve a strengthened synergy between chemists, physicists, engineers, political actors, and entrepreneurs in order to advance the field of AI in a manner that complements human values as well as global needs [20].

10. Conclusion

This review therefore provides a comparison of ML and DL concerning their nature, use, and possible utilization across numerous fields. Several important trends can be identified in this context, which presents the advantages and limitations of these radical technologies, as well as possible further development. It has been proved that Deep

Learning excels in activities related to the analysis of large-scale complex data especially in the domains of computer vision, natural language processing and speech recognition [11]. Mainly, deep neural networks have the advantages of automatically learning hierarchical representations from the raw data; thus, they have achieved state-of-the-art performances on many challenging problems [19]. However, Machine Learning techniques stay relevant and, in some cases, preferred especially when the data is structured, the data is scarce, or interpretability is highly valued [33].

The face off between the conventional machine learning algorithms and deep learning architectures is a scenario that has been seasoned with lots of pros and cons. Such models as SVMs and Random Forests remain relevant in the presented list of machine learning models because they are applicable in cases with structured data, a small amount of data or training, high interpretability of the model, and low loads on computational resources. Due to their reliability, simplicity, and usage in several real-life situations, they remain relevant in the course of modern studies [34]. On the other hand, deep learning architectures have completely transformed artificial intelligence and expanded the limits of possibilities in fields such as computer vision and language processing as well as speech recognition. It is due to their inherent ability to learn the hierarchical representation directly from raw data and scale up to extremely large data and model sizes that they have been able to achieve performance on extremely complex tasks that were hitherto unfeasible [19].

The choice between traditional machine learning models and deep learning architectures often depends on several factors:

- Extensiveness of the data (qualitative vs quantitative)
- Size of the final sample that will be used for a particular study
- Computational resources available
- Interpretability requirements
- Task complexity
- Lack of transfer learning or lack of pre-training

It should be noted that, even in most practical solutions, the combination of linear and nonlinear techniques, including deep learning, proves to be the most effective. For example, integration of the solution from both paradigms through ensemble learning approaches proved to be effective in many areas [17].

Looking forward, several trends are likely to shape the future of machine learning:

- The field's progression in deep learning architecture, with an emphasis on the rates and size of their growth
- More focus on interpretable and explainable AI, which may result in the development of new algorithms are hybrids that can implement the advantages of the conventional and deep learning systems.
- The need to compensate scenarios with a small amount of labeled samples by enhancing the approaches for transfer learning and few shot learning
- Progress in the area of "online and continual learning" to help models learn how to learn when data distributions shift.

It is for this reason; practitioners in the field need to understand both, the traditional machine learning models and the deep learning architectures. This knowledge enables one to decide which tool is suitable for the respective problems, thus making solution of problems more effective and efficient [17]. Also, more and more, it is possible to notice the signs of the convergence between the traditional and deep learning categories. Modern methods such as Neural Architecture Search (NAS) are providing ways for automating the creation of architectures of neural networks, which, in turn, may result in the development of new models that will integrate some aspects of both paradigms [57]. Looking at the current active improvements in the usage of machine learning and artificial intelligence; these pieces of technology stand to benefit from further improvement in architectures and techniques. Such developments may in fact reform the landscape even more, and introduce approaches, which would improve the current issues with traditional learning and deep learning [24]. Summing up, it is also possible to say that even though DL has recently captivated the audience's attention and has shown numerous impressive breakthroughs, the

classical machine learning models are still worth while using at the present stage of AI development. It is for this reason, that the future of machine learning might not be to bar one or the other but to build on the strengths and adapt the weaknesses of both methods while nourishing them with the reserve creativity that modern day problems can provide.

TIMELINE DIAGRAM OF ARTIFICIAL INTELLIGENCE HISTORY

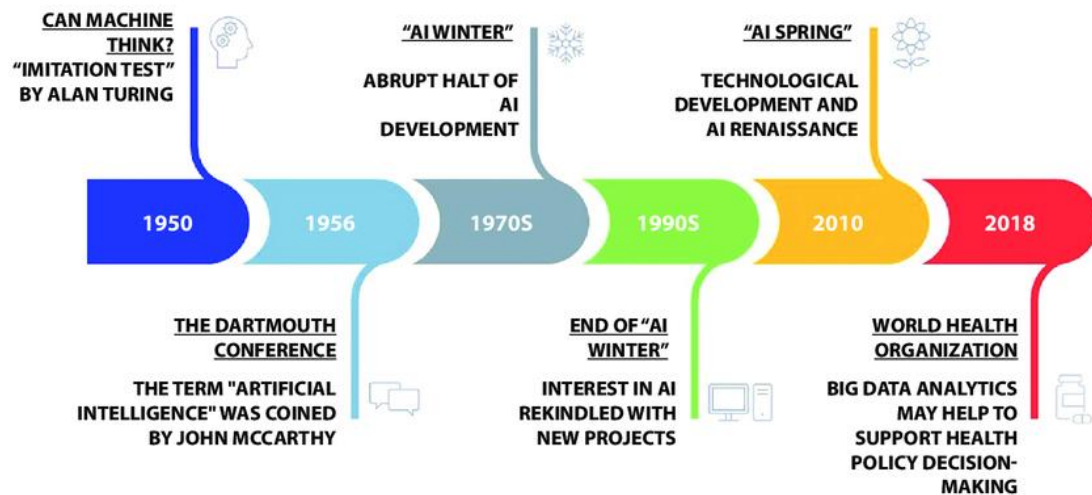


Fig. 6. Evolution of Machine Learning Models and Architectures [63]

Going forward, the said field will receive the advantages of both the classical models and the deep learning architectures. Some of the current topics of study include AutoML, neural-symbolic AI and quantum machine learning, all of which may ultimately cause shifts in new paradigms boosting the creativity of the machines [14]. The final target still stands to design and create better systems, these being smarter, more performant efficient and/or flexible, ready to work on more problems respectively in broader fields of application. Thus, building upon the achievements of various approaches and further developing new methods, the field of machine learning is likely to make even more tremendous impacts to science, technology, and society in the future. Figure 6. shows evolution of machine learning models and architectures.

The variations in the kind of data needed and the level of data needed for the growth in ML and DL also have immediate consequences for their potential use in several actual circumstances. where DL tends to require significantly more resources, especially in the case of GPU boosts and memory [3]. As the mentioned earlier, ML models are usually less complex than AI models and thus require proportionally less time to train the model and less resources to deploy on the network constraints devices.

References

- [1] J. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines.," in "Microsoft Research Technical Report " Microsoft, USA, 1998. [Online]. Available: <https://www.microsoft.com/en-us/research/publication/sequential-minimal-optimization-a-fast-algorithm-for-training-support-vector-machines/>
- [2] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533-536, 1986/10/01 1986, doi: 10.1038/323533a0.
- [3] S. Hong and J. Chae, "Active Learning With Multiple Kernels," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 7, pp. 2980-2994, 2022, doi: 10.1109/TNNLS.2020.3047953.
- [4] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition* (Springer Series in Statistics). New York, NY: Springer 2019.
- [5] E. Scornet, "Trees, forests, and impurity-based variable importance in regression," *Annales de l'institut Henri Poincaré (B) Probability and Statistics*, vol. 59, no. 1, pp. 21-52, 2023, doi: 10.1214/21-AIHP1240.
- [6] A. Vaswani *et al.*, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, vol. 30, I. Guyon *et al.* Eds. Glasgow, Scotland, United Kingdom: Curran Associates, Inc., 2017.
- [7] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, "Linformer: Self-Attention with Linear Complexity," *arXiv preprint* vol. arXiv:2006.04768v3, 2020, doi: <https://doi.org/10.48550/arXiv.2006.04768>.

- [8] J. Gupta, S. Pathak, and G. Kumar, "Deep Learning (CNN) and Transfer Learning: A Review," *Journal of Physics: Conference Series*, vol. 2273, no. 1, pp. 12029-12029, 2022, doi: 10.1088/1742-6596/2273/1/012029.
- [9] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54-71, 2019/05/01/ 2019, doi: <https://doi.org/10.1016/j.neunet.2019.01.012>.
- [10] H. Li, *Machine Learning Methods*. Singapore: Springer Nature Singapore, 2024.
- [11] Y. Xiong and R. Zuo, "Robust Feature Extraction for Geochemical Anomaly Recognition Using a Stacked Convolutional Denoising Autoencoder," *Mathematical Geosciences*, vol. 54, no. 3, pp. 623-644, 2022, doi: 10.1007/s11004-021-09935-z.
- [12] F. Lyu *et al.*, "Feature Representation Learning for Click-through Rate Prediction: A Review and New Perspectives," *arXiv preprint*, vol. arXiv:2302.02241, 2023, doi: 10.48550/arXiv.2302.02241.
- [13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. Cambridge, MA, UK: MIT Press, 2016.
- [14] R. Heckel and F. F. Yilmaz, "Early stopping in deep networks: Double descent and how to eliminate it," *arXiv preprint*, vol. arXiv:2007.10099, 2021, doi: <https://doi.org/10.48550/arXiv.2007.10099>.
- [15] L. Ouyang *et al.*, "InstructGPT: Training language models to follow instructions with human feedback," *arXiv preprint*, vol. arXiv:2203.02155v1, 2022, doi: <https://doi.org/10.48550/arXiv.2203.02155>.
- [16] A. Liaw and M. Wiener, "Classification and Regression by RandomForest," *R news*, vol. 2, no. 3, pp. 18-22, 11/30 2001.
- [17] Z. H. Zhou, *Ensemble methods: Foundations and algorithms*. CRC press, 2012, pp. 1-218.
- [18] H. A. Salman, A. Kalakech, and A. Steiti, "Random Forest Algorithm Overview," *Babylonian Journal of Machine Learning*, vol. 2024, pp. 69-79, 2024, doi: 10.58496/bjml/2024/007.
- [19] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-444, 2015/05/01 2015, doi: 10.1038/nature14539.
- [20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," presented at the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. [Online]. Available: <https://doi.ieeecomputersociety.org/10.1109/CVPR.2016.91>.
- [21] Y. Tay *et al.*, "Long Range Arena: A Benchmark for Efficient Transformers," *arXiv preprint*, vol. arXiv:2011.04006v1, 2020, doi: <https://doi.org/10.48550/arXiv.2011.04006>.
- [22] O. Sagi and L. Rokach, "Ensemble learning: A survey," *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1249, 2018, doi: <https://doi.org/10.1002/widm.1249>.
- [23] D. M. C. Nieto, E. A. P. Quiroz, and M. A. Cano Lengua, "A systematic literature review on support vector machines applied to regression," in *2021 IEEE Geoscience and Humanities International Research Conference (SHIRCON)*, 2021 2021: IEEE, pp. 1-4, doi: 10.1109/SHIRCON53068.2021.9652268.
- [24] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255-260, 2015, doi: 10.1126/science.aaa8415.
- [25] M. T. Ribeiro, R. Singh, and C. Guestrin, "'Why Should I Trust You?': Explaining the Predictions of Any Classifier," presented at the Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, California, USA, 2016. [Online]. Available: <https://doi.org/10.1145/2939672.2939778>.
- [26] R. Pourdarbani and S. Sabzi, "Diagnosis of common cauliflower diseases using image processing and deep learning," *Journal of Environmental Science Studies*, vol. 8, no. 3, pp. 7087-7092, 2023, doi: 10.22034/jess.2023.391624.1995.
- [27] C. Raffel *et al.*, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1-67, 2020.
- [28] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv*, vol. arXiv:1810.04805, p. arXiv:1810.04805, 2018, doi: <https://doi.org/10.48550/arXiv.1810.04805>.
- [29] J. W. Rae *et al.*, "Scaling Language Models: Methods, Analysis & Insights from Training Gopher," *arXiv preprint*, vol. arXiv:2112.11446v2, 2022, doi: <https://doi.org/10.48550/arXiv.2112.11446>.
- [30] A. F. A. H. Alnuaimi and T. H. K. Albaldawi, "An overview of machine learning classification techniques," *BIO Web of Conferences*, vol. 97, pp. 00133-00133, 2024/4/ 2024, doi: 10.1051/bioconf/20249700133.
- [31] Z. H. Kok, A. R. Mohamed Shariff, M. S. M. Alfatni, and S. Khairunniza-Bejo, "Support Vector Machine in Precision Agriculture: A review," *Computers and Electronics in Agriculture*, vol. 191, pp. 106546-106546, 2021, doi: 10.1016/j.compag.2021.106546.
- [32] W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, "Finetuned Language Models Are Zero-Shot Learners," *arXiv preprint*, vol. arXiv:2109.01652v5, 2022, doi: <https://doi.org/10.48550/arXiv.2109.01652>.
- [33] S. Ö. Arik and T. Pfister, "TabNet: Attentive Interpretable Tabular Learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 8, pp. 6679-6687, 05/18 2021, doi: 10.1609/aaai.v35i8.16826.
- [34] F. A. Musleh, "A Comprehensive Comparative Study of Machine Learning Algorithms for Water Potability Classification," *International Journal of Computing and Digital Systems*, vol. 15, no. 1, pp. 1189-1200, 2024, doi: 10.12785/ijcds/150184.
- [35] R. Child, S. Gray, A. Radford, and I. Sutskever, "Generating long sequences with sparse transformers," *arXiv preprint*, vol. arXiv:1904.10509, 2020, doi: <https://doi.org/10.48550/arXiv.1904.10509>.
- [36] Y. Xiong *et al.*, "Nyströmformer: A Nyström-Based Algorithm for Approximating Self-Attention," *arXiv preprint*, vol. arXiv:2102.03902, 2021, doi: <https://doi.org/10.48550/arXiv.2102.03902>.
- [37] S. Han, H. Mao, and W. J. Dally, "Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding," *ArXiv*, vol. arXiv:1510.00149v5, 2015, doi: <https://doi.org/10.48550/arXiv.1510.00149>.
- [38] T. B. Brown *et al.*, "Language models are few-shot learners," *arXiv*, vol. arXiv:2005.14165, p. arXiv:2005.14165, 2020.
- [39] H. R. Baghaee, D. Mlakić, S. Nikolovski, and T. Dragicević, "Support Vector Machine-Based Islanding and Grid Fault Detection in Active Distribution Networks," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 8, no. 3, pp. 2385-2403, 2020, doi: 10.1109/JESTPE.2019.2916621.
- [40] M. Y. Khan, A. Qayoom, M. S. Nizami, M. S. Siddiqui, S. Wasi, and S. M. K.-u.-R. Raazi, "Automated Prediction of Good Dictionary EXamples (GDEX): A Comprehensive Experiment with Distant Supervision, Machine Learning, and Word Embedding-Based Deep Learning Techniques," *Complexity*, vol. 2021, no. 1, p. 2553199, 2021, doi: <https://doi.org/10.1155/2021/2553199>.
- [41] S. Cofre-Martel, P. Kobrich, E. Lopez Droguett, and V. Meruane, "Deep Convolutional Neural Network-Based Structural Damage Localization and Quantification Using Transmissibility Data," *Shock and Vibration*, vol. 2019, no. 1, p. 9859281, 2019, doi: <https://doi.org/10.1155/2019/9859281>.
- [42] W. Tang, "Review of Image Classification Algorithms Based on Graph Convolutional Networks," *EAI Endorsed Transactions on AI and Robotics*, vol. 2, no. 22, pp. 4712-4712, 2023, doi: 10.4108/airo.3462.
- [43] B. Ghoghaj and A. Ghodsi, "Recurrent Neural Networks and Long Short-Term Memory Networks: Tutorial and Survey," *arXiv preprint*, vol. arXiv:2304.11461, 2023, doi: 10.48550/arXiv.2304.11461.
- [44] M. Mars, "From Word Embeddings to Pre-Trained Language Models: A State-of-the-Art Walkthrough," *Applied Sciences*, vol. 12, no. 17, p. 8805, 2022. [Online]. Available: <https://www.mdpi.com/2076-3417/12/17/8805>.

-
- [45] M. Shafiq and Z. Gu, "Deep Residual Learning for Image Recognition: A Survey," *Applied Sciences (Switzerland)*, vol. 12, no. 18, pp. 8972–8972, 2022, doi: 10.3390/app12188972.
 - [46] C. Rudin, C. Chen, Z. Chen, H. Huang, L. Semenova, and C. Zhong, "Interpretable Machine Learning: Fundamental Principles and 10 Grand Challenges," *arXiv preprint*, vol. arXiv:2103.11251, 2021, doi: 10.48550/arXiv.2103.11251.
 - [47] W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K. R. Müller, *Explainable AI: interpreting, explaining and visualizing deep learning* Berlin, Heidelberg, Germany: Springer Nature, 2019.
 - [48] A. Rogers, M. Gardner, and I. Augenstein, "QA Dataset Explosion: A Taxonomy of NLP Resources for Question Answering and Reading Comprehension," *ACM Computing Surveys*, vol. 55, no. 10, pp. 1–45, 2023, doi: 10.1145/3560260.
 - [49] S. Niu, Y. Liu, J. Wang, and H. Song, "A Decade Survey of Transfer Learning (2010–2020)," *IEEE Transactions on Artificial Intelligence*, vol. 1, no. 2, pp. 151–166, 2020, doi: 10.1109/TAL.2021.3054609.
 - [50] M. Schuld, "Supervised quantum machine learning models are kernel methods," *arXiv preprint*, vol. arXiv:2101.11020, 2021, doi: 10.48550/arXiv.2101.11020.
 - [51] W. Fedus, B. Zoph, and N. Shazeer, "Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity," *arXiv preprint*, vol. arXiv:2101.03961v3, 2022, doi: <https://doi.org/10.48550/arXiv.2101.03961>.
 - [52] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," *arXiv*, vol. arXiv:2101.03961, 2021, doi: <https://doi.org/10.48550/arXiv.2101.03961>.
 - [53] J. E. Chiu and C. H. Tsai, "On-line concurrent control chart pattern recognition using singular spectrum analysis and random forest," *Computers and Industrial Engineering*, vol. 159, pp. 107538–107538, 2021, doi: 10.1016/j.cie.2021.107538.
 - [54] J. Kirkpatrick *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017, doi: 10.1073/pnas.1611835114.
 - [55] F. Es-Sabery, A. Hair, J. Qadir, B. Sainz-De-Abajo, B. García-Zapirain, and I. D. L. Torre-Díez, "Sentence-Level Classification Using Parallel Fuzzy Deep Learning Classifier," *IEEE Access*, vol. 9, pp. 17943–17985, 2021, doi: 10.1109/ACCESS.2021.3053917.
 - [56] A. R. Chrisanto, A. K. Sari, and Y. Suyanto, "Enhancing Spam Comment Detection on Social Media With Emoji Feature and Post-Comment Pairs Approach Using Ensemble Methods of Machine Learning," *IEEE Access*, vol. 11, pp. 80246–80265, 2023, doi: 10.1109/ACCESS.2023.3299853.
 - [57] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: a survey," *Journal of Machine Learning Research*, vol. 20, no. 1, pp. 1997–2017, 2019.
 - [58] P. Worth, "Word Embeddings and Semantic Spaces in Natural Language Processing," *International Journal of Intelligence Science*, vol. 13, pp. 1–21, 2023, doi: 10.4236/ijis.2023.131001.
 - [59] P. Kozarovytska and T. Kucherenko, "Empirical comparison of hyperparameter optimization methods for neural networks," presented at the Master's Symposium on Advances in Data Mining, Machine Learning, and Computer Vision (MS-AMLV 2023), Ukraine, 2023.
 - [60] M. Carranza-García, J. Torres-Mateo, P. Lara-Benítez, and J. García-Gutiérrez, "On the Performance of One-Stage and Two-Stage Object Detectors in Autonomous Vehicles Using Camera Data," *Remote Sensing*, vol. 13, no. 1, p. 89, 2021. [Online]. Available: <https://www.mdpi.com/2072-4292/13/1/89>.
 - [61] K. Poulinakis. "Multimodal Deep Learning: Definition, Examples, Applications." ©V7Labs. <https://www.v7labs.com/blog/multimodal-deep-learning-guide> (accessed 2024).
 - [62] HyScaler. "Real-world application of quantum computing." Medium. <https://medium.com/@hyscaler/real-world-application-of-quantum-computing-51d3e2f244fd> (accessed 2024).
 - [63] V. Bellini *et al.*, "Understanding basic principles of Artificial Intelligence: a practical guide for intensivists," (in eng), *Acta bio-medica : Atenei Parmensis*, vol. 93, no. 5, p. e2022297, Oct 26 2022, doi: 10.23750/abm.v93i5.13626.