information

Article

A Comprehensive Review of Deep Learning: Architectures, Recent Advances, and Applications

Ibomoiye Domor Mienye *,† and **Theo G. Swart** †

Institute for Intelligent Systems, University of Johannesburg, Johannesburg 2006, South Africa; tgswart@uj.ac.za
* Correspondence: ibomoiyem@uj.ac.za
† These authors contributed equally to this work.

Abstract: Deep learning (DL) has become a core component of modern artificial intelligence (AI), driving significant advancements across diverse fields by facilitating the analysis of complex systems, from protein folding in biology to molecular discovery in chemistry and particle interactions in physics. However, the field of deep learning is constantly evolving, with recent innovations in both architectures and applications. Therefore, this paper provides a comprehensive review of recent DL advances, covering the evolution and applications of foundational models like convolutional neural networks (CNNs) and Recurrent Neural Networks (RNNs), as well as recent architectures such as transformers, generative adversarial networks (GANs), capsule networks, and graph neural networks (GNNs). Additionally, the paper discusses novel training techniques, including self-supervised learning, federated learning, and deep reinforcement learning, which further enhance the capabilities of deep learning models. By synthesizing recent developments and identifying current challenges, this paper provides insights into the state of the art and future directions of DL research, offering valuable guidance for both researchers and industry experts.

Keywords: deep learning; GAN; GRU; LLM; LSTM; machine learning; NLP; transformers

check for updates

Citation: Mienye, I.D.; Swart, T.G. A Comprehensive Review of Deep Learning: Architectures, Recent Advances, and Applications. *Information* **2024**, *15*, 755. https://doi.org/10.3390/info15120755

Academic Editor: Nicolas Guelfi

Received: 21 September 2024
Revised: 20 November 2024
Accepted: 21 November 2024
Published: 27 November 2024

1. Introduction

Deep learning (DL) has significantly transformed the field of artificial intelligence (AI), achieving excellent performance in different applications and demonstrating robust capabilities in handling vast amounts of data and complex computations [1–3]. This field, a subset of machine learning (ML), utilizes architectures comprising numerous layers of nodes or neurons, where each layer is designed to model increasingly complex patterns in data [4]. Initially centered around simple models like perceptrons, DL has evolved to include sophisticated neural networks capable of performing a variety of tasks, such as image recognition and natural language processing (NLP).

The evolution of these DL architectures has led to significant enhancements in model performance and expanded the application areas of deep learning [5–7]. In recognition of DL's interdisciplinary impact, recent milestones highlight its significant contributions to scientific fields. For instance, the 2023 Nobel Prize in Chemistry was awarded for innovations in DL applications within chemistry, particularly in molecular design and reaction predictions, which indicates DL's role in accelerating research and discovery in complex scientific domains.

Meanwhile, the field of deep learning is continually advancing, with novel architectures and improved learning algorithms emerging at a rapid pace. For instance, the introduction of models like transformers by Vaswani et al. [8] has reshaped the understanding of sequence processing, a crucial breakthrough in natural language processing. Similarly, advancements in areas such as reinforcement learning and generative models demonstrate the growing versatility and depth of neural networks [9,10]. Emerging trends in DL reflect a shift towards personalized and privacy-preserving applications. For instance,

personalized healthcare is gaining traction through DL-driven diagnostic and predictive models tailored to individual genetic and clinical data. In addition, edge AI and federated learning are prominent in addressing privacy concerns, particularly in healthcare and finance, by enabling model training directly on devices while safeguarding sensitive data. These trends indicate an ongoing evolution of DL technologies towards user-centric, secure, and accessible applications.

In the past, researchers have conducted several reviews, such as those by LeCun et al. [11], Bengio et al. [12], and Janiesch et al. [13], which introduced the basic concepts and building blocks of deep learning. Meanwhile, reviews by Matsuo et al. [14] and Xiao et al. [15] provided comprehensive overviews of DL, covering foundational models like convolutional neural networks (CNNs) and recurrent neural networks (RNNs). Other recent reviews have highlighted advancements and focused on specific applications, such as the integration of deep learning in autonomous systems [16–18], finance [19–22], and healthcare [23,24]. However, the rapid advancements in DL architectures and their widespread applications necessitate a review that captures the recent innovations, analyzes their implications, and anticipates future trends.

Therefore, this review aims to provide a comprehensive review that not only covers various architectural innovations but also explores innovative training techniques and connects these developments with their practical impacts and theoretical implications. Additionally, this review will also discuss the challenges faced by current models. This comprehensive approach will provide valuable insights for both researchers and industry experts. The study aims to cover foundation architectures, recent innovations, emerging architectures, and advances in training techniques. The specific architectures include CNNs, recurrent neural networks (RNNs), autoencoders, transformers, generative adversarial networks (GANs), Capsule networks, and graph neural networks (GNNs). Meanwhile, the advances in training techniques include self-supervised learning, transfer learning, federated learning, deep reinforcement learning, few-shot learning, and one-shot learning.

The paper is structured as follows: Section 2 presents a background and reviews related works. Section 3 provides an overview of deep learning, including key components of typical architectures. Section 4 discusses advances in DL architectures, from foundational models to recent innovations. Section 5 covers advancements in training techniques, Section 6 provides an overview of DL applications, and Section 7 focuses on recent applications in the literature. Section 8 addresses ongoing challenges and future research directions. Finally, Section 9 concludes the review.

2. Background

Deep learning architectures have been widely employed in the literature, with numerous studies contributing to its rapid evolution. One of the foundational studies in the development of CNNs was by LeCun et al. [25], who demonstrated the practical application of backpropagation for training multi-layer neural networks, specifically CNNs. This work was instrumental in applying CNNs to digit recognition tasks, particularly with the development of the LeNet architecture, which is instrumental in the widespread adoption of CNNs in image processing.

The introduction of AlexNet by Krizhevsky et al. [4] marked a significant milestone in deep learning. AlexNet's success in the ImageNet challenge demonstrated the potential of deep CNNs to handle complex image recognition tasks, leading to increased interest and research into deep learning across various domains. RNNs gained prominence with the work of Hochreiter and Schmidhuber [26], who introduced Long Short-Term Memory (LSTM) networks. LSTM networks were designed to overcome the vanishing gradient problem associated with standard RNNs, enabling them to learn long-range dependencies within data sequences. This capability has been critical in advancing sequence modeling tasks such as natural language processing and speech recognition.

The concept of self-supervised learning was further explored by Devlin et al. [27] in their development of Bidirectional Encoder Representations from Transformers (BERT),

which revolutionized natural language processing. BERT utilizes transformer architectures in a novel training approach that leverages unlabeled data, setting new state-of-the-art benchmarks for a variety of NLP tasks. Recent innovations have also focused on improving the efficiency and adaptability of DL models. Howard et al. [28] introduced MobileNets, which uses depth-wise separable convolutions to build lightweight deep neural networks for mobile and edge-device applications. This work shows the industry's shift towards developing computationally efficient models that do not compromise performance.

Furthermore, the integration of deep learning with reinforcement learning has led to the development of models capable of mastering complex games and tasks. Mnih et al. [29] presented a model that combined Q-learning with deep learning to create Deep Q-Networks (DQNs), enabling these models to perform at human-level capabilities on Atari games. This integration showcases the robustness of DL in understanding and interacting with environments in a way that mimics human learning. Other recent reviews have explored different architectures and applications [18–21,23]. While recent reviews have made significant contributions to our understanding of deep learning architectures, they often focus on specific subsets of the field, such as enhancements in neural network efficiency or applications within specific domains.

Therefore, this study provides a broader scope, offering a comprehensive review that encompasses a wider range of architectural innovations and training techniques. Moreover, unlike previous reviews that may not deeply connect architectural advances with their real-world applications, this review seeks to bridge this divide by providing detailed case studies and examples of how these technologies are being implemented across various industries. This approach is essential for providing a more holistic understanding of the current state of deep learning technology and potential future innovations, which will benefit both researchers and practitioners.

3. Overview of Deep Learning

Deep learning has evolved significantly over the years, fundamentally altering various fields with its capability to model complex patterns. Traditional DL models, particularly CNNs and RNNs, have laid the foundation for modern advancements in AI. CNNs, initially designed for image processing tasks, utilize layers of convolution operations to capture spatial hierarchies in data, leading to groundbreaking performance in areas such as image and video recognition [19]. Pioneering works like LeCun et al.'s [30] application of CNNs to digit recognition and Krizhevsky et al.'s [4] AlexNet model demonstrated the powerful capabilities of deep networks in handling complex visual tasks with much greater accuracy than previous techniques. On the other hand, RNNs were designed to handle sequence data, with applications ranging from speech recognition to language modeling [31]. The introduction of LSTM units addressed challenges related to learning long-range dependencies, enhancing the model's effectiveness by preventing the vanishing gradient problem typical of earlier versions.

A foundational theoretical insight underlying neural networks is their universal approximation property. The universal approximation theorem, first established by Hornik et al. [32], states that, given a non-linear activation function, a neural network with a single hidden layer containing a sufficient number of neurons can approximate any continuous function on a compact subset of $\mathbb{R}^n$ to any desired degree of accuracy. This theorem highlights the theoretical flexibility of neural networks to represent complex mappings, explaining why even relatively simple architectures can model intricate relationships in data, given enough capacity.

Meanwhile, neural networks are often split into shallow and deep learning. A fundamental distinction between shallow and deep learning models is in the number of hidden layers within the network. Shallow networks, typically with one or two hidden layers, are limited in their ability to learn complex features, while deep neural networks with multiple hidden layers are capable of hierarchical feature extraction and more sophisticated data representations [19]. Figure 1 illustrates this difference, showing a simple artificial neural network (ANN) with a single hidden layer compared to a deep neural network with

multiple hidden layers. The depth of these networks allows them to learn more intricate patterns in data, a principle that underlies much of the progress in modern AI.

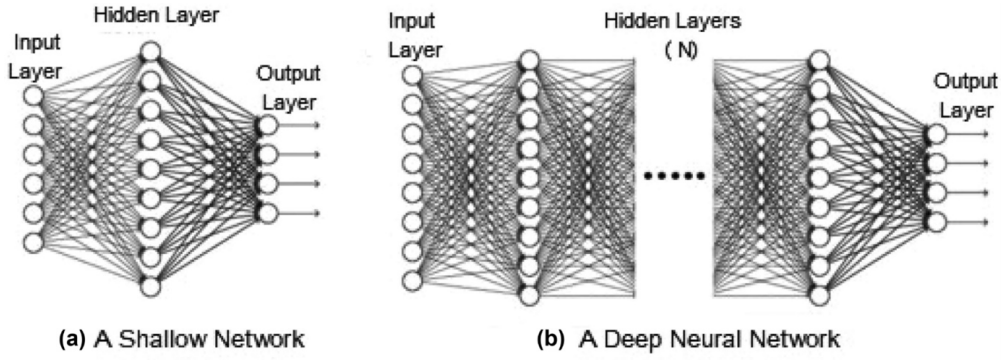

Figure 1. Comparison between (**a**) shallow network and (**b**) deep neural network [33].

3.1. Components of Deep Learning Models

Deep learning models are composed of various fundamental building blocks that enable them to learn from data, make predictions, and optimize performance. These components include layers, activation functions, loss functions, and optimization algorithms, each playing a critical role in the model's ability to learn and generalize from data.

3.1.1. Layers

Layers are the fundamental building blocks of deep learning models, determining how data are processed and transformed as they move through the network. A typical deep learning model consists of an input layer, one or more hidden layers, and an output layer [34]. The input layer receives the raw data, such as images or text, which are then passed through the hidden layers where various computations and transformations occur. The hidden layers are where the model learns to extract relevant features and patterns from the data. Finally, the output layer produces the final prediction or classification based on the processed information. In addition to these standard layers, deep learning models often incorporate specialized layers designed for specific tasks. For example, convolutional layers are used in CNNs to detect spatial hierarchies in image data by applying filters that learn to recognize patterns like edges and textures. Dense or fully connected layers, typically found towards the end of a network, connect each neuron to every neuron in the previous layer, allowing for high-level reasoning based on the features extracted by preceding layers [35]. Recurrent layers, such as those used in RNNs and their variants like LSTMs and gated recurrent units (GRUs), are designed to capture temporal dependencies in sequential data, making them suitable for tasks involving time series or natural language processing [34]. These specialized layers enable deep learning models to handle a wide range of complex tasks by effectively learning and representing intricate patterns in the data.

3.1.2. Activation Functions

Activation functions are crucial components of deep learning models as they introduce non-linearity into the network, enabling it to learn complex patterns and make sophisticated predictions. Without activation functions, a neural network would simply perform linear transformations, limiting its ability to model complex relationships in the data [36]. Common activation functions include the sigmoid function, Rectified Linear Unit (ReLU), and softmax function, each serving different purposes within a model. The sigmoid function maps input values to the range (0, 1), making it useful for binary classification problems where outputs are interpreted as probabilities [37]. However, sigmoid functions can suffer from vanishing gradients, which makes them less effective in deep networks. It is represented mathematically as:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{1}$$

Meanwhile, ReLU, defined as $\text{ReLU}(x) = \max(0, x)$, addresses this issue by outputting the input directly if it is positive and zero otherwise. ReLU is the most commonly used activation function in hidden layers of deep learning models due to its simplicity and effectiveness in mitigating the vanishing gradient problem, allowing for faster training and improved convergence [38]. Furthermore, the softmax function is represented as:

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}} \tag{2}$$

z_i represents the logit for class i, and the denominator normalizes the output to a probability distribution over all possible classes. It is typically used in the output layer of a neural network for multi-class classification problems. It converts raw model outputs (logits) into probabilities that sum to one, which is essential for interpreting the model's predictions as class probabilities [37]. Other activation functions, such as the hyperbolic tangent (tanh) and leaky ReLU, are also widely used. The tanh function is similar to the sigmoid but maps inputs to the range $(-1, 1)$, often providing better performance for hidden layers by centering the data around zero. The Leaky ReLU, a variation of the ReLU, allows a small gradient when the input is negative, addressing the "dying ReLU" problem where neurons can become inactive during training [39].

3.1.3. Loss Functions

Loss functions, also known as cost functions or objective functions, are fundamental components in training deep learning models as they quantify the difference between the model's predictions and the actual target values. Loss functions guide the optimization process by indicating how the model parameters should be adjusted to minimize errors and improve predictive accuracy [40]. The choice of loss function depends on the specific task and data characteristics, as it directly influences how the model learns during training. One of the most common loss functions is the Mean Squared Error (MSE), primarily used in regression tasks. MSE calculates the average squared difference between the actual target values y_i and the model's predictions $\hat{y}_i$:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{3}$$

where n is the number of samples, y_i represents the true value, and $\hat{y}_i$ denotes the predicted value [40,41]. MSE penalizes larger errors more heavily due to the squaring term, making it sensitive to outliers. This sensitivity is beneficial when large errors need to be significantly reduced, but it can also lead to challenges if the dataset contains outliers that skew the loss function. Another widely used loss function is the cross-entropy loss, which is effective for classification tasks [42]. Cross-entropy measures the difference between the true label distribution p and the predicted label distribution q, effectively capturing the dissimilarity between the actual and predicted class probabilities:

$$\text{Cross Entropy} = -\sum_i p(y_i) \log q(y_i) \tag{4}$$

In the context of binary classification, the cross-entropy loss simplifies to binary cross-entropy loss, which is used when the output is a probability representing two classes (0 or 1). For multi-class classification problems, categorical cross-entropy loss extends this concept to multiple classes, where $p(y_i)$ is a one-hot encoded vector representing the true class and $q(y_i)$ is the predicted probability distribution over all classes [40]. Cross-entropy loss is favored in classification problems because it heavily penalizes confident but incorrect predictions, encouraging the model to output probabilities that closely match the actual

class distributions. Beyond MSE and cross-entropy, other loss functions are tailored to specific types of tasks and data characteristics. For instance, the Huber Loss combines the best aspects of MSE and Mean Absolute Error (MAE) by being quadratic for small errors and linear for large errors [43]. It is defined as:

$$\text{Huber Loss} = \begin{cases} \frac{1}{2}(y_i - \hat{y}_i)^2 & \text{for } |y_i - \hat{y}_i| \leq \delta \\ \delta(|y_i - \hat{y}_i| - \frac{1}{2}\delta) & \text{otherwise} \end{cases} \tag{5}$$

where δ is a threshold parameter that determines the transition between the quadratic and linear regions. The Huber Loss is less sensitive to outliers than MSE, making it a robust choice for regression tasks with noisy data [43]. Selecting the appropriate loss function is crucial as it directly impacts the model's learning behavior and performance. The choice depends on the nature of the data, the specific requirements of the task, and the need for robustness against certain types of errors or data distributions.

3.1.4. Optimization Algorithms

Optimization algorithms are essential in deep learning as they adjust the model's parameters to minimize the loss function. Effective optimization is crucial for training deep learning models, as it directly influences the convergence speed and the quality of the final solution. Several optimization algorithms have been developed to improve training efficiency and performance, each with its advantages and considerations.

- Stochastic Gradient Descent: Stochastic Gradient Descent (SGD) is one of the foundational optimization algorithms in deep learning [35,44]. It updates the model parameters θ by computing the gradient of the loss function $\nabla_\theta \mathcal{L}(\theta)$ with respect to the model parameters and moving in the opposite direction of the gradient to minimize the loss. The update rule for SGD is given by:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{L}(\theta) \tag{6}$$

  where η is the learning rate, a hyperparameter that determines the step size of each update. Unlike batch gradient descent, which computes the gradient using the entire dataset, SGD approximates the gradient using a randomly selected subset of the data (a mini-batch), which reduces the computational cost and allows for faster iterations. However, SGD can be sensitive to the choice of learning rate and may have difficulty converging to the global minimum due to its stochastic nature.
- AdaGrad: The Adaptive Gradient Algorithm (AdaGrad) adapts the learning rate to the parameters, performing larger updates for infrequent and smaller updates for frequent parameters [45,46]. It is well suited for dealing with sparse data and performs well for sparse gradient problems such as natural language processing and image recognition. However, its learning rate continually decreases, which can cause the algorithm to stop learning prematurely. The update rule for AdaGrad is:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_{t,t} + \epsilon}} \nabla_\theta \mathcal{L}(\theta) \tag{7}$$

  where G_t is the diagonal matrix where each diagonal element $G_{t,t}$ is the sum of the squares of the gradients with respect to θ_t up to time step t [45].
- RMSProp: Root Mean Square Propagation (RMSProp) is an adaptive learning rate method that addresses the diminishing learning rates of AdaGrad by keeping a moving average of the squared gradients [47]. This prevents the learning rate from becoming too small and allows for more continuous training. The parameter update rule in RMSProp involves dividing the learning rate by the root of the mean squared gradients:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} \nabla_\theta \mathcal{L}(\theta) \tag{8}$$

where $E[g^2]_t$ is the exponentially weighted moving average of the squared gradients. RMSProp is effective in non-stationary environments where the loss function changes dynamically, and it is widely used in training RNNs.

- Adam Optimizer: The Adaptive Moment Estimation (Adam) optimizer is a more sophisticated optimization algorithm that combines the advantages of two other extensions of SGD: AdaGrad, which adapts the learning rate for each parameter based on its past gradients, and RMSProp, which scales the learning rate based on a moving average of recent gradients [48]. Adam maintains separate adaptive learning rates for each parameter by estimating the first moment (mean) $\hat{m}_t$ and the second moment (uncentered variance) $\hat{v}_t$ of the gradients. The parameter update rule for Adam is:

$$\theta_{t+1} = \theta_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{9}$$

where $\hat{m}_t$ and $\hat{v}_t$ are calculated as:

$$\hat{m}_t = \frac{\beta_1 m_t}{1 - \beta_1^t}, \quad \text{with} \quad m_t = \beta_1 m_{t-1} + (1 - \beta_1)\nabla_\theta \mathcal{L}(\theta) \tag{10}$$

$$\hat{v}_t = \frac{\beta_2 v_t}{1 - \beta_2^t}, \quad \text{with} \quad v_t = \beta_2 v_{t-1} + (1 - \beta_2)(\nabla_\theta \mathcal{L}(\theta))^2 \tag{11}$$

β_1 and β_2 are decay rates for the moving averages, and ϵ is a small constant to prevent division by zero. Adam has become one of the most popular optimizers in deep learning due to its ability to handle sparse gradients, adjust learning rates dynamically, and maintain stable updates [48,49]. It often achieves faster convergence than SGD and its variants, especially on problems with noisy or sparse gradients.

4. Advances in Deep Learning Architectures

DL has seen significant architectural innovations over the years, driving advances across numerous domains. This section explores foundational architectures that have laid the groundwork for modern DL and recent innovations that push the boundaries of what is possible with neural networks.

4.1. Foundational Architectures and Their Evolution

The development of deep learning has been marked by several key architectures that have fundamentally shaped the field. CNNs and RNNs are foundational architectures that have revolutionized how neural networks learn from data.

4.1.1. Convolutional Neural Networks

CNNs have been instrumental in advancing computer vision tasks such as image classification, object detection, and image segmentation. CNNs leverage convolutional layers to automatically learn spatial hierarchies of features from input data, allowing the network to capture patterns such as edges, textures, and complex shapes. The typical structure of a CNN involves a series of convolutional layers, pooling layers, and fully connected layers that culminate in a prediction [50]. Figure 2 illustrates a typical CNN architecture.

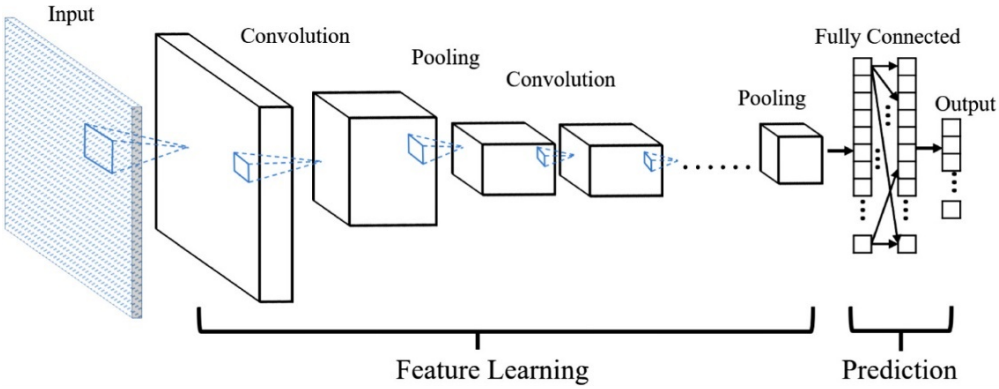

Figure 2. A CNN architecture [51].

In a CNN, the convolutional layer applies a set of learnable filters (or kernels) to the input data to produce a feature map. Mathematically, the output of a convolution operation for a given filter k can be expressed as:

$$f_k(x,y) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} w_{i,j} \cdot x_{(i+x),(j+y)} + b \tag{12}$$

where x is the input image, $w_{i,j}$ represents the filter weights, b is the bias term, and $f_k(x,y)$ is the resulting feature map [52]. The pooling layers, often max-pooling, downsample the feature maps to reduce their dimensionality, which helps to prevent overfitting and reduces computational complexity. This is achieved by taking the maximum (or average) value in each patch of the feature map:

$$p(x,y) = \max_{i=0}^{m-1} \max_{j=0}^{n-1} f(x+i, y+j) \tag{13}$$

After multiple convolutional and pooling layers, the network typically includes fully connected layers, which operate as in a standard neural network to combine features and make a final classification or regression prediction. Architectures like LeNet [30], AlexNet [4], VGGNet [53], and ResNet [54] have progressively improved the capability of CNNs to handle increasingly complex visual tasks. Innovations such as skip connections in ResNet allow gradients to flow more effectively through deeper networks, addressing issues like vanishing gradients and enabling the training of very deep networks. However, despite their success, CNNs are limited by their dependence on large amounts of labeled data and high computational requirements, which has driven the development of more efficient and flexible architectures.

4.1.2. Recurrent Neural Networks

RNNs are designed for sequential data processing, making them crucial for tasks involving time series or language data, such as natural language processing, language modeling, machine translation, and sentiment analysis [55]. Unlike feedforward neural networks, RNNs maintain a hidden state that captures information from previous time steps, allowing them to model temporal dependencies and context over sequences. Mathematically, the hidden state h_t at time step t in an RNN is computed as:

$$h_t = \sigma(W_{hh} h_{t-1} + W_{xh} x_t + b_h) \tag{14}$$

where x_t is the input at time step t, h_{t-1} is the hidden state from the previous time step, W_{xh} and W_{hh} are weight matrices, b_h is a bias vector, and σ is a non-linear activation function, typically tanh or ReLU. The output y_t at time step t is then calculated as:

$$y_t = \sigma(W_{hy} h_t + b_y) \tag{15}$$

where W_{hy} is the output weight matrix and b_y is the output bias. While traditional RNNs can theoretically learn dependencies over long sequences, in practice, they suffer from issues such as vanishing and exploding gradients, which make training challenging. To address these problems, advanced variants like LSTM networks, gated recurrent units (GRUs), bidirectional RNNs, and deep RNNs have been developed, each introducing unique mechanisms to improve performance on sequence modeling tasks.

- Long Short-Term Memory Networks: LSTM networks introduce a memory cell c_t that maintains its state over time, effectively managing long-range dependencies in sequences. The memory cell is regulated by three gates: the input gate i_t, forget gate f_t, and output gate o_t [56]. These gates control the flow of information into and out of the memory cell:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{16}$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{17}$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{18}$$
$$c_t = f_t * c_{t-1} + i_t * \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{19}$$
$$h_t = o_t * \tanh(c_t) \tag{20}$$

  where f_t, i_t, and o_t are the forget, input, and output gates, respectively, which control cell state updates [22]. LSTMs effectively handle the vanishing gradient problem, allowing gradients to flow back through time more effectively during training, making them well-suited for tasks like speech recognition and language modeling, where capturing long-term dependencies is crucial.
- Gated Recurrent Units: GRUs are a simplified variant of LSTMs, combining the forget and input gates into a single update gate z_t and incorporating a reset gate r_t [57]. GRUs reduce the complexity of LSTMs while maintaining similar performance, especially in scenarios where computational efficiency is a priority. The equations governing a GRU's operations are:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \tag{21}$$
$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \tag{22}$$
$$\tilde{h}_t = \tanh(W_h \cdot [r_t * h_{t-1}, x_t] + b_h) \tag{23}$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \tag{24}$$

  In GRUs, the update gate z_t controls how much of the previous hidden state is retained, and the reset gate r_t determines how much of the past information to forget [22]. This architecture simplifies the learning process by reducing the number of parameters, making GRUs computationally more efficient while still being effective for various sequence modeling tasks.
- Bidirectional RNNs: Bidirectional RNNs enhance the learning capacity of traditional RNNs by processing sequences in both forward and backward directions [58]. This bidirectional approach provides the network with additional context from both past and future time steps, which can significantly improve performance in tasks where context from both directions is crucial, such as in natural language understanding and speech processing. In bidirectional RNNs, two hidden layers are used: one for processing the input sequence from start to end (forward pass) and another from end to start (backward pass). The outputs of these two layers are then combined, often by concatenation, to form the final output [59]. This approach allows the model to utilize information from the entire input sequence, making it effective for tasks requiring comprehensive contextual understanding, such as named entity recognition (NER) and part-of-speech (POS) tagging. Bidirectional RNNs are beneficial in applications where the entire input sequence is available beforehand, and both past and future contexts are needed to make accurate predictions. However, they are not suitable for

real-time processing tasks where future inputs are not available, limiting their use in certain scenarios like real-time video processing or online speech recognition.

- Deep RNNs: Deep RNNs extend the architecture of standard RNNs by stacking multiple recurrent layers on top of each other [60]. This depth allows the model to capture more complex patterns and representations in sequential data, improving performance on tasks that require a hierarchical understanding of sequences, such as hierarchical language modeling or multi-step time series forecasting. By stacking multiple layers, deep RNNs can learn representations at multiple levels of abstraction, similar to how deep convolutional networks operate for image data. Each layer in a deep RNN can capture different levels of temporal dependencies, allowing the network to model complex dynamics over longer sequences more effectively. However, training deep RNNs poses challenges such as vanishing gradients and increased computational requirements. Techniques like residual connections and careful initialization are often used to mitigate these issues and ensure effective training.
  Despite their effectiveness, deep RNNs can be computationally intensive and challenging to optimize, especially for very deep architectures [61]. Recent advances have focused on improving their training efficiency and exploring new ways to incorporate depth without compromising stability, such as using skip connections or hybrid models that combine RNNs with other architectures like CNNs or transformers.

4.1.3. Autoencoders and Their Variants

Autoencoders (AEs) are a type of neural network designed to learn efficient representations of data, typically for the purpose of dimensionality reduction or feature learning [62,63]. The structure of an autoencoder consists of two main components: an encoder, which maps the input data to a latent space representation, and a decoder, which reconstructs the input from this latent representation. The training objective of an autoencoder is to minimize the reconstruction error, which is typically measured using the mean squared error between the input and the reconstructed output:

$$\mathcal{L}(x, \hat{x}) = \|x - \hat{x}\|^2 \tag{25}$$

where x is the input data and $\hat{x}$ is the reconstructed output from the decoder.

- Deep Autoencoders: Deep autoencoders extend the basic concept of autoencoders by employing multiple hidden layers in both the encoder and decoder. This deep architecture allows the model to learn more complex, hierarchical representations of the input data [64]. As data passes through multiple layers, each layer can capture progressively more abstract features, making deep AE useful in applications requiring feature extraction and data compression, such as image compression and denoising. The advantage of deep AE is their ability to model intricate patterns within the data, which shallow architectures might miss. For example, in image processing tasks, deep AE can be trained to reduce the dimensionality of high-resolution images while retaining essential details that are crucial for tasks like object recognition and image segmentation.
- Variational Autoencoders: Variational autoencoders (VAEs) introduce a probabilistic framework to the traditional Autoencoder. Unlike standard autoencoders that encode an input into a fixed latent space vector, VAEs encode the input into a distribution over the latent space [65]. Typically, this is a Gaussian distribution characterized by a mean and variance for each latent dimension. The encoder network outputs these parameters, and the latent variable is then sampled from this distribution:

$$z \sim q(z|x) = \mathcal{N}(\mu(x), \sigma^2(x)) \tag{26}$$

  The decoder reconstructs the input from the sampled latent vector. The training objective for VAEs includes two components: the reconstruction loss, which measures how well the decoder reconstructs the input from the latent space, and a regularization

term, which is the Kullback–Leibler (KL) divergence. This KL divergence ensures that the distribution $q(z|x)$ is close to a prior distribution $p(z)$, usually a standard normal distribution:

$$\mathcal{L}_{VAE} = \mathbb{E}_{q(z|x)}[\log p(x|z)] - KL(q(z|x)\|p(z)) \quad (27)$$

The regularization term acts as a form of regularization by penalizing deviations of the latent variable distribution from the prior, ensuring that the latent space is continuous and that similar inputs result in similar latent representations. This property allows VAEs to generate new data points by sampling from the latent space, making them powerful tools for generative modeling [65]. VAEs are widely used in applications such as generating synthetic data, image synthesis, and anomaly detection, where generating plausible data variations is essential.

- Denoising Autoencoders: Denoising autoencoders (DAEs) are an extension of traditional AEs, specifically designed to reconstruct clean inputs from noisy or corrupted versions of the data [66]. DAEs are trained by deliberately adding noise to the input data and optimizing the network to recover the original, uncorrupted data during reconstruction. The key objective of a DAE is to learn robust representations that capture the essential features of the input data, even when some aspects of the data are missing or distorted.
  The process of learning to remove noise improves the model's generalization ability, making DAEs effective in applications where data quality is affected by noise, such as image and audio processing. For instance, in image denoising tasks, DAEs learn to reconstruct clean images from noisy inputs, preserving fine details while removing the noise. In speech enhancement tasks, DAEs can remove background noise, enhancing the clarity of audio signals. This makes DAEs highly valuable in improving the performance of models operating in noisy environments, such as real-time image processing systems or voice recognition systems.
- Sparse Autoencoders: Sparse autoencoders (SAEs) introduce a sparsity constraint on the hidden layer activations, ensuring that only a few neurons are active at any given time. This sparsity constraint is typically achieved by adding a regularization term to the loss function that penalizes non-sparse activations, such as an L1 regularization term [67]. The motivation behind SAEs is to force the network to learn more efficient and compact representations of the input data, which can lead to the discovery of interpretable features that are relevant to the task at hand.
  The sparsity constraint makes SAEs particularly useful in applications where extracting meaningful features is essential. For example, in data compression tasks, sparse representations reduce the dimensionality of the input data, enabling efficient storage and transmission of information. In classification tasks, the sparse features learned by SAEs often capture the most important aspects of the data, improving the classifier's performance. SAEs are also beneficial for feature extraction in large-scale datasets, where compact, efficient representations of the data are crucial for downstream ML models.
- Contractive Autoencoders: Contractive autoencoders (CAEs) build upon the traditional autoencoder architecture by introducing a penalty to the loss function that encourages the model to learn representations that are insensitive to small changes in the input. This is achieved by adding a regularization term that penalizes the Frobenius norm of the Jacobian matrix of the hidden representation with respect to the input [68]. The Jacobian matrix quantifies how much the hidden representation changes in response to small perturbations in the input data, and minimizing this norm ensures that the learned representations are stable and robust to small variations. CAEs are useful for tasks requiring robust feature extraction in the presence of noise or small perturbations in the data. For instance, in image analysis, CAEs can learn invariant representations of objects, allowing the model to focus on the core structure of the image while ignoring small changes like shifts or distortions. This robustness

makes CAEs highly applicable in domains like medical image analysis, where the ability to capture the essential structure of the data while being robust to minor variations is crucial.

Autoencoders and their variants have been applied across various domains, leveraging their capabilities for dimensionality reduction, feature extraction, anomaly detection, and data generation [68]. They are highly effective in detecting anomalies by learning to reconstruct normal data patterns. Any significant deviation in reconstruction, indicating a poor match, can flag a data point as an anomaly. This approach is widely used in fraud detection, network intrusion detection, and medical diagnosis. Furthermore, DAEs are used to remove noise from data, improving data quality and usability, particularly in image processing tasks where the goal is to clean up corrupted or noisy images. Autoencoders are also used in data compression by reducing dimensionality, which is beneficial for the storage and transmission of data, especially in image and video compression. Meanwhile, VAEs and other generative AEs are powerful tools for generating synthetic data that can augment datasets, provide privacy-preserving data alternatives, and improve model training under data scarcity. Applications include generating realistic images, simulating potential outcomes in healthcare, and creating artificial data for training robust machine learning models.

4.2. Recent Innovations in Deep Learning Architectures

Building upon the foundation set by CNNs and RNNs, recent innovations in deep learning have introduced more sophisticated architectures designed to address their limitations and expand their applicability.

4.2.1. Transformers

The transformer architecture, introduced by Vaswani et al. [8], has revolutionized the approach to sequence processing tasks, particularly in NLP. This model diverges from traditional methods that relied on recurrent layers, instead employing self-attention mechanisms that allow for parallel processing of input data. This shift significantly enhances training efficiency and model performance on large datasets. The transformer is described by:

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V \tag{28}$$

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{29}$$

where X represents the input data, W_Q, W_K, W_V are the learned weight matrices, and d_k represents the dimensionality of the keys, enhancing the model's ability to focus on different parts of the input sequence simultaneously [8]. Algorithm 1 outlines the basic training procedure for a transformer model, specifically focusing on how the self-attention mechanism is integrated. Transformers have significantly impacted several domains beyond NLP, such as computer vision, where they have been adapted to handle image data in models like Vision transformer (ViT) [69]. In speech processing, the transformer architecture has enabled more accurate and faster speech recognition systems. Furthermore, their ability to handle sequential data efficiently makes them ideal for applications in financial forecasting where they can process time-series data for predictive analysis. The introduction of architectures like Bidirectional Encoder Representations from Transformers (BERT) and Generative Pre-trained Transformer (GPT) series has demonstrated the transformer's impact in NLP.

BERT's bidirectional training of transformers allows for a deeper sense of language context, improving performance on tasks such as sentence classification, entity recognition, and question answering [70]. GPT-3 and GPT-4, with their vast number of parameters, push the boundaries of language models in generating coherent and contextually relevant text on a near-human level. Moreover, transformers are facilitating advances in areas such as generative tasks and reinforcement learning, where their ability to model complex patterns and relationships in data is invaluable [71]. Their effectiveness and scalability are paving

the way for next-generation models that could offer even more profound insights across diverse domains, underlining their significant contribution to the advancement of both technology and deep learning methodologies.

Algorithm 1 Training the Transformer Model

Input: Training data $\{(X^{(i)}, Y^{(i)})\}_{i=1}^{N}$, where $X^{(i)}$ is the input sequence and $Y^{(i)}$ is the target sequence.
Initialize: Randomly initialize parameters W_Q, W_K, W_V, and other model parameters.
while not converged **do**
  for each batch **do**
    Compute attention scores for each element in the batch using:

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V$$

$$\text{Attention} = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

    Apply the attention mechanism to transform the input sequence.
    Compute the output sequence $\hat{Y}$ using the transformed input.
    Calculate loss L between $\hat{Y}$ and Y.
    Update model parameters using backpropagation to minimize L.
  end for
end while

4.2.2. Graph Neural Networks

GNNs have transformed the way data represented as graphs is processed and analyzed. This includes a vast array of applications where data are inherently structured in graphs, such as social networks, chemical molecules, and communication networks [72]. Traditional neural networks are less effective for such tasks due to their inability to directly process graph structures, which include nodes (or vertices) and edges. GNNs specifically address this challenge by updating the representation of each node based on the features of its neighboring nodes, as described by the following equation:

$$h_i^{(l+1)} = \sigma\left(\sum_{j \in \mathcal{N}(i)} \frac{1}{c_{ij}} W^{(l)} h_j^{(l)} + b^{(l)}\right) \tag{30}$$

where $\mathcal{N}(i)$ represents the set of neighbors of node i, $W^{(l)}$ are the weight matrices specific to layer l, $b^{(l)}$ are the biases, and σ is a nonlinear activation function, such as the ReLU [72,73]. This aggregation framework enables GNNs to learn complex patterns in the data by iteratively passing and transforming node features through the network. The capability of GNNs to capture both node and structural information makes them useful in several key applications, including node classification, link prediction, and graph classification. In node classification, GNNs can predict properties of a node based on its attributes and its connections, which is useful in social network analysis for identifying influential users or in bioinformatics for predicting protein functions [74]. Meanwhile, link prediction involves predicting the likelihood of a connection between two nodes, applicable in recommendation systems and social network growth analysis. Regarding graph classification, GNNs can determine the properties of entire graphs, which is useful in drug discovery, where molecules can be classified into different therapeutic categories based on their structural properties.

4.2.3. Capsule Networks

Capsule networks (CapsNets), introduced by Sabour et al. [75], address some limitations of CNNs such as the loss of spatial hierarchy due to pooling layers. CapsNets enhance the processing of information by using groups of neurons known as "capsules",

which handle more complex tasks than the typical neurons in CNNs. This architecture is represented by:

$$\mathrm{v}_j = \sum_i c_{ij}\mathrm{s}_j = \sum_i c_{ij}\sigma\left(\mathrm{W}_{ij}\mathrm{u}_i + \mathrm{b}_j\right) \tag{31}$$

where u_i represents the output vector from a lower-level capsule, W_{ij} are transformation matrices that capture spatial hierarchies between features, c_{ij} are the "routing coefficients" that determine how much each capsule contributes to higher-level capsules, and σ is a non-linear activation function [75]. This dynamic routing mechanism, effectively a sophisticated attention mechanism, allows the network to focus on salient spatial relationships within the data. CapsNets have demonstrated improved performance in tasks that demand a strong understanding of spatial relationships. These tasks include object detection and segmentation, where maintaining the spatial hierarchies enables the model to perform accurately even in complex scenes with overlapping objects. In medical imaging, CapsNets facilitate more precise identification and classification of tumors and organ segmentation, outperforming traditional CNNs, which might overlook crucial spatial relationships [76].

4.2.4. Generative Adversarial Networks

GANs are a groundbreaking class of deep learning models consisting of two interconnected neural networks: the generator and the discriminator [77]. These networks engage in a continuous game-theoretic competition during training. The generator's objective is to synthesize data that mimic real-world data, effectively deceiving the discriminator, which itself aims to distinguish between the genuine data received from the dataset and the fake data produced by the generator. The generator, initiating with a random noise vector z from a latent space, transforms this input into a data instance $G(z)$, striving to match the statistical properties of real data. Conversely, the discriminator evaluates each piece of data it receives by assigning a probability to the likelihood of the data being real, either as $D(x)$ for real data or $D(G(z))$ for generated data. The GAN architecture is shown in Figure 3.

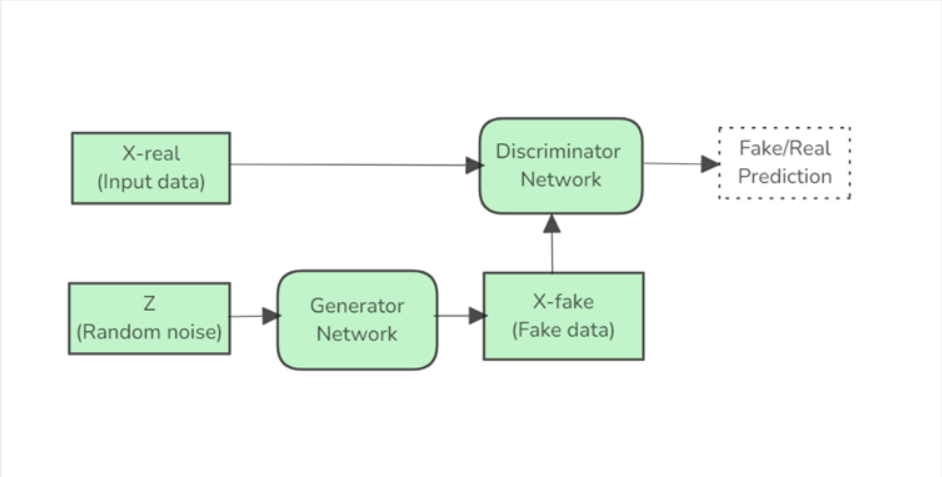

Figure 3. A GAN architecture.

This adversarial process is quantified through a minimax game described by the following objective function, which both the generator and discriminator aim to optimize from opposing perspectives:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{32}$$

Training GANs involves alternating updates to the discriminator and the generator, as outlined in Algorithm 2.

Algorithm 2 Training a Generative Adversarial Network

Initialize generator G and discriminator D with random weights.
while not converged **do**
  for each training step **do**
    Sample a minibatch of m noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from the noise prior $p_z(z)$.
    Sample a minibatch of m real data samples $\{x^{(1)}, \ldots, x^{(m)}\}$ from the data distribution $p_{data}(x)$.
    Compute the discriminator loss:

$$L_D = -\frac{1}{m}\sum_{i=1}^{m}\left[\log D(x^{(i)}) + \log(1 - D(G(z^{(i)})))\right]$$

    Update the discriminator by performing a gradient ascent step on L_D.
    Compute the generator loss:

$$L_G = -\frac{1}{m}\sum_{i=1}^{m}\log(D(G(z^{(i)})))$$

    Update the generator by performing a gradient descent step on L_G.
  end for
end while

The robustness of GANs has led to their effective use in creating synthetic financial datasets where genuine data are limited or biased. This application is vital in fields like credit scoring and market simulation, where robust models are crucial for accurate predictions and risk assessments. In these settings, GANs enrich the training datasets, thus enhancing the performance and reliability of predictive models used in financial decision-making.

4.3. Emerging Architectures

As the field of deep learning continues to evolve, new architectures are being developed to address the limitations of existing models and to push the boundaries of what neural networks can achieve. Emerging architectures such as Neural Ordinary Differential Equations, attention-based graph neural networks, and Neural Architecture Search models represent the next frontier in deep learning research. These architectures aim to provide more flexible, efficient, and interpretable solutions, offering new ways to handle complex data and optimize learning processes.

- Neural Ordinary Differential Equations: Neural Ordinary Differential Equations (Neural ODEs) are a class of models introduced by Chen et al. [78] that generalize the concept of residual networks (ResNets) by interpreting the layers of a neural network as discretized steps of a continuous transformation. Instead of stacking discrete layers, Neural ODEs model the hidden states as a continuous trajectory defined by an ordinary differential equation parameterized by a neural network. Given a hidden state $h(t)$ at time t, Neural ODEs define its evolution as:

$$\frac{dh(t)}{dt} = f(h(t), t; \theta) \tag{33}$$

  where f is a neural network with parameters θ [78]. The output at time t_1 can be computed from the initial state $h(t_0)$ by integrating over time:

$$h(t_1) = h(t_0) + \int_{t_0}^{t_1} f(h(t), t; \theta)\, dt \tag{34}$$

  This continuous formulation allows Neural ODEs to adaptively select the number of function evaluations required for a specific task, leading to potentially more efficient

models. Additionally, the continuous dynamics of Neural ODEs make them suitable for time-series data and systems governed by physical laws. They are useful for tasks requiring adaptive computation or modeling continuous processes, such as physics simulations and dynamic systems modeling.

- Attention-Based Graph Neural Networks: GNNs have been extended with attention mechanisms to better capture the dependencies and relationships within graph-structured data. Attention-based GNNs, such as the graph attention networks (GATs) introduced by Veličković et al. [79], use self-attention mechanisms to weigh the importance of neighboring nodes differently when aggregating information, allowing the model to focus on more relevant parts of the graph. In an attention-based GNN, the hidden representation h_i of a node i is updated by aggregating information from its neighbors $\mathcal{N}(i)$ using:

$$h_i' = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} W h_j\right) \tag{35}$$

where α_{ij} represents the attention coefficient that determines the importance of node j to node i, calculated as:

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(a^T[Wh_i || Wh_j]))}{\sum_{k \in \mathcal{N}(i)} \exp(\text{LeakyReLU}(a^T[Wh_i || Wh_k]))} \tag{36}$$

where W is a weight matrix applied to the node features, a is a learnable vector, and $||$ denotes concatenation [79]. The use of attention allows the model to dynamically focus on the most informative neighbors, enhancing its ability to learn from complex graph structures. This architecture has proven to be robust in tasks like node classification, link prediction, and graph classification, where relationships between nodes are not uniformly informative.

- Neural Architecture Search: Neural Architecture Search (NAS) is an automated approach to designing neural network architectures that optimizes both the architecture and its hyperparameters through a search algorithm [80]. NAS aims to discover architectures that achieve high performance on a given task without requiring extensive human expertise or manual experimentation. The NAS process involves defining a search space, a search strategy, and an evaluation strategy [81]. The search space represents the possible configurations of neural networks, including the types and connections of layers. The search strategy explores this space, often using reinforcement learning, evolutionary algorithms, or gradient-based optimization. The evaluation strategy assesses the performance of the candidate architectures. One popular NAS approach is the differentiable architecture search (DARTS) proposed by Liu et al. [82], which relaxes the search space to be continuous, allowing gradient-based optimization to be used for efficient exploration. In DARTS, the architecture parameters α are optimized simultaneously with the model weights θ using backpropagation:

$$\min_{\alpha} \mathcal{L}_{val}(\theta^*(\alpha), \alpha) \quad \text{where} \quad \theta^*(\alpha) = \arg\min_{\theta} \mathcal{L}_{train}(\theta, \alpha) \tag{37}$$

where $\mathcal{L}_{train}$ and $\mathcal{L}_{val}$ are the training and validation loss functions, respectively. The optimization process alternates between updating the architecture parameters and the network weights, enabling the discovery of optimal architectures that balance accuracy and efficiency. NAS has led to the development of state-of-the-art models for various tasks, including image classification, object detection, and language modeling [83]. The automatic design of neural networks through NAS not only reduces the time and effort required for manual architecture engineering but also uncovers novel architectures that outperform human-designed counterparts.

5. Advances in Training Techniques

Recent advancements in training techniques have significantly enhanced the efficiency and performance of deep learning models. This section explores key innovations such as self-supervised learning, few-shot learning, and federated learning, which address challenges like limited data availability, computational constraints, and model generalization.

5.1. Self-Supervised Learning

Self-supervised learning (SSL) represents a transformative approach in the training of deep learning models by utilizing unlabeled data to learn useful data representations [84]. This method is beneficial as it reduces the dependency on labeled data, which are often expensive and labor-intensive to obtain. SSL employs pretext tasks, where a model is trained to infer missing or hidden parts of the data based on the visible parts. For instance, in natural language processing, a model might predict the next word in a sentence or complete a missing part of a text. Mathematically, given an input sequence $x = (x_1, x_2, \ldots, x_n)$, the model predicts x_{n+1} using a function f parameterized by θ:

$$\hat{x}_{n+1} = f(x_1, x_2, \ldots, x_n; \theta) \tag{38}$$

The model parameters θ are optimized by minimizing a loss function $\mathcal{L}$, which typically assesses the difference between $\hat{x}_{n+1}$ and the actual subsequent item x_{n+1}. Another prevalent SSL method is contrastive learning, which involves learning to differentiate between similar (positive) and dissimilar (negative) data instances [85]. The contrastive loss function is expressed as:

$$\mathcal{L}(\theta) = -\log \frac{\exp(sim(z_i, z_j)/\tau)}{\sum_{k=1}^{N} \exp(sim(z_i, z_k)/\tau)} \tag{39}$$

where z_i, z_j represent embeddings of similar pairs, z_k represents embeddings of dissimilar pairs, sim is a similarity measure like cosine similarity, and τ is a temperature scaling parameter. Algorithm 3 describes self-supervised learning.

SSL has wide-ranging applications across various domains. For example, in NLP, SSL is used to pre-train models like BERT before fine-tuning on specific downstream tasks such as sentiment analysis and question-answering [86]. Meanwhile, in image processing, SSL helps in learning representations from unlabeled images, useful in tasks like object detection and image classification. Also, in audio processing, SSL techniques are employed to learn features from raw audio data, aiding in speech recognition and audio synthesis [87]. By enabling the extraction of meaningful patterns from unlabeled data, SSL not only enhances the efficiency of model training but also expands the potential applications of machine learning models, making significant contributions to the advancement of AI technologies.

Algorithm 3 Self-Supervised Learning Using Contrastive Loss

1: **Input:** Unlabeled dataset D
2: **Output:** Trained model f with parameters θ
3: Initialize model f with random weights θ
4: **while** not converged **do**
5: Sample a minibatch of inputs from D
6: Generate positive and negative pairs based on the pretext task
7: Compute embeddings z_i, z_j, z_k for all pairs in the batch
8: Compute contrastive loss using Equation (39)
9: Update model parameters θ using backpropagation
10: **end while**

5.2. Transfer Learning and Fine-Tuning

Transfer learning and fine-tuning are techniques that have become integral to the practice of deep learning, offering significant improvements in the way models are developed and deployed. Transfer learning typically involves taking a model that has been trained on a large, diverse dataset and adapting it to perform well on a more specialized, often smaller dataset, as shown in Figure 4. This approach allows for the leverage of previously learned features and knowledge, reducing the need for extensive data from the specialized domain and accelerating the training process [88]. Fine-tuning is a specific form of transfer learning where not only is the model adjusted to a new task, but more precisely, the pretrained model's layers are selectively retrained [89]. In many cases, the final layers or all layers of the model are fine-tuned to better suit the new task. This process involves initially keeping the general parameters θ_G fixed while training only the task-specific parameters θ_S:

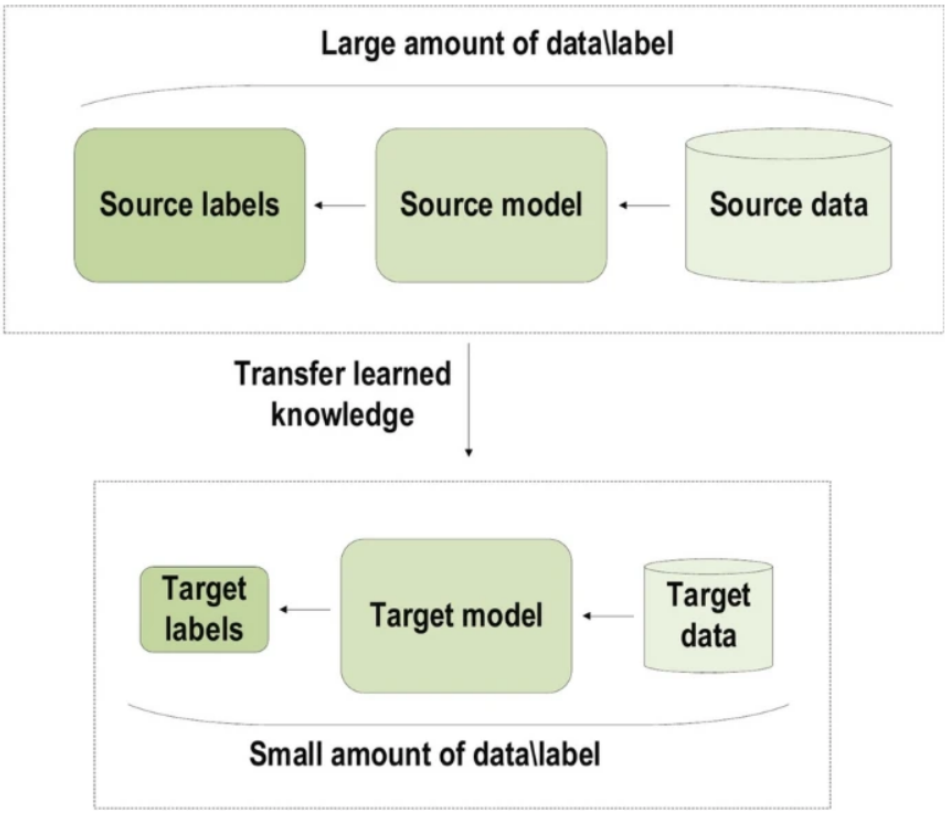

Figure 4. A conceptual transfer learning model [90].

$$\theta'_S = \arg\min_{\theta_S} \mathcal{L}(f(x; \theta_G, \theta_S)) \tag{40}$$

Once the task-specific parameters are adequately trained, the general parameters can also be adjusted to further improve performance on the new task:

$$\theta'_G, \theta''_S = \arg\min_{\theta_G, \theta_S} \mathcal{L}(f(x; \theta_G, \theta_S)) \tag{41}$$

This strategy is widely employed across various applications where models trained on generic tasks such as image recognition, language understanding, or general pattern recognition are fine-tuned to perform specialized functions like medical image analysis, sentiment analysis specific to a particular language, or identifying specific types of objects in aerial images. The ability to fine-tune models has made deep learning models not only more versatile but also more accessible, as it allows for customization without the need to

develop a new model from scratch. As a result, organizations can deploy state-of-the-art models tailored to their specific needs more quickly and efficiently.

5.3. Federated Learning

Federated learning is an innovative technique that enables the training of models across multiple decentralized edge devices or servers, each holding local data samples, without requiring the exchange of these data [91]. This approach is highly advantageous in environments where data privacy is a key concern, as it effectively minimizes the risk of data leakage by keeping all personal or sensitive information on the local device. In the federated learning framework, the training process typically unfolds over several rounds of computation. In each round, a central server disseminates the current model parameters θ to the participating devices. Upon receiving these parameters, each device uses its local data to compute an update $\Delta\theta_i$, which it then sends back to the central server [92]. Mathematically, the updates are aggregated by the server as follows:

$$\theta_{\text{new}} = \theta + \frac{1}{N}\sum_{i=1}^{N}\Delta\theta_i \tag{42}$$

where N represents the number of devices participating in the training [93]. This aggregation method, often achieved by averaging, ensures that each participating device contributes to the development of a more robust and widely applicable model without compromising the confidentiality of the data each device holds. Federated learning is especially relevant in scenarios such as mobile phone usage data, healthcare records analysis, and other fields where data security regulations, such as the European General Data Protection Regulation (GDPR), dictate stringent privacy measures [94]. By leveraging data from a wide range of sources without exposing individual data sets, federated learning facilitates more personalized and sensitive applications. This technique not only enhances privacy and data security but also harnesses the power of distributed data sources to create models that are both diverse in their learning and comprehensive in their application.

5.4. Deep Reinforcement Learning

Deep reinforcement learning (DRL) represents an advanced training technique that combines reinforcement learning principles with deep neural networks, allowing models to learn optimal behaviors through interaction with an environment [95]. Unlike traditional supervised or unsupervised learning methods, DRL relies on a feedback mechanism where the model, or agent, explores different actions to maximize a reward signal over time. The core of DRL's training methodology is the iterative learning process guided by the Bellman equation, which helps the agent estimate the optimal policy by updating its action-value function:

$$Q(s_t, a_t) = r_t + \gamma \max_a Q(s_{t+1}, a) \tag{43}$$

where $Q(s, a)$ represents the action-value function that calculates the expected return of taking action a in state s, r_t is the immediate reward received, and γ is the discount factor that values future rewards relative to immediate rewards [95]. This equation is vital for updating the policy that the agent follows to maximize its cumulative reward over time. Furthermore, DRL can be divided into several foundational approaches, including policy-based methods, value-based methods, and actor–critic methods. Policy-based methods, such as policy gradient methods, directly optimize the policy by adjusting its parameters to increase the probability of favorable actions. These methods are essential for handling environments with continuous action spaces, where defining discrete action values is impractical [96].

Actor-critic methods combine elements of both policy-based and value-based approaches. The actor component generates actions based on a policy, while the critic evaluates these actions by estimating the action-value function. This dual structure allows actor–critic models to achieve a balance between exploration and exploitation more effi-

ciently. Deep deterministic policy gradient (DDPG) and advantage actor–critic (A2C) are popular actor–critic methods that have been applied effectively in various domains such as autonomous robotics [97].

Multi-agent reinforcement learning (MARL) is another significant area within DRL, where multiple agents interact within a shared environment. This method is crucial for tasks involving collaborative or competitive settings, such as autonomous driving fleets, where agents must adapt to the actions of others in real time. MARL introduces challenges, including coordination and non-stationary environments, but has shown promising results in areas such as resource management and gaming [98]. An important innovation in policy optimization within DRL is proximal policy optimization (PPO), introduced by Schulman et al. [99]. PPO simplifies the training of DRL models by introducing a stable and efficient policy gradient update that prevents drastic changes to the policy, thereby enhancing training stability and performance. PPO has become a cornerstone in DRL applications, with successful implementations in areas such as game playing, robotic control, and navigation tasks.

DRL has been successfully applied across various domains, including robotics, where agents learn to navigate and manipulate objects, autonomous vehicles that optimize driving strategies, and strategic games that require evaluating complex decision spaces [100–102]. These applications demonstrate the power of DRL as an advanced training technique that enables models to learn from sequential decision-making processes, driving significant advancements in AI and machine learning.

5.5. Few-Shot Learning and One-Shot Learning

Few-shot learning and one-shot learning are advanced training techniques designed to enable models to learn from a very small amount of labeled data. These techniques are valuable in situations where acquiring large annotated datasets is impractical or impossible, such as in medical diagnosis, rare event detection, or personalized user experiences [103]. Few-shot learning refers to scenarios where models learn from a few examples per class, while one-shot learning is an extreme case where the model must learn from a single example. The primary goal of few-shot and one-shot learning is to generalize from limited data by leveraging prior knowledge gained from related tasks. This is achieved through various approaches, such as meta-learning, which trains models on a range of tasks so that they learn a strategy for rapid adaptation to new tasks with minimal data.

- Few-Shot Learning: Few-shot learning often employs methods like prototypical networks, which calculate a prototype (mean representation) for each class from the few available examples. These prototypes are then used to classify new instances based on their proximity to these prototypes in a shared embedding space [104]. The prototype $\mathbf{c}_k$ for a class k is calculated as:

$$\mathbf{c}_k = \frac{1}{N_k} \sum_{i=1}^{N_k} f_\theta(\mathbf{x}_i) \tag{44}$$

  where N_k is the number of examples in class k, $\mathbf{x}_i$ represents an example from class k, and f_θ is the embedding function parameterized by θ. To classify a new query point $\mathbf{x}$, the prototypical network computes the distance between the query point's embedding and each class prototype [104]. The squared Euclidean distance is often used as the distance metric:

$$d(\mathbf{x}, \mathbf{c}_k) = \|f_\theta(\mathbf{x}) - \mathbf{c}_k\|^2 \tag{45}$$

  The model predicts the class with the closest prototype. The loss function for training prototypical networks is the negative log-probability of the correct class under a softmax distribution over distances:

$$\mathcal{L}(\theta) = -\log \frac{\exp(-d(f_\theta(\mathbf{x}), \mathbf{c}_y))}{\sum_k \exp(-d(f_\theta(\mathbf{x}), \mathbf{c}_k))} \tag{46}$$

- One-Shot Learning: One-shot learning focuses on learning from a single example per class. This is significantly more challenging and requires the model to utilize transfer learning and prior knowledge efficiently [105]. Techniques such as Siamese networks are instrumental in this approach. Siamese networks learn to determine whether two inputs belong to the same class by learning a similarity function during training. The network consists of two identical subnetworks that embed two input samples x_1 and x_2 into a feature space:

$$h_1 = f_\theta(x_1), \quad h_2 = f_\theta(x_2) \tag{47}$$

The similarity score between the two embeddings is computed using a distance function, such as the Euclidean distance:

$$d(x_1, x_2) = \|h_1 - h_2\| \tag{48}$$

Furthermore, the Siamese network is trained using a contrastive loss function, which encourages the distance between similar examples to be small and dissimilar examples to be large [106]:

$$\mathcal{L}(\theta) = (1-y)\frac{1}{2}(d(\mathbf{x}_1, \mathbf{x}_2))^2 + (y)\frac{1}{2}\max(0, m - d(\mathbf{x}_1, \mathbf{x}_2))^2 \tag{49}$$

where y is a binary label indicating whether the pair $(\mathbf{x}_1, \mathbf{x}_2)$ is from the same class ($y = 0$) or different classes ($y = 1$), and m is a margin parameter.

6. Overview of Typical Deep Learning Applications

Deep learning has revolutionized various domains by leveraging its ability to model complex patterns and extract meaningful insights from vast datasets. This section provides an overview of typical applications of deep learning, highlighting its versatility and impact across different fields.

6.1. Computer Vision

Computer vision is one of the most prominent fields where deep learning has made a substantial impact. It involves developing algorithms and models that allow machines to interpret and understand visual data from the real world [107]. Within this broad domain, several specific applications have emerged as key areas of research and development.

- Image Classification: Image classification is one of the most prominent applications of deep learning, where the goal is to assign a label to an input image based on its content [108]. DL models, particularly CNNs, have significantly outperformed traditional methods in image classification tasks by automatically learning hierarchical features from raw data. CNNs can identify intricate patterns and structures in images, such as edges, textures, and objects, making them highly effective for diverse tasks ranging from medical diagnosis to social media content filtering [109]. Models like AlexNet, VGGNet, and ResNet have set benchmarks in this domain by achieving unprecedented accuracy on challenging datasets such as ImageNet.
- Object Detection: Object detection is an extension of image classification that involves identifying and localizing multiple objects within an image. Unlike image classification, which assigns a single label to an entire image, object detection provides both the category and the bounding box for each object in the image [110,111]. DL models, such as Faster R-CNN, You Only Look Once (YOLO), and Single-Shot MultiBox Detector (SSD), have transformed this field by achieving real-time performance with high accuracy [112]. These models use region proposal networks and anchor boxes to detect multiple objects of varying scales and sizes, making them suitable for applications in autonomous driving, surveillance, and robotics.

- Image Retrieval: Image retrieval involves searching and retrieving images from a large database based on their content or features. DL models, particularly CNNs, have been extensively used in content-based image retrieval systems, where they learn to extract high-dimensional feature vectors that capture the semantic content of images [113]. These feature vectors are then used to compare and retrieve similar images from a database. Techniques like deep hashing and metric learning further refine this process by mapping images into a compact binary code or optimizing the distance metric to enhance retrieval accuracy.

6.2. Natural Language Processing

Deep learning has significantly advanced the field of natural language processing (NLP), enabling machines to understand and generate human language with high accuracy. RNNs and their variants, such as LSTMs and GRUs, have been instrumental in modeling sequential data and capturing context and dependencies in text [114]. Transformer-based models, introduced by Vaswani et al. [8], have led to numerous innovations in NLP by employing self-attention mechanisms to capture long-range dependencies and contextual relationships more effectively. These models represent some of the most influential advancements in NLP.

Recent foundational models like GPT-4 and multimodal architectures such as Flamingo have expanded the scope of NLP applications. GPT-4, an extension of its predecessor GPT-3, has demonstrated remarkable capabilities in generating coherent, contextually relevant text and even solving complex reasoning tasks [115]. Flamingo integrates textual and visual inputs, enabling the model to perform tasks that require a nuanced understanding of both modalities, such as image captioning and question answering [116]. These advancements illustrate the growing impact of foundational models in addressing diverse challenges in NLP.

These advancements have led to substantial improvements in a wide range of NLP tasks, including machine translation, sentiment analysis, named-entity recognition, and text summarization. Models like BERT and GPT are pre-trained on vast corpora of text data and fine-tuned on specific downstream tasks, achieving state-of-the-art performance across various benchmarks. For example, multilingual extensions like mBERT and XLM-R have improved machine translation and cross-lingual information retrieval, enhancing accessibility for non-English languages.

In-context learning, a groundbreaking capability of GPT-4 and similar models, allows them to perform novel tasks without explicit retraining. By processing examples directly within the input prompt, these models adapt dynamically to diverse applications such as legal text summarization, medical report generation, and scientific paper drafting [71]. Fine-tuning and reinforcement learning with human feedback further enhance these models with specific user requirements, improving their relevance and safety in high-stakes domains like healthcare and law.

Another emerging trend is the integration of NLP with multimodal data, enhancing applications that require text, images, or other sensory data. For instance, Gato, a generalist model by Reed et al. [117], seamlessly processes textual, visual, and even robotic input, opening new frontiers in areas like autonomous navigation and interactive AI systems. The ability of these models to generate coherent and contextually relevant text has opened new avenues for applications in chatbots, virtual assistants, and automated content generation, significantly enhancing human-computer interaction.

6.3. Speech Recognition and Processing

Speech recognition and processing have been greatly enhanced by deep learning, leading to more accurate and robust speech-to-text systems. Traditional speech recognition models relied on handcrafted features and statistical methods, which often struggled with variability in accents, background noise, and speaking styles [118,119]. DL models, such as DNNs, CNNs, and LSTMs, have outperformed these traditional approaches by

automatically learning complex acoustic patterns from raw audio data, leading to significant reductions in word error rates. Recent advancements in speech processing have focused on end-to-end models, such as DeepSpeech and transformer-based architectures, which streamline the speech recognition pipeline by integrating all components into a single model [7,120]. These end-to-end models simplify training and improve generalization by directly optimizing for the desired output, such as text transcription.

7. Notable Applications of Deep Learning in the Literature

Deep learning has transformed a wide range of fields, providing state-of-the-art solutions across various domains. This section reviews significant applications of deep learning in areas such as healthcare, finance, autonomous systems, and agriculture.

7.1. Healthcare

Deep learning has substantially advanced healthcare, from diagnostic processes to personalized treatment approaches. This section covers some of those advancements.

7.1.1. Medical Imaging

Deep learning has improved medical imaging, enhancing both the accuracy and efficiency of diagnostic processes. The application of CNNs, in particular, has advanced image classification and anomaly detection across various types of radiographic images, contributing significantly to early disease detection and improved patient management. Recent studies have demonstrated the effectiveness of deep learning models in detecting various pathologies. For example, Sanida et al. [121] showed that deep learning algorithms could accurately identify pathological lung patterns in chest X-rays, which is crucial for the early detection and management of pulmonary diseases such as pneumonia and tuberculosis. The use of CNNs in this context not only accelerates the diagnostic process but also enhances its precision, which is essential for timely and effective treatment.

Moreover, a study by Manole et al. [122] focused on the application of deep learning to dermatological images, achieving dermatologist-level accuracy in classifying skin lesions. This study highlights how deep learning models can assist non-specialists in diagnosing skin conditions, thereby improving access to quality healthcare, particularly in underserved areas. Further advancements have been reported in the use of deep learning for MRI and CT scan analysis. McKinney et al. [123] developed a deep learning system for mammography that outperformed radiologists in breast cancer detection. Their research demonstrated that the model could reduce false positives and negatives, enhancing both the speed and accuracy of breast cancer screenings.

Additionally, in the field of neuroimaging, deep learning has been utilized to improve the diagnosis of neurological conditions. Elazab et al. [124] employed a deep learning framework to detect Alzheimer's disease from brain MRI scans. Their model demonstrated superior performance in distinguishing between healthy controls and patients with Alzheimer's compared to traditional machine learning methods.

A recent study by Gorenstein et al. [125] demonstrated the application of deep learning in lung cancer screening using low-dose CT scans. The study reported that the deep learning model could predict the malignancy of nodules with accuracy comparable to that of expert radiologists, suggesting its potential to serve as a reliable second reader in clinical settings, reducing the workload of radiologists and enhancing diagnostic accuracy.

In addition to cancer detection, deep learning has also been applied to improve cardiovascular imaging. A study by Ouyang et al. [126] utilized a video-based deep learning model to evaluate cardiac function through echocardiograms. This model demonstrated the ability to accurately predict cardiovascular outcomes, which could significantly impact the early detection and management of heart diseases. Other deep learning studies in medical imaging include the following: brain tumor detection by Liu and Wang [127] and Zhou et al. [128], lung cancer diagnosis by Hroub et al. [129], and breast cancer diagnosis by Naz et al. [130].

7.1.2. Genomics and Drug Discovery

Deep learning is a crucial tool in genomics and drug discovery, providing the means to analyze and interpret complex biological data that traditional methods struggle to decipher. One of the most notable advancements in this area is the use of deep learning models to predict protein structures and gene expression levels, enabling more accurate and rapid drug development. The work by Senior et al. [131] is a prime example, where deep learning models such as AlphaFold were shown to predict protein folding with unprecedented accuracy, marking a significant leap in computational biology.

In addition to protein structure prediction, deep learning has also been employed in understanding gene expression. For instance, a study by Avsec et al. [132] introduced a model called Enformer, which utilizes transformer architectures to predict gene expression and chromatin accessibility from DNA sequences. This approach has opened new avenues for exploring regulatory genomics and understanding gene regulation mechanisms at an unprecedented scale and resolution.

Recent advancements continue to expand the application of deep learning in genomics. For example, the ProtGPT2 model developed by Ferruz et al. [133] used generative transformer-based architectures to design novel protein sequences with potential therapeutic uses, marking a milestone in protein engineering. This capability is especially valuable in accelerating the development of targeted drugs by creating tailored protein structures suited to specific biological functions. Additionally, the EvoDiff model introduced by Alamdari et al. [134] combined evolutionary data with diffusion models to predict and design protein sequences, further enhancing the accuracy and applicability of protein generation in drug discovery.

Deep learning models have further been applied to virtual screening and drug repositioning. Stokes et al. [135] developed a deep learning model that identifies potential antibiotic candidates by predicting their ability to inhibit bacterial growth, leading to the discovery of Halicin, a potent new antibiotic. This model significantly accelerates the drug discovery process by rapidly screening thousands of compounds to find those with the highest therapeutic potential.

More recently, Irwin et al. [136] introduced ChemFormer, a transformer-based model for reaction prediction that uses extensive chemical reaction datasets to improve virtual screening and drug synthesis, demonstrating promising results in identifying viable drug compounds. Other studies, including [137–139], have also evaluated the ability of ChemFormer to parse large chemical datasets and predict reaction viability, adding a new layer of accuracy to virtual screening, ultimately accelerating the drug discovery pipeline.

Furthermore, deep learning has been integrated into the optimization of chemical reactions, an essential aspect of drug synthesis. The work by Schwaller et al. [140] employed a transformer-based model to predict the outcomes of chemical reactions, aiding in the optimization of synthetic routes and reducing the time required to develop new drugs. This model provides chemists with valuable insights into reaction mechanisms and potential yields, enhancing the efficiency of the drug development pipeline.

Another key area where deep learning is making strides is in the identification of biomarkers for disease prediction and drug response. Kuenzi et al. [141] utilized a deep learning approach to predict the sensitivity of cancer cell lines to various drugs based on genomic features, helping to identify potential biomarkers for personalized medicine. This application demonstrates how deep learning can contribute to precision oncology by tailoring treatments to the genetic profiles of individual patients.

Recently, a study by Li et al. [142] used a deep generative model for identifying prognostic biomarkers in genomics data, allowing for more accurate predictions of patient-specific drug responses. By leveraging patient genomic profiles, this model enhances the accuracy of drug response predictions, which is essential for developing personalized treatments in oncology and beyond.

Additionally, deep learning models are being used to simulate the effects of genetic mutations and their impacts on drug response. A study by Jumper et al. [143] focused

on understanding the consequences of mutations on protein function, which is crucial for developing targeted therapies. Their model helps predict how different mutations can alter protein structures and interactions, providing a foundation for designing drugs that are more effective against specific genetic variations.

Moreover, Chen et al. [144] introduced a mutation-based model that leverages DL to simulate genetic variant effects on protein stability and interactions, offering insights into how specific genetic changes impact drug efficacy. This model has been instrumental in advancing personalized medicine by predicting adverse drug reactions based on individual genetic variants, contributing to safer and more targeted drug therapies.

7.1.3. Patient Monitoring and Care

Deep learning is also advancing patient monitoring and care by enhancing the capabilities of wearable devices and mobile health technologies to detect and predict health events. These advancements enable continuous monitoring of patients, predicting health conditions with high accuracy and allowing for timely medical interventions. The integration of deep learning into patient monitoring provides proactive and predictive healthcare support, which is crucial for managing chronic conditions and preventing acute health episodes.

Recent studies have highlighted the effectiveness of deep learning models in predicting adverse health events using data from wearable devices. For example, Sivasubramaniam and Balamurugan [145] developed models capable of predicting cardiovascular events before they occurred by analyzing physiological data such as heart rate and activity levels collected from wearable devices. This capability allows for early intervention, significantly reducing the risk of severe outcomes such as heart attacks or strokes. Additionally, deep learning models have been applied to the management of diabetes through the prediction of blood glucose levels. Koca et al. [146] used data from continuous glucose monitors and smartphone sensors to forecast hypoglycemic events, providing alerts to patients before dangerous levels are reached. This proactive approach enhances patient safety and supports better self-management of diabetes.

Meanwhile, deep learning has been used to monitor neurodegenerative conditions. Davidashvilly et al. [147] developed a model that uses data from wearable sensors to detect early signs of Parkinson's disease by analyzing gait patterns and other movement-related features. The model demonstrated high accuracy in early detection, which is essential for the timely administration of treatments that can slow disease progression. In the field of mental health, deep learning has also been employed to predict episodes of depression and anxiety using data from mobile applications. For instance, Barua et al. [148] utilized smartphone usage patterns, such as typing speed and social media activity, to predict depressive episodes. These predictions enable healthcare providers to intervene early, providing necessary support and resources to those at risk.

Another significant application is in the prediction and management of respiratory conditions. A study by Igbal et al. [149] developed a deep learning model that monitors respiratory patterns using wearable sensors to predict asthma attacks. By providing timely alerts, the model helps patients manage their conditions more effectively, potentially reducing hospital admissions and improving quality of life. Deep learning is also being applied to detect arrhythmias using electrocardiogram (ECG) data. Ahmed et al. [150] demonstrated that a deep learning model could classify various types of arrhythmias from ECG signals with accuracy comparable to that of certified cardiologists. This model enables more accessible and frequent monitoring of heart conditions, improving patient outcomes through early detection and intervention.

In wearable technology, personalized medicine is becoming increasingly feasible with deep learning. A study by Baseer et al. [151] introduced an adaptive deep learning model for wearable devices, enabling real-time prediction of chronic disease escalations. This approach provides dynamic insights tailored to individual patients, enhancing proactive management strategies.

7.1.4. Federated Learning and Privacy-Preserving Healthcare Systems

Federated learning (FL) is a vital technique in healthcare, addressing privacy concerns while enabling collaborative advancements. FL allows healthcare institutions to train models collaboratively without sharing sensitive patient data, preserving privacy, and complying with regulations such as GDPR and HIPAA. For instance, Gaber et al. [152] demonstrated the use of FL to improve predictive models for diagnosing cardiovascular diseases by aggregating insights from multiple hospitals. This approach significantly enhanced model accuracy without compromising data privacy.

In cancer research, FL has been utilized to develop robust models for early detection. A study by Tan et al. [153] trained a federated deep learning model for breast cancer prediction using mammography data from various institutions, achieving improved diagnostic performance. Such models ensure equitable access to high-quality diagnostic tools, even in resource-constrained settings, while maintaining stringent privacy standards.

Another emerging application of FL in healthcare is in rare disease research. Due to the scarcity of data for rare conditions, FL enables the pooling of information from multiple institutions to train models without centralized data storage. For example, a study by Siganporia et al. [154] showed the potential of FL in improving diagnostic accuracy for rare genetic disorders by integrating genomic and clinical data from global healthcare networks. This methodology ensures equitable advancements in rare disease research while addressing data sensitivity issues.

Furthermore, FL has demonstrated significant potential in advancing mental health care. A recent study by Rehman et al. [155] used federated learning to develop predictive models for depression and anxiety disorders by combining smartphone usage patterns from multiple institutions. The study highlighted the capability of FL to train robust models that respect patient confidentiality, paving the way for scalable solutions in mental health diagnostics.

In the context of global health crises, FL has also proven instrumental in pandemic management. For instance, Saidi et al. [156] demonstrated the use of FL in combining data from hospitals worldwide to create an accurate and globally applicable model for early detection of viral mutations. This approach emphasized the utility of FL in facilitating cross-border collaborations for public health while maintaining stringent privacy protocols.

These advancements illustrate the broad applicability of deep learning in patient monitoring and care, enhancing the predictive power of health monitoring systems and supporting timely and personalized medical interventions.

Table 1 summarizes the different DL applications in healthcare.

Table 1. Summary of applications of deep learning in healthcare.

Application Domain	Reference	Year	Methods and Application
Medical Imaging	McKinney et al. [123]	2020	Developed a DL system for mammography, surpassing radiologists in breast cancer detection.
	Ouyang et al. [126]	2020	Employed video-based DL to evaluate cardiac function through echocardiograms.
	Gorenstein et al. [125]	2023	Used DL for lung cancer screening via low-dose CT, with performance comparable to expert radiologists.
	Sanida et al. [121]	2024	Utilized CNNs to detect lung disease patterns in chest X-rays for early diagnosis.
	Manole et al. [122]	2024	Applied DL to dermatological images, achieving dermatologist-level accuracy for skin lesions.
	Elazab et al. [124]	2024	Used DL for Alzheimer's diagnosis from brain MRI, outperforming traditional methods.
	Liu and Wang [127]	2024	Implemented DL models for brain tumor detection in MRI scans.
	Hroub et al. [129]	2024	Developed explainable DL models for lung cancer diagnosis.
	Naz et al. [130]	2024	Proposed efficient DL methods for breast cancer diagnosis using mammography images.

Table 1. *Cont.*

Application Domain	Reference	Year	Methods and Application
Genomics and Drug Discovery	Senior et al. [131]	2020	Utilized AlphaFold DL models for protein structure prediction, aiding drug discovery.
	Stokes et al. [135]	2020	Developed a DL model to identify new antibiotics, discovering Halicin.
	Schwaller et al. [140]	2020	Employed a transformer-based model for predicting chemical reactions, optimizing drug synthesis.
	Kuenzi et al. [141]	2020	Used DL to predict drug sensitivity from genomic data for personalized medicine.
	Avsec et al. [132]	2021	Introduced Enformer, a transformer-based model for gene expression prediction.
	Jumper et al. [143]	2021	Focused on genetic mutations' impact on protein function using DL for targeted therapies.
	Ferruz et al. [133]	2022	Used ProtGPT2, a transformer model, to generate novel protein sequences for drug discovery.
	Irwin et al. [136]	2022	Introduced ChemFormer, a transformer model for predicting chemical reactions in virtual screening.
	Alamdari et al. [134]	2023	Developed EvoDiff model, integrating evolutionary data with diffusion models for protein design.
	Das et al. [137]	2024	Enhanced ChemFormer's reaction prediction capabilities for drug discovery applications.
	Li et al. [142]	2024	Employed DL to identify prognostic biomarkers, enhancing drug response predictions in genomics data.
	Chen et al. [144]	2024	Developed mutation-based DL model for simulating genetic variant effects on drug efficacy.
Patient Monitoring and Care	Iqbal et al. [149]	2022	Developed DL models for respiratory monitoring via wearable sensors, predicting asthma attacks.
	Ahmed et al. [150]	2023	Used DL to classify arrhythmias from ECG data with cardiologist-level accuracy.
	Sivasubramaniam and Balamurugan [145]	2024	Applied DL using wearable data to forecast cardiovascular events.
	Koca et al. [146]	2024	Utilized DL for proactive glucose level prediction in diabetics, enhancing self-management.
	Davidashvilly et al. [147]	2024	Used DL for monitoring neurodegenerative conditions via wearable sensors.
	Barua et al. [148]	2024	Employed smartphone data in DL models to predict depressive and anxiety episodes.
	Baseer et al. [151]	2024	Designed an adaptive DL model for wearable devices for chronic disease prediction in real-time.
Federated Learning and Privacy	Rehman et al. [155]	2022	Applied FL for mental health prediction using cross-institutional smartphone usage data.
	Tan et al. [153]	2023	Trained FL models for breast cancer prediction from mammography data across multiple centers.
	Siganporia et al. [154]	2023	Used FL for rare disease diagnostics integrating genomic data from multiple institutions.
	Gaber et al. [152]	2024	Developed FL models for cardiovascular disease prediction using multi-institutional data.
	Saidi et al. [156]	2024	Demonstrated FL utility in pandemic management for early viral mutation detection.

7.2. Finance

Deep learning has significantly impacted the finance sector by enhancing the capabilities of various financial processes. Its ability to analyze vast amounts of data and detect complex patterns has proven invaluable for applications in algorithmic trading, fraud detection, and credit scoring. These advancements have not only improved efficiency but also enhanced the accuracy and reliability of financial systems.

7.2.1. Algorithmic Trading

Algorithmic trading involves the use of algorithms to execute trades at high speed and frequency, often based on complex mathematical models and large volumes of data. Deep learning has significantly transformed algorithmic trading by enabling models to predict market trends and make trading decisions based on vast datasets that capture complex patterns and dependencies in financial data.

Recent advancements in deep learning architectures, such as RNNs, CNNs, and reinforcement learning, have been particularly influential in this domain. Zhang et al. [157] demonstrated the effectiveness of RNNs in forecasting stock prices by capturing temporal dependencies in sequential financial data. Their study highlighted that RNNs, especially LSTM networks, can learn from historical price movements and predict future trends more accurately than traditional statistical methods. The model showed significant improvements in prediction accuracy, which suggests that deep learning techniques can enhance trading strategies by better anticipating market movements and reducing risks associated with trading.

In addition to RNNs, CNNs have also been applied to algorithmic trading. Song and Choi [158] explored the use of CNNs for extracting spatial patterns in financial time series data, allowing for more nuanced analysis of market trends. The study proposed a hybrid model combining CNNs and RNNs to leverage both spatial and temporal features. The results indicated that the hybrid model outperformed individual CNN and RNN models in predicting stock prices, showcasing the potential of deep learning to integrate multiple types of data representation for more effective trading strategies.

Reinforcement learning is another powerful tool in algorithmic trading, enabling models to learn optimal trading strategies through interaction with market environments. Theate and Ernst [159] developed a deep reinforcement learning model that adapts its trading strategy based on feedback from a simulated market environment. This model uses a reward system to optimize actions, learning to buy, sell, or hold assets in a manner that maximizes cumulative profit over time. The study found that this approach not only increased profitability but also improved risk management by allowing the model to adapt to changing market conditions and learn from both positive and negative outcomes. Further expanding on reinforcement learning, Majidi et al. [160] applied a deep Q-learning approach to cryptocurrency trading. The model continuously interacted with a simulated trading environment, updating its policy based on market fluctuations and volatility. By using a deep Q-network (DQN), the researchers were able to effectively manage the exploration–exploitation trade-off, enhancing the model's ability to capitalize on short-term opportunities while managing long-term risks.

Another notable study by Ma et al. [161] introduced a multi-agent reinforcement learning framework for trading multiple assets simultaneously. This approach modeled the trading environment as a multi-agent system where each agent represented a different asset or trading strategy. The agents learned to collaborate and compete, optimizing a global objective function related to portfolio returns and risk.

Recent advancements have also explored the application of generative models for financial forecasting and trading. Vuletic et al. [162] utilized GANs to simulate realistic financial time series data, which can be used to train trading models more effectively. The GAN-generated data provided additional training samples that helped models generalize better, especially in markets with limited historical data. This innovative approach demonstrated the potential of generative models to enhance algorithmic trading by providing robust synthetic data for model training and validation.

7.2.2. Fraud Detection

Fraud detection is a crucial area in finance where deep learning has proven particularly effective due to its ability to analyze vast amounts of transaction data in real-time and detect complex patterns indicative of fraudulent activity. As financial transactions become more frequent and digital, the need for robust, real-time fraud detection systems has grown significantly. Deep learning models offer substantial improvements over traditional methods by learning intricate patterns from data, allowing for more precise and timely detection of fraudulent activities.

A study by Gambo et al. [163] implemented a CNN model to detect fraudulent credit card transactions. The CNN was able to automatically learn hierarchical feature representations from transaction data, improving the detection accuracy compared to traditional

logistic regression models. The authors demonstrated that their model significantly reduced false positives, which is crucial for minimizing the inconvenience to legitimate customers.

Another recent study by Hiremath et al. [164] explored the application of GNNs for fraud detection in online payment systems. GNNs can model the relationships between different entities in a transaction network, capturing more complex fraud patterns, such as collusive behaviors in fraudulent transactions. By representing transactions as nodes and their relationships as edges, the GNN-based model was able to identify hidden patterns that traditional models often overlook, achieving higher precision and recall rates in fraud detection. Similarly, Dang et al. [165] proposed an adaptive fraud detection model using a reinforcement learning framework. This model dynamically adjusts its parameters based on real-time feedback from the environment, learning to optimize its fraud detection strategy over time. The reinforcement learning approach allowed the model to adapt to new fraud patterns as they emerged, maintaining high accuracy even as fraudulent behaviors evolved.

Furthermore, deep autoencoders were utilized by Koo et al. [62] to detect anomalies in transaction data. The study focused on using unsupervised learning to model normal transaction behavior, flagging deviations as potential fraud. This approach is particularly effective in cases where fraudulent transactions are rare and labeled data are limited. The autoencoder model demonstrated high accuracy in detecting fraudulent transactions with a low false positive rate, making it a valuable tool for fraud detection in environments with imbalanced datasets.

A novel approach using transformers for fraud detection was introduced by Yu et al. [166]. The transformer model employs its self-attention mechanism to focus on relevant parts of a transaction sequence, effectively capturing long-range dependencies in the data. This approach allowed the model to detect complex, temporally dispersed fraud patterns that other models might miss. The study showed that the transformer-based model outperformed traditional RNN and LSTM models, particularly in detecting fraud that involves coordinated activities over extended periods. Additionally, Yang et al. [167] proposed using BERT for fraud detection. Their study highlighted the effectiveness of transformer architectures in capturing complex dependencies across long sequences of transaction data. The attention mechanism in transformers allows the model to focus on relevant parts of a transaction sequence, improving detection performance compared to RNNs and other traditional models.

An ensemble learning approach combining multiple deep learning models, including LSTM, GRU, and CNN, was presented by Mienye and Sun [168]. By integrating different model types, the ensemble system leveraged the strengths of each, resulting in a more robust fraud detection mechanism. The ensemble model achieved higher sensitivity and specificity in detecting fraud compared to single-model approaches, highlighting the benefits of combining multiple deep learning architectures. A study by Esenogho et al. [56] explored the application of ensemble learning with deep learning models to improve fraud detection accuracy. The study used LSTM and GRU models as base learners in an AdaBoost implementation, and the ensemble approach provided a more robust detection mechanism that reduced the rate of false positives while maintaining high detection sensitivity.

Another study by Langevin et al. [169] applied GANs to enhance fraud detection performance by generating synthetic fraudulent transactions. These synthetic samples were used to augment the training data, helping to address the class imbalance problem often faced in fraud detection tasks. The GAN-augmented model demonstrated improved detection accuracy, particularly in identifying less common types of fraud that were underrepresented in the original dataset. Furthermore, Lin and Gao [170] introduced explainable AI techniques into fraud detection models to improve transparency and interpretability. Their model combined a deep learning framework with an explainability module that provided clear, understandable reasons for each fraud detection decision. The explainable AI approach helps build trust in automated fraud detection systems by ensuring that the decisions made by the model can be audited and understood by human analysts.

7.2.3. Credit Scoring

Credit scoring is a critical area in finance where deep learning has significantly advanced the ability to assess creditworthiness and predict defaults. A recent study by Mercep et al. [171] developed a DNN model that integrates both structured and unstructured data, including credit history and social media activity, to improve credit scoring accuracy. The study found that this model could better predict defaults compared to traditional scoring methods, particularly for customers with limited credit histories.

Further advancements in credit scoring have been achieved through the application of graph neural networks. Feng et al. [172] explored the use of GNNs to represent borrowers and their financial relationships as graphs, allowing the model to capture complex dependencies and interactions that traditional models might overlook. The study demonstrated that GNN-based models could provide more accurate assessments of credit risk, especially in peer-to-peer lending platforms where borrower information may be sparse or unstructured. In addition to GNNs, other advanced deep learning architectures have been explored for credit scoring. Chen et al. [173] utilized self-supervised learning to enhance the performance of credit scoring models. By pretraining the model on a large dataset of financial transactions and then fine-tuning it on credit scoring tasks, the model learned useful representations that improved prediction accuracy. This approach is beneficial for scenarios with limited labeled data, as it allows the model to learn from unlabeled data and generalize better to unseen cases.

Another notable approach is the use of RNNs and their variants, such as LSTM networks, to model temporal dependencies in credit data. A study by Ala'raj et al. [58] implemented LSTM networks to analyze time-series data of borrower behavior, such as transaction sequences and payment histories, improving the prediction of future defaults. This model captures the temporal dynamics of credit behavior, which is crucial for understanding changes in creditworthiness over time. Furthermore, a study by Plawiak et al. [174] proposed an ensemble approach that integrates different types of deep learning models, including CNNs, RNNs, and autoencoders, to capture various aspects of credit risk. This ensemble model achieved superior performance compared to individual models, reducing both false positives and false negatives in credit scoring predictions.

Autoencoders have also been used to improve credit scoring models by detecting anomalies in borrower behavior. For example, Wu et al. [63] used autoencoders to identify outliers in transaction data, which often correspond to risky credit behaviors. This method enhances traditional credit scoring by incorporating an additional layer of anomaly detection, thereby improving the robustness of credit risk assessments. In Table 2, the discussed DL applications in finance are summarized.

Table 2. Summary of applications of deep learning in finance.

Application Domain	Reference	Year	Methods and Application
Algorithmic Trading	Zhang et al. [157]	2020	Used RNNs for stock price forecasting, improving prediction accuracy over traditional methods.
	Theate and Ernst [159]	2021	Developed a DRL model for optimal trading strategies in simulated environments, enhancing profitability and risk management.
	Song and Choi [158]	2023	Combined CNNs and RNNs for stock price prediction, leveraging both spatial and temporal data features.
	Ma et al. [161]	2023	Introduced multi-agent DRL for trading multiple assets, outperforming single-agent models in returns and risk management.
	Majidi et al. [160]	2024	Applied deep Q-learning to cryptocurrency trading, managing exploration–exploitation in volatile markets.
	Vuletic et al. [162]	2024	Used GANs to simulate financial data for training trading models, enhancing model generalization with limited historical data.

Table 2. *Cont.*

Application Domain	Reference	Year	Methods and Application
Fraud Detection	Dang et al. [165]	2021	Proposed adaptive fraud detection using RL to adjust strategies dynamically based on real-time data.
	Gambo et al. [163]	2022	Implemented CNN for credit card fraud detection, reducing false positives compared to logistic regression.
	Esenogho et al. [56]	2022	Applied ensemble learning with LSTM and GRU for fraud detection, minimizing false positives while maintaining sensitivity.
	Lin and Gao [170]	2022	Integrated explainable AI into fraud detection for improved transparency and interpretability.
	Langevin et al. [169]	2022	Used GANs to create synthetic fraud data for model training, improving detection accuracy for rare fraud types.
	Hiremath et al. [164]	2024	Explored GNNs for fraud detection in online payment networks, achieving higher precision and recall by modeling transaction relationships.
	Koo et al. [62]	2024	Used deep autoencoders for unsupervised anomaly detection in transactions, achieving low false positives.
	Yu et al. [166]	2024	Introduced a transformer-based model for detecting fraud in transaction sequences, outperforming RNNs and LSTMs.
	Mienye and Sun [168]	2023	Combined LSTM, GRU, and CNN models in an ensemble for fraud detection, enhancing sensitivity and specificity.
Credit Scoring	Plawiak et al. [174]	2019	Proposed an ensemble model with CNNs, RNNs, and autoencoders for credit scoring, outperforming single models.
	Mercep et al. [171]	2020	Developed a DNN for credit scoring, integrating structured and unstructured data to improve default predictions.
	Ala'raj et al. [58]	2021	Used LSTM for temporal modeling in credit data, enhancing future default predictions.
	Feng et al. [172]	2022	Applied GNNs to model borrower relationships, capturing complex dependencies for better credit risk assessment.
	Chen et al. [173]	2024	Leveraged self-supervised learning to enhance credit scoring, using unlabeled data for improved generalization.
	Wu et al. [63]	2024	Used autoencoders for anomaly detection in borrower behavior, improving credit risk assessment robustness.

7.3. Autonomous Systems

Deep learning has been vital in the development of autonomous systems, significantly enhancing the capabilities of autonomous vehicles, drones, and robotics. These systems rely heavily on advanced perception algorithms for obstacle detection, navigation, and real-time decision-making in dynamic environments. In autonomous vehicles, deep learning plays a crucial role in perception systems, which are essential for tasks such as lane detection, pedestrian recognition, and obstacle avoidance. A study by Feng et al. [175] introduced a multitask deep learning framework that combines object detection, lane detection, and semantic segmentation into a single model. This integrated approach allows for more efficient processing and improved performance in complex driving scenarios, reducing the computational overhead typically associated with running multiple models simultaneously. Deep learning has also enhanced real-time decision-making in autonomous vehicles. Lee et al. [176] proposed an end-to-end learning approach using DNNs to predict steering angles directly from raw camera images. This model was trained on human driving data, enabling it to learn complex driving behaviors and make real-time decisions in various traffic conditions. This method reduces the reliance on pre-programmed rules, allowing for more adaptive and flexible driving strategies.

Deep learning has improved autonomous navigation and obstacle avoidance in complex environments, such as in drones. A study by Xue and Chen [177] developed a deep reinforcement learning model for autonomous drone navigation, which learns to navigate through dense forests by trial and error. The model outperforms traditional algorithms in terms of speed and collision avoidance, demonstrating the potential of DRL for real-time decision-making in unstructured environments. Meanwhile, deep learning models have been applied to swarm robotics, where multiple robots work together to achieve a common goal. A recent study by Hu et al. [178] used a combination of CNNs and GNNs to enable robots in a swarm to communicate and coordinate their actions in real-time, enhancing their ability to perform complex tasks such as search and rescue operations. This model

allows for decentralized control, reducing the need for a central coordinating unit and making the system more robust to failures.

Deep learning has also enabled robots to perform more sophisticated tasks, such as object manipulation and human–robot interaction. Han et al. [179] presented a deep learning-based grasping study that allows robots to identify and pick up objects of various shapes and sizes with high precision. The model uses a combination of visual and tactile data to improve grasp success rates, making it suitable for applications in warehouses and healthcare settings. Healthcare robotics has also benefited from advancements in deep learning. For example, Xu et al. [180] proposed a deep learning model for surgical robots to assist in minimally invasive procedures. The model uses deep reinforcement learning to optimize the robot's movements, reducing the risk of errors and improving surgical outcomes. This application demonstrates the potential of deep learning to enhance precision and safety in delicate medical procedures.

Autonomous underwater vehicles (AUVs) have seen significant improvements due to deep learning as well. A study by Zhou et al. [181] introduced a GAN-based sonar image classification system for AUVs, allowing for better detection and classification of underwater objects. This capability is crucial for applications such as underwater exploration and environmental monitoring. Furthermore, in agricultural robotics, deep learning has been utilized to enhance precision farming. Meanwhile, a study by Zheng et al. [182] developed a CNN-based deep learning model for fruit detection and harvesting in orchards. The model accurately identifies ripe fruits and guides the robotic arm to pick them, improving efficiency and reducing labor costs in agricultural operations.

Another innovative application is in the development of autonomous construction robots. Lee et al. [183] applied DNN to enable robots to perform complex construction tasks, such as bricklaying and painting. The model learns from 3D simulations of construction environments, allowing the robot to operate autonomously on construction sites with minimal human supervision. In the domain of aerial surveillance, deep learning models have been employed to improve the accuracy of object detection and tracking in video feeds from drones. Deepal et al. [184] proposed a deep learning-based framework that integrates object detection and multi-object tracking to enhance the performance of video surveillance systems in tracking multiple moving objects over large areas. Autonomous systems in disaster response have also been enhanced through deep learning [185].

7.4. Energy

Deep learning has become increasingly important in the energy sector, enhancing various aspects of production, distribution, and consumption. Its applications span across optimizing energy production processes, improving grid management through demand forecasting, and enabling predictive maintenance of equipment to prevent failures and reduce downtime. These advancements help ensure the stability, efficiency, and sustainability of energy systems in a rapidly evolving landscape.

One significant application of deep learning in the energy sector is predictive maintenance. A study by Vidal et al. [186] demonstrated the effectiveness of using LSTM networks for predicting failures in wind turbines. The model analyzed data from vibration sensors and temperature readings, significantly improving the accuracy of failure predictions compared to traditional statistical models, thereby reducing downtime and maintenance costs.

Another critical application of deep learning is in energy demand forecasting. Accurate demand forecasting is essential for grid stability and efficient energy distribution. Guo-Feng et al. [187] developed a hybrid model using RNNs and CNNs for short-term energy demand forecasting that outperformed traditional methods. Their approach effectively captured temporal dependencies in energy consumption data, leading to more reliable predictions. In addition to demand forecasting, deep learning models have been used to optimize energy distribution in smart grids. A study by Yu et al. [188] employed a deep reinforcement learning approach to optimize power flow in a smart grid. The model

learned to manage the distribution of energy resources dynamically, considering fluctuating demand and supply conditions. This method improved grid stability and reduced energy waste, highlighting the potential of deep learning in smart grid management.

Deep learning has also been applied to enhance the integration of renewable energy sources into the grid. Al-Ali et al. [189] proposed a hybrid model that combines CNN, LSTM, and transformer models for predicting solar energy production based on weather data. The model's ability to accurately predict solar output helped optimize the scheduling of energy resources, ensuring a more balanced integration of solar power into the grid. Furthermore, deep learning has been utilized in the optimization of energy storage systems. A study by Gwangwoo et al. [190] developed a model using deep reinforcement learning to optimize the charging and discharging cycles of battery storage systems in response to fluctuating energy prices and demand. The model improved the efficiency of energy storage operations, reducing costs and enhancing the economic viability of renewable energy storage solutions.

In the area of fault detection, deep learning models have shown promise in identifying faults in power grids. Dong-Hee and Jonghee [191] introduced a transformer-based model for real-time fault detection in power grids. The model used sequential data from sensors to detect anomalies and potential faults with high accuracy, enabling faster response times and reducing the risk of widespread outages.

Another innovative application of deep learning in the energy sector is in energy-efficient building management. A study by Elsisi et al. [192] used a deep learning approach to optimize heating, ventilation, and air conditioning (HVAC) systems in large buildings. The model predicted energy consumption based on occupancy patterns and external weather conditions, allowing for more efficient energy use and reduced operational costs.

7.5. Agriculture

Deep learning has become a powerful tool in agriculture, driving advancements in precision farming, crop management, and automation of farming equipment. These technologies enhance productivity and sustainability by enabling more precise assessments of crop health, predicting yields, detecting pests and diseases, and optimizing the use of resources.

One of the most impactful applications of deep learning in agriculture is precision farming. By analyzing crop imagery from drones or satellites, deep learning models can assess plant health with high accuracy. A study by Kamilaris and Prenafeta-Boldú [193] reviewed various deep learning techniques used in analyzing agricultural imagery to monitor plant growth and health. These models can detect subtle changes in color, texture, and shape, indicating the presence of diseases or nutrient deficiencies, thereby allowing for timely interventions.

Another significant application is in yield prediction, which is crucial for planning and resource allocation. Gastli et al. [194] developed a CNN model that uses satellite images to predict crop yields. The model outperformed traditional statistical methods by capturing complex patterns in the data that are often missed by simpler models, demonstrating the potential of deep learning to improve agricultural productivity.

Deep learning is also instrumental in pest and disease detection, which is vital for maintaining crop health. A study by Ferentinos [195] employed deep learning models to identify various plant diseases from leaf images. The models achieved high accuracy in detecting diseases like powdery mildew and rust, significantly reducing the need for chemical pesticides and supporting sustainable farming practices.

In addition to crop monitoring, deep learning models have been applied to optimize irrigation systems. A study by Sami et al. [196] introduced a deep learning-based irrigation management system that uses soil moisture data and weather forecasts to optimize water usage. The system helps conserve water while ensuring that crops receive the optimal amount of moisture, promoting sustainable water management in agriculture. Additionally, Mrutyunjay et al. [197] studied the use of deep learning models in autonomous tractors

that can navigate through fields, identify crops and weeds, and perform precision spraying. This technology reduces labor costs and increases operational efficiency.

In fruit detection and harvesting, DL has shown significant promise. As already mentioned in a previous section, Zheng et al. [182] employed a CNN-based approach for fruit detection and harvesting in orchards. The model identifies ripe fruits and guides a robotic arm to pick them. Similarly, Santos et al. [198] created a deep learning-based system for detecting and classifying fruits in orchards. The model uses a combination of RGB and hyperspectral images to identify fruits with high accuracy, enabling autonomous harvesters to pick fruits selectively, reducing waste and maximizing yield. Furthermore, deep learning models have been applied to predict soil properties, which are crucial for determining the suitability of land for different crops. A study by Pavlovic et al. [199] utilized a deep neural network to predict soil organic carbon content from remote sensing data. Accurate soil property predictions help farmers make informed decisions about crop rotation and fertilizer application, enhancing soil health and agricultural productivity.

Robotic weeding is another area where deep learning is making an impact. Mckay et al. [200] developed a deep learning model for a robotic weeder that can distinguish between crops and weeds in real time. The model enables precise weeding, reducing the need for herbicides and promoting sustainable farming practices.

Meanwhile, deep learning has been used to optimize supply chain management in agriculture. A study by Feng et al. [201] applied deep learning models to forecast demand and optimize the logistics of agricultural products, reducing waste and improving the efficiency of food distribution networks. Deep learning has also contributed to the development of smart farming technologies that integrate multiple sources of data, such as weather forecasts, soil conditions, and crop health.

Manufacturing

Deep learning has become instrumental in modernizing manufacturing, contributing significantly to quality inspection, predictive maintenance, and anomaly detection in production lines. These applications enhance production efficiency, reduce waste, and ensure high-quality output, addressing some of the industry's most critical challenges. (See Table 3).

In quality inspection, DL models have been widely adopted for automated defect detection. Recent advancements, such as the work by Li et al. [202], applied CNN and transformer architectures to identify surface defects in products, significantly outperforming traditional visual inspection methods. By training on diverse datasets, these models can detect subtle anomalies in textures and patterns that may not be visible to human inspectors, ensuring that defective products are identified early in the production process. Additionally, hybrid architectures combining CNNs with attention mechanisms have been shown to improve accuracy in identifying complex defect patterns, as demonstrated by Li et al. [203], who reported a marked increase in defect detection rates in semiconductor manufacturing.

Predictive maintenance has also seen substantial improvements through DL integration. By analyzing time-series data from equipment sensors, deep learning models can predict potential failures before they occur. For instance, Zhang et al. [204] developed a model using LSTM and attention mechanisms to monitor machinery health, enabling manufacturers to anticipate mechanical failures with high precision. This approach not only reduces downtime but also extends equipment lifespan by scheduling maintenance based on data-driven predictions rather than fixed intervals. Furthermore, the adoption of GANs for simulating machine states has provided robust training data for maintenance models, even in environments with scarce failure data [205,206].

In addition to predictive maintenance, DL is increasingly used for anomaly detection on production lines. Anomaly detection models identify deviations in production processes that may indicate inefficiencies or faults, thereby preventing potential quality issues. Recent studies, such as by Sorostinean et al. [207], have utilized unsupervised learning methods, like autoencoders, to detect anomalies in real-time. These models are trained to

recognize normal operational patterns, flagging any unusual behaviors or inconsistencies in data as anomalies. This approach has proven particularly useful in high-speed manufacturing environments where real-time monitoring is essential for maintaining optimal production flow.

Table 3. Summary of applications of deep learning in autonomous systems, energy, agriculture, and manufacturing.

Application Domain	Reference	Year	Methods and Application
Autonomous Systems	Xu et al. [180]	2019	Used DRL for surgical robots, optimizing minimally invasive procedures.
	Zhou et al. [181]	2019	Developed GAN-based sonar image classification for AUVs, enhancing underwater object detection.
	Feng et al. [175]	2020	Created multitask DL framework for object detection, lane detection, and segmentation in autonomous vehicles.
	Hu et al. [178]	2021	Combined CNNs and GNNs for real-time coordination in robot swarms for search and rescue.
	Lee et al. [176]	2023	Proposed end-to-end DNN model for steering angle prediction from raw images, improving driving decisions.
	Xue and Chen [177]	2023	Developed DRL model for drone navigation in dense environments, enhancing speed and collision avoidance.
	Han et al. [179]	2023	Implemented DL-based grasping in robots for precise object handling in warehouses and healthcare.
	Zheng et al. [182]	2019	Developed CNN for fruit detection and harvesting, improving agricultural robotic efficiency.
Energy	Yu et al. [188]	2019	Applied DRL for smart grid power flow optimization, reducing waste and enhancing stability.
	Gwangwoo et al. [190]	2021	Used DRL for optimizing battery storage, improving energy efficiency.
	Elsisi et al. [192]	2021	Utilized DL for HVAC optimization in buildings, lowering energy use and costs.
	Al-Ali et al. [189]	2022	Proposed hybrid model (CNN, LSTM, transformer) for solar energy prediction, improving grid integration.
	Vidal et al. [186]	2023	Applied LSTM for wind turbine predictive maintenance, enhancing failure prediction and reducing downtime.
	Guo-Feng et al. [187]	2024	Developed hybrid RNN-CNN for short-term energy demand forecasting, improving prediction reliability.
	Dong-Hee and Jonghee [191]	2024	Introduced transformer-based model for real-time fault detection in power grids, improving accuracy.
Agriculture	Kamilaris and Prenafeta-Boldú [193]	2018	Reviewed DL techniques for agricultural imagery analysis, enhancing plant health monitoring.
	Ferentinos [195]	2018	Applied DL for plant disease detection from leaf images, supporting sustainable farming by reducing pesticide use.
	Zheng et al. [182]	2019	Developed CNN for fruit detection and harvesting, improving agricultural robotic efficiency.
	Gastli et al. [194]	2021	Developed CNN for crop yield prediction using satellite images, outperforming traditional methods.
	Pavlovic et al. [199]	2021	Used DNN to predict soil properties from remote sensing data, aiding in agricultural decision-making.
	Mckay et al. [200]	2021	Created DL model for robotic weeding, reducing herbicide use.
	Sami et al. [196]	2022	Introduced DL-based irrigation management using soil moisture and weather data for optimal water usage.
	Feng et al. [201]	2023	Utilized DL for optimizing agricultural supply chain management, reducing waste and improving distribution.
	Mrutyunjay et al. [197]	2024	Studied DL in autonomous tractors for precision spraying, enhancing farming efficiency.
	Santos et al. [198]	2024	Developed DL system for fruit detection in orchards, enabling selective harvesting to reduce waste.
Manufacturing	Li et al. [202]	2022	Applied CNN and transformer models for automated defect detection in surface inspection, improving accuracy over traditional methods.
	Zhang et al. [204]	2020	Developed LSTM with attention for predictive maintenance, anticipating failures in machinery and reducing downtime.
	Sorostinean et al. [207]	2024	Used autoencoders for real-time anomaly detection in production lines, flagging deviations to prevent quality issues.
	Li et al. [205]	2024	Employed GANs for data augmentation in predictive maintenance, generating training data in environments with limited failure data.
	Yan et al. [206]	2024	Combined LSTM and GAN models to create a hybrid predictive maintenance system, enhancing fault prediction accuracy in manufacturing.

The application of DL in manufacturing has led to improvements in quality assurance, optimized maintenance schedules, and increased operational efficiency. As DL technologies continue to evolve, their role in manufacturing is expected to expand, offering manufacturers enhanced capabilities for managing quality control and operational risks.

8. Discussion, Challenges, and Future Research Directions

Deep learning is witnessing significant advancements that are transforming its applications across various domains. One notable trend is the rise of edge AI, which facilitates real-time processing on resource-constrained devices. Edge AI has enabled the deployment of deep learning models in Internet of Things (IoT) devices, autonomous vehicles, and wearable healthcare devices, providing faster decision-making while reducing latency and dependence on centralized servers. For instance, models optimized for edge AI are being applied to real-time patient monitoring, enhancing healthcare outcomes through continuous data analysis directly on devices.

Another important trend is the increasing adoption of federated learning, a privacy-preserving technique that allows collaborative model training across distributed datasets without sharing sensitive information. This approach is impactful in healthcare and finance, where privacy regulations such as GDPR and HIPAA demand strict data confidentiality. Federated learning enables hospitals to develop shared diagnostic models without exposing patient records, or banks to collaboratively improve fraud detection systems while preserving customer privacy. Recent advances in secure aggregation and differential privacy techniques are further strengthening the applicability of federated learning in these domains.

Sustainable AI is another crucial focus area, aiming to minimize the environmental impact of deep learning models. Energy-efficient training techniques, such as model pruning and quantization, are being extensively researched to reduce carbon footprints. Additionally, sustainable AI is finding applications in sectors like agriculture and energy management. In agriculture, resource optimization models powered by deep learning are improving irrigation efficiency and precision farming practices. Similarly, in energy management, smart grid systems enhanced by deep learning are optimizing power flow and reducing waste, aligning technological advancements with global sustainability goals.

While these advancements have demonstrated the immense potential of deep learning, several challenges remain that need to be addressed to further harness its capabilities. One major challenge is the heavy reliance on large-scale labeled datasets, which are often difficult and costly to obtain, particularly in specialized fields like medical imaging and climate science [19]. Although methods such as transfer learning and data augmentation have been developed to mitigate this issue, they are not always sufficient. Emerging techniques like few-shot learning and self-supervised learning present promising alternatives, enabling models to learn effectively with limited labeled data or to utilize vast amounts of unlabeled data. However, these techniques still require significant refinement to enhance their robustness and adaptability across diverse applications.

Additionally, as deep learning models grow more complex, the need for interpretability and transparency becomes increasingly critical. Understanding how these models make decisions is essential, especially in high-stakes areas such as healthcare and finance, where accountability and trust are paramount. Enhancing the interpretability of these models is crucial not only for user trust but also for ensuring ethical deployment [208]. Addressing issues related to model bias and fairness remains a pressing concern, as biases in data and algorithms can lead to inequitable outcomes. As deep learning continues to be integrated into sensitive applications, there is an urgent need to develop models that are both interpretable and fair, ensuring they are aligned with ethical standards.

Furthermore, the growing computational and energy demands of deep learning models pose significant challenges. As architectures become more sophisticated, they require more resources for training and deployment, which can limit their accessibility and sustainability. Research into efficient algorithms such as model pruning, quantization, and knowledge distillation is essential to reduce these computational costs without compromising

model performance. Additionally, exploring new hardware innovations, like neuromorphic computing, may offer pathways to enhance computational efficiency and enable the deployment of deep learning models in resource-constrained environments.

Based on the current challenges, future research directions could include:

- Improving Model Interpretability and Fairness: Developing interpretable models and enhancing existing explainability techniques are critical for the broader adoption of deep learning in high-stakes applications. Future research should prioritize the creation of frameworks that ensure transparency and fairness, thereby enabling greater trust in AI systems.
- Enhancing Computational Efficiency: There is a need to develop more efficient deep learning algorithms, such as those employing model compression techniques like pruning and quantization, to reduce computational demands. Research should also investigate the potential of advanced hardware solutions, such as neuromorphic computing, to further enhance efficiency.
- Enhancing Privacy and Security Measures: As deep learning models are increasingly deployed in sensitive applications, improving privacy and security is essential. Future research should focus on developing privacy-preserving algorithms, such as federated learning and differential privacy, and advancing techniques to defend against adversarial attacks, thereby ensuring the integrity and reliability of AI systems.
- Integrating Deep Learning with Other AI Techniques: Combining deep learning with other AI methods, such as symbolic reasoning or reinforcement learning, could lead to the development of hybrid models capable of handling complex tasks requiring both data-driven learning and logical reasoning. This integration could pave the way for more versatile and capable AI systems.
- Expanding Applications into New Domains: While deep learning has shown remarkable success in areas like computer vision and NLP, there is still significant potential for its application in other fields, such as climate science, social sciences, and education. Future research should focus on adapting deep learning techniques to address the unique challenges in these areas, offering innovative solutions to global issues.
- Advancing Few-Shot and Self-Supervised Learning: Further development of few-shot and self-supervised learning methods is essential to reduce the reliance on large labeled datasets. Research should focus on improving the adaptability and generalization of these models to a wide array of tasks and data types, enhancing their robustness and effectiveness.

9. Conclusions

The field of deep learning continues to evolve at an unprecedented pace, driven by advancements in architecture design, training methodologies, and computational capabilities. This paper has reviewed the foundational and emerging architectures that have shaped the field, from CNNs and RNNs to transformers and GANs, highlighting their respective contributions and limitations. The study also explored the integration of these architectures into various applications, such as healthcare diagnostics, autonomous systems, financial modeling, and natural language processing, and demonstrated their robustness and impact. Furthermore, this study highlighted the current challenges in deep learning, including the need for more efficient models, addressing data bias, enhancing model interpretability, and reducing computational demands.

Author Contributions: Conceptualization, I.D.M. and T.G.S.; methodology, I.D.M.; validation, I.D.M. and T.G.S.; investigation, I.D.M. and T.G.S.; writing—original draft preparation, I.D.M.; writing—review and editing, I.D.M. and T.G.S.; visualization, I.D.M.; supervision, T.G.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
AE	Autoencoder
ANN	Artificial Neural Network
BERT	Bidirectional Encoder Representations from Transformers
BiLSTM	Bidirectional Long Short-Term Memory
CAE	Convolutional Autoencoder
CNN	Convolutional Neural Network
DAE	Denoising Autoencoder
DL	Deep Learning
DRL	Deep Reinforcement Learning
GAN	Generative Adversarial Network
GRU	Gated Recurrent Unit
LSTM	Long Short-Term Memory
ML	Machine Learning
MANN	Memory-Augmented Neural Network
NAS	Neural Architecture Search
NLP	Natural Language Processing
ODE	Ordinary Differential Equation
RNN	Recurrent Neural Network
RL	Reinforcement Learning
SSL	Self-Supervised Learning
SNN	Spiking Neural Network
VAE	Variational Autoencoder
ViT	Vision Transformer

References

1. O'Halloran, T.; Obaido, G.; Otegbade, B.; Mienye, I.D. A deep learning approach for Maize Lethal Necrosis and Maize Streak Virus disease detection. *Mach. Learn. Appl.* **2024**, *16*, 100556. [CrossRef]
2. Peng, Y.; He, L.; Hu, D.; Liu, Y.; Yang, L.; Shang, S. Decoupling Deep Learning for Enhanced Image Recognition Interpretability. *ACM Trans. Multimed. Comput. Commun. Appl.* **2024** , *20*, 309. [CrossRef]
3. Obaido, G.; Achilonu, O.; Ogbuokiri, B.; Amadi, C.S.; Habeebullahi, L.; Ohalloran, T.; Chukwu, C.W.; Mienye, E.; Aliyu, M.; Fasawe, O.; et al. An Improved Framework for Detecting Thyroid Disease Using Filter-Based Feature Selection and Stacking Ensemble. *IEEE Access* **2024**, *12*, 89098–89112. [CrossRef]
4. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1–9. [CrossRef]
5. Al-Jumaili, A.H.A.; Muniyandi, R.C.; Hasan, M.K.; Paw, J.K.S.; Singh, M.J. Big data analytics using cloud computing based frameworks for power management systems: Status, constraints, and future recommendations. *Sensors* **2023**, *23*, 2952. [CrossRef]
6. Gill, S.S.; Wu, H.; Patros, P.; Ottaviani, C.; Arora, P.; Pujol, V.C.; Haunschild, D.; Parlikad, A.K.; Cetinkaya, O.; Lutfiyya, H.; et al. Modern computing: Vision and challenges. *Telemat. Inform. Rep.* **2024**, *13*, 100116. [CrossRef]
7. Mienye, I.D.; Obaido, G.; Emmanuel, I.D.; Ajani, A.A. A Survey of Bias and Fairness in Healthcare AI. In Proceedings of the 2024 IEEE 12th International Conference on Healthcare Informatics (ICHI), Orlando, FL, USA, 3–6 June 2024; pp. 642–650.
8. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems NIPS, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
9. Zioviris, G.; Kolomvatsos, K.; Stamoulis, G. An intelligent sequential fraud detection model based on deep learning. *J. Supercomput.* **2024**, *80*, 14824–14847. [CrossRef]
10. Mienye, I.D.; Swart, T.G. A Hybrid Deep Learning Approach with Generative Adversarial Network for Credit Card Fraud Detection. *Technologies* **2024**, *12*, 186. [CrossRef]
11. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
12. Bengio, Y.; Lecun, Y.; Hinton, G. Deep learning for AI. *Commun. ACM* **2021**, *64*, 58–65. [CrossRef]

13. Janiesch, C.; Zschech, P.; Heinrich, K. Machine learning and deep learning. *Electron. Mark.* **2021**, *31*, 685–695. [CrossRef]
14. Matsuo, Y.; LeCun, Y.; Sahani, M.; Precup, D.; Silver, D.; Sugiyama, M.; Uchibe, E.; Morimoto, J. Deep learning, reinforcement learning, and world models. *Neural Netw.* **2022**, *152*, 267–275. [CrossRef]
15. Xiao, Y.; Tian, Z.; Yu, J.; Zhang, Y.; Liu, S.; Du, S.; Lan, X. A review of object detection based on deep learning. *Multimed. Tools Appl.* **2020**, *79*, 23729–23791. [CrossRef]
16. Deng, Y.; Zhang, T.; Lou, G.; Zheng, X.; Jin, J.; Han, Q.L. Deep learning-based autonomous driving systems: A survey of attacks and defenses. *IEEE Trans. Ind. Inform.* **2021**, *17*, 7897–7912. [CrossRef]
17. Qiao, Y.; Yin, J.; Wang, W.; Duarte, F.; Yang, J.; Ratti, C. Survey of deep learning for autonomous surface vehicles in marine environments. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 3678–3701. [CrossRef]
18. Al-lQubaydhi, N.; Alenezi, A.; Alanazi, T.; Senyor, A.; Alanezi, N.; Alotaibi, B.; Alotaibi, M.; Razaque, A.; Hariri, S. Deep learning for unmanned aerial vehicles detection: A review. *Comput. Sci. Rev.* **2024**, *51*, 100614. [CrossRef]
19. Mienye, I.D.; Jere, N. Deep Learning for Credit Card Fraud Detection: A Review of Algorithms, Challenges, and Solutions. *IEEE Access* **2024**, *12*, 96893–96910. [CrossRef]
20. Zhang, C.; Sjarif, N.N.A.; Ibrahim, R. Deep learning models for price forecasting of financial time series: A review of recent advancements: 2020–2022. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2024**, *14*, e1519. [CrossRef]
21. Sahu, S.K.; Mokhade, A.; Bokde, N.D. An overview of machine learning, deep learning, and reinforcement learning-based techniques in quantitative finance: Recent progress and challenges. *Appl. Sci.* **2023**, *13*, 1956. [CrossRef]
22. Mienye, E.; Jere, N.; Obaido, G.; Mienye, I.D.; Aruleba, K. Deep Learning in Finance: A Survey of Applications and Techniques. *AI* **2024**, *5*, 2066–2091. [CrossRef]
23. Ajagbe, S.A.; Adigun, M.O. Deep learning techniques for detection and prediction of pandemic diseases: A systematic literature review. *Multimed. Tools Appl.* **2024**, *83*, 5893–5927. [CrossRef]
24. Askr, H.; Elgeldawi, E.; Aboul Ella, H.; Elshaier, Y.A.; Gomaa, M.M.; Hassanien, A.E. Deep learning in drug discovery: An integrative review and future challenges. *Artif. Intell. Rev.* **2023**, *56*, 5975–6037. [CrossRef]
25. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation applied to handwritten zip code recognition. *Neural Comput.* **1989**, *1*, 541–551. [CrossRef]
26. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
27. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2018**, arXiv:1810.04805.
28. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
29. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef]
30. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
31. Sherstinsky, A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Phys. D Nonlinear Phenom.* **2020**, *404*, 132306. [CrossRef]
32. Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Netw.* **1989**, *2*, 359–366. [CrossRef]
33. Sarker, I.H. AI-based modeling: Techniques, applications and research issues towards automation, intelligent and smart systems. *SN Comput. Sci.* **2022**, *3*, 158. [CrossRef] [PubMed]
34. Mienye, I.D.; Swart, T.G.; Obaido, G. Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications. *Information* **2024**, *15*, 517. [CrossRef]
35. Mienye, I.D.; Sun, Y.; Ileberi, E. Artificial intelligence and sustainable development in Africa: A comprehensive review. *Mach. Learn. Appl.* **2024**, *18*, 100591. [CrossRef]
36. Sharma, S.; Sharma, S.; Athaiya, A. Activation functions in neural networks. *Towards Data Sci.* **2017**, *6*, 310–316. [CrossRef]
37. Nwankpa, C.; Ijomah, W.; Gachagan, A.; Marshall, S. Activation functions: Comparison of trends in practice and research for deep learning. *arXiv* **2018**, arXiv:1811.03378.
38. Dubey, A.K.; Jain, V. Comparative study of convolution neural network's relu and leaky-relu activation functions. In *Applications of Computing, Automation and Wireless Systems in Electrical Engineering: Proceedings of MARC 2018*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 873–880.
39. Shyni, H.M.; Chitra, E. PulmonNet V1: Leveraging the benefit of Leaky ReLU activation for the local and multi-scale global feature integration of chest radiographs to classify pulmonary diseases. *Biomed. Signal Process. Control* **2024**, *96*, 106600.
40. Zhang, Z.; Sabuncu, M. Generalized cross entropy loss for training deep neural networks with noisy labels. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 1–11.
41. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
42. Connor, R.; Dearle, A.; Claydon, B.; Vadicamo, L. Correlations of cross-entropy loss in machine learning. *Entropy* **2024**, *26*, 491. [CrossRef]

43. Jadon, A.; Patil, A.; Jadon, S. A Comprehensive Survey of Regression-Based Loss Functions for Time Series Forecasting. In Proceedings of the International Conference on Data Management, Analytics & Innovation, Pune, India, 20–22 January 2014; Springer: Berlin/Heidelberg, Germany, 2024; pp. 117–147.
44. Zheng, W.; Chen, G. An accurate GRU-based power time-series prediction approach with selective state updating and stochastic optimization. *IEEE Trans. Cybern.* **2021**, *52*, 13902–13914. [CrossRef]
45. Wilson, A.C.; Roelofs, R.; Stern, M.; Srebro, N.; Recht, B. The marginal value of adaptive gradient methods in machine learning. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–11.
46. Liu, M.; Mroueh, Y.; Ross, J.; Zhang, W.; Cui, X.; Das, P.; Yang, T. Towards better understanding of adaptive gradient algorithms in generative adversarial nets. *arXiv* **2019**, arXiv:1912.11940.
47. Peng, Y.L.; Lee, W.P. Practical guidelines for resolving the loss divergence caused by the root-mean-squared propagation optimizer. *Appl. Soft Comput.* **2024**, *153*, 111335. [CrossRef]
48. Kingma, D.P. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
49. Zhang, Z. Improved adam optimizer for deep neural networks. In Proceedings of the 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS), Banff, AB, Canada, 4–6 June 2018; pp. 1–2.
50. Taye, M.M. Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions. *Computation* **2023**, *11*, 52. [CrossRef]
51. González-Rodríguez, L.; Plasencia-Salgueiro, A. Uncertainty-Aware autonomous mobile robot navigation with deep reinforcement learning. In *Deep Learning for Unmanned Systems*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 225–257.
52. Sarıgül, M.; Ozyildirim, B.M.; Avci, M. Differential convolutional neural network. *Neural Netw.* **2019**, *116*, 279–287. [CrossRef]
53. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
54. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
55. Tsantekidis, A.; Passalis, N.; Tefas, A. Recurrent neural networks. In *Deep Learning for Robot Perception and Cognition*; Elsevier: Amsterdam, The Netherlands, 2022; pp. 101–115.
56. Esenogho, E.; Mienye, I.D.; Swart, T.G.; Aruleba, K.; Obaido, G. A neural network ensemble with feature engineering for improved credit card fraud detection. *IEEE Access* **2022**, *10*, 16400–16407. [CrossRef]
57. Ni, Q.; Ji, J.; Feng, K.; Zhang, Y.; Lin, D.; Zheng, J. Data-driven bearing health management using a novel multi-scale fused feature and gated recurrent unit. *Reliab. Eng. Syst. Saf.* **2024**, *242*, 109753. [CrossRef]
58. Ala'raj, M.; Abbod, M.F.; Majdalawieh, M. Modelling customers credit card behaviour using bidirectional LSTM neural networks. *J. Big Data* **2021**, *8*, 69. [CrossRef]
59. Schuster, M.; Paliwal, K.K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681. [CrossRef]
60. Shi, H.; Xu, M.; Li, R. Deep learning for household load forecasting—A novel pooling deep RNN. *IEEE Trans. Smart Grid* **2017**, *9*, 5271–5280. [CrossRef]
61. Capra, M.; Bussolino, B.; Marchisio, A.; Masera, G.; Martina, M.; Shafique, M. Hardware and software optimizations for accelerating deep neural networks: Survey of current trends, challenges, and the road ahead. *IEEE Access* **2020**, *8*, 225134–225180. [CrossRef]
62. Koo, K.; Park, M.; Yoon, B. A suspicious financial transaction detection model using autoencoder and risk-based approach. *IEEE Access* **2024**, *12*, 68926–68939. [CrossRef]
63. Wu, D.; Hu, Y.; Liu, K.; Li, J.; Wang, X.; Deng, S.; Zheng, N.; Luo, X. An Outlier-Resilient Autoencoder for Representing High-Dimensional and Incomplete Data. *IEEE Trans. Emerg. Top. Comput. Intell.* **2024**, 1–13. [CrossRef]
64. Zhao, J.; Mathieu, M.; Goroshin, R.; Lecun, Y. Stacked what-where auto-encoders. *arXiv* **2015**, arXiv:1506.02351.
65. Hu, H.; Liao, M.; Mao, W.; Liu, W.; Zhang, C.; Jing, Y. Variational auto-encoder for text generation. In Proceedings of the 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 12–14 June 2020; pp. 595–598.
66. Badjie, B.; Cecílio, J.; Casimiro, A. Denoising Autoencoder-Based Defensive Distillation as an Adversarial Robustness Algorithm Against Data Poisoning Attacks. *ACM SIGAda Ada Lett.* **2024**, *43*, 30–35. [CrossRef]
67. Gao, L.; la Tour, T.D.; Tillman, H.; Goh, G.; Troll, R.; Radford, A.; Sutskever, I.; Leike, J.; Wu, J. Scaling and evaluating sparse autoencoders. *arXiv* **2024**, arXiv:2406.04093.
68. Berahmand, K.; Daneshfar, F.; Salehi, E.S.; Li, Y.; Xu, Y. Autoencoders and their applications in machine learning: A survey. *Artif. Intell. Rev.* **2024**, *57*, 28. [CrossRef]
69. Han, K.; Wang, Y.; Chen, H.; Chen, X.; Guo, J.; Liu, Z.; Tang, Y.; Xiao, A.; Xu, C.; Xu, Y.; et al. A survey on vision transformer. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 87–110. [CrossRef]
70. Van Aken, B.; Winter, B.; Löser, A.; Gers, F.A. How does bert answer questions? a layer-wise analysis of transformer representations. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 1823–1832.
71. Yenduri, G.; Ramalingam, M.; Selvi, G.C.; Supriya, Y.; Srivastava, G.; Maddikunta, P.K.R.; Raj, G.D.; Jhaveri, R.H.; Prabadevi, B.; Wang, W.; et al. Gpt (generative pre-trained transformer)—A comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions. *IEEE Access* **2024**, *12*, 54608–54649. [CrossRef]

72. Khemani, B.; Patil, S.; Kotecha, K.; Tanwar, S. A review of graph neural networks: Concepts, architectures, techniques, challenges, datasets, applications, and future directions. *J. Big Data* **2024**, *11*, 18. [CrossRef]
73. Waikhom, L.; Patgiri, R. A survey of graph neural networks in various learning paradigms: Methods, applications, and challenges. *Artif. Intell. Rev.* **2023**, *56*, 6295–6364. [CrossRef]
74. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Philip, S.Y. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4–24. [CrossRef] [PubMed]
75. Sabour, S.; Frosst, N.; Hinton, G.E. Dynamic routing between capsules. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–11.
76. Saif, A.; Imtiaz, T.; Rifat, S.; Shahnaz, C.; Zhu, W.P.; Ahmad, M.O. CapsCovNet: A modified capsule network to diagnose Covid-19 from multimodal medical imaging. *IEEE Trans. Artif. Intell.* **2021**, *2*, 608–617. [CrossRef]
77. Bengesi, S.; El-Sayed, H.; Sarker, M.K.; Houkpati, Y.; Irungu, J.; Oladunni, T. Advancements in Generative AI: A Comprehensive Review of GANs, GPT, Autoencoders, Diffusion Model, and Transformers. *IEEE Access* **2024**, *12*, 69812–69837. [CrossRef]
78. Chen, R.T.; Rubanova, Y.; Bettencourt, J.; Duvenaud, D.K. Neural Ordinary Differential Equations. *Adv. Neural Inf. Process. Syst.* **2018**, *31* . Available online: https://proceedings.neurips.cc/paper/2018/hash/69386f6bb1dfed68692a24c8686939b9-Abstract.html (accessed on 1 August 2024).
79. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.
80. Pham, H.; Guan, M.; Zoph, B.; Le, Q.; Dean, J. Efficient neural architecture search via parameters sharing. In Proceedings of the International Conference on Machine Learning, PMLR, Stockholm, Sweden, 10–15 July 2018; pp. 4095–4104.
81. Yu, K.; Sciuto, C.; Jaggi, M.; Musat, C.; Salzmann, M. Evaluating the search phase of neural architecture search. *arXiv* **2019**, arXiv:1902.08142.
82. Liu, H.; Simonyan, K.; Yang, Y. Darts: Differentiable architecture search. *arXiv* **2018**, arXiv:1806.09055.
83. Klyuchnikov, N.; Trofimov, I.; Artemova, E.; Salnikov, M.; Fedorov, M.; Filippov, A.; Burnaev, E. Nas-bench-nlp: Neural architecture search benchmark for natural language processing. *IEEE Access* **2022**, *10*, 45736–45747. [CrossRef]
84. Jin, W.; Derr, T.; Liu, H.; Wang, Y.; Wang, S.; Liu, Z.; Tang, J. Self-supervised learning on graphs: Deep insights and new direction. *arXiv* **2020**, arXiv:2006.10141.
85. Gui, J.; Chen, T.; Zhang, J.; Cao, Q.; Sun, Z.; Luo, H.; Tao, D. A Survey on Self-supervised Learning: Algorithms, Applications, and Future Trends. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 9052–9071. [CrossRef] [PubMed]
86. Penarrubia, C.; Valero-Mas, J.J.; Calvo-Zaragoza, J. Self-Supervised Learning for Text Recognition: A Critical Survey. *arXiv* **2024**, arXiv:2407.19889.
87. Sinha, S.; Dey, S.; Saha, G. Improving self-supervised learning model for audio spoofing detection with layer-conditioned embedding fusion. *Comput. Speech Lang.* **2024**, *86*, 101599. [CrossRef]
88. Li, E.; Bedi, S.; Melek, W. Anomaly detection in three-axis CNC machines using LSTM networks and transfer learning. *Int. J. Adv. Manuf. Technol.* **2023**, *127*, 5185–5198. [CrossRef]
89. Vrbančič, G.; Podgorelec, V. Transfer learning with adaptive fine-tuning. *IEEE Access* **2020**, *8*, 196197–196211. [CrossRef]
90. Alzubaidi, L.; Zhang, J.; Humaidi, A.J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaría, J.; Fadhel, M.A.; Al-Amidie, M.; Farhan, L. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *J. Big Data* **2021**, *8*, 53. [CrossRef]
91. Lim, W.Y.B.; Luong, N.C.; Hoang, D.T.; Jiao, Y.; Liang, Y.C.; Yang, Q.; Niyato, D.; Miao, C. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 2031–2063. [CrossRef]
92. Zhan, Y.; Zhang, J.; Hong, Z.; Wu, L.; Li, P.; Guo, S. A Survey of Incentive Mechanism Design for Federated Learning. *IEEE Trans. Emerg. Top. Comput.* **2022**, *10*, 1035–1044. [CrossRef]
93. Zhang, C.; Xie, Y.; Bai, H.; Yu, B.; Li, W.; Gao, Y. A survey on federated learning. *Knowl.-Based Syst.* **2021**, *216*, 106775. [CrossRef]
94. Abaoud, M.; Almuqrin, M.A.; Khan, M.F. Advancing federated learning through novel mechanism for privacy preservation in healthcare applications. *IEEE Access* **2023**, *11*, 83562–83579. [CrossRef]
95. Arulkumaran, K.; Deisenroth, M.P.; Brundage, M.; Bharath, A.A. Deep reinforcement learning: A brief survey. *IEEE Signal Process. Mag.* **2017**, *34*, 26–38. [CrossRef]
96. Sutton, R.S.; McAllester, D.; Singh, S.; Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. *Adv. Neural Inf. Process. Syst.* **1999**, *12*, 1057–1063.
97. Mnih, V. Asynchronous Methods for Deep Reinforcement Learning. *arXiv* **2016**, arXiv:1602.01783.
98. Lowe, R.; Wu, Y.I.; Tamar, A.; Harb, J.; Pieter Abbeel, O.; Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 6382–6393.
99. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347.
100. Wu, J.; Yang, H.; Yang, L.; Huang, Y.; He, X.; Lv, C. Human-Guided Deep Reinforcement Learning for Optimal Decision Making of Autonomous Vehicles. *IEEE Trans. Syst. Man Cybern. Syst.* **2024**, *54*, 6595–6609. [CrossRef]
101. Kim, Y.J.; Ahn, W.J.; Jang, S.H.; Lim, M.T.; Pae, D.S. A Reinforcement Learning Approach to Dynamic Trajectory Optimization with Consideration of Imbalanced Sub-Goals in Self-Driving Vehicles. *Appl. Sci.* **2024**, *14*, 5213. [CrossRef]
102. Xiao, W.; Yang, Y.; Mu, X.; Xie, Y.; Tang, X.; Cao, D.; Liu, T. Decision-making for autonomous vehicles in random task scenarios at unsignalized intersection using deep reinforcement learning. *IEEE Trans. Veh. Technol.* **2024**, *73*, 7812–7825. [CrossRef]

103. Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P.H.; Hospedales, T.M. Learning to compare: Relation network for few-shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1199–1208.
104. Snell, J.; Swersky, K.; Zemel, R. Prototypical networks for few-shot learning. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–11.
105. Sun, Q.; Liu, Y.; Chua, T.S.; Schiele, B. Meta-transfer learning for few-shot learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 403–412.
106. Chen, X.; He, K. Exploring simple siamese representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 15750–15758.
107. Khan, A.I.; Al-Habsi, S. Machine learning in computer vision. *Procedia Comput. Sci.* **2020**, *167*, 1444–1451. [CrossRef]
108. Li, S.; Kou, P.; Ma, M.; Yang, H.; Huang, S.; Yang, Z. Application of semi-supervised learning in image classification: Research on fusion of labeled and unlabeled data. *IEEE Access* **2024**, *12*, 27331–27343. [CrossRef]
109. Mienye, I.D.; Jere, N. Optimized ensemble learning approach with explainable AI for improved heart disease prediction. *Information* **2024**, *15*, 394. [CrossRef]
110. Yuvaraj, N.; Rajput, K.; Suganyadevi, K.; Aeri, M.; Shukla, R.P.; Gurjar, H. Multi-Scale Object Detection and Classification using Machine Learning and Image Processing. In Proceedings of the 2024 Second International Conference on Data Science and Information System (ICDSIS), Hassan, India, 17–18 May 2024; pp. 1–6. [CrossRef]
111. Zhao, Z.Q.; Zheng, P.; Xu, S.t.; Wu, X. Object detection with deep learning: A review. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3212–3232. [CrossRef]
112. Kumar, A.; Zhang, Z.J.; Lyu, H. Object detection in real time based on improved single shot multi-box detector algorithm. *EURASIP J. Wirel. Commun. Netw.* **2020**, *2020*, 204. [CrossRef]
113. Latif, A.; Rasheed, A.; Sajid, U.; Ahmed, J.; Ali, N.; Ratyal, N.I.; Zafar, B.; Dar, S.H.; Sajid, M.; Khalil, T. Content-Based Image Retrieval and Feature Extraction: A Comprehensive Review. *Math. Probl. Eng.* **2019**, *2019*, 9658350. [CrossRef]
114. Wang, Z.; Kim, S.; Joe, I. An Improved LSTM-Based Failure Classification Model for Financial Companies Using Natural Language Processing. *Appl. Sci.* **2023**, *13*, 7884. [CrossRef]
115. Munikoti, S.; Stewart, I.; Horawalavithana, S.; Kvinge, H.; Emerson, T.; Thompson, S.E.; Pazdernik, K. Generalist Multimodal AI: A Review of Architectures, Challenges and Opportunities. *arXiv* **2024**, arXiv:2406.05496.
116. Alayrac, J.B.; Donahue, J.; Luc, P.; Miech, A.; Barr, I.; Hasson, Y.; Lenc, K.; Mensch, A.; Millican, K.; Reynolds, M.; et al. Flamingo: A visual language model for few-shot learning. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 23716–23736.
117. Reed, S.; Zolna, K.; Parisotto, E.; Colmenarejo, S.G.; Novikov, A.; Barth-Maron, G.; Gimenez, M.; Sulsky, Y.; Kay, J.; Springenberg, J.T.; et al. A generalist agent. *arXiv* **2022**, arXiv:2205.06175.
118. Kheddar, H.; Hemis, M.; Himeur, Y. Automatic speech recognition using advanced deep learning approaches: A survey. *Inf. Fusion* **2024**, *109*, 102422. [CrossRef]
119. Kumar, Y.; Koul, A.; Singh, C. A deep learning approaches in text-to-speech system: A systematic review and recent research perspective. *Multimed. Tools Appl.* **2023**, *82*, 15171–15197. [CrossRef]
120. Amodei, D.; Ananthanarayanan, S.; Anubhai, R.; Bai, J.; Battenberg, E.; Case, C.; Casper, J.; Catanzaro, B.; Cheng, Q.; Chen, G.; et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In Proceedings of the International Conference on Machine Learning, PMLR, New York City, NY, USA, 19–24 June 2016; pp. 173–182.
121. Sanida, T.; Sanida, M.V.; Sideris, A.; Dasygenis, M. Optimizing Lung Condition Categorization through a Deep Learning Approach to Chest X-ray Image Analysis. *BioMedInformatics* **2024**, *4*, 2002–2021. [CrossRef]
122. Manole, I.; Butacu, A.I.; Bejan, R.N.; Tiplica, G.S. Enhancing Dermatological Diagnostics with EfficientNet: A Deep Learning Approach. *Bioengineering* **2024**, *11*, 810. [CrossRef]
123. McKinney, S.M.; Sieniek, M.; Godbole, V.; Godwin, J.; Antropova, N.; Ashrafian, H.; Back, T.; Chesus, M.; Corrado, G.; Darzi, A.; et al. International evaluation of an AI system for breast cancer screening. *Nature* **2020**, *577*, 89–94. [CrossRef]
124. Elazab, A.; Wang, C.; Abdelaziz, M.; Zhang, J.; Gu, J.; Gorriz, J.M.; Zhang, Y.; Chang, C. Alzheimer's disease diagnosis from single and multimodal data using machine and deep learning models: Achievements and future directions. *Expert Syst. Appl.* **2024**, *255*, 124780. [CrossRef]
125. Gorenstein, L.; Onn, A.; Green, M.; Mayer, A.; Segev, S.; Marom, E.M. A novel artificial intelligence based denoising method for ultra-low dose CT used for lung cancer screening. *Acad. Radiol.* **2023**, *30*, 2588–2597. [CrossRef]
126. Ouyang, D.; He, B.; Ghorbani, A.; Yuan, N.; Ebinger, J.; Langlotz, C.P.; Heidenreich, P.A.; Harrington, R.A.; Liang, D.H.; Ashley, E.A.; et al. Video-based AI for beat-to-beat assessment of cardiac function. *Nature* **2020**, *580*, 252–256. [CrossRef] [PubMed]
127. Liu, X.; Wang, Z. Deep learning in medical image classification from mri-based brain tumor images. *arXiv* **2024**, arXiv:2408.00636.
128. Zhou, L.; Wang, M.; Zhou, N. Distributed federated learning-based deep learning model for privacy mri brain tumor detection. *arXiv* **2024**, arXiv:2404.10026. [CrossRef]
129. Hroub, N.A.; Alsannaa, A.N.; Alowaifeer, M.; Alfarraj, M.; Okafor, E. Explainable deep learning diagnostic system for prediction of lung disease from medical images. *Comput. Biol. Med.* **2024**, *170*, 108012. [CrossRef] [PubMed]
130. Naz, A.; Khan, H.; Din, I.U.; Ali, A.; Husain, M. An Efficient Optimization System for Early Breast Cancer Diagnosis based on Internet of Medical Things and Deep Learning. *Eng. Technol. Appl. Sci. Res.* **2024**, *14*, 15957–15962. [CrossRef]
131. Senior, A.W.; Evans, R.; Jumper, J.; Kirkpatrick, J.; Sifre, L.; Green, T.; Qin, C.; Zidek, A.; Nelson, A.W.R.; Bridgland, A.; et al. Improved protein structure prediction using potentials from deep learning. *Nature* **2020**, *577*, 706–710. [CrossRef]

132. Avsec, Z.; Agarwal, V.; Visentin, D.; Ledsam, J.R.; Grabska-Barwinska, A.; Taylor, K.R.; Assael, Y.; Jumper, J.; Kohli, P.; Kelley, D.R. Effective gene expression prediction from sequence by integrating long-range interactions. *Nat. Methods* **2021**, *18*, 1196–1203. [CrossRef]
133. Ferruz, N.; Schmidt, S.; Höcker, B. ProtGPT2 is a deep unsupervised language model for protein design. *Nat. Commun.* **2022**, *13*, 4348. [CrossRef] [PubMed]
134. Alamdari, S.; Thakkar, N.; van den Berg, R.; Lu, A.X.; Fusi, N.; Amini, A.P.; Yang, K.K. Protein generation with evolutionary diffusion: Sequence is all you need. *BioRxiv* **2023**, 1–62. [CrossRef]
135. Stokes, J.M.; Yang, K.; Swanson, K.; Jin, W.; Cubillos-Ruiz, A.; Donghia, N.M.; MacNair, C.R.; French, S.; Carfrae, L.A.; Bloom-Ackermann, Z.; et al. A deep learning approach to antibiotic discovery. *Cell* **2020**, *180*, 688–702. [CrossRef]
136. Irwin, R.; Dimitriadis, S.; He, J.; Bjerrum, E.J. Chemformer: A pre-trained transformer for computational chemistry. *Mach. Learn. Sci. Technol.* **2022**, *3*, 015022. [CrossRef]
137. Das, M.; Ghosh, A.; Sunoj, R.B. Advances in machine learning with chemical language models in molecular property and reaction outcome predictions. *J. Comput. Chem.* **2024**, *45*, 1160–1176. [CrossRef]
138. Mswahili, M.E.; Jeong, Y.S. Transformer-Based Models For Chemical SMILES Representation: A Comprehensive Literature Review. *Heliyon* **2024**, *10*, e39038. . [CrossRef]
139. Sultan, A.; Sieg, J.; Mathea, M.; Volkamer, A. Transformers for molecular property prediction: Lessons learned from the past five years. *J. Chem. Inf. Model.* **2024**, *64*, 6259–6280. [CrossRef]
140. Schwaller, P.; Laino, T.; Gaudin, T.; Bolgar, P.; Hunter, C.A.; Bekas, C.; Lee, A.A. Predicting retrosynthetic pathways using transformer-based models and a hyper-graph exploration strategy. *Chem. Sci.* **2020**, *11*, 4475–4486. [CrossRef] [PubMed]
141. Kuenzi, B.M.; Park, J.; Fong, S.H.; Sanchez, K.S.; Lee, J.; Kreisberg, J.F.; Ma, J.; Ideker, T. Predicting drug response and synergy using a deep learning model of human cancer cells. *Cancer Cell* **2020**, *38*, 672–684. [CrossRef]
142. Li, K.; Shen, B.; Feng, F.; Li, X.; Wang, Y.; Feng, N.; Tang, Z.; Ma, L.; Li, H. A disentangled generative model for improved drug response prediction in patients via sample synthesis. *J. Pharm. Anal.* **2024**, 101128. [CrossRef]
143. Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Zidek, A.; Potapenko, A.; et al. Highly accurate protein structure prediction with AlphaFold. *Nature* **2021**, *596*, 583–589. [CrossRef]
144. Chen, H.; King, F.J.; Zhou, B.; Wang, Y.; Canedy, C.J.; Hayashi, J.; Zhong, Y.; Chang, M.W.; Pache, L.; Wong, J.L.; et al. Drug target prediction through deep learning functional representation of gene signatures. *Nat. Commun.* **2024**, *15*, 1853. [CrossRef] [PubMed]
145. Sivasubramaniam, S.; Balamurugan, S. Early detection and prediction of Heart Disease using Wearable devices and Deep Learning algorithms. *Multimed. Tools Appl.* **2024**, 1–15. [CrossRef]
146. Koca, Ö.A.; Kabak, H.Ö.; Kılıç, V. Attention-based multilayer GRU decoder for on-site glucose prediction on smartphone. *J. Supercomput.* **2024**, *80*, 25616–25639. [CrossRef]
147. Davidashvilly, S.; Cardei, M.; Hssayeni, M.; Chi, C.; Ghoraani, B. Deep neural networks for wearable sensor-based activity recognition in Parkinson's disease: Investigating generalizability and model complexity. *BioMedical Eng. OnLine* **2024**, *23*, 17. [CrossRef]
148. Barua, P.D.; Vicnesh, J.; Lih, O.S.; Palmer, E.E.; Yamakawa, T.; Kobayashi, M.; Acharya, U.R. Artificial intelligence assisted tools for the detection of anxiety and depression leading to suicidal ideation in adolescents: A review. *Cogn. Neurodyn.* **2024**, *18*, 1–22. [CrossRef]
149. Iqbal, T.; Simpkin, A.J.; Roshan, D.; Glynn, N.; Killilea, J.; Walsh, J.; Molloy, G.; Ganly, S.; Ryman, H.; Coen, E.; et al. Stress monitoring using wearable sensors: A pilot study and stress-predict dataset. *Sensors* **2022**, *22*, 8135. [CrossRef]
150. Ahmed, A.A.; Ali, W.; Abdullah, T.A.; Malebary, S.J. Classifying cardiac arrhythmia from ECG signal using 1D CNN deep learning model. *Mathematics* **2023**, *11*, 562. [CrossRef]
151. Baseer, K.; Sivakumar, K.; Veeraiah, D.; Chhabra, G.; Lakineni, P.K.; Pasha, M.J.; Gandikota, R.; Harikrishnan, G. Healthcare diagnostics with an adaptive deep learning model integrated with the Internet of medical Things (IoMT) for predicting heart disease. *Biomed. Signal Process. Control* **2024**, *92*, 105988. [CrossRef]
152. Gaber, A.; Abdeltwab, H.; Elbatt, T. FedCVD: Towards a Scalable, Privacy-Preserving Federated Learning Model for Cardiovascular Diseases Prediction. In Proceedings of the 2024 8th International Conference on Machine Learning and Soft Computing, Singapore, 26–28 January 2024; pp. 7–11. [CrossRef]
153. Tan, Y.N.; Tinh, V.P.; Lam, P.D.; Nam, N.H.; Khoa, T.A. A transfer learning approach to breast cancer classification in a federated learning framework. *IEEe Access* **2023**, *11*, 27462–27476. [CrossRef]
154. Siganporia, A.; Varia, M.; Gorimar, U.; Nanade, A. Privacy-Enhanced Federated Learning for Rare Genetic Disorder Classification with EHR. In Proceedings of the 2023 Global Conference on Information Technologies and Communications (GCITC), Bengaluru, India, 1–3 December 2023; pp. 1–8. [CrossRef]
155. Rehman, A.; Razzak, I.; Xu, G. Federated learning for privacy preservation of healthcare data from smartphone-based side-channel attacks. *IEEE J. Biomed. Health Inform.* **2022**, *27*, 684–690. [CrossRef]
156. Saidi, R.; Rahmany, I.; Dhahri, S.; Moulahi, T. A Privacy-Enhanced Framework for Chest Disease Classification using Federated Learning and Blockchain. *IEEE Access* **2024**. [CrossRef]
157. Zhang, Y.; Zheng, X.; Zhang, C. Deep learning in algorithmic trading: A study on S&P 500 index future forecasting. *J. Forecast.* **2020**, *39*, 675–688. [CrossRef]

158. Song, H.; Choi, H. Forecasting stock market indices using the recurrent neural network based hybrid models: CNN-LSTM, GRU-CNN, and ensemble models. *Appl. Sci.* **2023**, *13*, 4644. [CrossRef]
159. Théate, T.; Ernst, D. An application of deep reinforcement learning to algorithmic trading. *Expert Syst. Appl.* **2021**, *173*, 114632. [CrossRef]
160. Majidi, N.; Shamsi, M.; Marvasti, F. Algorithmic trading using continuous action space deep reinforcement learning. *Expert Syst. Appl.* **2024**, *235*, 121245. [CrossRef]
161. Ma, C.; Zhang, J.; Li, Z.; Xu, S. Multi-agent deep reinforcement learning algorithm with trend consistency regularization for portfolio management. *Neural Comput. Appl.* **2023**, *35*, 6589–6601. [CrossRef]
162. Vuletić, M.; Prenzel, F.; Cucuringu, M. Fin-gan: Forecasting and classifying financial time series via generative adversarial networks. *Quant. Financ.* **2024**, *24*, 175–199. [CrossRef]
163. Gambo, M.L.; Zainal, A.; Kassim, M.N. A convolutional neural network model for credit card fraud detection. In Proceedings of the 2022 International Conference on Data Science and Its Applications (ICoDSA), Bandung, Indonesia, 6–7 July 2022; pp. 198–202. [CrossRef]
164. Hiremath, A.C.; Arya, A.; Sriranga, L.; Reddy, K.V.S.R.; Nikhil, M. Ensemble of Graph Neural Networks for Enhanced Financial Fraud Detection. In Proceedings of the 2024 IEEE 9th International Conference for Convergence in Technology (I2CT), Pune, India, 5–7 April 2024; pp. 1–8. [CrossRef]
165. Dang, T.K.; Tran, T.C.; Tuan, L.M.; Tiep, M.V. Machine learning based on resampling approaches and deep reinforcement learning for credit card fraud detection systems. *Appl. Sci.* **2021**, *11*, 10004. [CrossRef]
166. Yu, C.; Xu, Y.; Cao, J.; Zhang, Y.; Jin, Y.; Zhu, M. Credit card fraud detection using advanced transformer model. *arXiv* **2024**, arXiv:2406.03733.
167. Yang, X.; Zhang, C.; Sun, Y.; Pang, K.; Jing, L.; Wa, S.; Lv, C. FinChain-BERT: A High-Accuracy Automatic Fraud Detection Model Based on NLP Methods for Financial Scenarios. *Information* **2023**, *14*, 499. [CrossRef]
168. Mienye, I.D.; Sun, Y. A deep learning ensemble with data resampling for credit card fraud detection. *IEEE Access* **2023**, *11*, 30628–30638. [CrossRef]
169. Langevin, A.; Cody, T.; Adams, S.; Beling, P. Generative adversarial networks for data augmentation and transfer in credit card fraud detection. *J. Oper. Res. Soc.* **2022**, *73*, 153–180. [CrossRef]
170. Lin, K.; Gao, Y. Model interpretability of financial fraud detection by group SHAP. *Expert Syst. Appl.* **2022**, *210*, 118354. [CrossRef]
171. Merćep, A.; Mrčela, L.; Birov, M.; Kostanjčar, Z. Deep neural networks for behavioral credit rating. *Entropy* **2020**, *23*, 27. [CrossRef]
172. Feng, B.; Xu, H.; Xue, W.; Xue, B. Every corporation owns its structure: Corporate credit rating via graph neural networks. In Proceedings of the Chinese Conference on Pattern Recognition and Computer Vision (PRCV), Shenzhen, China, 4–7 November 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 688–699. [CrossRef]
173. Chen, C.T.; Lee, C.; Huang, S.H.; Peng, W.C. Credit Card Fraud Detection via Intelligent Sampling and Self-supervised Learning. *ACM Trans. Intell. Syst. Technol.* **2024**, *15*, 1–29. [CrossRef]
174. Pławiak, P.; Abdar, M.; Acharya, U.R. Application of new deep genetic cascade ensemble of SVM classifiers to predict the Australian credit scoring. *Appl. Soft Comput.* **2019**, *84*, 105740. [CrossRef]
175. Feng, D.; Haase-Schütz, C.; Rosenbaum, L.; Hertlein, H.; Glaeser, C.; Timm, F.; Wiesbeck, W.; Dietmayer, K. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 1341–1360. [CrossRef]
176. Lee, D.H.; Liu, J.L. End-to-end deep learning of lane detection and path prediction for real-time autonomous driving. *Signal Image Video Process.* **2023**, *17*, 199–205. [CrossRef]
177. Xue, Y.; Chen, W. Multi-agent deep reinforcement learning for UAVs navigation in unknown complex environment. *IEEE Trans. Intell. Veh.* **2023**, *9*, 2290–2303. [CrossRef]
178. Hu, T.K.; Gama, F.; Chen, T.; Zheng, W.; Wang, Z.; Ribeiro, A.; Sadler, B.M. Scalable perception-action-communication loops with convolutional and graph neural networks. *IEEE Trans. Signal Inf. Process. Netw.* **2021**, *8*, 12–24. [CrossRef]
179. Han, D.; Mulyana, B.; Stankovic, V.; Cheng, S. A survey on deep reinforcement learning algorithms for robotic manipulation. *Sensors* **2023**, *23*, 3762. [CrossRef]
180. Xu, D.; Fan, H.; Deng, J.; Wei, H.; Xu, P.; Zhang, Y. Virtual Muscle-Model-Modified Reinforcement Learning for Tilt Motion Control of a Novel Magnetic Actuated Flexible-Joint Robotic Surgical Camera System. *IEEE/ASME Trans. Mechatron.* **2024**. [CrossRef]
181. Zhou, J.; Li, Y.; Qin, H.; Dai, P.; Zhao, Z.; Hu, M. Sonar image generation by MFA-CycleGAN for boosting underwater object detection of AUVs. *IEEE J. Ocean. Eng.* **2024**, *49*, 905–919. [CrossRef]
182. Zheng, Y.Y.; Kong, J.L.; Jin, X.B.; Wang, X.Y.; Su, T.L.; Zuo, M. CropDeep: The Crop Vision Dataset for Deep-Learning-Based Classification and Detection in Precision Agriculture. *Sensors* **2019**, *19*, 1058. [CrossRef]
183. Lee, D.; Lee, S.; Masoud, N.; Krishnan, M.; Li, V.C. Digital twin-driven deep reinforcement learning for adaptive task allocation in robotic construction. *Adv. Eng. Inform.* **2022**, *53*, 101710. [CrossRef]
184. Jain, D.K.; Zhao, X.; Gan, C.; Shukla, P.K.; Jain, A.; Sharma, S. Fusion-driven deep feature network for enhanced object detection and tracking in video surveillance systems. *Inf. Fusion* **2024**, *109*, 102429. [CrossRef]
185. Akhyar, A.; Zulkifley, M.A.; Lee, J.; Song, T.; Han, J.; Cho, C.; Hyun, S.; Son, Y.; Hong, B.W. Deep artificial intelligence applications for natural disaster management systems: A methodological review. *Ecol. Indic.* **2024**, *163*, 112067. [CrossRef]

186. Vidal, Y.; Puruncajas, B.; Castellani, F.; Tutivén, C. Predictive maintenance of wind turbine's main bearing using wind farm SCADA data and LSTM neural networks. *J. Phys. Conf. Ser.* **2023**, *2507*, 012024. [CrossRef]
187. Fan, G.F.; Han, Y.Y.; Li, J.W.; Peng, L.L.; Yeh, Y.H.; Hong, W.C. A hybrid model for deep learning short-term power load forecasting based on feature extraction statistics techniques. *Expert Syst. Appl.* **2024**, *238*, 122012. [CrossRef]
188. Yu, L.; Xie, W.; Xie, D.; Zou, Y.; Zhang, D.; Sun, Z.; Zhang, L.; Zhang, Y.; Jiang, T. Deep Reinforcement Learning for Smart Home Energy Management. *IEEE Internet Things J.* **2020**, *7*, 2751–2762. [CrossRef]
189. Al-Ali, E.M.; Hajji, Y.; Said, Y.; Hleili, M.; Alanzi, A.M.; Laatar, A.H.; Atri, M. Solar Energy Production Forecasting Based on a Hybrid CNN-LSTM-Transformer Model. *Mathematics* **2023**, *11*, 676. [CrossRef]
190. Han, G.; Lee, S.; Lee, J.; Lee, K.; Bae, J. Deep-learning- and reinforcement-learning-based profitable strategy of a grid-level energy storage system for the smart grid. *J. Energy Storage* **2021**, *41*, 102868. [CrossRef]
191. Yoon, D.H.; Yoon, J. Development of a real-time fault detection method for electric power system via transformer-based deep learning model. *Int. J. Electr. Power Energy Syst.* **2024**, *159*, 110069. [CrossRef]
192. Elsisi, M.; Tran, M.Q.; Mahmoud, K.; Lehtonen, M.; Darwish, M.M.F. Deep Learning-Based Industry 4.0 and Internet of Things towards Effective Energy Management for Smart Buildings. *Sensors* **2021**, *21*, 1038. [CrossRef]
193. Kamilaris, A.; Prenafeta-Boldú, F.X. Deep Learning in Agriculture: A Survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90. [CrossRef]
194. Gastli, M.S.; Nassar, L.; Karray, F. Satellite Images and Deep Learning Tools for Crop Yield Prediction and Price Forecasting. In Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN), Virtual, 18–22 July 2021; pp. 1–8. [CrossRef]
195. Ferentinos, K.P. Deep Learning Models for Plant Disease Detection and Diagnosis. *Comput. Electron. Agric.* **2018**, *145*, 311–318. [CrossRef]
196. Sami, M.; Khan, S.Q.; Khurram, M.; Farooq, M.U.; Anjum, R.; Aziz, S.; Qureshi, R.; Sadak, F. A Deep Learning-Based Sensor Modeling for Smart Irrigation System. *Agronomy* **2022**, *12*, 212. [CrossRef]
197. Padhiary, M.; Saha, D.; Kumar, R.; Sethi, L.N.; Kumar, A. Enhancing precision agriculture: A comprehensive review of machine learning and AI vision applications in all-terrain vehicle for farm automation. *Smart Agric. Technol.* **2024**, *8*, 100483. [CrossRef]
198. Santos, T.T.; de Souza, K.X.; Camargo Neto, J.; Koenigkan, L.V.; Moreira, A.S.; Ternes, S. Multiple orange detection and tracking with 3-D fruit relocalization and neural-net based yield regression in commercial sweet orange orchards. *Comput. Electron. Agric.* **2024**, *224*, 109199. [CrossRef]
199. Pavlovic, M.; Ilic, S.; Ralevic, N.; Antonic, N.; Raffa, D.W.; Bandecchi, M.; Culibrk, D. A Deep Learning Approach to Estimate Soil Organic Carbon from Remote Sensing. *Remote Sens.* **2024**, *16*, 655. [CrossRef]
200. Mckay, M.; Danilevicz, M.F.; Ashworth, M.B.; Rocha, R.L.; Upadhyaya, S.R.; Bennamoun, M.; Edwards, D. Focus on the Crop Not the Weed: Canola Identification for Precision Weed Management Using Deep Learning. *Remote Sens.* **2024**, *16*, 2041. [CrossRef]
201. Feng, Y.; Mei, D.; Zhao, H. Auction-based deep learning-driven smart agricultural supply chain mechanism. *Appl. Soft Comput.* **2023**, *149*, 111009. [CrossRef]
202. Li, Y.; Xiang, Y.; Guo, H.; Liu, P.; Liu, C. Swin transformer combined with convolution neural network for surface defect detection. *Machines* **2022**, *10*, 1083. [CrossRef]
203. Li, S.; Wu, C.; Xiong, N. Hybrid architecture based on CNN and transformer for strip steel surface defect classification. *Electronics* **2022**, *11*, 1200. [CrossRef]
204. Zhang, H.; Zhang, Q.; Shao, S.; Niu, T.; Yang, X. Attention-based LSTM network for rotatory machine remaining useful life prediction. *IEEE Access* **2020**, *8*, 132188–132199. [CrossRef]
205. Li, X.; Yue, C.; Liu, X.; Zhou, J.; Wang, L. ACWGAN-GP for milling tool breakage monitoring with imbalanced data. *Robot. Comput.-Integr. Manuf.* **2024**, *85*, 102624. [CrossRef]
206. Yan, K.; Lu, C.; Ma, X.; Ji, Z.; Huang, J. Intelligent fault diagnosis for air handing units based on improved generative adversarial network and deep reinforcement learning. *Expert Syst. Appl.* **2024**, *240*, 122545. [CrossRef]
207. Sorostinean, R.; Burghelea, Z.; Gellert, A. Anomaly Detection in Smart Industrial Machinery through Hidden Markov Models and Autoencoders. *IEEE Access* **2024**, *12*, 69217–69228. [CrossRef]
208. Mienye, I.D.; Swart, T.G.; Obaido, G. Fairness Metrics in AI Healthcare Applications: A Review. In Proceedings of the 2024 IEEE International Conference on Information Reuse and Integration for Data Science (IRI), San Jose, CA, USA, 7–9 August 2024; pp. 284–289.