IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 15, No. 6, December 2026, pp. 5016~5030
ISSN: 2252-8938, DOI: 10.11591/ijai.v15.i6.pp5016-5030

Overview of deep learning: techniques, approaches, and applications

Hamza Rabiai[1], Sanae Khali Issa[2], Kaouthar El Fassi[1]
[1]Laboratory of Mathematics and Applications, Faculty of Science and Technologies of Tangier, Abdelmalek Essaadi University, Tangier, Morocco
[2]Laboratory of Innovative Technologies, Faculty of Science and Technologies of Tangier, Abdelmalek Essaadi University, Tangier, Morocco

Article Info

Article history:

Received Jan 1, 2025
Revised Aug 10, 2026
Accepted Aug 31, 2026

Keywords:

Convolutional neural networks
Deep learning
Deep learning applications
Neural network architectures
Training algorithms

ABSTRACT

Deep learning has emerged as the dominant paradigm in machine learning, revolutionizing artificial intelligence through its ability to automatically learn hierarchical data representations. This paper presents a comprehensive survey of deep learning methodologies, systematically reviewing core architectures including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and long short-term memory (LSTM) networks, along with their mathematical foundations and training algorithms. The exploratory analysis across five critical application domains, computer vision, medical image processing, time-series forecasting, natural language processing (NLP), and big data analytics, reveals breakthrough performances in speech recognition, image understanding, and decision-making tasks. This review provides researchers and practitioners with a consolidated understanding of current techniques and highlights emerging trends that will shape the next generation of intelligent systems, particularly in addressing challenges of multimodal learning and limited dataset scenarios.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Hamza Rabiai
Laboratory of Mathematics and Applications, Faculty of Science and Technologies of Tangier
Abdelmalek Essaadi University
Old Airport Road, Km 10, Ziaten, Tangier, Morocco
Email: hamza.rabiai@etu.uae.ac.ma

1. INTRODUCTION

Over the past decade, machine learning has attracted growing interest from both the research community and industry. Applications such as multimedia content retrieval, image classification, video recommendation, social network analysis, and text mining illustrate this widespread adoption. Among the various machine learning paradigms employed in these applications, deep learning, also known as representation learning [1], has emerged as particularly prominent.

As a machine learning paradigm, neural networks draw their inspiration from the structure and function of the human brain and nervous system. Their processing units are arranged into input, hidden, and output layers, with each unit in a given layer connected to units in the layer beneath it through weighted links. Every unit computes a weighted sum of its inputs, which is subsequently passed through an activation function, commonly a sigmoid, hyperbolic tangent, or rectified linear unit (ReLU), to produce its output. These functions are favored in part because their derivatives are mathematically convenient [2], which eases the calculation of error-delta partial derivatives with respect to individual weights during training. Owing to this flexibility, neural networks have been applied across a broad

spectrum of tasks, ranging from pattern recognition, classification, clustering, and dimensionality reduction to computer vision, natural language processing (NLP), regression, and predictive analytics. Given its remarkable achievements across numerous domains, deep learning ranks among today's most active research areas within the machine learning community.

In this survey, deep learning is viewed from a variety of angles, including history, challenges, deep neural network (DNN) architectures, algorithms, frameworks, and applications, as well as parallel and distributed computing methods. Grasping where deep learning stands today requires an understanding of the conceptual and historical path that led to it. Deep learning has evolved significantly through history, overcoming numerous challenges. Its roots trace back to Aristotle's "associationism", which explored human cognition. The foundations of the modern era were laid in 1943, when McCulloch and Pitts [3] proposed the first artificial neural prototype, simulating human reasoning using "threshold logic" without learning capabilities. Progress accelerated in 1957 when Rosenblatt [4] introduced the perceptron, a two-layer network recognizing simple patterns. However, neural networks faced setbacks, such as XOR problem in 1969 [5] and debates on their necessity following the universal approximation theorem [6]. The "AI winter" ensued due to limited training methods and understanding of DNNs. This changed with the backpropagation algorithm, proposed in the 1970s and widely applied by the 1980s [7]. Breakthroughs like the "neocognitron" (1980) [8], recurrent neural networks (RNNs) (1986), and LeNet (1990s) [9] advanced DNNs, though hardware limitations hindered large-scale applications.

A turning point occurred in 2006, when deep belief networks (DBNs) introduced an effective pretraining strategy that enabled deeper architectures to be trained successfully, giving rise to the term "deep learning" itself [10]. Building on artificial neural networks (ANNs), deep learning has since become one of the most effective machine learning techniques, driven by inexpensive computing resources like general-purpose computing on graphics processing units (GPGPUs) and vast datasets. GPGPUs, with numerous parallel cores, outperform central processing units (CPUs) in DNN tasks [2]. Today, deep learning impacts diverse fields, including education, healthcare, and marketing, and continues to reshape industries [11]. Having explored the historical development of deep learning, we now examine the fundamental building blocks that make these systems possible.

An ANN is essentially a set of interconnected neurons that work as parallel information processors. Collectively, they can approximate an unknown "black-box" mapping between inputs and outputs, or divide the input space into finite classes. When stacked, these multiplexed neurons form a network capable of exchanging information with its environment through a designated layer of neurons. Learning rules and design objectives govern how information is propagated between the layers positioned inside this black box, corresponding to the network's hidden layers, while certain units feed processed information back out to the environment, forming the network's output layers. Within a hidden layer, each neuron receives weighted contributions from some or all neurons of the preceding layer. Formally, the output is an m-dimensional vector obtained by multiplying the input vector by an m×n weight matrix, so that an n-dimensional input is mapped onto an m-dimensional output, with each individual neuron behaving as a linear classifier. Collectively, the m neurons define an m-dimensional hyperplane that separates the output space into two classes [12]. Weight values are typically estimated using the least mean-squares (LMS) criterion, which minimizes the error between the network's computed outputs and the true outputs observed in the training data, analogous to fitting a linear regression.

Stacking multiple purely linear layers offers no representational advantage over a single layer, since the composition of linear transformations reduces mathematically to one equivalent matrix multiplication. To introduce genuine nonlinearity, each neuron adds a bias term to the weighted sum of its inputs before passing the result through a nonlinear activation function, thereby enabling nonlinear classification or regression. The activation function, a design parameter, is a monotonically increasing function with a zero inflection point, shifted by the bias to center it. Neurons can produce pulses through scaling and subtraction, and the ANN learns approximations by dividing the input space into samples to minimize the discrepancy between predicted and target outputs. According to the universal approximation theorem, an ANN with enough neurons and layers can approximate any functional mapping with arbitrary accuracy [13], [14]. Figure 1 depicts a single-neuron perceptron, in which the nonlinear activation function σ converts the weighted-sum output Σ into the neuron's final output value. The output network is optimized to minimize the error function by considering input-output pairings, layers, neurons, activation function shape, and error function, resulting in an optimal approximation of known data. Given these foundational principles and the rapid advancement of deep learning technologies, this research addresses several key objectives.

Few developments in computer science over the past years rival the impact of deep learning. Nearly every scientific field has been impacted by it. Businesses and entire industries are already being reshaped by it, and leading economies and technology companies alike are racing to push the field further. In a number of domains, including movie rating prediction, loan application approval, and car delivery time, deep learning has already surpassed human performance [2]. Its significance was formally recognized on 27 March 2019,

when Yoshua Bengio, Geoffrey Hinton, and Yann LeCun, three pioneers of deep learning, received the turing award, often described as computing's equivalent of the nobel prize. Deep learning still has potential for development, despite tremendous advancements.

Deep learning is utilized in many cases where intelligent machines could be useful, including vision, speech recognition, language comprehension, biometrics, navigation on Mars without a human expert, and personalization for solutions in specific situations [15]. This study offers a broad yet coherent overview of deep learning by consolidating knowledge that is otherwise scattered across the literature. Its contribution lies in synthesizing the work of leading researchers across the breadth and depth of the field. Various deep-network architectures are introduced across multiple application domains, including image processing and NLP, various types of deep networks are introduced. The article also discusses and contrasts several well-known deep learning technologies, as well as their respective optimization methods. Additionally, a number of deep learning applications are examined to aid other academics in broadening their perspectives on the field.

The remainder of this paper is organized as follows. Popular deep learning networks, and frameworks are briefly introduced in section 2. A range of deep learning techniques, and methods, are covered in section 3. Training algorithms are highlighted in section 4, and section 5 provides a summary of deep learning applications. The study's conclusion and future work is provided in section 6.

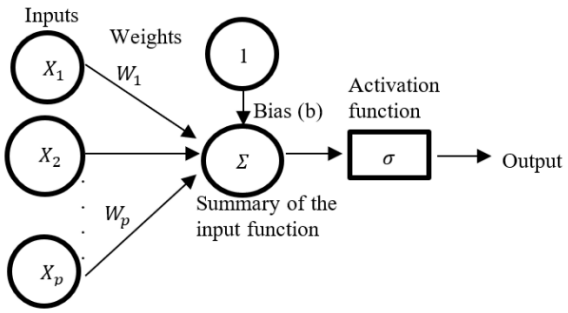

Figure 1. The perceptron learning model

2. CLASSIFICATION OF NEURAL NETWORKS

Neural networks can be broadly classified into various types, each suited for different kinds of tasks and data structures. These include models that range from simple feedforward architectures to more complex designs, such as recurrent networks and self-organizing maps (SOMs). By understanding the unique characteristics of each type, this study can better leverage their strengths for specific applications in machine learning and artificial intelligence. This study explore the main categories of neural networks in detail. Information moves from the input layer to the output layer of a feedforward neural network only in one direction, passing via hidden nodes. They don't make any loops or circles. As shown in Figure 2, a specific kind of multilayer feedforward neural network implementation is depicted in Figure 2(a) [2], where values and functions are computed along the forward pass path. In addition, a time-unrolled representation of RNN is shown in Figure 2(b). Z stands for the inputs' weighted sum, and y denotes Z's non-linear activation function (f) for each layer. The unit's bias value is denoted by b, while the weights between the two units in the adjacent layers are represented by W and the subscript letters, respectively.

This architecture continues to be widely favored by both academics and across various disciplines because of its simplicity and effective training in this paradigm. This model is trained using the most used learning algorithm, backpropagation utilizing gradient descent (GD) [7]. GD is used to update the weight parameters so as to reduce the prediction error after the algorithm first initializes them at random. Multiple consecutive forward and backward passes are made during the learning process. Through several hidden layers of nonlinearity, this study forward the input toward the output in a forward pass. In the end, a comparison is drawn between the predicted output and the actual target value for the same input, after which the partial derivatives of the resulting error with respect to the network parameters are backpropagated in the backward pass to modify the weights and minimize output error. Until achieve the required improvement in the model prediction, the process is repeated several times. Denoting the input to layer i as X_i, its corresponding output can be written as X_{i+1}, f_i is the nonlinear activation function. This relationship is expressed in (1).

$$X_{i+1} = f_i(W_i X_i + b_i) \quad (1)$$

X_{i+1}, since this serves as the following layer's input. The parameters that link layer i to the layer before it are W_i and b_i. These parameters can be modified within the backward pass. The updated weights and biases are given by (2) and (3), respectively.

$$W_{new} = w - \eta \partial E \,/\, \partial w \tag{2}$$

$$b_{new} = b - \eta \partial E \,/\, \partial b \tag{3}$$

Where E denotes the loss function, η is the learning rate step, while W_{new} and b_{new} correspond to the updated values of W and b, respectively. The specific form the model's cost function takes depends on the task to be completed, such as regression or classification. Similar to the root mean square error in regression and the softmax function in classification.

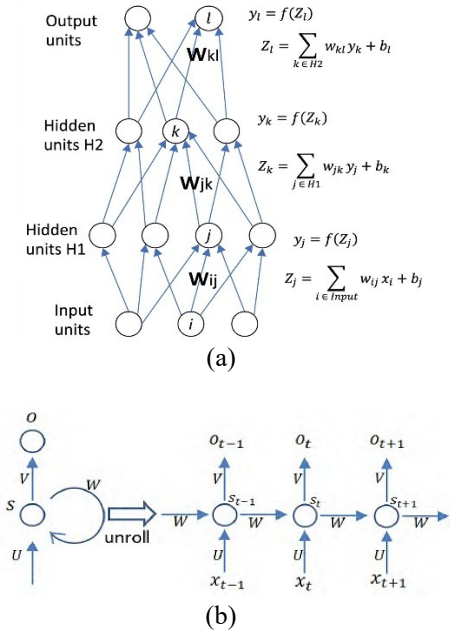

Figure 2. Overview of representative neural network architectures: (a) multilayer feedforward neural network with forward propagation and (b) time-unrolled RNN showing temporal dependencies

Unlike feedforward neural networks, RNNs have a cycle of processing units. A feedback loop is formed when a layer's output becomes an input to itself and the input supplied to the next layer, which is typically the network's only layer. This enables the network to retain information about past states and utilize it to affect the output at that moment. The ability of RNN to map a sequence of inputs onto a corresponding sequence of outputs, in contrast to feedforward neural networks, is a significant result of this difference. This makes RNN very useful for applications including speech recognition and frame-by-frame video classification, which entail processing time-sequential input data. A time-unrolled RNN representation is shown in Figure 2(b). For example, if an input consists of a string of three words, each individual word corresponds to a distinct layer, with the network unrolled three separate times to create a three-layer RNN. The underlying mathematical formulation is provided as follows: the input at time t is represented by x_t. The learnt parameters that are shared at every time step are denoted U, V, and W. The output at time t is O_t. The state at time t is represented by S_t, which may be calculated as given in (4), where f is the activation function, such as ReLU [16].

$$s_t = f(\cup x_t + w s_{t-1}) \tag{4}$$

Function of radial basis neural networks are used to solve problems related to time series prediction, function approximation, and classification. There are three layers in it: input, concealed, and output. Every node in the hidden layer represents a cluster center, and the radial basis function (expressed as a Gaussian function) is included. The output layer combines the radial basis function and weight parameter outputs to carry out classification or inference, and the network learns to assign the input to a center [17]. The Kohonen

SOM constitutes a type of ANN that relies on unsupervised learning to generate a reduced-dimensionality representation of the input data, while maintaining its topological properties. SOMs function by projecting high-dimensional data onto a grid of reduced dimensionality, generally taking the form of a two-dimensional lattice, which facilitates the exploration and grouping of intricate datasets through visual means. During the training process, neuron weights undergo adjustment as a function of the input data presented to the network. Neurons compete to best match the input, and the winning neuron, along with its neighbors, is updated to more closely resemble the input using the following formula given in (5).

$$w_i(t+1) = w_i(t) + \alpha(t)\eta_{ji}\big(x(t) - w_i(t)\big) \quad (5)$$

Modular neural networks (MNNs) are composed of several interconnected neural networks that work together to solve complex problems. Each module can specialize in different aspects of the task, enhancing overall performance. The use of MNNs is driven by the need to manage complexity in artificial intelligence problems. By dividing tasks among specialized networks, MNNs can improve efficiency and accuracy in learning and decision-making processes. MNNs can be categorized based on their architecture and learning algorithms, which helps in understanding the various approaches to building and training these networks [18].

Deep learning techniques can be classified into three principal categories: supervised, unsupervised, and semi-supervised learning. Moreover, another kind of learning strategy is a paradigm known as deep reinforcement learning, more commonly abbreviated as DRL or RL. It is generally accepted to belong to the class of partially supervised (and occasionally unsupervised) learning strategies [19].

2.1. Deep supervised learning

This method handles data that has been labeled. The environment has a set of inputs and outputs as a result of using such a technique (x_t, y_t). As an example, the intelligent agent determines if the input is x_t and returns the resulting loss value $L(\hat{y}_t, y_t)$. So as to derive a more precise approximation of the intended outputs, the agent then continually modifies the network parameters. Following a successful learning phase, the agent gains the capacity to find the appropriate responses to the questions posed by the environment. Some supervised learning methods for deep learning include DNNs, convolutional neural networks (CNNs), and RNNs. The latter category also encompassing long short-term memory (LSTM) methods and gated recurrent units (GRUs). This technique's principal strength lies in its capacity to gather data or produce data outputs based on historical data. This method's disadvantage is that the boundary separating decision regions can be surpassed whenever the training dataset is devoid of samples that fit into a given category. This approach for achieving high performance is generally simpler to utilize than other approaches.

2.2. Deep semi-supervised learning

This approach bases its learning process on semi-labeled datasets. This approach is sometimes used with generative adversarial networks (GANs) and DRL. Additionally, semi supervised learning also makes use of RNNs, which include LSTMs and GRUs. One benefit associated with this method lies in the fact that less labeled data is needed. However, a drawback of this approach resides in its potential to yield incorrect conclusions because of irrelevant input features in the training set. The text document classifier is one of the most popular uses of semi-supervised learning, as it is challenging to gather a large number of labeled text documents, making this task particularly well suited to the semi-supervised approach. When insufficient representative training data is available, the model may produce unreliable classifications.

2.3. Deep unsupervised learning

This method allows the process of learning to be implemented even in the lack of labeled data that is readily available (i.e., labels are not necessary). In order to determine the unknown structure or relationships in the input data, the agent must first learn the key characteristics or inner representation. Generative networks, dimensionality reduction, and clustering techniques are commonly included in the unsupervised learning category. Recent developments in the deep learning family include auto-encoders, GANs, and restricted Boltzmann machines (RBM), which have all shown excellent performance on non-linear dimensionality reduction and clustering problems. Moreover, unsupervised learning has made use of a range of RNN applications, including GRUs and LSTM methods. Unsupervised learning's processing complexity and incapacity to provide accurate information on data sorting are its main disadvantages. One of the most often used unsupervised learning techniques is clustering [20].

2.4. Deep reinforcement learning

Reinforcement learning involves interacting with the environment, while supervised learning relies on provided sample data. Introduced by Google DeepMind in 2013 [21], this approach led to improved

methods using reinforcement learning. Given environment samples $x_t \sim$ ρ, the agent predicts $\hat{y}_t = x(f_t)$, receives a cost $c_t \sim P(c_t|x_t, \hat{y}_t)$, P represents the unidentified probability distribution, and returns a noisy score. Known as semi-supervised learning, it inspired many supervised and unsupervised techniques [19]. Unlike supervised learning, reinforcement learning lacks a simple loss function, making it more challenging. Additionally, it differs fundamentally in two ways: the function is inaccessible and requires interaction to optimize, and the state depends on previous actions within the environment (the input (x_t) is determined by the activities taken before it) [22]. DNNs face two primary challenges: overfitting and premature convergence. When the weights and biases settle at a local optimum without reaching the global minima in the multidimensional space, premature convergence takes place. Overfitting happens when the network becomes overly tailored to the training dataset, making it unsuitable and inflexible for other test datasets.

This study have a variety of machine learning frameworks, libraries that have facilitated model training, and training styles, algorithms, and architectures. Without requiring you to develop them from scratch, these frameworks provide access to statistical models, training methods, and complicated mathematical calculations. A comparative overview of popular frameworks is presented in Table 1, which is introduced here to summarize their institutional origins, licensing models, and release timelines.

Table 1. Commonly used frameworks and libraries for deep learning

Framework	Institution	License	1st Release
Caffe	Berkeley Research	AI BSD / Free	2015
Microsoft	Microsoft	MIT License / Free	2016
Gluon	AWS and Microsoft	Open Source	2017
Keras	Individual Author	MIT License / Free	2015
MXNet	Apache Software Foundation	Apache 2.0 / Free	2015
TensorFlow	Google Brain	Apache 2.0 / Free	2015
Theano	University of Montreal	BSD / Free	2002
Torch	Ronan Collobert *et al*.	BSD / Free	2002
PyTorch	Facebook	BSD / Free	2016
Chainer	Preferred Networks	BSD / Free	2015
Deeplearning4j	Adam Gibson *et al*.	Apache 2.0 / Free	2014

3. DEEP NEURAL NETWORK ARCHITECTURES

Several well-known deep learning networks are covered in this section. These include CNN, recursive neural network (RvNN), RNN, and deep generative models. However, because of the rapid growth of deep learning, numerous new networks and designs are created every few months. Each architecture is designed to address specific learning tasks and computational challenges.

3.1. Recursive neural networks

Using compositional vectors, RvNN can classify the outputs and make predictions in a hierarchical structure. Recursive auto-associative memory (RAAM) served as the primary inspiration for the creation of RvNN. The RvNN architecture is designed to process items having randomly formed structures, like trees or graphs. This technique creates a fixed-width distributed representation from a variable-size recursive data structure. The network is trained using a newly developed back-propagation through structure (BTS) learning algorithm [23]. The BTS system employs the same technique as the general back-propagation algorithm and can support a tree-like structure. Through auto-association, the network is taught to produce the input-layer pattern at the output layer. When used in an NLP setting, RvNN is very effective. RvNN architecture was first presented by Socher *et al.* [24] and is intended to handle inputs from several modalities. The authors present two scenarios for the classification of phrases in natural language: one in which each sentence is divided into words and images of nature, and the other in which each image is divided into different interest segments. RvNN builds a syntactic tree and determines two scores that are most likely to merge. Moreover, RvNN calculates for every pair of units a score pertaining to the merging likelihood. A composition vector is created by combining the pair with the highest score. After each merging, RvNN produces three outputs: i) a larger area with multiple units; ii) a compositional vector of the area; and iii) a class label (for example, if two units are noun words, then the class label for the new area will be a noun phrase). A RvNN tree example can be seen in Figure 3 [19]. The compositional vector for the entire area is represented by the root of the RvNN tree structure.

3.2. Convolutional neural networks

A popular deep learning algorithm, CNN is known for its capacity to automatically recognize important components without the need for human interaction [25]. CNNs are used in a variety of fields, including face recognition, voice processing, and computer vision. Inspired by the visual cortex in a cat's brain [26], Because of shared parameters, sparse interactions, and similar representations, CNNs perform exceptionally well [27]. Unlike fully connected (FC) networks, CNNs utilize local connections and shared

weights, minimizing the training procedure and expediting operations by exploiting 2D input structures like images. CNNs typically consist of several convolution layers, pooling layers and FC layers, as illustrated in Figure 4 [19]. Each layer's input (x) in a CNN is represented in three dimensions: depth (channels), width, and height, or $m \times m \times r$. Convolutional layers use filters (kernels), denoted by k, These filters have three dimensions ($n \times n \times q$), just like the input picture. With q being smaller than or equal to r, n must be less than m. Kernels form the basis for local connections, convolving with inputs and sharing characteristics (weight w_k and bias b_k) to produce k feature maps h_k with a size of $(m - n - 1)$. The convolution layer computes a dot product between small input portions and weights as in (1), followed by applying an activation function or nonlinearity to the output." to establish local connections, generating feature maps by performing dot products between inputs and weights, followed by an activation function for output processing as given in (6).

$$h_k = f(w_k * x + b_k) \tag{6}$$

All feature maps in sub-sampling layers are down-sampled, reducing network parameters to speed up training and address overfitting. A pooling function (e.g., max or average) is applied to a neighboring area of size $p \times p$, where p is the kernel size. FC layers then use mid- and low-level data to create high-level abstraction, while the final layer, such as softmax or SVMs, generates classification scores, indicating the probability of each class.

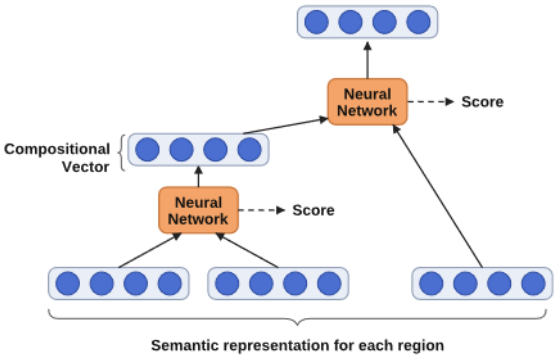

Figure 3. Example of RvNN tree structure

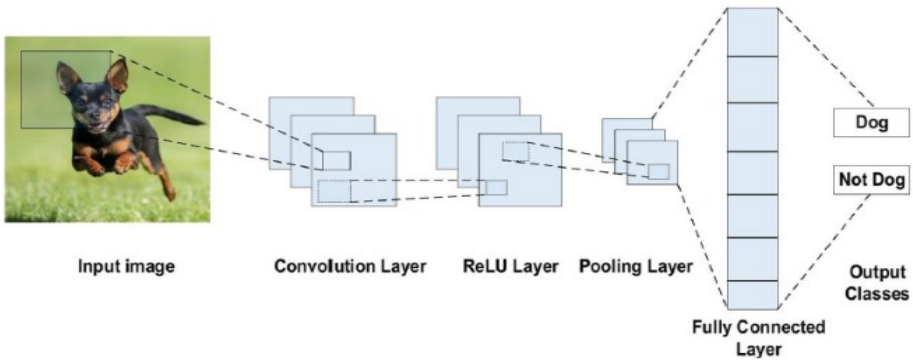

Figure 4. A CNN architecture sample used for image classification

3.3. Multilayer perceptron networks

An ANN that employs feed-forward is called a multilayer perceptron (MLP). This study use MLP networks or feedforward networks for supervised learning issues. An MLP's architecture is made up of at least three layers of nodes: an input layer, an output layer, and propagation for training, and one or more hidden layers that are used for computation in the dataset's hidden layers to capture complex connections. The network is trained using supervised learning via backpropagation. Every node has a nonlinear activation function and is a neuron, with the exception of the input nodes. The most basic kind of DNN, also called a "deep feedforward neural network" (DFN), is the multilayer perceptron, as shown in Figure 5 [28].

3.4. Recurrent neural networks

In the field of deep learning, RNNs are a well-known and often used algorithm [29]. NLP and speech processing parameters are the primary areas in which RNN is used. Unlike traditional networks, RNNs use sequential data in their networks. This characteristic is crucial to many different applications because the inherent structure in the data sequence provides useful information. To ascertain the meaning of a particular word in a sentence, for example, it is necessary to comprehend the statement's context. Thus, the RNN can be thought of as a short-term memory unit where s stands for the state (hidden) layer, y for the output layer, and x for the input layer. Three distinct kinds of deep RNN techniques "Hidden-to-hidden", "Hidden-to-output", and "Input-to-hidden" were presented by Pascanu *et al*. [30]. Based on these three methods, a deep RNN is presented that lessens the learning of the deep network challenge and offers the advantages of a deeper RNN. Nevertheless, one of the primary concerns with this strategy is that RNN is sensitive to the disappearing and exploding gradients [31]. More particular, the gradients may decay or burst exponentially throughout the training phase due to the reduplications of several big or small derivatives. Over time, this sensitivity lowers since the network stops considering the first inputs when new ones come in. Moreover, LSTM [32] can be used to solve this problem. Recurrent connections to network memory blocks are provided by this method. A number of memory cells, found in each memory block, are capable of storing the network's temporal states. It also has gated units to regulate the information flow. Residual connections can also significantly lessen the effect of the impact of the vanishing gradient issue in extremely deep networks [33].

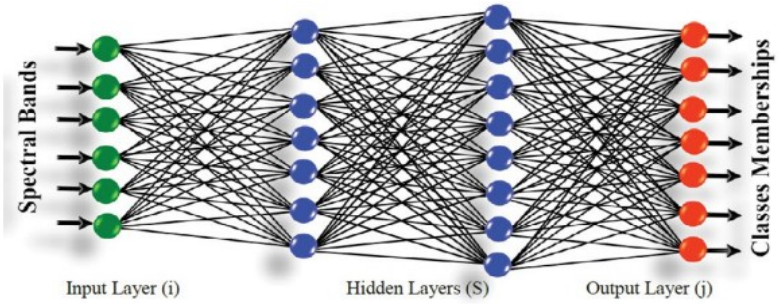

Figure 5. MLP networks architecture

3.5. Long short-term memory

In 1997, Hochreiter and Schmidhuber [32] introduced the LSTM system, which is an application of the RNN. In contrast to the feed forward network designs previously discussed, LSTM may be trained for tasks requiring memory or state awareness, and it can retain knowledge of previous states. Gradient vanishing is a key shortcoming of RNN, which is somewhat addressed by LSTM by permitting gradients to flow through unmodified. An LSTM is made up of memory cell state blocks that let signals pass through while being controlled by input, forget, and output gates. What is written, read, and saved on the cell is managed by these gates. Google, Apple, and Amazon all use LSTM in their voice recognition systems. Figure 6 [2] shows the cell, input, and output values as C, x, and h. To put it another way, t represents the current block values, whereas $t-1$ represents the preceding LSTM block (or time $t-1$). The time step value is indicated by the subscript t. $tanh$ represents the hyperbolic tangent function, while σ represents the sigmoid function. The operator + represents the summation of element-wise, while x represents the multiplication of element-wise. The (7) to (11) explain how the gates are computed [34].

$$f_t = \sigma\left(W_f x_t + w_f h_{t-1} + b_f\right) \tag{7}$$

$$i_t = \sigma(W_i x_t + w_i h_{t-1} + b_i) \tag{8}$$

$$o_t = \sigma(W_o x_t + w_o h_{t-1} + b_o) \tag{9}$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \sigma_c(W_c x_t + w_c h_{t-1} + b_c) \tag{10}$$

$$h_t = o_t \otimes \sigma_h(c_t) \tag{11}$$

Where the forget input and output gate vectors, are denoted by the letters f, i, and o, respectively. W, w, b, and $\otimes$ stand for input weights, recurrent output weights, bias and element-wise multiplication respectively.

GRUs are a more compact version of the LSTM. Because they lack an output gate, GRUs are smaller than LSTM and are only as effective on certain basic datasets. Long-term dependencies can be tracked using RNNs, such as LSTMs. As a result, they work well for developing models that depend on context and previous states and for learning from sequence incoming data. The LSTM cell block keeps important details from earlier states. The input, forget, and output gates control, respectively, what goes into the cell, what remains in the cell, and the cell values used to calculate the output of the LSTM block [32]. LSTM and GRU based autoencoders for automatic feature extractions were presented [35].

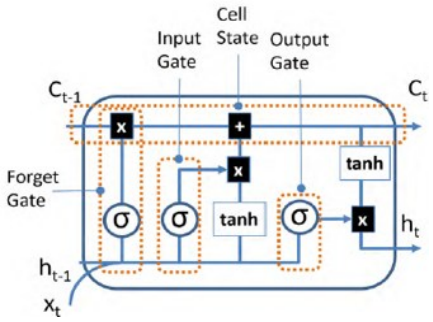

Figure 6. block of LSTM with gates and a memory cell

3.6. Autoencoder

The autoencoder algorithm, introduced by Rumelhart *et al*. [7] in 1986, is an unsupervised learning technique that uses backpropagation to set target values equal to inputs, enabling the extraction of hidden features by reconstructing input data. Its basic structure consists of three layers: the input layer (x), hidden layer (h), and output layer (y), where the input and output layers are of the same scale. The autoencoder maximizes similarity between input and output data by minimizing training error, represented as mean square error or cross-entropy. In this process, the input signal x is transformed into z in the hidden layer using (12).

$$z = g(Wx + b) \tag{12}$$

Where W represents weights, b is the bias, and $g(\cdot)$ is a nonlinear function like the sigmoid or ReLU. The decoded signal y, regarded as a prediction of x, is computed as given in (13).

$$y = f(W'z + b') \tag{13}$$

Where W' is typically the transpose of W, b' is the output layer's bias, and $f(\cdot)$ is another nonlinear function. Deep autoencoders, with multiple hidden layers, address challenges in training by treating each layer as a simple autoencoder [10]. They have been effectively used in a number of fields, including document encoding for faster retrieval, image retrieval, and generating effective speech features.

3.7. Comparison of deep neural networks

Feedforward neural networks, including CNNs, are trained using supervised learning and GD-based backpropagation for tasks like image recognition and classification, often utilizing datasets such as Modified National Institute of Standards and Technology (MNIST) and frameworks like TensorFlow, Caffe, and PyTorch. Residual networks also follow supervised training with backpropagation and are commonly applied to image recognition with datasets like ImageNet. Autoencoders, trained unsupervised with backpropagation, focus on dimensionality reduction and encoding using datasets like MNIST. Adversarial networks generate realistic fake data and improve images through unsupervised backpropagation, using datasets such as CIFAR10. RNNs, particularly LSTMs, handle NLP and language translation, utilizing GD and backpropagation through time with datasets like MNIST stroke sequence. Radial basis function networks can be trained in both supervised and unsupervised modes for function approximation and time series prediction, while Kohonen networks, arranged in hexagonal or rectangular grids, employ competitive learning for clustering and optimization.

4. TRAINING ALGORITHMS

Deep learning's primary component is the learning algorithm. DNNs differ from shallow ones in terms of the number of layers. It gets deeper the more levels there are in it. It is possible to specialize each layer to identify a particular element or trait. In order to solve a set of problems in a domain, the learning algorithm seeks to determine the ideal weight vector values.

4.1. Gradient descent

GD serves as the foundational concept in a significant number of machine learning and deep learning algorithms. This technique is grounded in the principles of Newton's algorithm for locating the roots or zeroes of a two-dimensional function. To achieve this, a point is arbitrarily chosen on the curve, and one moves leftward or rightward along the x-axis based on the function's derivative or slope at the selected point, either positively or negatively estimated. This process continues until the value of the y-axis, i.e., the function or f(x), reaches zero.

The same concept is employed in GD, where we navigate or descend along a specific path in a weight space with multiple dimensions. The study continue this process until the cost function continues to decline and stops when the error rate no longer decreases. If the function's derivative at the present position is zero, Newton's approach is prone to get trapped in local minima. GD, a technique used in the backpropagation method employs the chain rule and partial derivatives to determine the error delta for each weight in response to changes in its value [16]. The cost function is decreased by modifying the individual weights following each training data set learning iteration, which eventually yields a multi-dimensional weight landscape with numerous dimensions.

4.2. Stochastic gradient descent

The stochastic gradient descent (SGD) approach is used to train DNNs because the main disadvantage of the traditional GD approach is its lengthy training period [36]. GD involves going over each sample in the training dataset iteratively and then updating the weights. In contrast, updates are made in SGD after a minibatch of n samples is finished. The higher frequency of weight changes in SGD allows us to get closer to the global minimum faster.

4.3. Momentum

Momentum is a technique that aids in speeding the SGD approach training process. Using the gradient's moving average rather than just its current true value is the basic concept behind it. Mathematically, this study can represent this with the (14) and (15).

$$v_t = \gamma v_{t-1} - \eta \nabla \mathcal{F}(\theta_{t-1}), \tag{14}$$

$$\theta_t = \theta_{t-1} + v_t\,, \tag{15}$$

γ represents momentum and η represents the learning rate for the tth training cycle. Using momentum during training has the primary benefit of keeping the network from becoming trapped in a local minimum. The momentum values are $\gamma \in (0, 1]$. A larger momentum value is shown to overshoot its minimum, potentially leading to network instability. is often set to 0.5 until the initial learning becomes stable, at which point it is raised to 0.9 or higher [37].

4.4. Backpropagation through time

The well-known back-propagation technique is used to train DNN using SGD [38]. Using directed acyclic graphs, or computation graphs, this study can simply represent neural networks models in the case of MLPs. This study can quickly compute the gradient between the top and bottom layers with back-propagation for that deep learning representation using the chain rule [7], [22].

5. DEEP LEARNING APPLICATION

Deep learning is being applied in various fields such as NLP (e.g., sentence classification, and translation), computer vision, multimedia data analysis, visual data processing, speech and audio processing (e.g., enhancement and recognition), social network analysis, and healthcare. Table 2 presents a detailed overview of the methods utilized for each application area, categorized by network architecture. Table 3 provides a collection of recent reviews in several deep learning domains.

Table 2. Articles distribution according to application areas

Networks	Application area	Authors
RvNNs	Data-driven forecasting of chaotic dynamical systems and complex system modeling	Vlachas *et al*. [39], 2018
	Speech recognition, and solving complex real-world problems that involve structured input sequences.	Zelios *et al*. [40], 2022
	High-speed plate collision emulation,	Pogorelko *et al*. [41], 2024
CNNs	Face recognition	Li *et al*. [42], 2020
	Medical data analysis for tasks like diagnosis, prognosis, and disease classification	Dai *et al*. [43], 2024
	Object recognition, image processing, computer vision, and face recognition	Taye [44], 2023
	X-Ray automated identification and classification of COVID-19	Thakur and Kumar [45], 2021
MLP	Applications for MLP are found in many different industries, particularly in biology.	Zhu [46], 2023
networks	Prediction of financial distress	Wu *et al*. [47], 2022
	Application in hydrological forecasting,	Valles [48], 2024
	Application in predicting solar irradiance patterns at various time intervals,	Obiora *et al*. [49], 2023
	Deepfake detection	Kolagati *et al*. [50], 2022
RNN	Prediction of the COVID-19 mortality risk change	Villegas *et al*. [51], 2023
	Methodology for predicting crop rotation	Dupuis *et al*. [52], 2023
	Pore-water pressure prediction using time series	Wei *et al*. [53], 2021
	RNN is applied in seizure classification of electroencephalogram (EEG) signals,	Dutta *et al*. [54], 2022
	Text sentiment analysis	Onan [55], 2022
LSTM	Positive COVID-19 forecasting	Sunjaya *et al*. [56], 2023
	Robot cell predictive maintenance application	Joseph *et al*. [57], 2022
	Application in climate prediction for temperature, precipitation, and humidity forecasting	Chu *et al*. [58], 2023
	Application in time series forecasting	Yadav *et al*. [59], 2023
	Groundwater quality forecasting for irrigation	Gorgij *et al*. [60], 2023
	Application in automatic speech recognition (ASR) tasks	Oruh *et al*. [61], 2022
Autoencoder	Defect diagnosis in industrial processes	Qian *et al*. [62], 2022
	Biomedical applications, including diagnosis of rare diseases.	Pratella *et al*. [63], 2021
	Stock prediction	Wu *et al*. [64], 2022
	Underwater target recognition	Khishe [65], 2022
	Self-supervised medical image representations	Prabhakar *et al*. [66], 2023

Table 3. Collection of recent papers on deep learning and their application fields

Topic	Methodology	Specified objects	Papers
Computer vision and image classification	LSTM and RNN	Text and image classification	[67]
	CNN, RNN, LSTM, and auto-encoders	Biometric recognition (facial, iris, fingerprint, and palmprint recognition)	[68]
	CNN	Application in computer vision for image classification tasks.	[44]
	You only look once (YOLO), spatial pyramid pooling network (SPPNet), and fast RCNN	Vehicle detection, classification, and other intelligent transportation systems applications.	[69]
Forecasting	CNN and RNN	Deep learning for time-series analysis	[70]
	Hybrid model: GRU and one-dimensional convolutional neural network (1DCNN)	development of a hybrid model, and improve forecasting accuracy	[71]
	MLP, LSTM, GRU, autoencoder - bidirectional gated recurrent unit (BiGRU) - Swish, autoregressive integrated moving average (ARIMA)	forecasting financial series, specifically exchange rates	[72]
Medical applications	CNN models (residual network (ResNet)18, ResNet50, Vgg16, ResNet101, VGG19) and SVM	Images from an X-ray to identify COVID-19	[73]
	CNN, GAN	Medical imaging for tasks like image segmentation, classification	[74]
	CNN, RNN, and GAN	Arrhythmia detection, cardiac signals analysis	[75]
Adversarial situations	Defensive strategies	evaluation of adversarial attacks and defenses in machine learning systems.	[76]
	Defense method Attack modes: FGSM, DeepFool	Protecting neural networks from error-inducing disturbances in image data	[77]
	Gaussian data augmentation (GDA) Spatial smoothing (SS)	Enhancing model robustness against attacks by incorporating feature-denoising blocks	[78]
	BERT, bidirectional LSTM, k-nearest neighbors (KNN), and naive Bayes	Email spam detection	[79]
	BERT, LSTM	Multilingual sentiment analysis	[80]
Big data analytics	CNNs for visual data, RNNs for sequential data	The creation of an adaptive learning environment	[81]
	CNN, RNN, and ANN algorithms	Feature extraction and classification, to enhance reliability in analyzing massive datasets.	[82]
	Deep learning techniques in big data analytics	Application in various big data analytics fields including IoT data application	[83]

6. CONCLUSION

Deep learning is a key area of machine learning research that marks the beginning of artificial intelligence research. This study have offered a thorough review of neural networks and deep learning and their recent applications. The history of ANNs is presented first, followed by a broad description of neural networks. This study have outlined the different classifications of deep learning, as well as the frameworks in this area, and explored the popular training algorithms and architectures. Various supervised deep learning methods such as RNN, CNN, and DNN have been discussed, along with unsupervised methods such as autoencoder and RBM. The goal of this study is to highlight the importance of deep learning research throughout the last few decades, as numerous new fields that potentially benefit from deep learning have emerged during this time. This article aims to highlight the most important deep learning applications that have experienced significant adoption recently and are anticipated to be heavily featured in upcoming studies. Readers may use these examples as a foundation for further investigation. Still, there are several challenges with deep learning that need to be resolved. Simultaneous modeling of many intricate data modalities is still a problem for deep learning, with multimodal deep learning being a popular path in current research. In contrast to human brains, deep learning requires large datasets for training, ideally labeled data, in order to predict unseen data. When limited datasets are available, such as healthcare data, or when real-time data processing is required, the challenge becomes more difficult. Furthermore, in order to deal with actual data without manual human labels, machine learning is increasingly moving toward unsupervised and semi-supervised learning, whereas the majority of deep learning methods currently in use are supervised algorithms.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Hamza Rabiai	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓
Sanae Khali Issa	✓	✓		✓						✓	✓	✓	✓	
Kaouthar El Fassi	✓	✓								✓	✓	✓	✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

INFORMED CONSENT

This study does not involve human participants or personal data. Therefore, informed consent is not required in this study.

DATA AVAILABILITY

Data sharing is not applicable to this article as no new data were created or analyzed in this study. All data discussed are available in the cited references.

REFERENCES

[1] L. Deng, "A tutorial survey of architectures, algorithms, and applications for deep learning," *APSIPA Transactions on Signal and Information Processing*, vol. 3, no. 1, 2014, doi: 10.1017/atsip.2013.9.
[2] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures," *IEEE Access*, vol. 7, pp. 53040–53065, 2019, doi: 10.1109/ACCESS.2019.2912200.
[3] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, Dec. 1943, doi: 10.1007/BF02478259.

[4] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain.," *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958, doi: 10.1037/h0042519.

[5] M. Minsky and S. A. Papert, *An introduction to computational geometry*, Cambridge, United Kingdom: The Science Press, vol. 479, no. 480, 1969.

[6] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991, doi: 10.1016/0893-6080(91)90009-T.

[7] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986, doi: 10.1038/323533a0.

[8] K. Fukushima, "Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, Apr. 1980, doi: 10.1007/BF00344251.

[9] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, doi: 10.1109/5.726791.

[10] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006, doi: 10.1162/neco.2006.18.7.1527.

[11] M. I. Jordan and T. M. Mitchell, "Machine learning: trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015, doi: 10.1126/science.aaa8415.

[12] S. Sengupta *et al.*, "A review of deep learning with special emphasis on architectures, applications and recent trends," *Knowledge-Based Systems*, vol. 194, 2020, doi: 10.1016/j.knosys.2020.105596.

[13] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems*, vol. 2, no. 4, pp. 303–314, Dec. 1989, doi: 10.1007/BF02551274.

[14] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks*, vol. 61, pp. 85–117, Jan. 2015, doi: 10.1016/j.neunet.2014.09.003.

[15] N. Sharma, R. Sharma, and N. Jindal, "Machine learning and deep learning applications-a vision," *Global Transitions Proceedings*, vol. 2, no. 1, pp. 24–28, Jun. 2021, doi: 10.1016/j.gltp.2021.01.004.

[16] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.

[17] M. D. Buhmann, *Radial basis functions: theory and implementations*, 1st ed. Cambridge, United Kingdom: Cambridge University Press, 2003, doi: 10.1017/CBO9780511543241.

[18] K. Chen, "Deep and modular neural networks," in *Springer Handbook of Computational Intelligence*, Berlin, Germany: Springer, 2015, pp. 473–494, doi: 10.1007/978-3-662-43505-2_28.

[19] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, Mar. 2021, doi: 10.1186/s40537-021-00444-8.

[20] M. M. Saeed, Z. A. Aghbari, and M. Alsharidah, "Big data clustering techniques based on Spark: a literature review," *PeerJ Computer Science*, vol. 6, Nov. 2020, doi: 10.7717/peerj-cs.321.

[21] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015, doi: 10.1038/nature14236.

[22] M. Z. Alom *et al.*, "A state-of-the-art survey on deep learning theory and architectures," *Electronics*, vol. 8, no. 3, Mar. 2019, doi: 10.3390/electronics8030292.

[23] C. Goller and A. Kuchler, "Learning task-dependent distributed representations by backpropagation through structure," in *International Conference on Neural Networks (ICNN'96)*, 1996, pp. 347–352, doi: 10.1109/ICNN.1996.548916.

[24] R. Socher, C. C. Lin, A. Y. Ng, and C. D. Manning, "Parsing natural scenes and natural language with recursive neural networks," in *28th international conference on machine learning*, 2011, pp. 129–136.

[25] J. Gu *et al.*, "Recent advances in convolutional neural networks," *Pattern recognition*, vol. 77, pp. 354–377, 2018, doi: 10.1016/j.patcog.2017.10.013.

[26] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of Physiology*, vol. 160, no. 1, pp. 106–154, Jan. 1962, doi: 10.1113/jphysiol.1962.sp006837.

[27] I. Goodfellow, Y. Bengio, and A. Courville, "Convolutional networks," in *Deep Learning*, Cambridge, United Kingdom: MIT Press, 2016, pp. 326–366.

[28] S. Ahmed, "Studying the effect of activation function on classification accuracy using deep artificial neural networks," *Journal of Remote Sensing & GIS*, vol. 6, no. 3, 2017, doi: 10.4172/2469-4134.1000203.

[29] H. Hewamalage, C. Bergmeir, and K. Bandara, "Recurrent neural networks for time series forecasting: current status and future directions," *International Journal of Forecasting*, vol. 37, no. 1, pp. 388–427, Jan. 2021, doi: 10.1016/j.ijforecast.2020.06.008.

[30] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," 2014, *arXiv: 1312.6026*.

[31] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Journal of Machine Learning Research*, pp. 249–256, 2010.

[32] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[34] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: continual prediction with LSTM," *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000, doi: 10.1162/089976600300015015.

[35] B. Naul, J. S. Bloom, F. Pérez, and S. V. D. Walt, "A recurrent neural network for classification of unevenly sampled variable stars," *Nature Astronomy*, vol. 2, no. 2, pp. 151–155, Nov. 2017, doi: 10.1038/s41550-017-0321-z.

[36] L. Bottou, "Stochastic gradient descent tricks," in *Neural Networks: Tricks of the Trade*, Berlin, Germany: Springer Berlin Heidelberg, 2012, pp. 421–436, doi: 10.1007/978-3-642-35289-8_25.

[37] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *30th International Conference on Machine Learning (ICML 2013)*, 2013, pp. 2176–2184.

[38] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for boltzmann machines*," *Cognitive Science*, vol. 9, no. 1, pp. 147–169, Jan. 1985, doi: 10.1207/s15516709cog0901_7.

[39] P. R. Vlachas, W. Byeon, Z. Y. Wan, T. P. Sapsis, and P. Koumoutsakos, "Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 474, no. 2213, May 2018, doi: 10.1098/rspa.2017.0844.

[40] A. Zelios, A. Grammenos, M. Papatsimouli, N. Asimopoulos, and G. Fragulis, "Recursive neural networks: recent results and applications," *SHS Web of Conferences*, vol. 139, 2022, doi: 10.1051/shsconf/202213903007.

[41] V. V. Pogorelko, A. E. Mayer, and E. V. Fedorov, "Recursive neural network as a high-speed plate collision emulator," *Chelyabinsk Physical and Mathematical Journal*, vol. 9, no. 1, pp. 134–143, 2024, doi: 10.47475/2500-0101-2024-9-1-134-143.
[42] H.-C. Li, Z.-Y. Deng, and H.-H. Chiang, "Lightweight and resource-constrained learning network for face recognition with performance optimization," *Sensors*, vol. 20, no. 21, Oct. 2020, doi: 10.3390/s20216114.
[43] L. Dai, M. Zhou, and H. Liu, "Recent applications of convolutional neural networks in medical data analysis," in *Federated Learning and AI for Healthcare 5.0*, New York, United States: IGI Global Scientific Publishing, 2024, pp. 119–131.
[44] M. M. Taye, "Theoretical understanding of convolutional neural network: concepts, architectures, applications, future directions," *Computation*, vol. 11, no. 3, Mar. 2023, doi: 10.3390/computation11030052.
[45] S. Thakur and A. Kumar, "X-ray and CT-scan-based automated detection and classification of covid-19 using convolutional neural networks (CNN)," *Biomedical Signal Processing and Control*, vol. 69, Aug. 2021, doi: 10.1016/j.bspc.2021.102920.
[46] H. Zhu, "Multi-layered perceptron and its applications in biotechnology," *Theoretical and Natural Science*, vol. 20, no. 1, pp. 159–165, Dec. 2023, doi: 10.54254/2753-8818/20/20230753.
[47] D. Wu, X. Ma, and D. L. Olson, "Financial distress prediction using integrated Z-score and multilayer perceptron neural networks," *Decision Support Systems*, vol. 159, Aug. 2022, doi: 10.1016/j.dss.2022.113814.
[48] J. Valles, "Application of a multilayer perceptron artificial neural network (MLP-ANN) in hydrological forecasting in El Salvador," in *Advanced Hydroinformatics: Machine Learning and Optimization for Water Resources*, 1st ed., Hoboken, United States: Wiley, 2024, pp. 213–239, doi: 10.1002/9781119639268.ch8.
[49] C. N. Obiora, A. Ali, and A. N. Hasan, "Using the multilayer perceptron (MLP) model in predicting the patterns of solar irradiance at several time intervals," in *2023 31st Southern African Universities Power Engineering Conference (SAUPEC)*, Jan. 2023, pp. 1–6, doi: 10.1109/SAUPEC57889.2023.10057744.
[50] S. Kolagati, T. Priyadharshini, and V. M. A. Rajam, "Exposing deepfakes using a deep multilayer perceptron – convolutional neural network model," *International Journal of Information Management Data Insights*, vol. 2, no. 1, Apr. 2022, doi: 10.1016/j.jjimei.2021.100054.
[51] M. Villegas *et al.*, "Predicting the evolution of COVID-19 mortality risk: a recurrent neural network approach," *Computer Methods and Programs in Biomedicine Update*, vol. 3, 2023, doi: 10.1016/j.cmpbup.2022.100089.
[52] A. Dupuis, C. Dadouchi, and B. Agard, "Methodology for multi-temporal prediction of crop rotations using recurrent neural networks," *Smart Agricultural Technology*, vol. 4, 2023, doi: 10.1016/j.atech.2022.100152.
[53] X. Wei, L. Zhang, H.-Q. Yang, L. Zhang, and Y.-P. Yao, "Machine learning for pore-water pressure time-series prediction: Application of recurrent neural networks," *Geoscience Frontiers*, vol. 12, no. 1, pp. 453–467, 2021, doi: 10.1016/j.gsf.2020.04.011.
[54] K. K. Dutta, P. Sridharan, and S. A. S. Bellary, "Recurrent neural networks and their application in seizure classification," in *Deep Learning in Visual Computing and Signal Processing*, 1st ed., Boca Raton, United States: Apple Academic Press, 2022, pp. 165–203, doi: 10.1201/9781003277224-8.
[55] A. Onan, "Bidirectional convolutional recurrent neural network architecture with group-wise enhancement mechanism for text sentiment classification," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 5, pp. 2098–2117, May 2022, doi: 10.1016/j.jksuci.2022.02.025.
[56] B. A. Sunjaya, S. D. Permai, and A. A. S. Gunawan, "Forecasting of covid-19 positive cases in Indonesia using long short-term memory (LSTM)," *Procedia Computer Science*, vol. 216, pp. 177–185, 2023, doi: 10.1016/j.procs.2022.12.125.
[57] D. Joseph, T. Gallege, E. T. Bekar, C. Dudas, and A. Skoogh, "A predictive maintenance application for a robot cell using LSTM model," *IFAC-PapersOnLine*, vol. 55, no. 19, pp. 115–120, 2022, doi: 10.1016/j.ifacol.2022.09.193.
[58] H. Chu, J. Wu, W. Wu, and J. Wei, "A dynamic classification-based long short-term memory network model for daily streamflow forecasting in different climate regions," *Ecological Indicators*, vol. 148, 2023, doi: 10.1016/j.ecolind.2023.110092.
[59] D. Yadav, L. Sahoo, S. K. Mandal, G. Ravivarman, P. Vijayaraghavan, and B. Prasad, "Using long short-term memory units for time series forecasting," in *2023 2nd International Conference on Futuristic Technologies (INCOFT)*, IEEE, Nov. 2023, pp. 1–6, doi: 10.1109/INCOFT60753.2023.10425756.
[60] A. D. Gorgij, G. Askari, A. A. Taghipour, M. Jami, and M. Mirfardi, "Spatiotemporal forecasting of the groundwater quality for irrigation purposes, using deep learning method: long short-term memory (LSTM)," *Agricultural Water Management*, vol. 277, Mar. 2023, doi: 10.1016/j.agwat.2022.108088.
[61] J. Oruh, S. Viriri, and A. Adegun, "Long short-term memory recurrent neural network for automatic speech recognition," *IEEE Access*, vol. 10, pp. 30069–30079, 2022, doi: 10.1109/ACCESS.2022.3159339.
[62] J. Qian, Z. Song, Y. Yao, Z. Zhu, and X. Zhang, "A review on autoencoder based representation learning for fault detection and diagnosis in industrial processes," *Chemometrics and Intelligent Laboratory Systems*, vol. 231, 2022, doi: 10.1016/j.chemolab.2022.104711.
[63] D. Pratella, S. A. Saadi, S. Bannwarth, V. P.-Fluckinger, and S. Bottini, "A survey of autoencoder algorithms to pave the diagnosis of rare diseases," *International Journal of Molecular Sciences*, vol. 22, no. 19, 2021, doi: 10.3390/ijms221910891.
[64] D. Wu, X. Wang, and S. Wu, "A hybrid framework based on extreme learning machine, discrete wavelet transform, and autoencoder with feature penalty for stock prediction," *Expert Systems with Applications*, vol. 207, Nov. 2022, doi: 10.1016/j.eswa.2022.118006.
[65] M. Khishe, "DRW-AE: a deep recurrent-wavelet autoencoder for underwater target recognition," *IEEE Journal of Oceanic Engineering*, vol. 47, no. 4, pp. 1083–1098, Oct. 2022, doi: 10.1109/JOE.2022.3180764.
[66] C. Prabhakar, H. Li, J. Yang, S. Shit, B. Wiestler, and B. Menze, "ViT-AE++: improving vision transformer autoencoder for self-supervised medical image representations," in *Proceedings of Machine Learning Research,* 2024, pp. 666–679.
[67] A. S. Gaafar, J. M. Dahr, and A. K. Hamoud, "Comparative analysis of performance of deep learning classification approach based on LSTM-RNN for textual and image datasets," *Informatica*, vol. 46, no. 5, pp. 21–28, 2022, doi: 10.31449/inf.v46i5.3872.
[68] S. Minaee, A. Abdolrashidi, H. Su, M. Bennamoun, and D. Zhang, "Biometrics recognition using deep learning: a survey," *Artificial Intelligence Review*, vol. 56, no. 8, pp. 8647–8695, Aug. 2023, doi: 10.1007/s10462-022-10237-x.
[69] M. A. Berwo *et al.*, "Deep learning techniques for vehicle detection and classification from images/videos: a survey," *Sensors*, vol. 23, no. 10, May 2023, doi: 10.3390/s23104832.
[70] B. Lim and S. Zohren, "Time-series forecasting with deep learning: a survey," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 379, no. 2194, Apr. 2021, doi: 10.1098/rsta.2020.0209.
[71] A. E. Zaar, A. Mansouri, N. Benaya, A. E. Allati, and T. Bakir, "A contribution to time series analysis and forecasting using deep learning approaches," in *2024 International Conference on Control, Automation and Diagnosis (ICCAD)*, IEEE, May 2024, pp. 1–6, doi: 10.1109/ICCAD60883.2024.10553976.
[72] H. Makika, J. M. T. Romano, and R. Ballini, "Deep learning techniques for time series forecasting: an application for exchange rates," in *2023 IEEE 33rd International Workshop on Machine Learning for Signal Processing (MLSP)*, IEEE, 2023, pp. 1–6, doi: 10.1109/MLSP55844.2023.10285930.
[73] N. H. Ajam and Z. S. Jumaa, "Convolutional neural network in the classification of COVID-19," *Periodicals of Engineering and Natural Sciences*, vol. 10, no. 2, pp. 241–250, Apr. 2022, doi: 10.21533/pen.v10.i2.602.

[74] Y. Han, "Deep learning methods and corresponding applications in medical imaging," *Applied and Computational Engineering*, vol. 46, pp. 79–83, Mar. 2024, doi: 10.54254/2755-2721/46/20241106.
[75] P. Das, D. Vetrithangam, G. H. Krishnan, A. Kumar, and R. V. K. Reddy, "Deep learning techniques revolutionizing biomedical applications: arrhythmia detection, cardiac sensed signals, and cell-free synthetic biology," in *Applications of Synthetic Biology in Health, Energy, and Environment,* New York, United States: IGI Global, 2023, pp. 68–91, doi: 10.4018/978-1-6684-6577-6.ch004.
[76] S. Y. Khamaiseh, D. Bagagem, A. Al-Alaj, M. Mancino, and H. W. Alomari, "Adversarial deep learning: a survey on adversarial attacks and defense mechanisms on image classification," *IEEE Access*, vol. 10, pp. 102266–102291, 2022, doi: 10.1109/ACCESS.2022.3208131.
[77] Z. Wang and Y. Wang, "A neural network model for adversarial defense based on deep learning," in *Third International Conference on Image Processing and Intelligent Control (IPIC 2023)*, Aug. 2023, doi: 10.1117/12.3000789.
[78] D. Vyas and V. V. Kapadia, "Designing defensive techniques to handle adversarial attack on deep learning based model," *PeerJ Computer Science*, vol. 10, Mar. 2024, doi: 10.7717/peerj-cs.1868.
[79] I. AbdulNabi and Q. Yaseen, "Spam email detection using deep learning techniques," *Procedia Computer Science*, vol. 184, pp. 853–858, 2021, doi: 10.1016/j.procs.2021.03.107.
[80] J. Liu *et al.*, "Application of deep learning-based natural language processing in multilingual sentiment analysis," *Mediterranean Journal of Basic and Applied Sciences*, vol. 08, no. 02, pp. 243–260, 2024, doi: 10.46382/MJBAS.2024.8219.
[81] R. Yu, T. Yao, and F. Bai, "Educational big data analytics for futuristic smart learning using deep learning techniques," *Scalable Computing: Practice and Experience*, vol. 25, no. 4, pp. 2728–2735, Jun. 2024, doi: 10.12694/scpe.v25i4.2988.
[82] A. K. Sharma and S. Gupta, "Reduce latency and transmission overhead for massive big data using convolutional neural networks," in *2023 International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*, Ballar, India: IEEE, Apr. 2023, pp. 1–7, doi: 10.1109/ICDCECE57866.2023.10151350.
[83] H. A. Selmy, H. K. Mohamed, and W. Medhat, "Big data analytics deep learning techniques and applications: a survey," *Information Systems*, vol. 120, Feb. 2024, doi: 10.1016/j.is.2023.102318.

BIOGRAPHIES OF AUTHORS



Hamza Rabiai holds a bachelor in Mathematics and application at Faculty of Science and Technology in Errachidia, master in Applied Analysis and Statistical Engineering. He is currently a Ph.D. student in Statistics and Artificial Intelligence at the Department of Mathematics in the Faculty of Science and Technology, Abdelmalek Essaadi University, Tangier, Morocco. His research areas of interest include artificial intelligence, deep learning and sentiment analysis. His research focuses on current issues in artificial intelligence, namely, how emotions might be extracted from unstructured data. She investigates ways to use neural network designs to collect and model pertinent emotional information by utilizing advanced deep learning techniques. He can be contacted at email: hamza.rabiai@etu.uae.ac.ma.



Sanae Khali Issa is a research professor at Abdelmalek Essaadi University since April 2016, attached to the Department of Computer Engineering, Faculty of Science and Technology in Tangier. She obtained her Ph.D. in Business Intelligence in June 2015. Her thesis was on risk management, with a focus on modeling a decision support system for managing fire risks in buildings. After completing her doctorate, she joined the "networks, systems, and artificial intelligence" research team at the Laboratory of Innovative Technologies (LTI) at the National School of Applied Sciences in Tangier, Abdelmalek Essaadi University to continue her research in the field of artificial intelligence and machine learning. In July 2024, she obtained her university qualification to direct research work. She has been working on a fascinating research project on the use of artificial intelligence tools to improve the quality of life of human beings in various fields: education, health, and security. She can be contacted at email: skhaliissa@uae.ac.ma.



Dr. Kaouthar El Fassi is a researcher in applied mathematics, renowned for her contributions to non-parametric functional estimation and survival analysis. Since September 2010, she has been a professor at the Faculty of Science and Technologies, Abdelmalek Essaadi University. Holding a Ph.D. from Paris VI University, her early work focused on improving statistical smoothing methods, particularly local polynomial fitting, to reduce bias and enhance estimation accuracy. These techniques have found significant applications in health data modeling and quality of life assessments. Recently, her research has shifted toward modern challenges in artificial intelligence, particularly the analysis of emotions from unstructured data. By applying advanced deep learning techniques, she explores how to extract and model relevant emotional information using neural network architectures. In parallel, she continues to collaborate with international experts on projects related to survival analysis, applying non-parametric models to address complex problems in reliability and risk modeling. Her ability to integrate traditional statistical approaches with innovations in artificial intelligence highlights the diversity and modernity of her scientific contributions. She can be contacted at email: k.elfassi@uae.ac.ma.