Annals of Emerging Technologies in Computing (AETiC)
Vol. 10, No. 1, 2026

Review Article

Review and Utilisation of AI in Signal Processing

Johan Note[1], Maaruf Ali[1,2,*] and Lekë Pepkolaj[1]

[1]Universiteti Metropolitan Tirana (UMT), Albania
jnote@umt.edu.al; maaruf.ali@umt.edu.al; lpepkolaj@umt.edu.al
[2]University of Wales Trinity Saint David (UWTSD), Wales, UK
Muhammad-Maaruf.Ali@uwtsd.ac.uk
*Correspondence: maaruf.ali@umt.edu.al

Received: 14 November 2025; Accepted: 22 December 2025; Published: 1 January 2026

Abstract: Numerous challenges persist within Signal Processing (SP) in communication systems, i.e. maintaining system stability, complexity of multi-dimensional SP and filtering. Many resources are required to cope with these growing problems. The application of AI offers promising solutions. AI is already being applied to solve engineering, medical and scientific problems, including utilisation in SP and Digital Signal Processing (DSP). The research methodology began with common supervised AI algorithms in SP: Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNNs). Regular unsupervised AI learning algorithms investigated included: Sparse Auto-encoder, Deep Auto-encoder, Convolutional Auto-encoder and the De-noising Auto-encoder. Their benefits were ascertained for their utilisation in an intelligent, scalable and secure communication system. The paper also provides over 100 references. The conclusion is that using specifically identified AI algorithms will make communication systems more scalable, thus mitigating overloading, instability and premature breaking point failure. AI use will also eventually make communication systems less expensive, more agile and far more intelligent - essential for Cognitive Radio and SDR (Software Defined Radio). The communication channels will become more reliable with AI SP and aid in its increased robustness in noise, quality and error recovery, compared with using traditional SP. These benefits will cause a shift toward using new methods in SP such as the Advantage Actor Critic and Trust Region Policy Gradient Algorithms. Two different datasets comprising the MNIST (Modified National Institute of Standards and Technology) database consisting of 70,000 28×28 pixels greyscale images of handwritten integers and the CIFAR-10 (Canadian Institute for Advanced Research) database containing 60,000 32×32 pixels small colour images were utilised. These were then processed using the aforementioned algorithms to test their accuracy. Because of 6G radio channel data scarcity for training purposes, image datasets were used to check the feasibility of the effectiveness of AI in SP in general radio communications, due to their two-dimensional nature. The result showed that all seven AI models performed well: 80% - 99.77% (accuracies). The CCN-Autoencoder gave the best result overall using both datasets: 94.5%-99.77% (accuracies). Future work will use real radio communication channel data and other AI algorithms.

Keywords: *AI; Artificial Neural Networks; Convolutional Neural Networks; Denoising Auto-encoder; Signal Processing; Sparse Auto-encoder*

1. Introduction

Currently, with the preponderance of technological devices, often having to work together whilst sharing the overcrowded Electromagnetic Spectrum (EM), Signal Processing (SP) faces many difficulties in trying to make them all work under extremely noisy conditions. Table 1, below, lists the top SP challenges faced in digital and analogue signal processing in electrical, electronic and computer engineering in descending order. Tables 1-3 were composed based on querying the AI agent "poe.com".

Table 2 lists in descending order, the current challenges in SP in modern communication systems, with the prevailing solutions or strategies to mitigate them.

Johan Note, Maaruf Ali and Lekë Pepkolaj, "Review and Utilisation of Artificial Intelligence in Signal Processing", *Annals of Emerging Technologies in Computing (AETiC)*, Print ISSN: 2516-0281, Online ISSN: 2516-029X, pp. 58-82, Vol. 10, No. 1, 1 January 2026, Published by International Association for Educators and Researchers (IAER), DOI: 10.33166/AETiC.2026.01.004, Available: https://aetic.theiaer.org/archive/v10/v10n1/p4.html.

Table 1. Current Challenges in Signal Processing[1]

Order №	Analogue and DSP Challenges	Nature of Problem
1st	Computational Complexity [1-2]	Advanced algorithms can be resource-intensive, impacting processing time and power consumption.
2nd	Real-time Processing [3]	Many applications require instantaneous responses, making it difficult to achieve real-time performance with complex algorithms.
3rd	Data Volume Management [4]	The rapid increase in data from IoT (Internet-of-Thing) devices and sensors necessitates efficient handling and processing strategies.
4th	Noise and Distortion Management [5]	Signals are often affected by noise and distortion, which can significantly degrade performance.
5th	Signal Integrity [6]	Maintaining signal quality over long distances is challenging due to attenuation and interference.
6th	Integration with Digital Systems [7]	Bridging analogue and digital systems can lead to compatibility issues, particularly in ADCs (analogue-to-digital converters) and DACs (digital-to-analogue converters).
7th	Power Consumption [8]	High power usage in analogue circuits, especially in high-frequency applications, is a major concern.
8th	Component Variability [9]	Variations in manufacturing processes can affect the performance of analogue components, necessitating calibration.
9th	Machine Learning (ML) Integration [10]	Incorporating Machine Learning (ML) into signal processing workflows is complex and can demand significant resources.
10th	Environmental Sensitivity [11]	Analogue systems can be affected by environmental changes, impacting reliability and performance – necessitating the use of self-testing and self-calibrating circuits [11].
11th	Interference and Jitter [12]	Both analogue and digital systems face challenges from interference, which can compromise signal quality.
12th	Security Concerns [13]	Ensuring the security and integrity of data in Signal Processing (SP) systems is increasingly critical.
13th	Integration of and with Emerging Technologies [14]	Adapting to new technologies (e.g. 5G, 6G, AI) requires significant adjustments and can complicate existing frameworks.

Table 2. Current Challenges and Solutions in Signal Processing in Modern Communication Systems[2]

Order №	SP Comms Challenge Type	Challenge	Solution
1st	Interference and Noise Management	Increasing interference from various sources can degrade the signal quality.	▪ Implement adaptive filtering techniques to dynamically suppress interference [15]. ▪ Utilise advanced modulation schemes that are more resilient to noise, such as OFDM (Orthogonal Frequency Division Multiplexing) [16].
2nd	Spectral Efficiency	The growing demand for bandwidth leads to a shortage of available spectrum.	▪ Employ Multi-input Multi-output (MIMO) technology to increase capacity without requiring additional spectrum [17]. ▪ Use Intelligent Radio [18] and Cognitive Radio [19] techniques to dynamically access underutilised spectrum bands.
3rd	Latency Reduction [20]	Real-time applications require low latency, which can be challenging in complex systems.	▪ Optimise signal processing algorithms for speed and efficiency. ▪ Utilise edge computing [21] to process data closer to the source, reducing transmission delays.
4th	Channel Estimation and Equalisation	Variability in channel conditions complicates signal recovery.	▪ Implement machine learning algorithms for more accurate channel estimation [22]. ▪ Use advanced equalisation techniques to mitigate the effects of multipath fading [23].
5th	Energy Efficiency	The need for low-power solutions in mobile and IoT devices is critical.	▪ Develop energy-efficient algorithms and hardware, such as low-power signal processing circuits [24]. ▪ Implement sleep modes and energy-harvesting [25] technologies to extend battery life.
6th	Security and Privacy	Increasing threats to data integrity and privacy in communication systems.	▪ Incorporate robust encryption techniques to secure transmitted data [26]. ▪ Use advanced signal processing techniques for anomaly detection [27] to identify potential security breaches.

[1] Poe.com, "What are the current challenges in digital and analog signal processing in electrical, electronic and computer engineering?", Poe.com, Available: https://www.poe.com/ [Accessed: 13 April 2025].

[2] Poe.com, "What are the current challenges in signal processing in modern communication systems? List the main ones in descending order offering solutions or strategies to mitigate them?", Poe.com, Available: https://www.poe.com/ [Accessed: 13 April 2025].

7th	Scalability	As systems grow, maintaining performance becomes more complex.	▪ Design modular systems that can be easily expanded. ▪ Leverage cloud-based processing to handle increased load without degrading performance, i.e. by load balancing [28].
8th	Integration of Emerging Technologies	Incorporating technologies like 5G, 6G, IoT and AI [29] into existing systems.	▪ Foster interdisciplinary collaboration to ensure seamless integration. ▪ Develop standardised protocols to facilitate interoperability between different technologies.

In this paper we will investigate the use of Artificial Intelligence (AI) in one of the many ways to improve both analogue and DSP. We want to show that this processing provides a better accuracy than before. It will be done by implementing AI and Machine Learning (ML) Algorithms. AI will cause a revolution in the field of telecommunication. Table 3, below, shows the challenges of SP in modern communication systems, listed in descending order of significance, along with the application of AI to mitigate them. The use of AI as shown in Table 3, does show a different ranking order compared to Table 2 in the use of AI in SP for communications, notably the second position is the use of AI for channel estimation and equalisation and the third being to increase spectral efficiency through advanced AI modulation schemes and Cognitive Radio [18-19] technologies.

Table 3. Current Challenges and AI Solutions in SP in Modern Communication Systems[3]

№	**SP Comms Challenge Type**	**Challenge**	**Artificial Intelligence (AI) Solution**
1st	Interference and Noise Management	Increased interference from various sources can degrade signal quality.	▪ Use deep learning algorithms for adaptive noise cancellation [30] and interference suppression [31]. ▪ Implement AI-driven signal classification [32] to identify and mitigate specific interference patterns.
2nd	Channel Estimation and Equalization	Variability in channel conditions complicates signal recovery.	▪ Employ machine learning models to predict channel conditions [22] and optimise equalisation techniques [33]. ▪ Utilise reinforcement learning for dynamic adaptation of equalisation algorithms [34] based on real-time channel feedback.
3rd	Spectral Efficiency	Growing demand for bandwidth leads to a shortage of available spectrum.	▪ Implement AI algorithms for dynamic spectrum management [18] and cognitive radio [19] technologies to optimise spectrum usage. ▪ Use generative models to design new modulation schemes [35] that maximise spectral efficiency.
4th	Latency Reduction	Real-time applications require low latency, which can be challenging.	▪ Optimise signal processing pipelines using AI to predict and reduce processing delays [36]. ▪ Employ edge AI for local processing [37], minimising the need for long-distance data transmission.
5th	Energy Efficiency	The need for low-power solutions in mobile and IoT devices is critical.	▪ Utilise AI algorithms to optimise resource allocation and power management in communication devices [38]. ▪ Implement predictive models to dynamically adjust transmission power based on network conditions [39].
6th	Security and Privacy	Increasing threats to data integrity and privacy in communication systems.	▪ Use ML for anomaly detection to identify unusual patterns that may indicate security breaches [40]. ▪ Implement AI-driven encryption techniques that adapt to evolving threats in real time [41].
7th	Scalability	As systems grow, maintaining performance becomes more complex.	▪ Develop AI-based resource management systems [42] that automatically scale processing power and bandwidth as needed. ▪ Utilise cloud-based AI solutions to handle increased data loads efficiently [43].
8th	Integration of Emerging Technologies	Incorporating technologies like 5G, 6G. IoT and AI into existing systems.	▪ Foster collaborative AI frameworks [44] that ensure interoperability between different technologies. ▪ Use AI for simulation and modelling to predict the impact of integrating new technologies before implementation [45].

[3] Poe.com, "What are the current challenges in signal processing in modern communication systems? List the main ones in descending order offering solutions or strategies to mitigate them using AI?", Poe.com, Available: https://www.poe.com/ [Accessed: 13 April 2025].

This paper discusses the accuracy of SP because of the implementation of these types of AI SP algorithms. It shows the accuracy using Deep Learning (DL) algorithms and Autoencoders. There exist many techniques in SP such as: Filtering, Fourier Transform and Modulation. However, these techniques are outdated now - when working with big datasets these techniques have a low accuracy and performance. It is good to improve the techniques of the prevailing SP algorithms by incorporating these new AI algorithms. The implementation of DL models in SP has shown favourable results by overcoming the different limitations and offering new prospects in this field. DL algorithms have been able to exhibit extraordinary potentials in the exploitation of compounded patterns. Incorporating these AI algorithms with existing SP algorithms will make SP more robust.

2. Signal Processing (SP)

2.1. Signal Processing Techniques

A signal is just a function which varies according to time. Its purpose is to carry information based on the phenomenon nature or behaviour. A signal has a mathematical representation as a function of a variable (in this case it is an independent variable) t which symbolizes the time. So, we can denote a signal as x(t). A signal can be continuous or discrete based on the value of the variable t. If the variable t is continuous, then the signal x(t) will be continuous. On the other hand, if the variable t is discrete the signal x(t) will be discrete. Usually, a discrete signal is described as a sequence of numbers because the discrete signals are usually represented at discrete intervals of time [46].

SP includes the data conversion and data transformation in such an approach that we are able to perceive things which we cannot see through direct scrutiny. It is able to permit many scientists and engineers to improve, examine and to rectify signals, involving streams in audio, videos, scientific data and images. There are several categories of SP which includes: Analogue, Digital, Continuous Time, Discrete Time, Nonlinear and Statistical.

One of the techniques used for SP is called Filtering. During filtering we are able to omit undesired components from our signal and keep only the components that we want. Based on the frequency of the signals, we have four types of filters which are called: High-Pass Filters, Low-Pass Filters, Band-Pass Filters and Band-Stop Filters. High-Pass Filters allow only the signals with high frequency to pass through it, while the signals with low frequency are blocked. Low-Pass Filters allow only the signals with low frequency to pass through it and blocks the signals that have a high frequency. Band-Pass Filters allow only a given range of signals to pass through, while blocking the others. Band–Stop Filters block a given range of signals to pass through while allowing others [47]. The four cardinal types of filters are shown below in Fig. 1.

Another technique used for SP is known as Fourier Analysis. It is a category of mathematical analysis which aims to recognize different cycles or patterns inside a dataset of time series that has previously been normalized [47]. This technique wants to make the data (which may be complex) simpler.

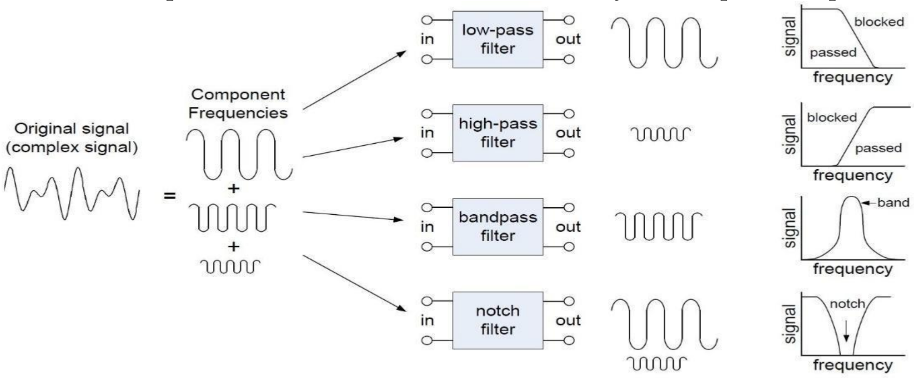

Figure 1. The Four Major Filter Types[4]

This is done by decomposing this complex data into an exponential function series or trigonometric series e.g. sinusoidal waves. Every one of these sinusoidal waves has a certain amplitude, cycle length and

[4] N. Davis, "An Introduction to Filters", All About Filters, 30th September 2023, Available: https://www.allaboutcircuits.com/technical-articles/an-introduction-to-filters/ [Accessed: 21 June 2025].

a phase relationship with the other sinusoidal waves, that later being able to reconstruct the data that we observed [48]. Fourier Analysis comprised different approaches such as: Continuous Fourier Transform (CFT), Discrete Fourier Transform (DFT), Fast Fourier Transform (FFT) and Short-Time Fourier Transform. Continuous Fourier Transform is used to transform a continuous-time signal (also known as an analogue signal) into the representation of their continuous-frequency approach. The formula of CFT for a signal x(t) is:

$$X(j\omega) = \int_{-\infty}^{\infty} x(t) * e^{-j\omega t} dt \tag{1}$$

Where $X(j\omega)$ represent the Continuous Frequency approach of the signal, $\omega = 2\pi f$ and represents the angular frequency while j represents the imaginary part. The limits of the integral are from $-\infty$ to $+\infty$ because the CFT needs an infinite amount of frequency and time ranges [48]. The DFT is used to convert discrete-time signals into the representation of their discrete-frequency approach. Usually, DFT uses FFT. The formula of DFT is:

$$X[k] = \sum_{n=0}^{N-1} x[n] * e^{-\frac{j2\pi kn}{N}} \tag{2}$$

Where $X[k]$ represents the Discrete Signal in its frequency domain, $x[n]$ represents the signal in its time domain, n represents the number of samples, and k represents the frequency.

The FFT transforms a signal into discrete spectral units and as a result, it is able to issue information about the signal. The FFT is employed in many different areas of SP such as control of quality, monitoring of systems or machines and fault analysis. It is an advanced algorithm used for DFT implementation. When we sample a signal during a time period, the signal is split into its frequency units. These units are sine oscillations at marked frequencies, every one of them possessing its phase and amplitude. Because of this, FFT is able to decrease the complexity of the computation of the DFT [49].

Short Fourier Transform (SFT) is employed to establish the phase content and the frequency of a sinusoidal signal while it alternates gradually. During this method, a longer signal is split into segments which are shorter and the length of these segments is the same. After this, we perform the Fourier Transformation on every one of these short segments.

Another process of SP is "modulation". Modulation is the conversion of information into waves of radio by the addition of data to an optical or electromagnetic (EM) carrier signal. A carrier signal is a signal with a stable waveform. This means that a carrier signal has its frequency, height and amplitude constant, before it is changed or "modulated" by the information/data signal. The data can be appended onto the carrier signal by differing its: polarization, frequency, phase, amplitude or a combination of these. There are different types of modulations such as Amplitude Modulation (AM), Phase Modulation (PM), Frequency Modulation (FM), Quadrature Amplitude Modulation (QAM) and Phase Shifting Keying (PSK). AM is employed in order to broadcast messages with the help of a radio wave. In this form of modulation, the wave amplitude is varied in sympathy to the ratio of the signal of the message. The message signal carries the required data which may be video or audio and on the other side, the transmission frequency is determined by the carrier signal. The signal data are held by a lower sideband, an upper sideband and a carrier wave. PM is another modulation type during which the carrier signal's phase is modified based on the amplitude of the message signal. This type of modulation is usually employed in communication between satellites. FM takes place when the frequency of a carrier signal is modified based on the amplitude of the message signal. FM is able to issue a better quality for the signal since it is less susceptible to interference and noise. QAM employs both phase components and amplitude components in order to issue a modulation form which can provide very high levels of efficiency. So, in other words, this is a combination of both PM and AM. QAM may be employed for both Analogue and Digital Modulation. When it is employed for radio waves transmission, it can transfer very large data rates compared to the usual PM schemes and AM schemes. While for basic signals, there are only two points in which the transfer is allowed, by employing QAM there are many more different positions in which the signal may be transferred, every one of them having a certain phase and amplitude. When we plot a graph of QAM based on phase and amplitude these positions are denoted as points and based on the number of points in these graphs we are able to determine what kind of QAM it is. It may be: 2-QAM, 4-QAM, 8-QAM, 16-QAM, 64-QAM or 256-QAM, etc. [50]. PSK is a modulation process that transfers information by modifying the phase of a constant-frequency carrier signal. This type of modulation is achieved by differing the inputs of sine and cosine at an accurate time. PSK includes: Binary Phase Shift Keying (BPSK), in which the carrier signal contains two phases; Quadrature Phase Shift Keying (QPSK), in which the carrier signal contains four phases; 8-Phase Shift Keying (8-PSK), in which the carrier signal contains eight phases, etc. Every phase represents a bit sequence, which permits the information transmission to be as efficient as possible [51]. Fig. 2, below, shows the major modulation classifications based on a digital carrier being used.

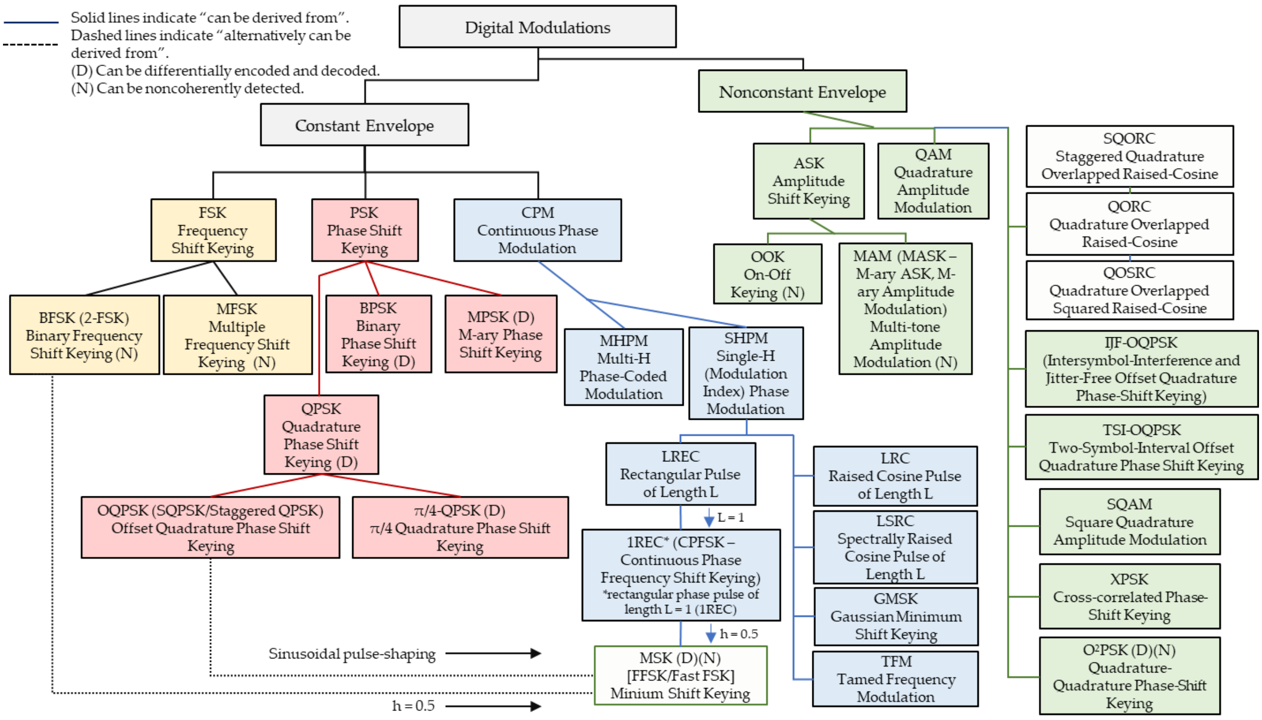

Figure 2. The Principal Modulation Types. (Adapted from [52])

3. AI Algorithms

AI is the ability of machines to perform tasks without being supervised all the time by people. It is very powerful and it is very useful nowadays. The prime subset of AI is ML. ML is a term used for algorithms that can make a device learn without programming it in an explicit way. In Supervised Learning, the data is labelled before being taken under processing. It means that if we have different images of apples and oranges and want to classify which images will be oranges and which will be apples, then the data is divided between images which are apples and the images which are oranges. By doing this, after each data entering in these algorithms, it will be able to recognize which image belongs to specific class. Differently from that when we are using Unsupervised Learning there is no labelling of the data before processing. Based on the example above, all of the images of apples and oranges are put inside the algorithms and if another image comes, it will only be recognisable by the program based on the similarity of the image in comparison to all the images. Reinforcement Learning is another type of ML algorithms. In this algorithm we have an agent being put inside a specific environment. Based on the achievements it will make through its actions; it will also take different rewards. The algorithms or RL are usually employed for computer games.

DL Algorithms are a subset of ML. Differently from usual ML algorithms these types of algorithms contain more than one-layer, specifically at least three-layers in their structure. The DL Algorithms may be supervised, such as: Artificial Neural Networks (ANNs) [54], Convolutional Neural Networks (CNNs) [55] and Recurrent Neural Networks (RNNs) [56]; unsupervised, such as: Auto-encoder [57] or part of Reinforcement Learning algorithms [58], such as Deep RL [59][60] or Deep Q-Learning (DQN) [61]. Fig. 3, below, shows the top five ML algorithm classifications and their typical application scenarios.

ANNs [54] have taken their name from biological neurons. The structure of an artificial neuron is shown in Fig. 4, below.

The ANN [54] begins with an input data x which may be taken from a previous layer inside the structure or from an outsider factor. These input data contain a numerical value. After these input nodes we have *weight*. Weight is the framework that is able to convert the input data inside the hidden layer as well as to reduce the error. After all the input data is summed-up we have the bias b. Bias is a constant that is added to the sum of the products of the input data and weight. After that we have the *activation function*. The *activation function* decides if our function should be activated or not. There are different types of *activation functions* such as: Reactive Linear Unit (ReLU) [63], Sigmoid [64], LeakyReLU [65], SoftMax [66], etc. ReLU [63] is a function that for each $x < 0 \ then \, y = 0$ and for each $x \geq 0 \ then \, y = x$. Sigmoid function [64] is a function that for each x we have $y = \frac{e^x}{e^x+1}$. LeakyRelu [65] activation function has its equation for each: $x < 0 \, then \, y = 0.1 * x \, and \, for \, each \, x \geq 0 \ then \, y = x$. SoftMax [66] is an activation function that is able to scale the numbers into probabilities.

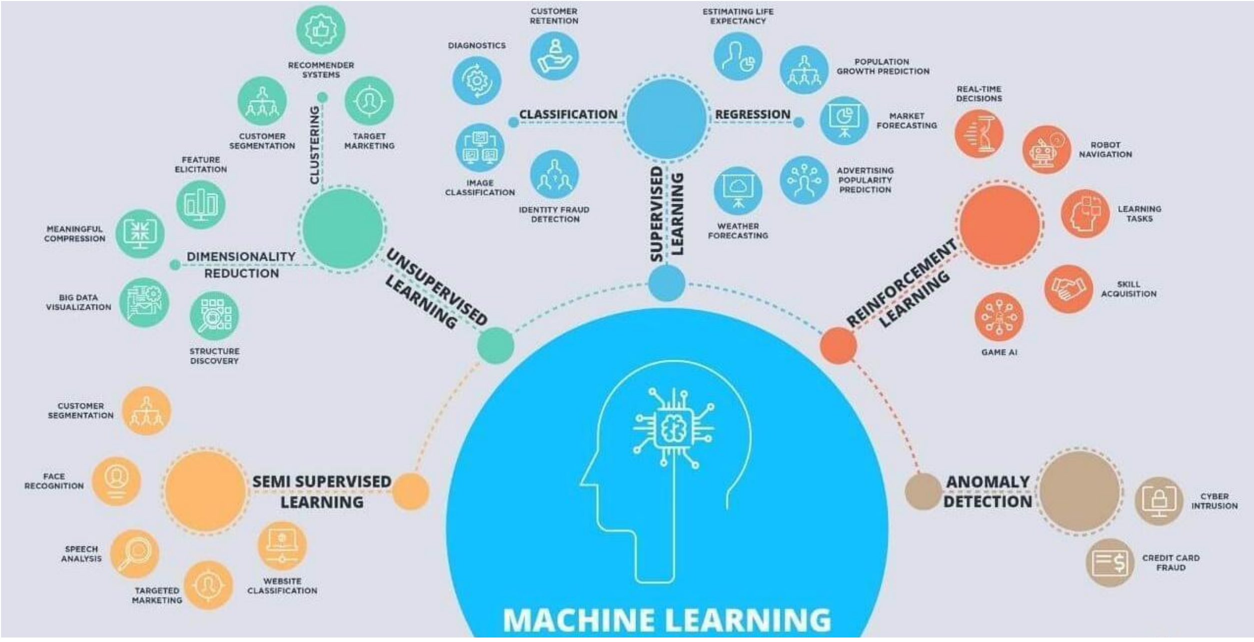

Figure 3. The Top Five Powerful ML Algorithm Categories[5]

Its mathematical function for each $x \ is \ y_j = \frac{e^{x_j}}{\sum_{i=1}^{N} e^{x_i}}$ where N is the number of different classes inside of a classifier, which is multilevel. The final part of an artificial neuron is the *output*, which is the predicted value of all process inside the neuron. So, the function of ANN [54] is:

$$y = \sum_{i=1}^{N} x_i * w_i + b \tag{3}$$

in which y represents the *output*, x represents the *input data*, w represent the *weight*, b represents the *bias*, i represents the *number of iterations* and N represents the *total number of the inputs*.

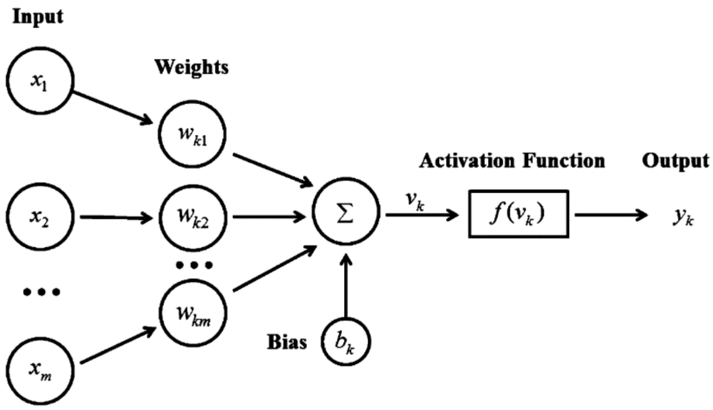

Figure 4. The Structure of a Neuron [62]

ANN [54] is feed-forward, so it means that the movement of data inside its neural network structure is from the input to the hidden layer/s until the output. There is no turning back from the hidden layer to the input layer or from the output layer to the hidden layer.

[5] Filip Projcheski, "List of Top 5 Powerful Machine Learning Algorithms That Will Solve 99% of Your Problems", Laconic Machine Learning, 18 July 2020, Available: https://laconicml.com/machine-learning-algorithms/ [Accessed: 24 June 2024].

Another type of ANN [54] is the CNN [55]. These structures are employed in many fields of today's technology such as Computer Vision, Medicine, etc. Differently from ANNs [54], CNNs contain types of layers such as *Convolutional Layer*, *Pooling Layer* and *Flatten Layer*. The *Convolutional Layer* can apply a convolution computation inside the input layer, proceeding the result to the following layer. The convolution transforms the whole pixels to its accessible field all together into a single value, e.g. if we could perform an image convolution, we would be reducing the size of the image, moreover bringing the whole information inside the field all together into a single pixel. A *filter* or also known as a *kernel*, implements an element-wise multiplication over the input data. After we sum up all the results, we take only a single output pixel. The next layer in the CNN is the *Pooling Layer*. The aim of the *Pooling Layer* is to reduce the hidden layer dimensions, by uniting the neuron clusters' output from the preceding layer to a single neuron into to the succeeding layer. There exist two different forms of pooling operation, which are: *Average Pooling* and *Max Pooling*. In the Average Pooling layer, we calculate the average of the input values and take their mean. In Max Pooling, later we take the largest value of the input values. The *Flatten Layer* is employed to transform all the resultant 2-D arrays from maps of pool attribute into a line vector, which is continuous [67]. Fig. 5, below, is an abstraction of a typical CNN architecture.

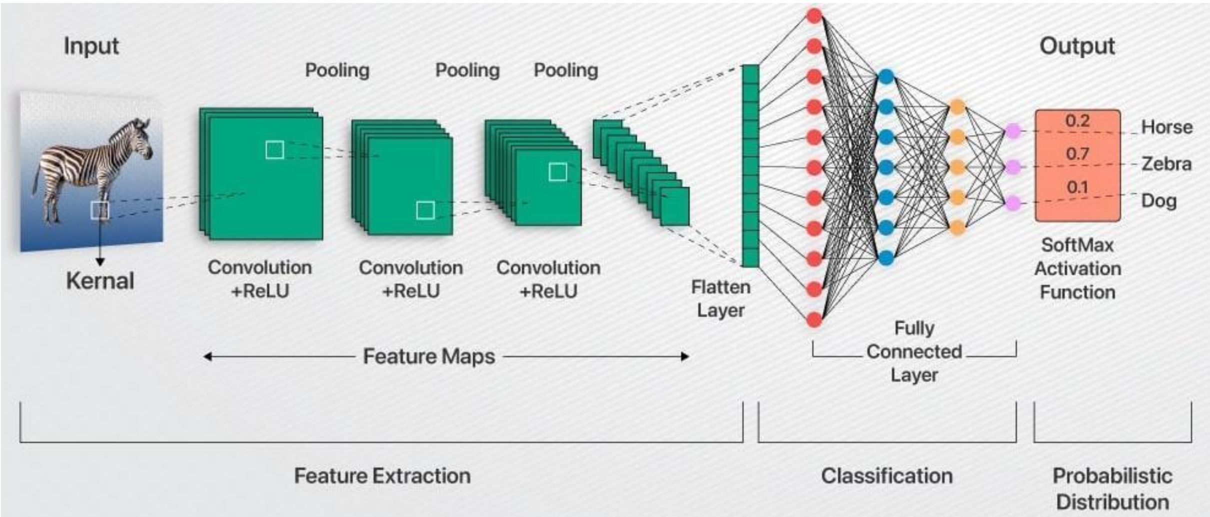

Figure 5. A Typical CNN (Convolutional Neural Network) [55, 67] Architecture[6]

Another type of ANN [54] is RNNs [56]. Unlike the previous types of ANNs [54], these type of ANNs [54] are Feed-backward, so the data flows not only in one direction from the input to the output, but also backwards, from the hidden layer to the input, also from the output to hidden layer/s or directly from the output to the input. A typical RNN [56] is shown in Fig. 6, below.

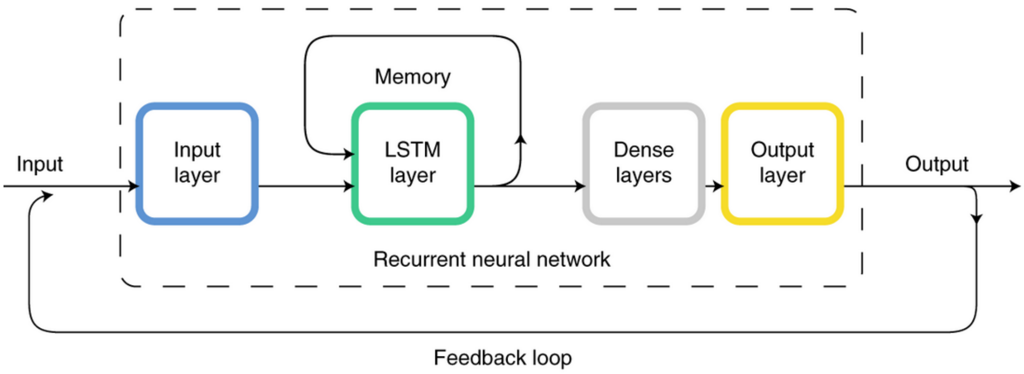

Figure 6. A Typical Recurrent Neural Network Architecture [74]

In RNNs [56], this movement of data usually form a loop. For this reason, these algorithms are good for Speech Recognition [68] and Natural Language Processing (NLP) [69-70]. Two of the most well-known layers of RNNs are Long Short-Term Memory (LSTM) [69, 71-72] and Gated Recurrent Unit (GRU) [69, 72]. The LSTM [71] layer is composed of a *cell*, an *input gate*, a *forget gate* and an *output gate*. The *cell* is able to

[6] Analytixlabs, "Convolutional Neural Networks – Definition, Architecture, Types, Applications, and more", 8 January 2024, Available: https://www.analytixlabs.co.in/blog/convolutional-neural-network/ [Accessed: 24 June 2024].

recall the values above intervals of time. The *input gate*, the *forget gate* and the *output gate* control the flux of information inside and outside of the *cell*. The *input gate* determines what chunks of information should be loaded in the present state by giving a value of 0 (not to load the information) or 1 (to load the information).

The *forget gate* determines what chunks of information should be discarded from the previous state by giving them a value of 0 (to discard the information) or 1 (to keep the information). The *output gate* determines what chunks of information into the present state should be output by giving a value of 0 (which represents the previous state) or 1 (which represents the current state). LSTM [71-72] can preserve valuable, long-term dependencies to make predictions, both in present and future time-steps, by selectively outputting accurate information from its present state. Gated Recurrent Unit (GRU) [69, 72] is just a version of LSTM [71-72]; however, it is much less complex than LSTM [71-72]. Unlike LSTM [71-72], GRU [69, 72] does not contain on output gate [73].

3.1. Advantages of Using CNNs over RNNs for Denoising and Classification in Signal Channels

The advantages of CNNs over RNNs when used for denoising or classification tasks in signal channels include:

- **Spatial Hierarchy**: CNNs are designed to capture spatial hierarchies in data through convolutional layers. This capability allows them to effectively learn features from local regions of input data, making them particularly suited for image-like data in signal processing.
- **Parameter Sharing**: CNNs use shared weights for convolutional filters, which significantly reduces the number of parameters in the model. This leads to faster training times and less risk of over-fitting; this is especially beneficial in scenarios where limited data is available.
- **Translation Invariance**: The pooling layers in CNNs enable the model to become invariant to small translations in the input signal. This property is crucial for applications in signal processing, where the exact location of a signal feature may vary.
- **Reduced Complexity**: For tasks like denoising, CNNs can process entire batches of data in parallel, enhancing computational efficiency. In contrast, RNNs process sequences step-by-step, which can be slower and less efficient for high-dimensional data typical in signal processing.

Another type of ANN [54] is the Generative Adversarial Networks (GANs) [75]. GANs [75] are a procedure which includes the implementation of ANNs [54] into Generative Modelling [76-77]. Generative Modelling [76-77] uses unsupervised learning, which includes finding and learning patterns or regularities inside the input information so that the model may be employed in order to generate new samples that probably may have been taken from the original dataset. GANs [75] are a smart method of generative modelling [76-77] training by formulating this type of issue as a supervised learning issue, which contains two sub-models: a generator model [78], which is trained in order to generate new samples and a discriminator model [78], which attempts to segregate the samples into real (which belong to the domain) and fake (which are generated). A modern GAN generator and discriminator architecture is shown in Fig. 7, below. Fig. 8, below, shows a typical complete GAN [75] architecture. We train these two types of models together inside a zero-sum game [79], till the point that the model of the discriminator is deceived for about half a time, which means that the model of the generator is still generating possible samples. GANs [75] are employed in many fields such as image-to-image translation [78] and for generating photorealistic images [80].

Auto-encoders [57] are also DL [63, 73] algorithms but function differently from traditional ANNs [54]. These types of algorithms are unsupervised learning algorithms. These algorithms are very useful for dimensionality reduction [81-82] and data compression [83]. Auto-encoders [57] are able to learn image reconstruction, text reconstruction, also different types of data reconstruction, from different compressed varieties of themselves.

The structure of an auto-encoder is composed of three layers, as shown in Fig. 9, which are the *Encoder Layer* (input cells), *Code Layer* (hidden cells) and the *Decoder Layer* (match input output cells). The function of the *Encoder Layer* is to compress the input data to a representation of latent space. This means that the *Encoder Layer* can encode the input data like a representation which is compressed inside a dimension which is reduced. The compressed data is a twisted version of the original data. The next layer that is the *Code Layer*, illustrates the compressed data input which is sent to the *Decoder Layer*. The *Decoder Layer* is able to decode the encoded data back into its initial dimensions. This decoded data is re-established from the representation of the latent space. There are several types of auto-encoders [84].

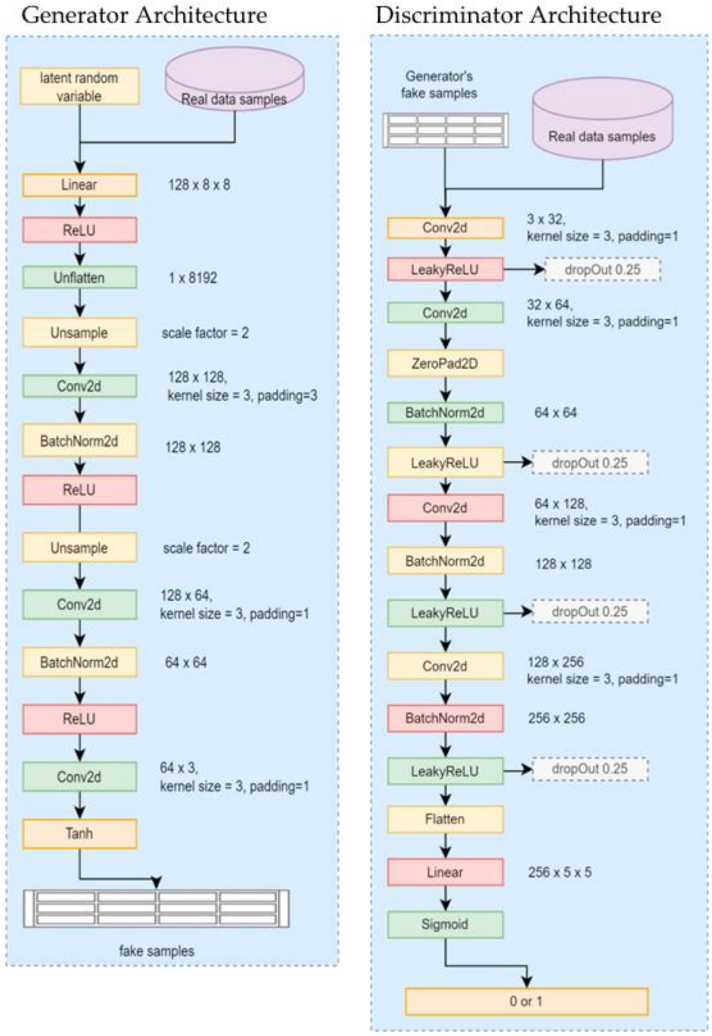

Figure 7. A Modern GAN Generator and Discriminator Architecture [78]

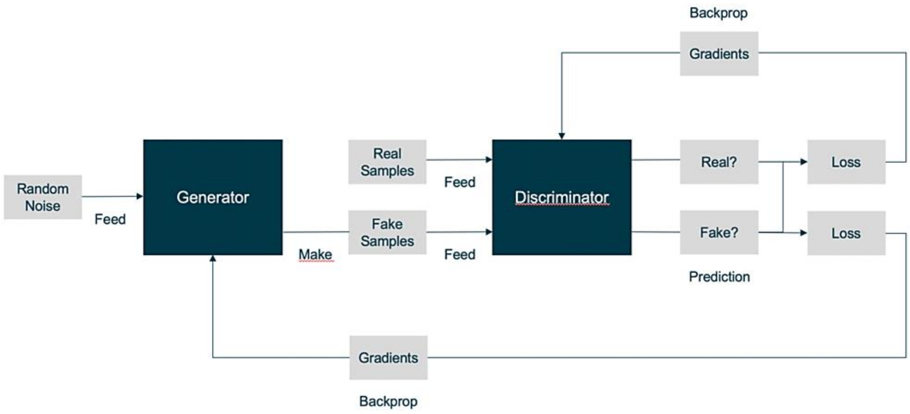

Figure 8. A Typical Generative Adversarial Network (GAN) Architecture[7]

One type of Auto-encoder is De-noising Auto-encoder [85]. This type of Auto-encoder is able to generate a distorted copy of the original input by adding noise. This is helpful in order to evade auto-encoders from making copies of the original input into the output without learning data attributes. De-noising auto-encoders [85] extract a relatively distorted input during the time of training to retrieve the

[7] Stephan Müller, "Generative Adversarial Networks: How Data Can Be Generated With Neural Networks", statworx(R) blog, 10th October, 2020, Available: https://www.statworx.com/en/content-hub/blog/generative-adversarial-networks-how-data-can-be-generated-with-neural-networks/ [Accessed: 24 June 2024].

initial input. In order to remove the extra noise, this auto-encoder learns a field of vectors for shifting the input data to a lower dimensionality manifold that represents the real data [84].

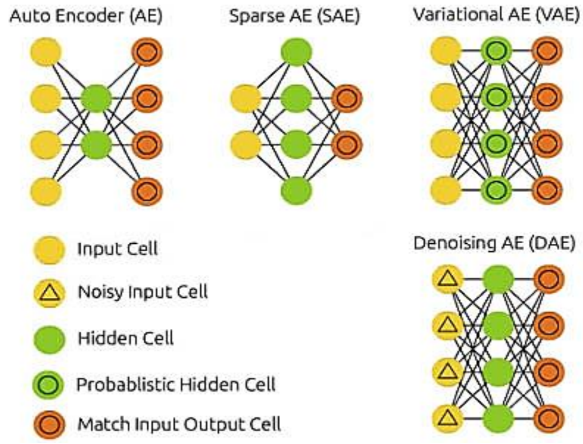

Figure 9. Architecture of four Types of Autoencoders[8]

Another type of Auto-encoder is the Sparse Auto-encoder [86]. In Sparse Auto-encoder [86], the hidden vertexes are more sizeable compared to input vertexes. These Auto-encoders [86] are able to find essential attributes from data. A common Sparse Auto-encoder is conceptualized where the complexity of a vertex matches with the activation level. Restriction of sparsity is established inside the hidden layer. This is done in order to obviate the output layer to copy data from the input layer. Sparsity can be established by further terms inside the loss function in the course of the process of training, either through probability distribution comparison of hidden layer activations, or by making all the strongest hidden layer activations zero [87].

Another type of Auto-encoder is the Convolutional Auto-encoder (CAE) [88]. These Auto-encoders [84] employ the convolution function [89] in order to utilize this examination. They first learn to transform the input into a collection of straightforward signals before trying to reconstruct its input data from these signals or alter the resulting image's geometry or reflectivity. They are cutting-edge tools for convolutional filter unsupervised learning. These filters can be used to extract attributes from any input once they have been trained. Therefore, these attributes can be applied to any task that calls for a condensed representation for the data, such as classification.

RNN auto-encoders (LSTM auto-encoders) [90] is the application of a sequence data auto-encoder by employing the LSTM [69, 71-72] in an encoder-decoder architecture. The LSTM layer [69, 71-72] is trained for reading the input data, encoding the input data, decoding the input data and finally rewriting it. The model performance is estimated according to its capability to reconstruct the input data.

Deep Reinforcement Learning (DRL) [59-61, 91-92] algorithms are algorithms that are connected to both DL [55] and Reinforcement Learning (RL) [58]. As part of RL algorithms, it employs an agent inside an environment. But unlike the usual RL [58], it also incorporates Neural Networks [64] algorithms inside it. These algorithms are able to grab very big data and determine on what actions must be accomplished in order to improve the intensions.

4. Materials and Methods

4.1. Applicability of MNIST and CIFAR-10 Datasets to Signal Processing

The MNIST[9] (Modified National Institute of Standards and Technology) dataset is a dataset which contains different handwritten digits. This dataset is composed of 70,000 digits, 60,000 of which are employed for training while 10,000 of them are employed for testing. The size of the digits has been normalized. These digits are concentrated around an image whose size is fixed. The dimensions of this dataset are 28×28 pixels. This dataset is usually used for image processing. The MNIST dataset was

[8] Tallaswapna, "Types of Autoencoders in Deep Learning", Medium, 10 July 2023, Available: https://medium.com/@tallaswapna9/types-of-autoencoders-in-deep-learning-383cfecc4d0e [Accessed: 24 June 2024].

[9] Kaggle, "MNIST Dataset", Available: https://www.kaggle.com/datasets/hojjatk/mnist-dataset [last accessed: 27 June 2025].

generated by the rearrangement of the National Institute of Standards and Technology (NIST) dataset samples [87].

A much more advanced dataset is the CIFAR-100[10] (Canadian Institute for Advanced Research) dataset. The CIFAR-100 dataset comprises 60,000 images. Unlike the MNIST dataset, these images are coloured images. 50,000 of these images are employed for training while the other 10,000 are employed for testing [93]. The MNIST and CIFAR-10 datasets offer valuable insights into signal processing for the following reasons, given below.

4.1.1. Pattern Recognition

Both datasets focus on object recognition, akin to identifying patterns in signals. Signal processing often involves recognizing and classifying patterns amidst noise, making these datasets relevant for training models in similar tasks.

4.1.2. Image Characteristics

The images in MNIST (28×28 grayscale) and CIFAR-10 (32×32 colour) are structured similarly to signals. Their two-dimensional data representation aligns well with signal processing techniques that analyse spatial and temporal dimensions.

4.1.3. Benchmarking

These datasets serve as benchmarks for evaluating the performance of AI algorithms. The insights gained from training on such well-established datasets can be extrapolated to real-world signal processing applications, enhancing model reliability and accuracy.

4.2. Library Used for Deep Learning (DL)

The library used for calling DL algorithm is Keras[11] [87]. Keras is a freely available library written in Python™[12] which is able to implement ANNs [54]. It is able to run easily regardless if we are implementing this library for a GPU (Graphics Processing Unit) or a CPU (Central Processing Unit). All models of ANNs [54] are accessible in this library. This library is modular and can be used for research in the field of innovation. In order to build a Keras model in programming, there are six steps that should be followed. The first step is the establishment of the network, along with layers and their inter-connections. Keras has two model types which are *Functional* and *Sequential*. The second step of establishing a Keras [87] model is by compiling this network that we built. This implies that the code that we have written must be transformed in appropriate form so that the machine will be able to understand. The third steps involve compilation of the model. This is carried out by specifying three parameters which are: the *optimizer*, which is used to minimize the losses; *metrics*, which is employed in order to discover the model's accuracy; and the *loss function*, which is used to compute the model's losses. After compiling the model, the fourth step is to fit the data. This method is used in order to train the model. The fifth step is to make the model evaluation. This is done in order to evaluate the error of the model. The sixth and final step is to make predictions: 'model.predict()' is used, we then employ the model on the new data in order to make these predictions.

4.3. Model Training Architecture Variations and Optimisation Techniques in AI for SP

4.3.1. Training Techniques

Effective training of CNNs for signal processing tasks involves techniques such as data augmentation, which helps in generating additional training samples by transforming existing ones (e.g. by flipping or rotation). This is particularly useful in scenarios where the training dataset is small.

4.3.2. Architecture Variations

Variations in CNN architecture can significantly impact performance. For instance, deeper networks with more convolutional layers can capture more complex features but may require more extensive training data to avoid overfitting. Using residual connections can also help in training deeper models effectively.

4.3.3. Regularization Techniques

[10] University of Toronto, "The CIFAR-100 dataset", Available: https://www.cs.toronto.edu/~kriz/cifar.html [Accessed: 27 June 2025].

[11] Keras, "A superpower for ML developers", Available: https://keras.io/ [Accessed: 26 June 2025].

[12] Python Software Foundation, python™, Available: https://www.python.org/ [Accessed: 26 June 2025].

Regularization methods such as L2 regularization and Dropout are crucial in preventing overfitting. Dropout, which randomly deactivates a subset of neurons during training, encourages the network to learn more robust features, enhancing generalization.

4.3.4. Optimization Techniques

The choice of optimizer can greatly influence training efficiency. Techniques like Adam or RMSprop dynamically adjust learning rates during training, allowing for faster convergence. Additionally, the use of learning rate schedules can help in fine-tuning the model as training progresses.

4.4. Comparison of AI Models and Traditional Signal Processing Approaches

While traditional signal processing techniques such as Fast Fourier Transform (FFT), filtering and modulation have been foundational in the field, AI models present distinct advantages and challenges.

4.4.1. Advantages of AI Models

Adaptability: AI models, particularly DL algorithms, can adapt to complex, non-linear relationships in data. Unlike traditional methods that rely on predefined mathematical models, AI can learn from the data itself, making it suitable for dynamic environments.

Feature Extraction: AI models, especially CNNs, excel at automatic feature extraction. Traditional techniques require manual feature engineering, which can be time-consuming and may not capture all relevant information.

Performance on Large Datasets: With the increasing availability of large datasets, AI models can leverage vast amounts of information to improve performance. Traditional methods might struggle with high-dimensional data, whereas AI models can efficiently process and learn from it.

4.4.2. Limitations of AI Models Compared to Traditional Approaches

Data Dependency: AI models often require large amounts of labelled data for training. In contrast, traditional methods can be effective with smaller datasets since they rely on established mathematical principles.

Computational Resources: Training AI models, particularly deep learning architectures, demands significant computational power and can be time-consuming. Traditional signal processing techniques can often be implemented more quickly and with less computational overhead.

Interpretability: Traditional methods provide clear mathematical foundations that allow for greater interpretability. AI models, particularly deep neural networks, can operate as "black boxes", making it challenging to understand how decisions are made.

Overfitting: AI models, particularly deep learning architectures, can easily overfit to training data, especially when datasets are small or noisy. Overfitting occurs when a model learns the noise in the training data rather than the underlying patterns, leading to poor performance on unseen data. Regularization techniques and cross-validation are essential to mitigate this issue, but they can increase complexity.

Generalizability: The ability of an AI model to generalize to new, unseen data is critical for its practical application. However, models trained on specific datasets may not perform well when applied to different environments or conditions. This limitation is particularly pronounced in signal processing, where variations in noise, interference, or system parameters can significantly impact model performance.

Model Robustness: Many AI models can be sensitive to minor changes in input data, resulting in decreased performance in real-world applications. Robustness is crucial in signal processing, where signals can be affected by various factors such as environmental conditions or hardware imperfections. Developing more robust models that can maintain performance under varying conditions remains a key challenge.

4.5. AI Implementations of Signal Processing (AI SP)

In supervised DL algorithms, the model type is *Sequential*, i.e. the data is able to proceed from one layer (starting from the input layer) to another one, in specific order, till it reaches the final layer (output layer). For ANNs [54] in this experiment, we have used fully-connected layers, which are simply known as Dense Layers [94]. "Dense" is a neurons' layer that is fully connected to the previous layer, which indicates that the neurons inside the layer are linked to each neuron of the previous layer. The Dense Layer of each artificial neuron inside of a model obtains its output from each artificial neuron of its previous layer, in which a matrix vector multiplication is implemented by neurons. In order for this multiplication to take place, the output row vector of the previous layers should be equivalent to the currents dense layer's column vector. So, the formula for this multiplication is:

$$Ax = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & \dots \\ \dots & \dots & & \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}_{m\times n} * \begin{bmatrix} x_{11} \\ x_{21} \\ \dots \\ x_{n1} \end{bmatrix}_{n\times 1} = \begin{bmatrix} b_{11} \\ b_{21} \\ \dots \\ b_{m1} \end{bmatrix}_{m\times 1} = b \tag{4}$$

In which:
A represents the values inside the matrix;
x represents the values inside the vector;
b represent the values inside their product;
$m \times n$ represents the dimensions of the matrix which has m-rows and n-columns;
$n \times 1$ represents the dimensions of the vector which has n-rows and one column and
$m \times 1$ represents the dimensions of the product vector which has m-rows and one column.

The values that are inside the matrix are the parameters that are trained on the previous layers. Their parameters may be upgraded using the method of Back-propagation [95]. Inside the ANNs [54], this method calculates the loss function gradient in regard to network weights for single output or input. The Dense layer is employed in order to alternate the vector dimensions by employing each neuron. Inside the Dense layers, two parameters, which are: *Units* and *Activation*. *Units* have been used to determine the output size through the Dense Layer. Usually in these algorithms the number of *Units* is a multiplier of the number two on input layer and hidden layers while in the output layer it will have the value of different samples used for classification. In both cases using the MNIST dataset and CIFAR-10[13] dataset we had ten different classified groups, so the number in both output layers was ten. *Activation* determines the *activation function* which will convert the values of neuron inputs. Because the Dense layer is linear, we use the Rectified Linear Unit (ReLU) [63][65] activation function in all the hidden layers. In the output layer, we use the SoftMax [66] activation function because we want to determine the percentage of accuracy in this algorithm.

Convolutional Neural Networks (CNNs) [55, 67], besides these fully connected layers, include different layers such as 2D Convolutional Layer [96] (Conv2D), 2D MaxPooling Layers [96, 97] (MaxPooling2D) and a Flatten layer [98]. Inside the 2D-Convolutional layer [96], a filter (also known as a *kernel*) is employed to accomplish the element-wise multiplication of the input data. After that, all the results are summed up in just one output pixel. This the filter is able to accomplish this multiplication in each area that it slides over, thus converting the 2-Dimensional matrix of attributes into a divergent 2-Dimensional matrix of attributes. Two other elements inside the 2-D CNN are *strides* and *padding* [96]. *Strides* show up to which degree the filter is able to move in just a single direction. It means that if the strides number is (1, 1), it means that the filter is able to move one pixel horizontally and one pixel vertically. Similarly, if the number of strides is (2,2), it means that the filter is able to move two pixels horizontally and two pixels vertically at a certain time. So, the number of strides determine the steps number which we are proceeding in every convolution. *Padding* is important for the creation of Convolutional Neural Networks [55, 67]. It is employed in order to maintain the dimensions of inside an input image after operations of convolution are performed inside an attribute map. *Padding* includes the addition of further pixels throughout the border of the input attribute map before the time of the convolution. In our model, we have used *padding* 'same' because it is employed in order to maintain the dimensions of the attribute map. The other method of padding is called 'valid' and it is employed in order to decrease the dimensions of the attribute map. 2-D MaxPooling Layer [96-97] is the operation of computing the biggest value of every patch inside every attribute map. It also has a kernel which has the same function as the kernel of the Convolutional layer. The final *hidden layer* is the Flatten Layer [98] which is employed to transform the input spatial dimensions to the channel dimension. It has no elements inside its layer.

Another Layer besides the fully connected layer that is used in RNNs [56, 69, 74] is the LSTM layer [69, 71, 72]. LSTM is a type of RNNs which is able to learn long-term dependences, distinctly in problems which involve sequence predictions. Differently from the previous layers which have only feed-forward [54]

[13] University of Toronto, "The CIFAR-100 dataset", Available: https://www.cs.toronto.edu/~kriz/cifar.html [Accessed: 27 June 2025].

connections, this layer has also feed-backward [56] connections. The activation function of LSTM layers is the Rectified Linear Unit [63, 65]. It has a high level of accuracy as an algorithm.

In Sparse Auto-encoders [86] we use the same layers such as in ANN (the fully-connected layers dense). Differently from ANNs in Sparse auto-encoders [86] the layers are divided in three parts which as previously mentioned are the Encoded part, Code part and Decoded Part. The first part of the structure is the Decoded part. In these layers, the number of units is divided by two, compared to the previous layer. The second part is the Code part which is composed of just one layer and it has the smallest number of units compared to other layers. It is used to transfer the information from the Encoded Part to the Decoded part. The number of layers in the decoded part is equal to the number of layers in the Decoded part. In the decoded part the units of each layer are divided by two and the final hidden layer in the decoded part has the same number of units as the input layer inside the sparse auto-encoder [86]. The activation function is the same as ANN.

In De-noising Auto-encoder [85] we add a noise factor at the input. Here we use Convolutional Layers, MaxPooling Layers in the encoded part and in the decoded part we use Convolutional layers and Upsampling layers in the decoded part. Differently from the previous auto-encoder in this type of auto-encoder we multiply by two the units in the encoded part and the units are divided by two in the decoded part. Upsampling is employed in order to return the dimensions of the dataset at the output to the same as in the input. A parameter which defines the Upsampling layer is the size which determine in what scale the decompressed image will be resized.

In CNN-autoencoder [99] we use the same layers that we mentioned earlier which are Convolutional and MaxPooling in Encoder and Convolutional and Upsampling into decoder. In CNN, padding is the same. The units of Encoder are multiplied by two after each CNN layer and it is divided by two in the decoder after each CNN layer.

In RNN autoencoder [100] besides LSTM we use another layer which is Repeat-Vector. Repeat-Vector is employed in order to repeat the set number's input as many times as we want. In the encoded part, the LSTM layer units are divided by two after each layer and in the decoded part, they are multiplied by two after each layer until they reach the original shape.

The optimization algorithm used for all these implementations, is "Adam" (Adaptive Moment Estimation) [101]. This algorithm is able to integrate the advantages of two different algorithm of optimization which are RMSProp (Root Mean Squared Propagation) [102] and Adagrad (Adaptive Gradient) [103-104]. This algorithm is well known in terms of the efficiency it has. It is robust and is employed in many different tasks of DL. These algorithms calculate individual learning rates for various frameworks by computing the gradients first moments and second moments. This optimizer integrates manners of momentum by using the mean (the first moment) of the gradients, furthermore it examines the gradients' scale across the variance (the second moment). These characteristics make Adam one of the most powerful optimizers [101].

5. Results and Discussion

As we see from the results obtained using the two datasets show in Tables 4 and 5, the seven algorithms investigated, provide a high level of accuracy for SP. These algorithms are powerful and scalable. Because of their complexity and their multilayered structure, they are highly efficient in this topic. As the results show, all seven algorithms provide at least 80% rate of accuracy - which is very good for these kinds of problems. Their activation functions that we choose for this study such as ReLU (Rectified Linear Unit) and LeakyReLU (Leaky Rectified Linear Unit) in input and hidden layers provide as well as SoftMax (a probability function) are very helpful for their efficiency and the classification of different types of signals. Another reason for this high level of accuracy is the optimization algorithm, Adam, that we have used.

Table 4. MNIST Dataset

Order №	Algorithms	Accuracy Score
1	CNN Auto-encoder	99.77%
2	Artificial Neural Networks (ANNs)	99.13%
3	Sparse Auto-encoder	99%
4	Convolutional Neural Networks (CNNs)	98.39%
5	Recurrent Neural Networks (RNNs)	97.86%
6	De-noising Auto-encoder	97.59%
7	Auto-encoder RNN	96.46%

Table 5. CIFAR-10 Dataset

Order №	Algorithms	Accuracy
1	RNNs	96%
2	CNN Auto-encoder	94.5%
3	ANNs	93.79%
4	De-Noising Auto-encoder	92%
5	Sparse Auto-Encoder	84%
6	CNNs	82%
7	Auto-Encoder RNN	80%

5.1. Comparison of Algorithms Based on Key Parameters

Table 6, below provides a comparison of the key algorithms in terms of their complexity, data requirements, processing time and effectiveness (filtering, prediction and noise reduction).

Table 6. Comparison of Algorithms based on Key Parameters

Algorithm	Complexity	Data Requirements	Processing Time	Effectiveness for Filtering	Effectiveness for Prediction	Effectiveness for Noise Reduction
CNN	Low to Medium	Moderate	Low	High	High	High
RNN	Medium to High	High	Medium	Medium	High	Medium
ANN	Low	Moderate	Low	Medium	Medium	Low
Sparse Auto-encoder	Medium	Low	Medium	Medium	Medium	High
De-noising Auto-encoder	Medium	Moderate	Medium	High	Medium	High
CNN-Autoencoder	Medium	Moderate	Medium	High	High	Very High

5.2. Real-world Deployments of AI in Signal Processing

AI technologies have been successfully deployed in various real-world applications within signal processing, including:

- **AI-Enhanced 5G Receivers**: These systems utilize AI algorithms for adaptive channel estimation and interference cancellation, improving the reliability and quality of wireless communication. AI enables the receivers to dynamically adapt to changing signal conditions, optimizing performance.
- **Adaptive Beamforming Using Neural Networks**: In telecommunications, neural networks are employed for adaptive beamforming, allowing antennas to focus on specific signal sources while minimizing interference from others. This approach enhances signal clarity and quality in environments with multiple signal sources.
- **Medical Imaging**: AI algorithms are increasingly being used in medical imaging to enhance signal processing tasks such as noise reduction and feature extraction from MRI or ultrasound images, resulting in better diagnosis and treatment planning.

6. Conclusions and Recommendations

Both the Autoencoders and ANN algorithms have shown high accuracies. However, there are several disadvantages using these algorithms. Firstly, their applications are still in their infancy and require further research to avoid bias and hallucinations; secondly, powerful hardware is required for their real-time execution - whilst traditional algorithms have a long-tested application history with the required hardware for their real-time execution. Another problem that is added with the time spent implementing these algorithms (through research, coding, training and testing) is the computing power they expend in comparison to traditional algorithms. However, this hardware constraint may be seen to be temporary as GPUs continue to evolve, becoming faster in their implementations of AI algorithms. In the field of SP, our results and that from the literature have shown and continue to show a lot of benefits using AI, especially in terms of: signal accuracy, detection, recognition, reconstruction and integrity. As was shown from our results exploring seven of the latest AI algorithms, these algorithms are powerful compared to traditional algorithms and are already revolutionizing the industry of SP in applications such as SDR (Software Defined Radio) and Cognitive Radio. Especially neural networks have shown a great efficiency in this field. As we see these AI algorithms are taking the place of traditional SP algorithms such as FFT or DFT in the field of SP. So, it is possible that in the future these techniques will become more sophisticated and consequently faster with evolving hardware than they are now with ubiquity.

The research in particular concluded that specially auto-encoders can be used for SP, with the CNN-Autoencoder and De-noising Encoder obtaining a minimum of 92% accuracy. Future work will involve: testing with real 6G radio communication channel data and other AI algorithms such as Deep Q Learning and Deep Temporal Difference Learning.

6.1. Future Research

To address the limitations and enhance the application of AI in signal processing, several targeted research avenues can be pursued, as discussed below.

6.1.1. Hybrid Approaches

Future research could focus on developing hybrid models that integrate traditional signal processing techniques with AI. Such models could leverage the strengths of both approaches, using AI for adaptive feature extraction while maintaining the interpretability and efficiency of traditional methods.

6.1.2. Transfer Learning

Exploring transfer learning strategies could help improve model generalizability. By training models on diverse datasets and transferring knowledge to new tasks, researchers can enhance performance in specific signal processing applications with limited labelled data.

6.1.3. Model Robustness Enhancements

Investigating techniques to enhance model robustness is critical. Research could explore adversarial training, where models are trained on intentionally perturbed data, or develop new architectures designed to be more resilient to variations in input data.

6.1.4. Explainable AI (XAI)

Developing frameworks for explainable AI in signal processing can help bridge the gap between performance and interpretability. Research in this area can focus on methods that provide insights into model decisions, making it easier for practitioners to trust and apply AI solutions in critical applications.

6.1.5. Real-time Processing Solutions

With the demand for real-time signal processing increasing, research should focus on optimizing AI models for deployment in real-time systems. This includes developing lightweight models that can run efficiently on edge devices while maintaining performance.

CRediT Author Contribution Statement

Johan Note: Conceptualization, Methodology, Formal Analysis, Investigation, Data Curation, Software, Writing - Original Draft; Maaruf Ali: Supervision, Validation, Writing - Review and Editing, Project Administration; Lekë Pepkolaj: Formal Analysis, Validation, Resources, Writing - Review and Editing.

Conflict of Interest

The research authors declare that they have no conflict of interest in this work.

Acknowledgement

We would like to acknowledge the valuable feedback provided by Dr Elton Domnori of UMT.

References

[1] William G. Bath, "Evolution and Challenges in Signal Processing", *John Hopkins APL Technical Digest*, ISSN: 1930-0530, 1988, Vol. 9, No. 3, pp. 261-268, Published by The Johns Hopkins University Applied Physics Laboratory (JHU/APL) LLC, Maryland, USA, Available: https://secwww.jhuapl.edu/techdigest/Content/techdigest/pdf/V09-N03/09-03-Bath.pdf [Accessed: 15 April 2025].

[2] Changyang She, Peng Cheng, Ang Li and Yonghui Li, "Grand Challenges in Signal Processing for Communications", *Frontiers in Signal Processing*, Electronic ISSN: 2673-8198, 12 April 2021, Vol. 1, Article No. 664331, Published by Frontiers Media SA, Lausanne, Switzerland, DOI: 10.3389/frsip.2021.664331, Available: https://doi.org/10.3389/frsip.2021.664331.

[3] Carnevale John, "Real-time Digital Signal Processing: Challenges and Solutions", *Journal of Telecommunication System and Management*, ISSN: 2167-0919, 30 March 2024, Vol. 13, No. 2, p. 428, Published by Hilaris SRL, Brussels, Belgium, DOI: 10.37421/2167-0919.2024.13.428, Available: https://www.hilarispublisher.com/open-access/realtime-digital-signal-processing-challenges-and-solutions.pdf.

[4] Sruthi Sathyamoorthy, Ugochukwu Okwudili Matthew, Temitope Samson Adekunle and Nwamaka U. Okafor, "Advances and Challenges in IoT Sensors Data Handling and Processing in Environmental Monitoring Systems", *HAFED POLY Journal of Science, Management and Technology*, eISSN: 2811-2598, print ISSN: 1597-7463, 9 May 2024, Vol. 5, No. 2, pp. 40–60, Published by African Journals Online, Makhanda, South Africa, DOI: 10.4314/hpjsmt.v5i2.3, Available: https://www.ajol.info/index.php/hpjsmt/article/view/270133/256938.

[5] Juya Zhang, Yu Guo, Xinming Dong, Tong Wang, Jinhai Wang *et al*., "Opportunities and challenges of noise interference suppression algorithms for dynamic ECG signals in wearable devices: A review", *Measurement*, ISSN: 0263-2241, 15 June 2025, Vol. 250, Article No. 117067, Published by Elsevier B.V., Amsterdam, Netherlands, DOI: 10.1016/j.measurement.2025.117067, Available: https://doi.org/10.1016/j.measurement.2025.117067.

[6] Samuel H. Russ, *Signal Integrity*, 2nd ed. Cham, Switzerland: Springer, 3 January 2022, eBook ISBN: 978-3-030-86927-4, DOI: 10.1007/978-3-030-86927-4, Available: https://link.springer.com/book/10.1007/978-3-030-86927-4.

[7] Shoab Ahmed Khan, *Digital Design of Signal Processing Systems: A Practical Approach*, 1st ed. London, UK: John Wiley & Sons Ltd., 31 January 2011, Print ISBN: 9780470741832, Online ISBN: 9780470974681, DOI: 10.1002/9780470974681, Available: https://onlinelibrary.wiley.com/doi/book/10.1002/9780470974681.

[8] Anna Richelli, Paolo Faustini, Andrea Rosa and Luigi Colalongo, "An Investigation of the Operating Principles and Power Consumption of Digital-Based Analog Amplifiers", *Journal of Low Power Electronics and Applications*, 8 September 2023, Vol. 13, No. 3, Article No. 51, Published by: MDPI (Multidisciplinary Digital Publishing Institute), Basel, Switzerland, DOI: 10.3390/jlpea13030051, Available: https://www.mdpi.com/2079-9268/13/3/51.

[9] Kadaba R. Lakshmikumar, Robert A. Hadaway and Miles A. Copeland, "Characterisation and Modeling of Mismatch in MOS Transistors for Precision Analog Design", in *IEEE Journal of Solid-State Circuits*, Electronic ISSN: 1558-173X; Print ISSN: 0018-9200, 31 December 1986, Vol. 21, No. 6, pp. 1057-1066, Published by The Institute of Electrical and Electronics Engineers (IEEE), New York City, USA, DOI: 10.1109/JSSC.1986.1052648, Available: https://ieeexplore.ieee.org/document/1052648.

[10] Paulo S.r. Diniz, Ed., *Signal Processing and Machine Learning Theory*, 1st ed. New York, USA: Academic Press, 2023, ISBN: 978-0-323-91772-8, DOI: 10.1016/C2021-0-01229-3, Available: https://www.sciencedirect.com/book/edited-volume/9780323917728/signal-processing-and-machine-learning-theory.

[11] Tao Chen and Degang Chen, "Built-in self-test and self-calibration for analog and mixed signal circuits", in *Proceedings of the 2019 IEEE International Test Conference (ITC)*, 9-15 November 2019, Washington, DC, USA, Electronic ISBN: 978-1-7281-4823-6, Print on Demand (PoD) ISBN: 978-1-7281-4824-3, Electronic ISSN: 2378-2250, PoD ISSN: 1089-3539, pp. 1-8, Published by The Institute of Electrical and Electronics Engineers (IEEE), New York City, USA, DOI: 10.1109/ITC44170.2019.9000120, Available: https://ieeexplore.ieee.org/document/9000120.

[12] Zegeye Mekasha Kidane and Waltenegus Dargie, "Cross-technology interference: detection, avoidance, and coexistence mechanisms in the ISM bands", *CCF [China Computer Federation] Transactions on Pervasive Computing and Interaction*, Electronic ISSN: 2524-5228, Print ISSN: 2524-521X, September 2025, Vol. 7, pp. 356-375, Published by Springer Nature, Berlin, Germany, DOI: 10.1007/s42486-025-00185-0, Available: https://link.springer.com/article/10.1007/s42486-025-00185-0.

[13] Lalitha Sankar, Wade Trappe, Kannan Ramchandran, H. Vincent Poor and Merouane Debbah, "The Role of Signal Processing in Meeting Privacy Challenges: An Overview", in *IEEE Signal Processing Magazine*, Print ISSN: 1053-5888, Electronic ISSN: 1558-0792, September 2013, Vol. 30, No. 5, pp. 95-106, Published by The Institute of Electrical and Electronics Engineers (IEEE), New York City, USA, DOI: 10.1109/MSP.2013.2264541, Available: https://ieeexplore.ieee.org/document/6582707.

[14] Mir Mahdi Safari and Jafar Pourrostam, "The role of analog signal processing in upcoming telecommunication systems: Concept, challenges, and outlook", *Signal Processing*, Online ISSN: 1872-7557, Print ISSN: 0165-1684, July 2024, Vol. 220, Article No. 109446, Published by Published by Elsevier B.V., Amsterdam, Netherlands, DOI: 10.1016/j.sigpro.2024.109446, Available: https://www.sciencedirect.com/science/article/abs/pii/S0165168424000653.

[15] Cristina Despina-Stoian, Angela Digulescu-Popescu, Serbanescu Alexandra, Roua Youssef and Emanuel Radoi, "Comparison of Adaptive Filtering Strategies for Self-Interference Cancellation in LTE Communication Systems", in *Proceedings of the 2020 13th International Conference on Communications (COMM)*, 18-20 June 2020, Bucharest, Romania, Electronic ISBN: 978-1-7281-5611-8, USB ISBN: 978-1-7281-5610-1, Print on Demand (PoD) ISBN: 978-1-7281-5612-5, pp. 73-76, Published by The Institute of Electrical and Electronics Engineers (IEEE), New York City, USA, DOI: 10.1109/COMM48946.2020.9141965, Available: https://ieeexplore.ieee.org/document/9141965.

[16] Neelam Singh and Sumit J. Darak, "Enhancing Wireless PHY With Adaptive OFDM and Multiarmed Bandit Learning on Zynq System-on-Chip", in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Print ISSN: 1063-8210, Electronic ISSN: 1557-9999, Vol. 33, No. 6, June 2025, pp. 1651-1664, Published by The Institute of Electrical and Electronics Engineers (IEEE), New York City, USA, DOI: 10.1109/TVLSI.2025.3528865, Available: https://dl.acm.org/doi/10.1109/TVLSI.2025.3528865.

[17] Amarender Reddy Byreddy and Edeswaran Logashanmugam, "Energy and spectral efficiency improvement using improved shark smell-coyote optimization for massive MIMO system", *International Journal of Communication Systems*, eISSN: 1099-1131, January 2023, Vol. 36, No. 2, Article No. e5381, Published by John Wiley and Sons, Ltd., West Sussex, UK, DOI: 10.1002/dac.5381, Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/dac.5381.

[18] Xuechao Fan, "Intelligent Radio Spectrum Management and Dynamic Spectrum Allocation Algorithm", in *Proceedings of the 2024 International Conference on Power, Electrical Engineering, Electronics and Control (PEEEC)*, 14-16 August 2024, Athens, Greece, Electronic ISBN: 979-8-3503-7891-7, Print on Demand (PoD) ISBN: 979-8-3503-7892-4, pp. 836-841, Published by The Institute of Electrical and Electronics Engineers (IEEE), New York City, USA, DOI: 10.1109/PEEEC63877.2024.00156, Available: https://ieeexplore.ieee.org/document/10836105.

[19] Jianfeng Liu, Xin-Lin Huang, Fei Hu and Cheng Li, "Adaptive Compressive Spectrum Sensing Using a Deterministic Estimation Model for Wideband Cognitive Radios", in *IEEE Transactions on Wireless Communications*, e-ISSN: 1558-2248, Vol. 24, No. 11, pp. 9099-9111, Published by: Published by The Institute of Electrical and Electronics Engineers (IEEE), New York City, USA, DOI: 10.1109/TWC.2025.3570760, Available: https://ieeexplore.ieee.org/document/11014594.

[20] Nayuta Yamane, Jin Nakazato, Koki Ito, Manabu Mikami, Takahiro Tsuchiya *et al*., "Low Latency Redundant Network Architecture for Enhancing 5G Mobile Communication Quality", in *Proceedings of the 2025 International Conference on Information Networking (ICOIN)*, 15-17 January 2025, Chiang Mai, Thailand, Electronic ISBN: 979-8-3315-0694-0, Print on Demand (PoD) ISBN: 979-8-3315-0695-7, Electronic ISSN: 2996-1580, Print on Demand (PoD) ISSN: 2996-1572, pp. 88-92, Published by The Institute of Electrical and Electronics Engineers (IEEE), New York City, USA, DOI: 10.1109/ICOIN63865.2025.10993150, Available: https://ieeexplore.ieee.org/document/10993150.

[21] K. Deepthi, Dillip Narayan Sahu, Kuldeep Singh Rana, Asha V, Ippa Sumalatha *et al*., Analyzing the Effect of Edge Computing on Real-time Data Processing and Latency Reduction", in *Proceedings of the 2023 10th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, 1-3 December 2023, Gautam Buddha Nagar, India, ISBN: 979-8-3503-8247-1, Print on Demand (PoD) ISBN: 979-8-3503-8248-8, Electronic ISSN: 2687-7767, Print on Demand (PoD) ISSN: 2687-7759, pp. 730-734, Published by The Institute of Electrical and Electronics Engineers (IEEE), New York City, USA, DOI: 10.1109/UPCON59197.2023.10434297, Available: https://ieeexplore.ieee.org/document/10434297.

[22] Kithmini Weththasinghe, Beeshanga Jayawickrama and Ying He, "Machine Learning-Based Channel Estimation for 5G New Radio", in *IEEE Wireless Communications Letters*, Print ISSN: 2162-2337, Electronic ISSN: 2162-2345, April 2024, Vol. 13, No. 4, pp. 1133-1137, Published by The Institute of Electrical and Electronics Engineers (IEEE), New York City, USA, DOI: 10.1109/LWC.2024.3362963, Available: https://ieeexplore.ieee.org/document/10423294.

[23] Liu Yang, Dan Ding, Zhicheng Shen, Liming Hu and Hao Chen, "Parallel Combinatory Spread Spectrum System Based on Single Carrier Frequency Domain Equalization in Multipath Fading Channels", in *Proceedings of the 2023 8th International Conference on Communication, Image and Signal Processing (CCISP)*, 17-19 November 2023, Chengdu, China, Electronic ISBN: 979-8-3503-0583-8, Print on Demand (PoD) ISBN: 979-8-3503-0584-5, pp. 528-533, Published by The Institute of Electrical and Electronics Engineers (IEEE), New York City, USA, DOI: 10.1109/CCISP59915.2023.10355813, Available: https://ieeexplore.ieee.org/document/10355813.

[24] Muhammad Yasir Faheem, Muhammad Basit Azeem, Abid Ali Minhas, Shun'an Zhong and Xinghua Wang, "Key Techniques of Ultra-Low-Power ADC and Miniaturized RF Transceiver Circuits for 4G/LTE Applications", *Microelectronics International*, ISSN: 1356-5362, EISSN: 1758-812X, 23 March 2022, Vol. 39, No. 4, pp. 152-165, Published by Emerald Publishing Ltd., Leeds, England, UK, DOI: 10.1108/MI-06-2021-0054, Available: https://www.emerald.com/mi/article-abstract/39/4/152/452578/Key-techniques-of-ultra-low-power-ADC-and.

[25] Qingqing Wu, Yufei Wang, Di Zhai, Yang Lu, Cheng Zhong *et al*., "A low-power communication strategy for terminal sensors in power status monitoring", *Sensors*, ISSN: 1424-8220, 21 February, 2025, Vol. 25, No. 5, Article

1317, Published by: MDPI (Multidisciplinary Digital Publishing Institute), Basel, Switzerland, DOI: 10.3390/s25051317, Available: https://www.mdpi.com/1424-8220/25/5/1317.

[26] P. Deepanramkumar, A. Helen Sharmila, Niranchana Radhakrishnan, Devulapalli Sudheer, Jeethu V. Devasia *et al*., "Enhancing Connectivity in Rural Areas: Secure Spectrum Access in 6G Networks Using Advanced Encryption and Spectrum Sensing Techniques", in *IT Professional*, ISSN: 1520-9202, July-August 2024, Vol. 26, No. 4, pp. 22-28, Published by The Institute of Electrical and Electronics Engineers (IEEE), USA, DOI: 10.1109/MITP.2024.3433402, Available: https://www.computer.org/csdl/magazine/it/2024/04/10693585/20wD9Jm0hGg.

[27] Junjie Zeng, Xiangli Liu and Zan Li, "Radio Anomaly Detection Based on Improved Denoising Diffusion Probabilistic Models", *IEEE Communication Letters*, Print ISSN: 1089-7798; Electronic ISSN: 1558-2558; CD: 2373-7891, 28 June 2023, Vol. 27, No. 8, pp. 1979-1983, Published by The Institute of Electrical and Electronics Engineers (IEEE), USA, DOI: 10.1109/LCOMM.2023.3290813, Available: https://ieeexplore.ieee.org/document/10168113.

[28] Chin Tsai and Melody Moh, "Load balancing in 5G cloud radio access networks supporting IoT communications for smart communities", in *Proceedings of the 2017 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, 18-20 December 2017, Bilbao, Spain, Electronic ISBN: 978-1-5386-4662-5, PoD ISBN: 978-1-5386-4663-2, pp. 259-264, Published by The Institute of Electrical and Electronics Engineers (IEEE), New York City, USA, DOI: 10.1109/ISSPIT.2017.8388652, Available: https://ieeexplore.ieee.org/document/8388652.

[29] Navneet Kaur and Lav Gupta, "Enhancing IoT Security in 6G Environment With Transparent AI: Leveraging XGBoost, SHAP and LIME", in *Proceedings of the 2024 IEEE 10th International Conference on Network Softwarization (NetSoft)*, 24-28 June 2024, Saint Louis, MO, USA, Electronic ISBN: 979-8-3503-6958-8, Print on Demand (PoD) ISBN: 979-8-3503-6959-5, Electronic ISSN: 2693-9789, Print on Demand (PoD) ISSN: 2693-9770, pp. 180-184, Published by The Institute of Electrical and Electronics Engineers (IEEE), New York City, USA, DOI: 10.1109/NetSoft60951.2024.10588922, Available: https://ieeexplore.ieee.org/document/10588922.

[30] Donghang Wu, Xihong Wu and Tianshu Qu, "A Hybrid Deep-Online Learning Based Method for Active Noise Control in Wave Domain", in *Proceedings of the ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 14-19 April 2024, Seoul, Republic of Korea, ISBN: 979-8-3503-4485-1, Print on Demand (PoD) ISBN: 979-8-3503-4486-8, Electronic ISSN: 2379-190X, Print on Demand (PoD) ISSN: 1520-6149, pp. 1301-1305, Published by The Institute of Electrical and Electronics Engineers (IEEE), New York City, USA, DOI: 10.1109/ICASSP48485.2024.10446791, Available: https://ieeexplore.ieee.org/document/10446791.

[31] Taiwo Oyedare, Vijay K. Shah, Daniel J. Jakubisin and Jeffrey H. Reed, "Interference Suppression Using Deep Learning: Current Approaches and Open Challenges", in *IEEE Access*, eISSN: 2169-3536, 21 June 2022, Vol. 10, pp. 66238-66266, Published by The Institute of Electrical and Electronics Engineers (IEEE), New York City, USA, DOI: 10.1109/ACCESS.2022.3185124, Available: https://ieeexplore.ieee.org/document/9802083.

[32] Raju Hazari, Gaurav Singh, Devika Renjith, Divya Krishnan, Pavanitha B. *et al*., "Spec-SCAN: Spectrum Learning in Shared Channel using Neural Networks", in *Proceedings of the 2025 IEEE 22nd Consumer Communications & Networking Conference (CCNC)*, 10-13 January 2025, Las Vegas, NV, USA, Electronic ISBN: 979-8-3315-0805-0, Print on Demand (PoD) ISBN: 979-8-3315-0806-7, Electronic ISSN: 2331-9860, Print on Demand (PoD) ISSN: 2331-9852, pp. 1-6, Published by The Institute of Electrical and Electronics Engineers (IEEE), New York City, USA, DOI: 10.1109/CCNC54725.2025.10976013, Available: https://ieeexplore.ieee.org/document/10976013.

[33] E. Iradier, I. Bilbao, J. Montalban, P. Angueira, S-I. Park *et al*., "Improving ATSC 3.0 Channel Estimation using AI Techniques", in *Proceedings of the 2024 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, 19-21 June 2024, Toronto, Ontario, Canada, Electronic ISBN: 979-8-3503-6426-2, Print on Demand (PoD) ISBN: 979-8-3503-6427-9, Electronic ISSN: 2155-5052, Print on Demand (PoD) ISSN: 2155-5044, pp. 1-6, Published by The Institute of Electrical and Electronics Engineers (IEEE), New York City, USA, DOI: 10.1109/BMSB62888.2024.10608264, Available: https://ieeexplore.ieee.org/document/10608264?signout=success.

[34] Humam Hussein, Nouf Abd Elmunim, Ferhat Atasoy, Bal S. Virdee, Taha A. Elwi *et al*., "A Novel MIMO Antenna Integrated With a Solar Panel and Employing AI-Equalization for 5G Wireless Communication Networks", in *IEEE Access*, eISSN: 2169-3536, 12 August 2024, Vol. 12, pp. 114382-114393, Published by The Institute of Electrical and Electronics Engineers (IEEE), New York City, USA, DOI: 10.1109/ACCESS.2024.3441830, Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10633711.

[35] Matthew McCaskey, Austin Feydt, Robert Corrigan, Kul Bhasin and David Chelmins, "Implementation of a Machine Learning Based Modulation Scheme in GNURadio for Over-the-Air Packet Communications", in *Proceedings of the 36th International Satellite Communications Systems Conference (ICSSC 2018)*, 15-18 October 2018, Niagra Falls, Ontario, Canada, ISBN: 978-1-83953-065-4, e-ISBN: 978-1-83953-065-4, pp. 1-7, Published by The Institution of Engineering and Technology (IET), Stevenage, UK, DOI: 10.1049/cp.2018.1712, Available: https://digital-library.theiet.org/doi/10.1049/cp.2018.1712.

[36] Usman Iqbal, "AI Techniques for Enhancing Latency Reduction in Distributed Data Pipeline Systems", *AITOZ Multi-Disciplinary Review (AMR)*, 7 June 2023, Vol. 2, No. 1, pp. 216-223, Published by AITOZ Research Publications, Available: https://aitozresearch.com/index.php/amr/article/view/93/87 [Accessed: 28 November 2025].

[37] Tianyu Wang, Jinyang Guo, Bowen Zhang, Ge Yang and Dong Li, "Deploying AI on Edge: Advancement and Challenges in Edge Intelligence", *Mathematics*, ISSN: 2227-7390, 4 June 2025, Vol. 13, No. 11, Article ID. 1878, Published by: MDPI (Multidisciplinary Digital Publishing Institute), Basel, Switzerland, DOI: 10.3390/math13111878 DOI: Available: https://www.mdpi.com/2227-7390/13/11/1878.
[38] K. Shyam Sunder Reddy, M. Manohara, K. Shailaja, P. Revathy, Thota Mahesh Kumar *et al*., "Power management using AI-based IOT systems", *Measurement: Sensors*, ISSN: 2665-9174, December 2022, Vol. 24, Article No. 100551, Published by Elsevier B.V., Amsterdam, Netherlands, DOI: 10.1016/j.measen.2022.100551, Available: https://www.sciencedirect.com/science/article/pii/S2665917422001854.
[39] Waseem Raza, Muhammad Umar Bin Farooq, Aneeqa Ijaz, Marvin Manalastas and Ali Imran, "AI-Powered Resilience: A Dual-Approach for Outage Management in Dense Cellular Networks", *Computer Communications*, ISSN: 0140-3664, April 2025, Vol. 236, Article No. 108129, Published by Elsevier B.V., Amsterdam, DOI: 10.1016/j.comcom.2025.108129, Available: https://dl.acm.org/doi/10.1016/j.comcom.2025.108129.
[40] Haider W. Oleiwi, Doaa N. Mhawi and Hamed Al-Raweshidy, "MLTs-ADCNs: Machine Learning Techniques for Anomaly Detection in Communication Networks", in *IEEE Access*, Electronic ISSN: 2169-3536, 26 August 2022, Vol. 10, pp. 91006-91017, Published by The Institute of Electrical and Electronics Engineers (IEEE), New York City, USA, DOI: 10.1109/ACCESS.2022.3201869, Available: https://ieeexplore.ieee.org/document/9867983.
[41] Rajashekhar Reddy Kethireddy, "AI-Driven Encryption Techniques for Data Security in Cloud Computing", *Journal of Recent Trends in Computer Science and Engineering (JRTCSE)*, ISSN: 2322-0872, Jan-June 2021, Vol. 9, No. 1, pp. 27-38, Published by TechMan Script (Publications), Tamilnadu, India, DOI: 10.70589/JRTCSE.2021.1.3, Available: https://jrtcse.com/index.php/home/article/view/JRTCSE.2021.1.3/JRTCSE.2021.1.3.
[42] Mengting Lin and Youping Zhao, "Artificial intelligence-empowered resource management for future wireless communications: A survey", in *China Communications*, Print ISSN: 1673-5447, March 2020, Vol. 17, No. 3, pp. 58-77, Published by China Communications Magazine Co. Ltd., Beijing, China, DOI: 10.23919/JCC.2020.03.006, Available: https://ieeexplore.ieee.org/document/9058606.
[43] Ibrahim Aqeel, Ibrahim Mohsen Khormi, Surbhi Bhatia Khan, Mohammed Shuaib, Ahlam Almusharraf *et al*., "Load Balancing Using Artificial Intelligence for Cloud-Enabled Internet of Everything in Healthcare Domain", *Sensors*, ISSN: 1424-8220, 5 June 2023, Vol. 23, No. 11, Article No. 5349, Published by: MDPI (Multidisciplinary Digital Publishing Institute), Basel, Switzerland, DOI: 10.3390/s23115349, Available: https://www.mdpi.com/1424-8220/23/11/5349.
[44] Jieke Lin, Wanyu Wang, Longxiang Yin and Yinhe Han, "KAITIAN: Communication for Heterogeneous Accelerators in Embodied AI", in *Proceedings of the 2025 Workshop on Architecture Support for Embodied AI Systems (Arch4EAI)*, 21 June 2025, Tokyo, Japan, Available: https://arch4eai.github.io/file/ISCA25_workshop_JiekeLin_KaiTian.pdf [Accessed: 31 December 2025].
[45] Seyed Mohammad Mehdi Sajadieh and Sang Do Noh, "From Simulation to Autonomy: Reviews of the Integration of Artificial Intelligence and Digital Twins", *International Journal of Precision Engineering and Manufacturing-Green Technology*, Electronic ISSN: 2198-0810, Print ISSN: 2288-6206, September 2025, Vol. 12, pp. 1597-1628, Published by Springer Nature, Berlin, Germany, DOI: 10.1007/s40684-025-00750-z, Available: https://link.springer.com/content/pdf/10.1007/s40684-025-00750-z.pdf.
[46] Baolong Guo and Juanjuan Zhu, *Signals and Systems*, 1st ed. Boston, USA: Walter de Gruyter, 24 September 2018, ISBN-13: 978-3110595413, DOI: 10.1515/9783110593907, Available: https://www.degruyterbrill.com/document/doi/10.1515/9783110593907/html.
[47] Lorenzo Mucchi, Shahriar Shahabuddin, Mahmoud A.M. Albreem, Saeed Abdallah, Stefano Caputo *et al*., "Signal Processing Techniques for 6G", *Journal of Signal Processing Systems*, E-ISSN: 1939-8115; P-ISSN: 1939-8018, 2 February 2023, Vol. 95, pp. 435-457, Published by Springer Nature, Berlin, Germany, DOI: 10.1007/s11265-022-01827-7, Available: https://link.springer.com/content/pdf/10.1007/s11265-022-01827-7.pdf.
[48] Adnan Rashid and Osman Hasan, "Formal analysis of continuous-time systems using Fourier transform", *Journal of Symbolic Computation*, Online ISSN: 1095-855X; Print ISSN: 0747-7171, January-February 2019, Vol. 90, pp. 65–88, Published by Elsevier B.V., Amsterdam, Netherlands, DOI: 10.1016/j.jsc.2018.04.004, Available: https://www.sciencedirect.com/science/article/pii/S0747717118300336.
[49] Gerling Plonka, Daniel Potts, Gabriele Steidl and Manfred Tasche, "Fast Fourier Transforms", In: *Numerical Fourier Analysis. Applied and Numerical Harmonic Analysis*, 2nd ed. Cham, Switzerland: Birkhäuser, 8 November 2023, Online ISBN: 978-3-031-35005-4, Print ISBN: 978-3-031-35004-7, Chapter 5, pp. 265-337, DOI: 10.1007/978-3-031-35005-4_5, Available: https://link.springer.com/chapter/10.1007/978-3-031-35005-4_5.
[50] Agilent Technologies, *Agilent AN 1298 Digital Modulation in Communications Systems — An Introduction Application Note*, 1st ed. California, USA: Agilent Technologies, June, 2000, pp. 1-48, 2000.5965-7160E, Available: https:// https://www.csd.uoc.gr/~hy530/DigitalModulation.pdf [Accessed: 28 November 2025].
[51] Julius Ssimbwa, Byungju Lim, Ju-Hyung Lee and Young-Chai Ko, "A Survey on Robust Modulation Requirements for the Next Generation Personal Satellite Communications", *Frontiers in Communications and Networks*, ISSN: 2673-530X, 17 May 2022, Vol. 3, Article No. 850781, Published by Frontiers Media SA, Lausanne, Switzerland, DOI:

10.3389/frcmn.2022.850781, Available: https://www.frontiersin.org/journals/communications-and-networks/articles/10.3389/frcmn.2022.850781.

[52] Fuqin Xiong, "Modem techniques in satellite communications", in *IEEE Communications Magazine*, Print ISSN: 0163-6804, Electronic ISSN: 1558-1896, 31 August 1994, Vol. 32, No. 8, pp. 84-98, Published by The Institute of Electrical and Electronics Engineers (IEEE), New York City, USA, DOI: 10.1109/35.299842, Available: https://ieeexplore.ieee.org/document/299842.

[53] Dipankar Dasgupta, Zahid Akhtar and Sajib Sen, "Machine learning in cybersecurity: a comprehensive survey", *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, ISSN: 1548-5129; Online ISSN: 1557-380X, 19 September 2020, Vol. 19, No. 1, pp. 57-106, Published by Sage Publishing, California, USA, DOI: 10.1177/1548512920951275, Available: https://journals.sagepub.com/doi/10.1177/1548512920951275.

[54] Nilanjan Saha, Aleena Swetapadma and Mahadev Mondal, "A Brief Review on Artificial Neural Network: Network Structures and Applications", in *Proceedings of the 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 17-18 March 2023, Coimbatore, India, Electronic ISBN: 979-8-3503-9737-6, DVD ISBN: 979-8-3503-9735-2, Print on Demand (PoD) ISBN: 979-8-3503-9738-3, Electronic ISSN: 2575-7288, Print on Demand (PoD) ISSN: 2469-5556, pp. 1974-1979, Published by The Institute of Electrical and Electronics Engineers (IEEE), New York City, USA, DOI: 10.1109/ICACCS57279.2023.10112753, Available: https://ieeexplore.ieee.org/document/10112753.

[55] Laith Alzubaidi, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili, Ye Duan *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions", *Journal of Big Data*, 31 March 2021, Electronic ISSN: 2196-1115, Vol. 8, Article No. 53, Published by Springer Nature, Berlin, Germany, DOI: 10.1186/s40537-021-00444-8, Available: https://link.springer.com/article/10.1186/s40537-021-00444-8.

[56] Fathi M. Salem, *Recurrent Neural Networks From Simple to Gated Architectures*, 1st ed. Cham, Switzerland: Springer, 5 January 2023, ISBN: 978-3-030-89931-8, DOI: 10.1007/978-3-030-89929-5, Available: https://link.springer.com/book/10.1007/978-3-030-89929-5.

[57] Kamal Berahmand, Fatemeh Daneshfar, Elaheh Sadat Salehi, Yuefeng Li and Yue Xu, "Autoencoders and their applications in machine learning: a survey", *Artificial Intelligence Review*, Electronic ISSN: 1573-7462, Print ISSN: 0269-2821, 3 February 2024, Vol. 57, Article No. 28, Published by Springer Nature, Berlin, Germany, DOI: 10.1007/s10462-023-10662-6, Available: https://link.springer.com/content/pdf/10.1007/s10462-023-10662-6.pdf.

[58] Ashish Kumar Shakya, Gopinatha Pillai and Sohom Chakrabarty, "Reinforcement learning algorithms: A brief survey", *Expert Systems with Applications*, ISSN: 0957-4174, 30 November 2023, Vol. 231, No. C, Article No. 120495, Published by Elsevier B.V., Amsterdam, Netherlands, DOI: 10.1016/j.eswa.2023.120495, Available: https://www.sciencedirect.com/science/article/abs/pii/S0957417423009971.

[59] Vinod K. Mishra, *An Introduction to Deep Reinforcement Learning*, 1st ed. New York, USA: Chapman and Hall/CRC, 17 December 2025, eBook ISBN: 9781032659800, DOI: 10.1201/9781032659800. Available: https://www.taylorfrancis.com/books/mono/10.1201/9781032659800/introduction-deep-reinforcement-learning-vinod-mishra.

[60] Aske Plaat, *Deep Reinforcement Learning*, 1st ed. Singapore: Springer, 12 June 2022, Softcover ISBN: 978-981-19-0637-4, eBook ISBN: 978-981-19-0638-1, DOI: 10.1007/978-981-19-0638-1, Available: https://link.springer.com/book/10.1007/978-981-19-0638-1.

[61] Hado van Hasselt, Arthur Guez and David Silver, "Deep Reinforcement Learning With Double Q-Learning", *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI'16)*, 12-17 February 2016, Phoenix, Arizona, USA, ISSN: 2374-3468, Vol. 30, No. 1, pp. 2094-2100, Available: https://ojs.aaai.org/index.php/AAAI/article/view/10295/10154 [Accessed: 22 June 2025].

[62] Chenhao Jin, Shinae Jang, Xiaorong Sun, Jingcheng Li and Richard Christenson, "Damage detection of a highway bridge under severe temperature changes using extended Kalman filter trained neural network", *Journal of Civil Structural Health Monitoring*, Print ISSN: 2190-5452; Online ISSN: 2190-5479, 9 May 2016, Vol. 6, No. 3, pp. 545-560, Published by Springer Nature, Berlin, Germany, DOI: 10.1007/s13349-016-0173-8, Available: https://link.springer.com/article/10.1007/s13349-016-0173-8.

[63] Abien Fred Agarap, "Deep Learning using Rectified Linear Units (ReLU)", arXiv:1803.08375v2 [cs.NE], 7th February, 2019. DOI: 10.48550/arXiv.1803.08375, Available: https://arxiv.org/pdf/1803.08375.

[64] Yunji Chen, Ling Li, Wei Li, Qi Guo, Zidong Du *et al.*, "Fundamentals of neural networks", in AI Computing Systems An Application-Driven Perspective, San Francisco, California, USA: Morgan Kaufmann, 2023, ISBN: 978-0-323-95399-3, pp. 17-51. DOI: 10.1016/B978-0-32-395399-3.00008-1, Available: https://www.sciencedirect.com/science/chapter/monograph/abs/pii/B9780323953993000081.

[65] Bing Xu, Naiyan Wang, Tianqi Chen and Mu Li, "Empirical Evaluation of Rectified Activations in Convolutional Network", in *Proceedings of the Deep Learning Workshop, 2025 International Conference on Machine Learning (ICML '15)*, 6-11 July 2015, Lille, France, DOI: 10.48550/arXiv.1505.00853, Available: https://arxiv.org/abs/1505.00853.

[66] Jiuxiang Gu, Chenyang Li, Yingyu Liang, Zhenmei Shi and Zhao Song, "Exploring the Frontiers of Softmax: Provable Optimization, Applications in Diffusion Model, and Beyond", arXiv:2405.03251v1 [cs.LG], 6 May 2024, DOI: 10.48550/arXiv.2405.03251, Available: https://arxiv.org/abs/2405.03251.

[67] Arush Saxena, "An Introduction to Convolutional Neural Networks", *International Journal of Research in Applied Science & Engineering Technology (IJRASET)*, ISSN: 2321-9653, December 2022, Vol. 10, No. 12, pp. 943–947, Published by Sonepat, India, DOI: 10.22214/ijraset.2022.47789, Available: https://www.ijraset.com/best-journal/convolution-neural-network.

[68] Jane Oruh, Serestina Viriri and Adekanmi Adegun, "Long Short-Term Memory Recurrent Neural Network for Automatic Speech Recognition", in *IEEE Access*, Electronic ISSN: 2169-3536, 14 March 2022, Vol. 10, pp. 30069-30079, Published by The Institute of Electrical and Electronics Engineers (IEEE), New York City, USA, DOI: 10.1109/ACCESS.2022.3159339, Available: https://ieeexplore.ieee.org/document/9733937.

[69] Ibomoiye Domor Mienye, Theo G. Swart and George Obaido, "Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications", *Information*, ISSN: 2078-2489, 25 August 2024, Vol. 15, No. 9, Article No. 517, Published by: MDPI (Multidisciplinary Digital Publishing Institute), Basel, Switzerland, DOI: 10.3390/info15090517, Available: https://www.mdpi.com/2078-2489/15/9/517.

[70] Meijian Guan, Samuel Cho, Robin Petro, Wei Zhang, Boris Pasche *et al*., "Natural language processing and recurrent network models for identifying genomic mutation-associated cancer treatment change from patient progress notes", *Journal of the American Medical Informatics Association open* (*JAMIA open)*, Online ISSN: 2574-2531, April 2019, Vol. 2, No. 1, pp. 139-149, Published by the Oxford University Press, Oxford, UK, DOI: 10.1093/jamiaopen/ooy061, Available: https://academic.oup.com/jamiaopen/article/2/1/139/5272605.

[71] Sepp Hochreiter and Jürgen Schmidhuber, "Long Short-Term Memory", in *Neural Computation*, ISSN: 0899-7667, EISSN: 1530-888X, 15 November 1997, Vol. 9, No. 8, pp. 1735-1780, Published by MIT Press Direct, Massachusetts, USA, DOI: 10.1162/neco.1997.9.8.1735, Available: https://direct.mit.edu/neco/article-abstract/9/8/1735/6109/Long-Short-Term-Memory.

[72] K.E. ArunKumar, Dinesh V. Kalaga, Ch. Mohan Sai Kumar, Masahiro Kawaji and Timothy M. Brenza, "Comparative analysis of Gated Recurrent Units (GRU), long Short-Term memory (LSTM) cells, autoregressive Integrated moving average (ARIMA), seasonal autoregressive Integrated moving average (SARIMA) for forecasting COVID-19 trends", *Alexandria Engineering Journal*, Online ISSN: 2090-2670, Print ISSN: 1110-0168, October 2022, Vol. 61, No. 10, pp. 7585-7603, Publisher: Alexandria Amsterdam Faculty of Engineering, Alexandria University Elsevier, Egypt, DOI: 10.1016/j.aej.2022.01.011, Available: https://www.sciencedirect.com/science/article/pii/S1110016822000138.

[73] Delip Rao and Brian McMahan, *Natural Language Processing with Pytorch: Build Intelligent Language Applications Using Deep Learning*, 1st ed. Sebastopol, California, USA: O'Reilly Media, Inc., 5 February 2019. ISBN-13: 978-1491978238, Available: https://www.oreilly.com/library/view/natural-language-processing/9781491978221/.

[74] Salmela, L., Tsipinakis, N., Foi, A., Billet, C., Dudley, J.M. and Gentry, G., "Predicting ultrafast nonlinear dynamics in fibre optics with a recurrent neural network", *Nature Machine Intelligence*, EISSN: 2522-5839, 18 April, 2021, Vol. 3, pp. 344–354, Published by Springer Nature, Berlin, Germany, DOI: 10.1038/s42256-021-00297-z, Available: https://www.nature.com/articles/s42256-021-00297-z.

[75] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley *et al*., "Generative adversarial networks", *Communications of the ACM*, ISSN: 0001-0782; EISSN: 1557-7317, November 2020, Vol. 63, No. 11, pp. 139-144, Published by Association for Computing Machinery, New York, NY, United States, DOI: 10.1145/3422622, Available: https://dl.acm.org/doi/10.1145/3422622.

[76] Joe Benton, Generative Models: Theory and Applications, PhD thesis, September 2023, University of Oxford, UK, Available: https://ora.ox.ac.uk/objects/uuid:18001411-29e0-4f92-9245-f0c6d427dc2b/files/dxg94hq178 [Accessed: 19 October 2025].

[77] Maria Nareklishvili, Nick Polson and Vadim Sokolov, "Generative Modeling: A Review", arXiv:2501.05458v2 [stat.CO], 16 May 2025. DOI: 10.48550/arXiv.2501.05458, Available: https://arxiv.org/abs/2501.05458.

[78] Smit Gandhi, Hemit Rana and Nikita Bhatt, "Conditional GANs in Image-to-Image Translation: Improving Accuracy and Contextual Relevance in Diverse Datasets", *Procedia Computer Science*, ISSN: 1877-0509, 1 January 2025, Vol. 252, No. C, pp. 954-963, Published by Elsevier Science Publishers B. V., Amsterdam, Netherlands, DOI: 10.1016/j.procs.2025.01.056, Available: https://dl.acm.org/doi/10.1016/j.procs.2025.01.056.

[79] Michael Bacharach, "Zero-sum Games", in: John Eatwell, Murray Milgate and Peter Newman, Eds., *The World of Economics*, London, UK: The New Palgrave, Palgrave Macmillan, 13 May 1991, Softcover ISBN: 978-0-333-55177-6, eBook ISBN978-1-349-21315-3, Series ISSN 2946-9805, Series E-ISSN 2946-9813, pp. 721-731, DOI: 10.1007/978-1-349-21315-3_100, Available: https://link.springer.com/chapter/10.1007/978-1-349-21315-3_100.

[80] Ayman A. Aboalndr, Mohamed M.Abbassy, Waleed M.Ead, Abdellah I. Elfeky and Amr E. El-Shora, "A New Approach to Improving Photorealistic Image Quality Using Generative Adversarial Networks (GANs)", *Journal of Posthumanism*, Print ISSN: 2634-3576; Online ISSN: 2634-3584, 30 December 2024, Vol. 4, No. 3, pp. 301–324,

Published by Transnational Press London Ltd., Sale, UK, DOI: 10.63332/joph.v4i3.987, Available: https://posthumanism.co.uk/jp/article/view/987/2006.
[81] Yasi Wang, Hongxun Yao and Sicheng Zhao, "Auto-encoder based dimensionality reduction", *Neurocomputing*, Print ISSN: 0925-2312, Online ISSN: 1872-8286, 5 April 2016, Vol. 184, pp. 232-242, Published by Elsevier B.V., Amsterdam, Netherlands, DOI: 10.1016/j.neucom.2015.08.104, Available: https://www.sciencedirect.com/science/article/abs/pii/S0925231215017671.
[82] Sunhee Kim, Sang-Ho Chu, Yong-Jin Park and Chang-Yong Lee, "A tied-weight autoencoder for the linear dimensionality reduction of sample data", *Scientific Reports*, Online ISSN: 2045-2322, 5 November 2024, Vol. 14, Article No. 26801, Published by Springer Nature, Berlin, Germany, DOI: 10.1038/s41598-024-77080-8, Available: https://www.nature.com/articles/s41598-024-77080-8.
[83] Kevin Kögler, Alexander Shevchenko, Hamed Hassani and Marco Mondelli, "Compression of structured data with autoencoders: provable benefit of nonlinearities and depth", in *Proceedings of the 41st International Conference on Machine Learning (ICML'24)*, 21-27 July 2024, Vienna, Austria, Vol. 235, JMLR.org, Article 1000, pp. 24964–25015, Available: https://raw.githubusercontent.com/mlresearch/v235/main/assets/kogler24a/kogler24a.pdf [Accessed: 23 June 2025].
[84] Walter Hugo Lopez Pinaya, Sandra Vieira, Rafael Garcia-Dias and Andrea Mechelli, "Autoencoders", in: Andrea Michelli and Sandra Vieira, Eds., *Machine Learning: Methods and Applications to Brain Disorder*, London, UK: Academic Press, 19 November 2019, ISBN: 978-0-12-8157398, pp. 193–208, DOI: 10.1016/B978-0-12-815739-8.00011-0, Available: https://www.sciencedirect.com/science/chapter/edited-volume/abs/pii/B9780128157398000110.
[85] Komal Bajaj, Dushyant Kumar Singh and Mohd. Aquib Ansari, "Autoencoders Based Deep Learner for Image Denoising", *Procedia Computer Science*, ISSN: 1877-0509, 2020, Vol. 171, pp. 1535-1541, Published by Elsevier Science Publishers B. V., Amsterdam, Netherlands, DOI: 10.1016/j.procs.2020.04.164, Available: https://www.sciencedirect.com/science/article/pii/S1877050920311431.
[86] Yuxiao Li, Eric J. Michaud, David D. Baek, Joshua Engels, Xiaoqing Sun *et al.*, "The Geometry of Concepts: Sparse Autoencoder Feature Structure", *Entropy*, ISSN: 1099-4300, 27 March 2025, Vol. 27, No. 4, Article No. 344, Published by MDPI (Multidisciplinary Digital Publishing Institute), Basel, Switzerland, DOI: 10.3390/e27040344, Available: https://www.mdpi.com/1099-4300/27/4/344.
[87] Aurélien Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and Tensorflow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 3rd ed. Sebastopol, California, USA: O'Reilly Media Inc., 8 November 2022, ISBN-13: 978-1098125974, Available: https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/.
[88] Emmanuel Pintelas, Ioannis E. Livieris and Panagiotis E. Pintelas, "A Convolutional Autoencoder Topology for Classification in High-Dimensional Noisy Image Datasets", *Sensors (Basel)*, ISSN: 1424-8220, 20 November 2021, Vol. 21, No. 22, Article 7731, Published by MDPI (Multidisciplinary Digital Publishing Institute), Basel, Switzerland, DOI: 10.3390/s21227731, Available: https://www.mdpi.com/1424-8220/21/22/7731.
[89] Lars Hörmander, "Convolution", in: *The Analysis of Linear Partial Differential Operators I. Distribution Theory and Fourier Analysis, Classics in Mathematics*, 2nd ed. Berlin, Heidelberg Germany: Springer, 7 July 2003, Print ISBN: 978-3-540-00662-6, Online ISBN: 978-3-642-61497-2, Series ISSN: 1431-0821, Series E-ISSN 2512-5257, pp. 87-125, DOI: 10.1007/978-3-642-61497-2_5, Available: https://link.springer.com/chapter/10.1007/978-3-642-61497-2_5.
[90] Pooyan Mobtahej, Xulong Zhang, Maryam Hamidi and Jing Zhang, "An LSTM-Autoencoder Architecture for Anomaly Detection Applied on Compressors Audio Data", *Computational and Mathematical Methods*, Online ISSN: 2577-7408, Article No. 3622426, pp. 1–22, 29 September 2022, Published by John Wiley & Sons Inc., Oxford, England, DOI: 10.1155/2022/3622426, Available: https://onlinelibrary.wiley.com/doi/10.1155/2022/3622426.
[91] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage and Anil Anthony Bharath, "Deep Reinforcement Learning: A Brief Survey", *IEEE Signal Processing Magazine*, ISSN: 1558-0792, 9 November 2017, Vol. 34, No. 6, pp. 26–38, Published by The Institute of Electrical and Electronics Engineers (IEEE), New York City, USA, DOI: 10.1109/MSP.2017.2743240, Available: https://ieeexplore.ieee.org/document/8103164.
[92] Juan Terven, "Deep Reinforcement Learning: A Chronological Overview and Methods", *AI*, ISSN: 2673-2688, Vol. 6, No. 3, Article No. 46, 24 February 2025, Published by MDPI (Multidisciplinary Digital Publishing Institute), Basel, Switzerland, DOI: 10.3390/ai6030046, Available: https://www.mdpi.com/2673-2688/6/3/46.
[93] Tien Ho-Phuoc, "CIFAR10 to Compare Visual Recognition Performance between Deep Neural Networks and Humans", arXiv:1811.07270v2 [cs.CV], 15 August 2019, DOI: 10.48550/arXiv.1811.07270, Available: https://arxiv.org/abs/1811.07270.
[94] Farnaz Nazari and Wei Yan, "Convolutional versus Dense Neural Networks: Comparing the Two Neural Networks' Performance in Predicting Building Operational Energy Use Based on the Building Shape", *Proceedings of the 17th IBPSA Conference*, 1-3 September 2021, Bruges, Belgium, pp. 495-502, Published by the International Building Performance Simulation Association (IBPSA) Publications, Oxford, UK, DOI: 10.26868/25222708.2021.30735, Available: https://publications.ibpsa.org/conference/paper/?id=bs2021_30735.

[95] David E. Rumelhart, Geoffrey E. Hinton and Ronald J. Williams, "Learning representations by back-propagating errors", *Nature*, Online ISSN: 1476-4687, Print ISSN: 0028-0836, Vol. 323, pp. 533–536, 9 October 1986, Published by Springer Nature, Berlin, Germany, DOI: 10.1038/323533a0, Available: https://www.nature.com/articles/323533a0.

[96] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do and Kaori Togashi, "Convolutional neural networks: an overview and application in radiology", *Insights into Imaging*, Electronic ISSN: 1869-4101, August 2018, Vol. 9, pp. 611–629, Published by Springer Nature, Berlin, Germany, DOI: 10.1007/s13244-018-0639-9, Available: https://link.springer.com/article/10.1007/s13244-018-0639-9.

[97] Hossein Gholamalinezhad and Hossein Khosravi, "Pooling Methods in Deep Neural Networks, a Review", arXiv:2009.07485v1 [cs.CV], 16 September 2020, DOI: 10.48550/arXiv.2009.07485, Available: https://arxiv.org/abs/2009.07485.

[98] Ernest Jeczmionek and Piotr A. Kowalski, "Flattening Layer Pruning in Convolutional Neural Networks", *Symmetry*, EISSN 2073-8994, Vol. 13, No. 7, Article No. 1147, 27 June 2021, Published by MDPI (Multidisciplinary Digital Publishing Institute), Basel, Switzerland, DOI: 10.3390/sym13071147, Available: https://www.mdpi.com/2073-8994/13/7/1147.

[99] Mattia Perrone, D'Mar Moore, Daisuke Ukeba and John T. Martin, "A CNN Autoencoder for Learning Latent Disc Geometry from Segmented Lumbar Spine MRI", *Annals of Biomedical Engineering*, November 2025, Vol. 53. No. 11, pp. 2911-2919, Published by Springer Nature, Berlin, Germany, DOI: 10.1007/s10439-025-03840-w, Available: https://pubmed.ncbi.nlm.nih.gov/40983777/.

[100] Arun Ranganarsimhaiah, Marija Lindner, Gabriela Ventura Silva and Christoph Herrmann, "Comparison of recurrent neural network-based autoencoders for condition monitoring of individual devices from summarized electrical power signals", *Procedia CIRP [College International pour la Recherche en Productique]*, ISSN: 2212-8271, 27 November 2024, Vol. 130, pp. 1881-1886, Published by Elsevier B.V., Amsterdam, Netherlands, DOI: 10.1016/j.procir.2024.10.332, Available: https://www.sciencedirect.com/science/article/pii/S2212827124014902.

[101] Diederik P. Kingma and Jimmy L. Ba, 'Adam: A Method for Stochastic Optimization', in *Proceedings of the* 3rd *International Conference for Learning Representations*, 7-9 May 2015, San Diego, USA, pp. 1–15, 2015, DOI: 10.48550/arXiv.1412.6980, Available: https://arxiv.org/abs/1412.6980.

[102] Reham Elshamy, Osama Abu-Elnasr, Mohamed Elhoseny and Samir Elmougy, "Improving the efficiency of RMSProp optimizer by utilizing Nestrove in deep learning", *Scientific Reports*, Online ISSN 2045-2322, 31 May 2023, Vol. 13, No. 1, Article No. 8814, Published by Springer Nature, Berlin, Germany, DOI: 10.1038/s41598-023-35663-x, Available: https://www.nature.com/articles/s41598-023-35663-x.

[103] John Duchi, Elad Hazan and Yoram Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization", *The Journal of Machine Learning Research*, e-ISSN: 1533-7928; p-ISSN: 1532-4435, 11 July 2011, Vol. 12, No. 61, pp. 2121-2159, Published by Microtome Publishing, Massachusetts, USA, Available: https://www.jmlr.org/papers/volume12/duchi11a/duchi11a.pdf [Accessed: 20 October 2025].

[104] Ruinan Jin, Xiao Li, Yaoliang Yu and Baoxiang Wang, "A Comprehensive Framework for Analyzing the Convergence of Adam: Bridging the Gap with SGD", in *Proceedings of the 42nd International Conference on Machine Learning (ICML '25)*, 13-19 July 2025, Vancouver, BC, Canada, Article No. PMLR 267, pp. 27979-28030, Available: https://proceedings.mlr.press/v267/jin25c.html [Accessed: 30 December 2025].